



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castellet

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROYECTO FINAL DE CARRERA

TÍTULO DEL PFC: Viabilidad de realidad aumentada para dar soporte a personas con dificultades de memoria

TITULACIÓN: Ingeniería de Telecomunicación (segundo ciclo)

AUTOR: Bàrbara Morey Pascual

DIRECTOR: Roc Meseguer Pallarès

FECHA: 12 de julio de 2012

Title: Viability of Augmented Reality for assisting people suffering from memory loss

Author: Bàrbara Morey Pascual

Director: Roc Meseguer Pallarès

Date: July, 12th 2012

Overview

The number of people suffering from memory loss has risen considerably due to the increase in life expectancy in developed countries. Diseases such as Alzheimer are very disruptive to daily life, especially in elderly people.

People suffering from memory loss tend to forget daily tasks, and thus require other people to help them at certain times.

Technological breakthroughs happen all the time, and some of them have led to certain tools with which to help these people. A very common item in today's society is used for this project: a Smartphone.

The idea behind this project is to use an Android app that captures information about the environment via the mobile device's camera. This information is then processed, and using a database and a self-learning system the aforementioned app is able to show on screen recommendations regarding the next task to be performed by the user.

The goal of this project is to show the technological capabilities of an augmented reality tool. Which is accomplished by explaining how such an App has been implemented, and how it learns from the user's preferences, thus allowing it to adapt to the user and offer a reliable response.

Título: Viabilidad de realidad aumentada para dar soporte a personas con dificultades de memoria

Autor: Bàrbara Morey Pascual

Director: Roc Meseguer Pallarès

Fecha: 12 de julio de 2012

Resumen

A consecuencia del aumento de la esperanza de vida en los países desarrollados se ha incrementado el número de personas con trastornos de la memoria. Enfermedades como el alzhéimer causan dificultades en la vida diaria, sobretodo en gente de avanzada edad.

Las personas que padecen estos trastornos tienden a olvidar tareas cotidianas y por tanto, requieren de personas que les ayuden en determinados momentos.

La tecnología avanza rápidamente y gracias a ello han surgido herramientas con las que ayudar a estas personas. En este proyecto se hace uso de algo muy habitual a día de hoy, un *smartphone*.

Se plantea, mediante una aplicación Android, captar información del entorno real con la cámara del dispositivo móvil. Esta información, se procesa y, haciendo uso de una base de datos y de un sistema de aprendizaje automático, la aplicación es capaz de mostrar por pantalla recomendaciones o información acerca de la próxima tarea a realizar por el usuario.

En este proyecto se pretenden mostrar las posibilidades tecnológicas de una herramienta de realidad aumentada. Para ello en esta memoria se detalla cómo se ha implementado esta aplicación y se muestra como ésta aprende de las preferencias del usuario, permitiendo de este modo adaptarse mejor al mismo y mostrar una respuesta fiable.

A Roc, por haberme dado la oportunidad de realizar este proyecto y por las horas dedicadas a ayudarme.

A mi familia y a Marc, por el apoyo, la comprensión y la ayuda ofrecidos.

A los profesores que han aportado su grano de sabiduría.

A los amigos y compañeros de clase por apoyarme y darme consejo.

A todos ellos muchas gracias.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. REALIDAD AUMENTADA.....	3
1.1. ¿Qué es realidad aumentada?	3
1.2. ¿En qué consiste un sistema AR basado en marcadores?	3
1.3. Introducción a AndAR.....	4
1.3.1 ¿Qué es?	4
1.3.2 ¿Cómo funciona?	5
CAPÍTULO 2. IMPLEMENTANDO LA APLICACIÓN	6
2.1. ¿En qué consiste la aplicación?	6
2.2. Implementación de la administración web	9
2.3. Implementación de un sistema basado en marcadores con AndAR	12
2.3.1. Implementando el control de acceso.....	12
2.3.2. Detección de marcadores.....	15
2.3.3. Devolución de información del marcador detectado	16
2.3.4. Registrando el feedback del usuario	17
CAPÍTULO 3. APRENDIZAJE AUTOMÁTICO.....	20
3.1. ¿Qué es aprendizaje automático?	20
3.2. ¿Qué es Weka?	20
3.3. Implementación del aprendizaje automático en la aplicación Android realizada	21
CAPÍTULO 4. EVALUACIÓN EXPERIMENTAL	25
4.1. Escenario de pruebas	25
4.1.1. Hardware	25
4.1.2. Software.....	25
4.1.3. Configuración.....	25
4.2. Requisitos	26
4.2.1 Requisitos funcionales.....	26
4.3. Usabilidad.....	26
4.3.1. Usuarios avanzados	27
4.3.2. Usuarios normales.....	28
CAPÍTULO 5. CONCLUSIONES Y LÍNEAS FUTURAS.....	30
5.1. Impacto medioambiental	30
5.2. Análisis de los objetivos marcados	30
5.3. Conclusiones del proyecto.....	31

5.4. Conclusión personal	31
5.5. Líneas futuras	32
BIBLIOGRAFÍA	33

ÍNDICE DE FIGURAS

Figura 1.1 Ejemplo práctico de realidad aumentada.	3
Figura 1.2 Sistema AR basado en marcadores.	4
Figura 2.1 Arquitectura cliente-servidor.	6
Figura 2.2 Diagrama de flujo de la aplicación.	7
Figura 2.3 Diagrama de cómo funciona la aplicación.	8
Figura 2.4 Diagrama de secuencia de la aplicación.	9
Figura 2.5 Diagrama de base de datos.	10
Figura 2.6 Formulario de acceso a la administración web.	10
Figura 2.7 Listado de marcadores.	11
Figura 2.8 Listado de textos del marcador.	11
Figura 2.9 Formulario de autenticación.	12
Figura 2.10 Declaración de la interfaz inicial.	13
Figura 2.11 Objetos <code>HttpPost</code> y <code>HttpClient</code> , código.	13
Figura 2.12 Definición de los datos a enviar mediante <code>POST</code> , código.	14
Figura 2.13 Script <code>PHP</code> utilizado para la autenticación.	15
Figura 2.14 Definición del marcador, código.	16
Figura 2.15 Consulta para obtener la información del marcador visualizado. ...	16
Figura 2.16 Muestra del resultado, código.	17
Figura 2.17 Resultado al visualizar un marcador.	17
Figura 2.18 Menú, código.	17
Figura 2.19 Enviar <i>feedback</i> del usuario a la base de datos, código.	18
Figura 2.20 Consulta para insertar el <i>feedback</i> del usuario en la base de datos.	19
Figura 2.21 Menú para poder registrar el <i>feedback</i> del usuario.	19
Figura 3.1 Aprendizaje automático por similitud.	20
Figura 3.2 Diagrama de base de datos.	21
Figura 3.3 Creación del fichero <code>.arff</code> utilizado para entrenar <code>Weka</code>	23
Figura 3.4 Código <code>Weka</code> en <code>Java</code> utilizado para predecir resultados.	24

INTRODUCCIÓN

El concepto de realidad aumentada ha estado presente desde hace décadas. Sin embargo, hasta principios del siglo XXI no han empezado a aparecer aplicaciones al alcance de todos. La evolución de los teléfonos móviles a los llamados *smartphones* ha permitido la expansión de esta tecnología dando lugar a que los usuarios puedan acceder a multitud de aplicaciones que hacen uso de la realidad aumentada.

Esta tecnología, que consiste en combinar en tiempo real elementos virtuales y reales en el espacio, es utilizada en multitud de campos como por ejemplo la medicina, la educación y el ocio entre otros. Su rápida evolución no ha sido debida únicamente a las prestaciones de los *smartphones* sino que han entrado en juego otros factores a nivel de software. Estos factores son el sistema operativo Android y su lenguaje de programación, Java, ambos de libre acceso al desarrollo. Todo este conjunto ha permitido que desarrolladores de software se volcaran con Android y empezaran a aparecer nuevas aplicaciones.

La motivación de este proyecto ha sido introducirse en el mundo de la programación en Android y comprobar si es posible realizar una aplicación de realidad aumentada utilizando recursos de acceso común, como son un *smartphone* y un ordenador.

Dos aplicaciones móviles son las que han alimentado la motivación de este proyecto ya que su objetivo es similar.

La primera de ellas, el proyecto *OnTheBus* [1] es una aplicación desarrollada para la plataforma Android que ha sido implementada por el grupo GABiTAP (Grupo de Aplicaciones Biomédicas y Tecnologías para la Autonomía Personal) de la Universitat Autònoma de Barcelona.

Esta aplicación es plenamente accesible y no discriminatoria para poder ser utilizada por el mayor número de personas posible. Ha sido diseñada para el colectivo que sufre algún tipo de limitación funcional (ya sea visual, cognitiva, auditiva y/o de movilidad). Su principal objetivo es el de mejorar la accesibilidad y movilidad de las personas, guiando a personas (desde un punto de origen a un punto de destino) haciendo uso del transporte público.

La segunda se denomina *Mobile Accessibility* [2]. Esta aplicación ha sido desarrollada por *codefactory* para la plataforma Android y diseñada para dar soporte a personas con deficiencias visuales. Su objetivo es permitir a personas ciegas o con baja visión utilizar un teléfono Android de forma intuitiva, fácil y simple.

Esta aplicación permite realizar y contestar llamadas, administrar los contactos, redactar/leer SMS, configurar alarmas, utilizar el navegador web y el correo electrónico entre otras.

En este proyecto se pretenden descubrir las posibilidades que esta tecnología ofrece y qué mejor manera que implementando una aplicación y que además ésta sirva para ayudar a la gente, y más concretamente, a personas con dificultades de memoria, siguiendo el ejemplo de las aplicaciones anteriores.

La intención es evaluar la viabilidad de realizar una aplicación similar a las comentadas. Esta aplicación implementada debe ser capaz de aprender patrones de comportamiento del usuario y por tanto, realizar recomendaciones adaptándose a las necesidades de dicho usuario. Para llevarlo a cabo se combinará el uso de la realidad aumentada y los marcadores.

A razón de lo anterior, los objetivos marcados en el proyecto son:

- Implementar una aplicación, mediante un sistema de realidad aumentada basado en marcadores, la cual permita dar soporte a personas con dificultades de memoria. Esta aplicación debe incluir: control de acceso de usuarios, detección de marcadores, devolución de una respuesta fiable y registro del *feedback* del usuario.
- Hacer uso de algún mecanismo de aprendizaje automático para poder devolver información útil al usuario en función de la que le ha sido de utilidad y la que no.
- Implementar una administración web a través de la cuál poder configurar el acceso de usuarios a la aplicación y los marcadores con toda su información.
- Realizar una evaluación experimental de la aplicación implementada analizando los requisitos funcionales y la usabilidad, teniendo en cuenta la opinión de usuarios normales y de expertos.

En el primer capítulo se habla de realidad aumentada, realizando una explicación de lo qué es la realidad aumentada, cuáles son sus componentes y en qué consiste un sistema de realidad aumentada basado en marcadores, como AndAR. También se realiza una explicación teórica de qué es AndAR y se explica qué es necesario llevar a cabo para realizar una aplicación utilizando dicha librería.

En el segundo capítulo se detalla la aplicación creada explicando las dos partes que la componen por separado, primero se explica la administración web y a continuación la aplicación en Android implementada.

El tercer capítulo explica en qué consiste el aprendizaje automático y cómo éste ha sido incluido en la aplicación.

El cuarto capítulo contiene la evaluación experimental de la aplicación en donde se incluye el escenario de pruebas utilizado, los requisitos funcionales de la aplicación y la usabilidad de la misma.

Finalmente, se exponen las principales conclusiones del trabajo realizado.

CAPÍTULO 1. REALIDAD AUMENTADA

1.1. ¿Qué es realidad aumentada?

La realidad aumentada (AR) [3] es un término utilizado en los dispositivos móviles que permite superponer objetos virtuales en el mundo real, de forma que para el usuario parezca que ambos objetos coexisten. La Figura 1.1 muestra un ejemplo práctico de qué es realidad aumentada.

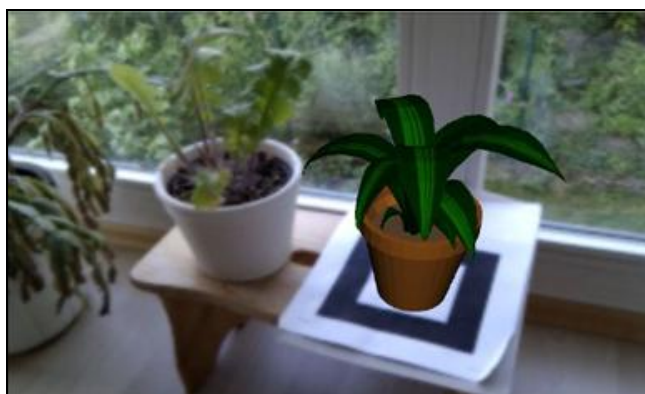


Figura 1.1 Ejemplo práctico de realidad aumentada.

Los componentes de la realidad aumentada son:

- **Pantalla:** donde poder ver reflejado la combinación de lo virtual y lo real.
- **Cámara:** dispositivo que toma información del mundo real y la transmite al software de realidad aumentada.
- **Aplicación:** programa que toma los datos reales y los transforma.
- **Marcadores:** hojas de papel con un patrón impreso en ellas interpretadas por la aplicación que devuelven una respuesta específica (mostrar una imagen en 3D o información por ejemplo).

1.2. ¿En qué consiste un sistema AR basado en marcadores?

Un sistema AR basado en marcadores consiste en utilizar una cámara que permita captar el escenario real y cuando en dicho escenario aparece un marcador, este es reconocido por la aplicación, la cual superpone a la imagen del marcador un objeto virtual. La Figura 1.2 muestra un ejemplo de utilización de los marcadores y el resultado obtenido.

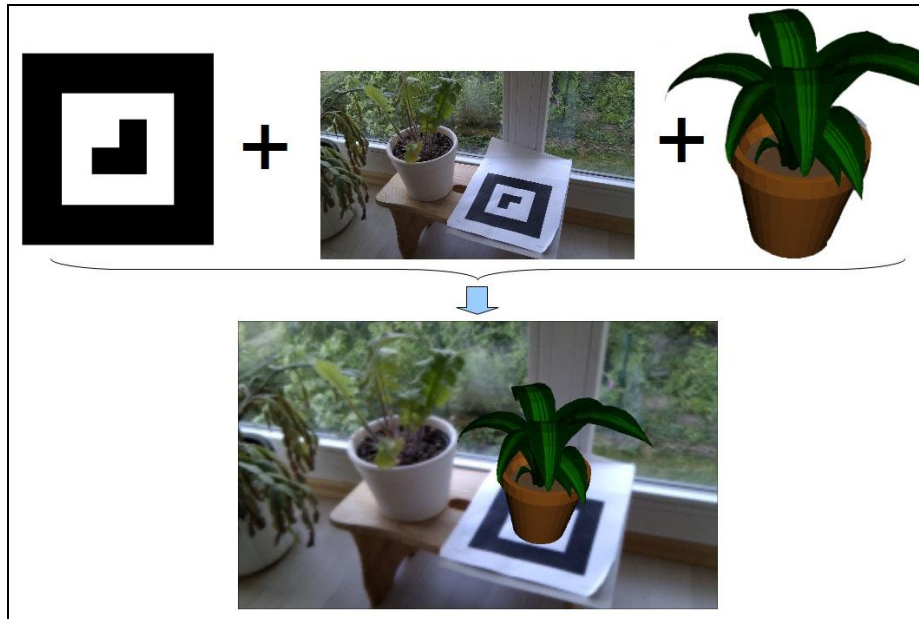


Figura 1.2 Sistema AR basado en marcadores [4].

1.3. Introducción a AndAR

Aclaro que en este proyecto no se ha realizado un estudio de las herramientas de realidad aumentada existentes. El tutor de este proyecto propuso la utilización de la librería AndAR y, tras un estudio para descubrir qué funcionalidades ofrecía esta herramienta se decidió su utilización en el proyecto ya que cubría todas las necesidades.

1.3.1 ¿Qué es?

AndAR [4] es una biblioteca de software de Java que permite realizar aplicaciones de realidad aumentada en la plataforma Android. El proyecto AndAR está publicado bajo la licencia GNU *General Public License*, que permite poder utilizar este software libremente.

Para poder realizar una aplicación basada en AndAR es necesario seguir una serie de pasos antes de empezar a programar [5]:

- Descargar e instalar eclipse [6].
- Instalar el plugin de Android para eclipse [7].
- Descargar la aplicación ejemplo basada en AndAR [8], descomprimirla e importarla en el eclipse.

1.3.2 ¿Cómo funciona?

El primer paso a realizar para poder crear una aplicación de realidad aumentada es crear una actividad, que representa una cosa concreta que puede hacer el usuario y se corresponde con una pantalla de la interfaz de usuario. Seguidamente es necesario extender dicha actividad a la clase abstracta `AndARActivity`. Esta clase permite encender la cámara, detectar los marcadores y renderizar objetos 3D en el marco de vídeo. Con sólo esto la aplicación ya funcionaría, encendiendo la cámara, pero no sería capaz de detectar ningún marcador.

Para poder llevar a cabo la detección de marcadores es necesario registrar los `ARObject` (cada objeto creado para cada marcador) como instancia de `ARToolkit`. Esto es porque se utiliza su sistema de detección de marcadores.

Los marcadores utilizados por `ARToolkit` son plantillas cuadradas de color negro con un cuadrado blanco en el centro, el cuál contiene una figura. Esta figura se denomina patrón, y es lo que será utilizado para reconocer el marcador.

El proceso de reconocimiento de marcadores consiste en digitalizar la imagen para determinar el recuadro negro del marcador encontrando grupos de píxeles conectados por debajo de un determinado umbral de grises. Luego se detecta la figura del interior y se muestrea en una cuadrícula de $N \times N$ (en este caso de 16×16) donde cada posición contiene su respectivo valor de gris. A la hora de detectar un marcador los valores de grises obtenidos son comparados con las plantillas de marcadores creadas con antelación.

Para poder crear plantillas de los marcadores deseados es necesario utilizar una herramienta llamada `mk_patt` [9] [10], que forma parte de `ARToolkit`. Esta herramienta es muy sencilla de utilizar; a continuación se muestran los pasos a seguir para poder crear tus propios marcadores en Windows:

- Descargar la herramienta `mk_patt` [9] [10].
- Descargar el `glut32.dll` y copiar dicho archivo en la carpeta `bin` [11].
- Ejecutar el archivo `mk_patt.exe`. Esto abrirá una ventana en modo consola. Pulsa la tecla `Intro`. Se abrirá una ventana de vídeo.
- Sostén el marcador para que se visualice en la cámara. Rótalo hasta que visualices un marco de color rojo en la parte superior del marcador y uno verde en su parte inferior. Al ser detectado, haz clic sobre la imagen con el ratón y a continuación se te pedirá un nombre para el patrón.
- Copia el archivo creado en la carpeta `assets` de tu proyecto en el eclipse para que la aplicación pueda detectar dicho marcador.
- Utiliza este nombre a la hora de registrar el `ARObject`.

CAPÍTULO 2. IMPLEMENTANDO LA APLICACIÓN

2.1. ¿En qué consiste la aplicación?

La aplicación implementada ha sido creada para dar soporte a personas con dificultades de memoria, facilitando información al usuario sobre su entorno, que será de gran utilidad para ayudarle a ubicarse. Además de lo anterior, pretende ayudar a recordar las tareas del día a día, una cosa que a simple vista parece muy sencilla pero que puede convertirse en un calvario.

La aplicación consiste en una arquitectura cliente-servidor. El cliente es una aplicación desarrollada para la plataforma Android. Esta aplicación contiene código desarrollado por la autora de esta memoria así como tres librerías: AndAR (para incorporar la realidad aumentada), Weka (utilizada para el aprendizaje automático) y MySQL Connector (que permite realizar una comunicación con una base de datos MySQL). El servidor tiene instalado WampServer, que es una plataforma de desarrollo web con Apache2, PHP y MySQL.

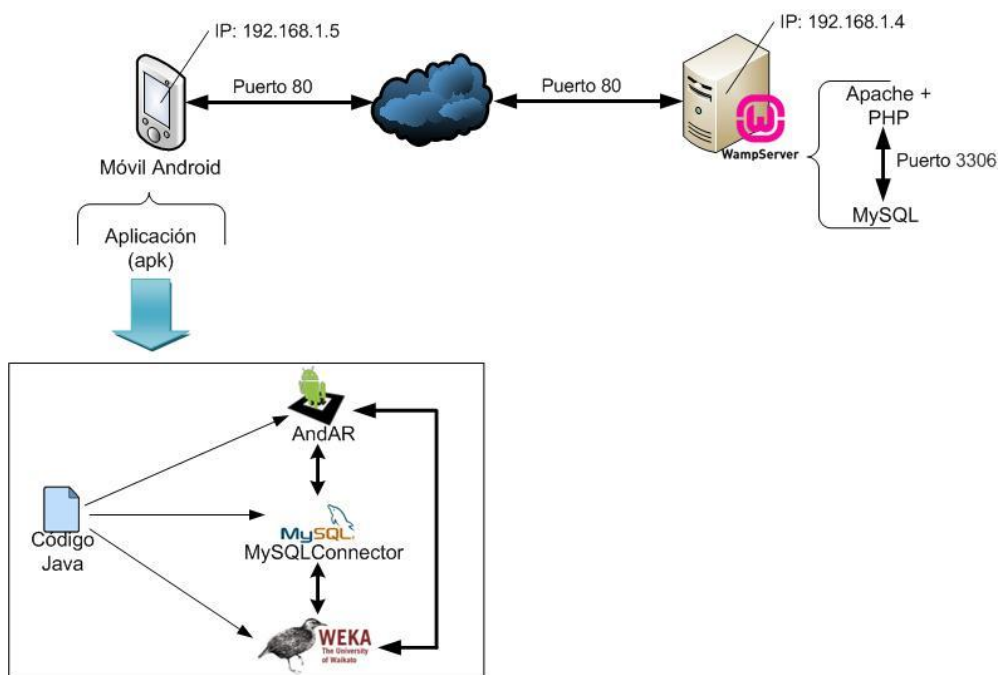


Figura 2.1 Arquitectura cliente-servidor.

La aplicación requiere de una autenticación previa a base de datos. Una vez validada, el usuario puede enfocar con la cámara a un determinado marcador y obtener cierta información. Una vez obtenida, se le ofrece la posibilidad de poder introducir un *feedback*.

La aplicación, gracias a la utilización de la librería AndAR, permite superponer una alerta (objeto virtual) sobre un marcador (objeto real) visualizado a través de la cámara del móvil.

Utilizando la librería Weka se incorpora el aprendizaje automático. Con esto, en función de diferentes factores como: día, hora, lugar (dentro o fuera de casa), compañía (sólo o acompañado) y valoraciones previas del usuario, se devuelven unos resultados u otros.

El diagrama de flujo de la Figura 2.2 refleja la anterior explicación de la aplicación.

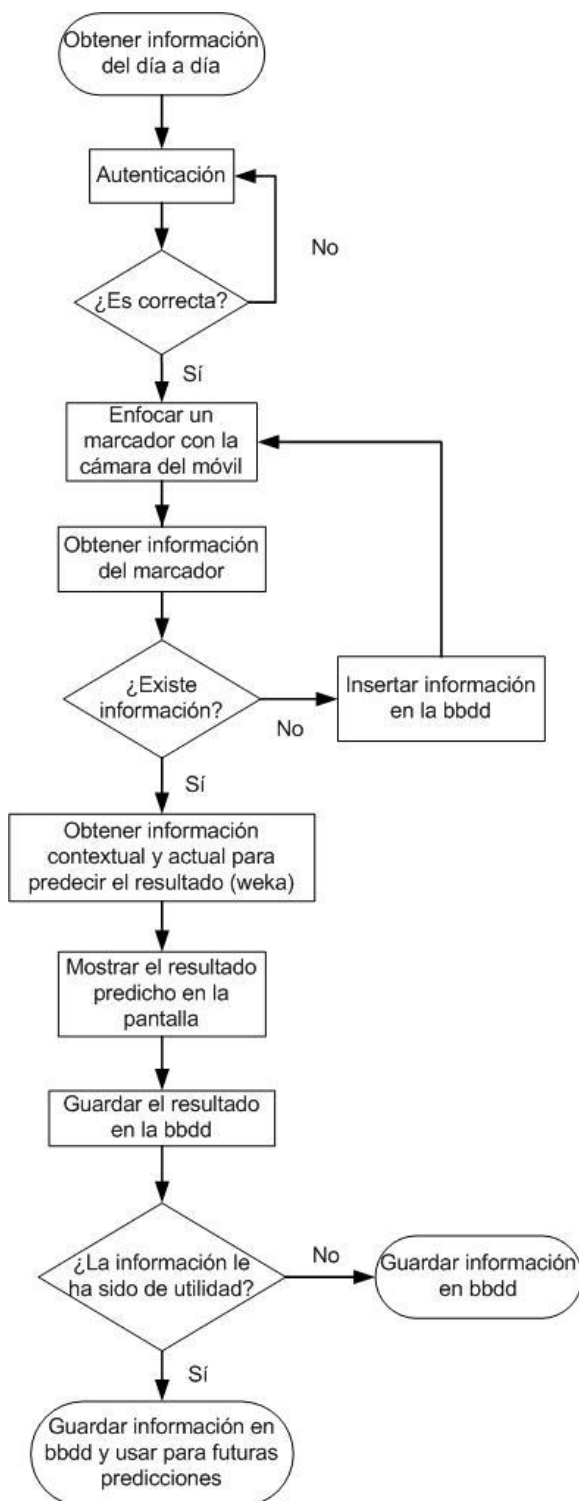


Figura 2.2 Diagrama de flujo de la aplicación.

La Figura 2.3 muestra un diagrama que detalla cómo se realizan las operaciones en la aplicación y en qué momento sucede cada una de ellas.

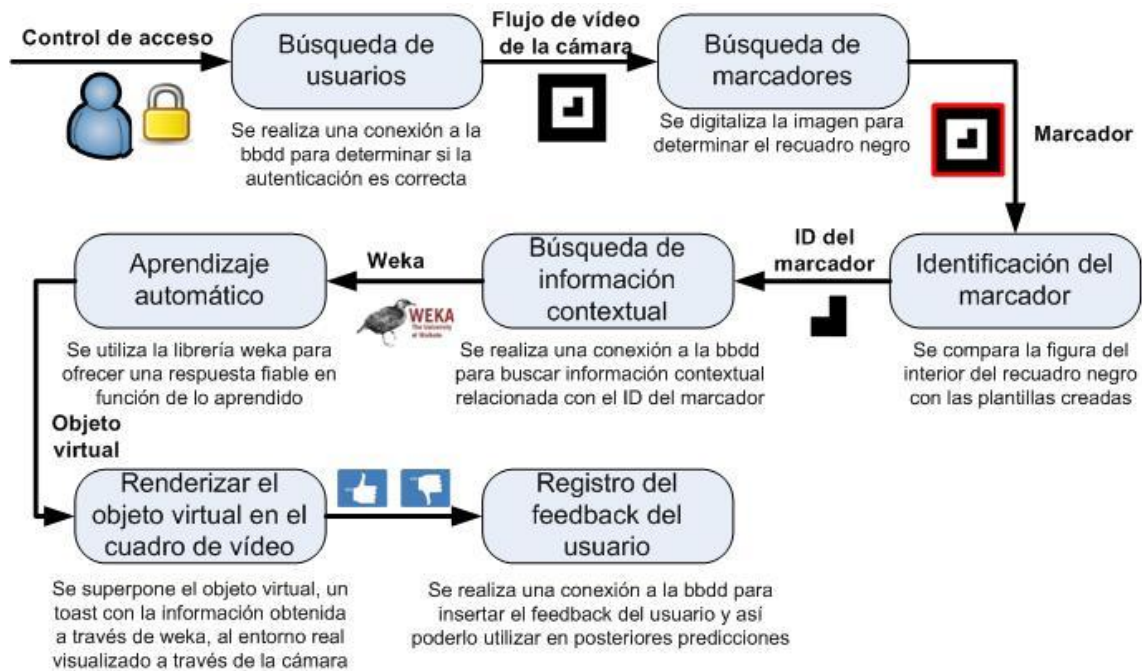


Figura 2.3 Diagrama de cómo funciona la aplicación.

La Figura 2.4 muestra un diagrama de secuencia de la aplicación que permite observar la interacción entre elementos a través del tiempo. En esta aplicación se distinguen seis elementos:

- *Login*: para el control de acceso de los usuarios.
- *CustomActivity*: es la base de cualquier aplicación en Android con interfaz de usuario ya que está relacionado con la pantalla visible.
- *ARToolkit*: contiene la parte de realidad aumentada.
- *CustomContext*: incluye la parte que permite generar el mensaje sobre el entorno real visualizado a través de la cámara.
- *Weka*: para el aprendizaje automático.
- La base de datos MySQL.

De la Figura 2.4 se extrae que el usuario simplemente debe ejecutar la aplicación, autenticarse y enfocar con la cámara a un marcador para obtener una respuesta sobre qué tarea le tocaría hacer a continuación. Finalmente y de forma opcional, el usuario puede dar un *feedback* acerca de la información que se le ha proporcionado.

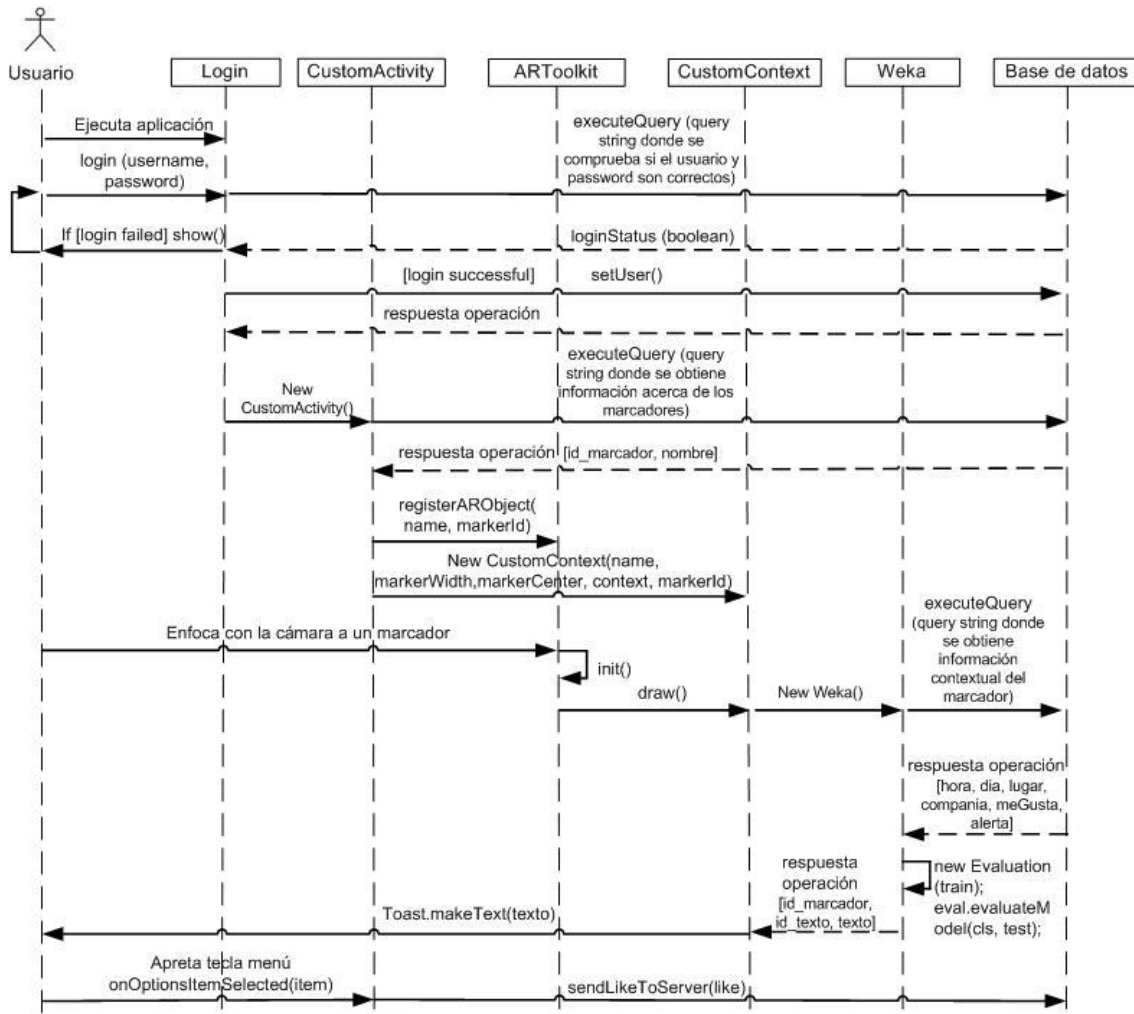


Figura 2.4 Diagrama de secuencia de la aplicación.

2.2. Implementación de la administración web

Para configurar qué usuarios pueden usar la aplicación, qué marcadores existen, qué información mostrar al detectarlos y a quién, se ha instalado y configurado WampServer en su versión 2.1 e [12]. WampServer es un entorno de desarrollo web que permite crear aplicaciones web y ejecutarlas con Apache, PHP y una base de datos MySQL. Una vez instalado, se ha creado una base de datos con todas las tablas necesarias para poder realizar las acciones anteriormente comentadas.

El diagrama de base de datos creado puede verse en la Figura 2.5 en la cual se han definido cuatro tablas. La primera de ellas, “usuario”, es la que se encarga de saber qué usuarios están registrados y por lo tanto pueden utilizar la aplicación. La segunda, “marcador”, es en donde se define el nombre del marcador; dicha tabla está relacionada con la tabla “usuario” para saber qué usuario a creado qué marcador y también con la tabla “rel_marcador_atributos” en donde se definen las características del marcador (información a mostrar y en qué fecha estará activo con dicha información). La última tabla,

“rel_atributos_usuario”, almacena qué usuarios pueden ver qué información y de qué marcadores así como si la información ofrecida les ha resultado útil o no.

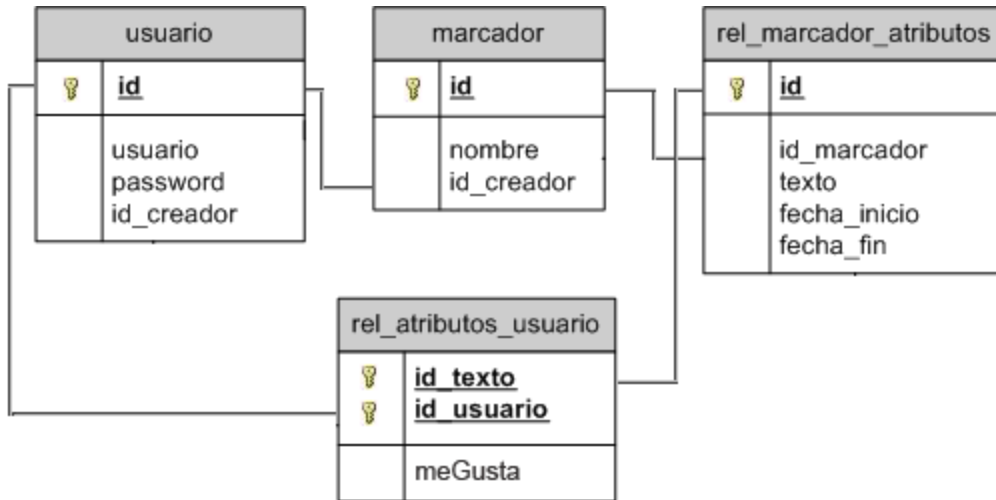


Figura 2.5 Diagrama de base de datos.

Una vez expuesto el esquema de base de datos creado, se puede empezar a explicar la administración web, en la cual se ha creado una página principal de acceso, en donde será requerido que el usuario se autentique para poder entrar. La Figura 2.6 muestra una imagen de esta pantalla.

Figura 2.6 Formulario de acceso a la administración web.

Al realizar la autenticación de forma correcta se accede a una página web en donde aparece un listado de los marcadores creados por dicho usuario. Además se ofrece al usuario la posibilidad de poder crear un nuevo marcador (+), editar uno ya existente (✏), eliminarlo (🗑) o configurar una información acerca de este marcador para los usuarios que desee (💬). La Figura 2.7 muestra un ejemplo de lo comentado anteriormente.









MARCADORES		USUARIOS
LISTADO DE MARCADORES		
ID	NOMBRE	OPERACIONES
1	androidMarker	 
2	eleMarker	 
3	pongMarker	 
 Añadir marcador		

Figura 2.7 Listado de marcadores.

En el caso en el que el usuario quiera configurar información acerca de un determinado marcador, una vez pulsado el botón () del marcador deseado, le aparecerá un listado con todos los textos definidos para dicho marcador. Estos textos serán únicos para determinados usuarios y para un determinado periodo de tiempo. Además, se podrán crear tantos textos como se desee. La Figura 2.8 muestra un ejemplo de lo comentado.






MARCADORES		USUARIOS
LISTADO DE TEXTOS DEL MARCADOR: androidMarker		
ID	TEXTO	OPERACIONES
1	Figura androiddd	 
4	bla bla	 
 Añadir texto		

Figura 2.8 Listado de textos del marcador.

En la administración también existe un módulo de usuarios en el que es posible crear, editar y eliminar los usuarios que tienen acceso a la aplicación. Las opciones del módulo de usuarios son las mismas que las que se pueden observar en la Figura 2.8.

2.3. Implementación de un sistema basado en marcadores con AndAR

Para poder realizar un sistema basado en marcadores se ha utilizado la librería AndAR (para plataforma Android) como base, aunque se le han aplicado varias modificaciones:

- Se ha incluido un control de acceso de usuarios.
- Se realizan múltiples conexiones a una base de datos creada para así realizar la autenticación, obtener información acerca de los marcadores a los que el usuario tiene acceso y guardar un *feedback* del usuario acerca de la información proporcionada por la aplicación.
- Se ha adaptado la aplicación para que a la hora de detectar un marcador no muestre un objeto en 3D sino que muestre un objeto 2D con información, en este caso un *toast*.
- Se ha incluido una librería de aprendizaje automático para que la información proporcionada al usuario sea dinámica y permita adaptarse a él. Esta última modificación será explicada en el apartado 3.3 de esta memoria.

2.3.1. Implementando el control de acceso

Para implementar el control de acceso de usuarios es necesario crear la pantalla de interfaz de usuario por medio de un archivo xml, denominado en este caso `control_login.xml`. Esta pantalla estará formada por dos entradas de texto, una para introducir el usuario y otra para la contraseña, y por un botón, que realizará la autenticación. La Figura 2.9 muestra como se visualiza dicha pantalla en la aplicación.



Figura 2.9 Formulario de autenticación.

Para que esta pantalla aparezca como interfaz inicial al acceder a la aplicación, es necesario crear un filtro por intención (*intent filter*) en el archivo `AndroidManifest.xml` con las líneas mostradas en la Figura 2.10.

```
<activity
    android:name=".Login"
    android:launchMode="singleTask"
    android:label="@string/app_name"
    android:screenOrientation="portrait">
    <intent-filter android:priority="100">
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Figura 2.10 Declaración de la interfaz inicial.

Ahora es necesario tratar los distintos datos del formulario y realizar la comprobación de la autenticación. Para ello es necesario comunicar la aplicación con la base de datos MySQL.

Para conectar a una base de datos MySQL remota desde un dispositivo Android se ha utilizado un script PHP que permite gestionar y ejecutar una secuencia de comandos mediante el protocolo HTTP desde el sistema Android.

Para permitir a la aplicación conectarse a Internet es necesario declarar dichos permisos en el archivo `AndroidManifest.xml` por medio de la siguiente línea:

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

Una vez realizado lo anterior es necesario definir los objetos *HttpClient* y *HttpPost* para poder ejecutar la petición POST deseada una vez pulsado el botón "Login". El código completo es el mostrado en la Figura 2.11.

```
HttpClient httpClient=new DefaultHttpClient();

//en HttpPost se introduce el destino, es decir, el script PHP a ejecutar
HttpPost httpPost= new HttpPost("http://192.168.1.4/login.php");
```

Figura 2.11 Objetos `HttpPost` y `HttpClient`, código.

A continuación es necesario definir qué datos se mandarán mediante el método POST. Ya que se trata de la autenticación del usuario, en este caso será necesario enviar el usuario y la contraseña para su verificación. Estos datos a

mandar mediante POST al script PHP se definen por medio de la utilización de una lista de *NameValuePair*. El código es el mostrado en la Figura 2.12.

```
//se obtienen los valores del nombre de usuario y contraseña del formulario
final EditText user = (EditText) findViewById(id. TxtUsuario);
final EditText pass = (EditText) findViewById(id. TxtPassword);

//se codifica el password mediante el algoritmo MD5
String password = pass.getText().toString();
String contrasena = "";
try {
    MessageDigest md;
    md = MessageDigest.getInstance("MD5");
    byte[] md5hash = new byte[32];
    md.update(password.getBytes("iso-8859-1"), 0, pass.length());
    md5hash = md.digest();
    contrasena = convertToHex(md5hash);
} catch (UnsupportedEncodingException e1) {
    e1.printStackTrace();
} catch (NoSuchAlgorithmException e) {
    e.printStackTrace();
}

//se declara la lista y se añaden los datos de usuario y contraseña a la lista
List<NameValuePair> pairs = new ArrayList<NameValuePair>();
pairs.add(new BasicNameValuePair("user", user.getText().toString()));
pairs.add(new BasicNameValuePair("pass", contrasena));
httppost.setEntity(new UrlEncodedFormEntity(pairs));
```

Figura 2.12 Definición de los datos a enviar mediante POST, código.

Finalmente es necesario ejecutar la petición POST. Esto devuelve un objeto *HttpResponse* que permite extraer y analizar sus datos. El código utilizado se muestra en la siguiente línea:

```
HttpResponse response = httpClient.execute(httppost);
```

Una vez explicado cómo enviar datos por POST a un script PHP es necesario describir cómo debe ser este script. El script PHP utilizado se muestra en la Figura 2.13 y permite:

- Conectar con la base de datos MySQL.
- Ejecutar una consulta en función de los valores de los datos enviados por POST.
- Devolver los datos de la consulta realizada utilizando el formato JSON.

```
<?php

//se realiza la conexión a la base de datos indicando el servidor MySQL, el nombre de
//usuario y el password
$conexion = mysql_connect("localhost", "root", "android");

//se selecciona la base de datos sobre la cual se va a ejecutar la consulta, en este
//caso se denomina admin
mysql_select_db("admin", $conexion);

//se comprueba si el usuario y el password introducidos son correctos mediante los
//datos recibidos por POST
$query = "SELECT * FROM usuario WHERE usuario='".$_$_POST['user']."' AND
password='".$_$_POST['pass']."'";

//se ejecuta la consulta
$res = mysql_query($query, $conexion) or die(mysql_error());

while($e=mysql_fetch_assoc($res))
    $output[]=$e;

//se envían los datos
print(json_encode($output));

//se cierra la conexión con el servidor
mysql_close();

?>
```

Figura 2.13 Script PHP utilizado para la autenticación.

El código de la Figura 2.13 devuelve los datos en formato JSON de la forma en la que se muestra a continuación.

```
[{"id":"id","usuario":"username","password":"md5(password)","id_creador":"id_user"}]
```

En donde se pueden observar todos los campos de la tabla usuario con sus respectivos valores.

Para terminar, es necesario parsear estos datos para que puedan ser utilizados por la aplicación. Como por ejemplo para poder guardar en sesión el usuario autenticado.

2.3.2. Detección de marcadores

Para poder llevar a cabo la definición y detección de los marcadores se utiliza el fichero *CustomActivity.java* que se encarga de registrar un objeto distinto para cada marcador y definir qué acción se llevará a cabo al detectar el

marcador, que será la de visualizar cierta información acerca de él en la pantalla. Para ello se utiliza el código mostrado en la Figura 2.14.

```
//registra un objeto para cada tipo de marcador
ARToolkit artoolkit = super.getArtoolkit();

//define qué acción se llevará a cabo al detectar el marcador
CustomContext context = new CustomContext(
json_data.getString("nombre"),80.0, new
double[]{0,0},getApplicationContext(),json_data.getInt("id_marcador"));

//vincula el resultado al objeto
artoolkit.registerARObject(context);
```

Figura 2.14 Definición del marcador, código.

Los parámetros de la Figura 2.14 que contienen `json_data.getString("")` son datos obtenidos a través de un script PHP. En este caso, los datos obtenidos son los relacionados con la información guardada en la base de datos sobre el marcador (id y nombre del marcador).

El script PHP para obtener la información acerca de los marcadores contiene la consulta que se muestra en la Figura 2.15.

```
//se selecciona la informacion de los marcadores a los que tiene acceso el usuario
//conectado teniendo en cuenta la fecha y hora actuales

$query = "SELECT m.nombre, r.id_marcador FROM marcador m LEFT JOIN
rel_marcador_atributos r ON m.id=r.id_marcador LEFT JOIN rel_atributos_usuario r2
ON r.id=r2.id_texto LEFT JOIN usuario u ON r2.id_usuario = u.id WHERE
u.usuario='".$_$_POST['user']."' AND r.fecha_inicio<=NOW() AND r.fecha_fin>NOW()";
```

Figura 2.15 Consulta para obtener la información del marcador visualizado.

2.3.3. Devolución de información del marcador detectado

El archivo *CustomContext.java* define como se mostrará el resultado una vez detectado el marcador. En este caso se define un *toast*, que es una notificación en Android a la cual se le pasa el texto a mostrar y la duración de la notificación. El código para realizar lo anterior se puede visualizar en la Figura 2.16.


```
public CustomContext(String name, double markerWidth, double[] markerCenter,
Context context, int markerId) {

    super(name, markerWidth, markerCenter);

    Toast.makeText(context, "texto a mostrar", Toast.LENGTH_SHORT).show();
}
```

Figura 2.16 Muestra del resultado, código.

El resultado en la aplicación es el que se muestra en la Figura 2.17.

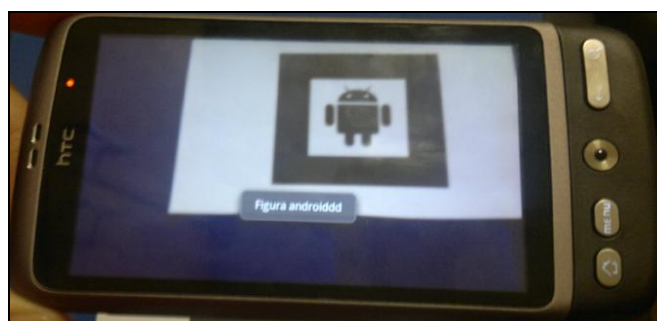


Figura 2.17 Resultado al visualizar un marcador.

2.3.4. Registrando el feedback del usuario

Para recibir un *feedback* del usuario se ha creado un menú en donde el usuario tiene la opción de expresar si la información le ha sido útil o no. Esto servirá a posteriori cuando el sistema implemente el aprendizaje automático y ofrezca al usuario recomendaciones, filtrando las mismas en función de las que le han sido de utilidad y las que no. Para poder recibir un *feedback*, ha sido necesario crear un menú en Android el cual está formado por dos botones, "Me gusta" y "No me gusta", tal y como muestra el código de la Figura 2.18.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/like"
        android:icon="@drawable/like"
        android:title="Me gusta" />
  <item android:id="@+id/dislike"
        android:icon="@drawable/dislike"
        android:title="No me gusta" />
</menu>
```

Figura 2.18 Menú, código.

Además de crear el menú, es necesario definir qué pasará al pulsar alguno de los dos botones del mismo. En este caso, se guardará un 1 o un 0 en la base de datos para conocer si al usuario le ha sido útil la información o no, respectivamente. Para ello se ha utilizado el código de la Figura 2.19.

```

public static PersonalizedObject perObject,
perObject = new PersonalizedObject());

//se instancia el menú con ambos botones
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    return true;
}

//se define qué ocurrirá al pulsar algún botón del menú.
public boolean onOptionsItemSelected(MenuItem item) {
    if(Login.getUser()!="" && perObject.getMarkerId()!=0 &&
perObject.getTextId()!=0){
        if (item.getItemId() == R.id.like){
            sendLikeToServer("1");
            return true;
        }else if (item.getItemId() == R.id.dislike){
            sendLikeToServer("0");
            return true;
        }
    }
    return false;
}

//esta función se encarga de enviar un 1 0 un 0 a la base de datos en función de si la
//información le ha sido útil o no.
public void sendLikeToServer(String like){
    List<NameValuePair> pairs = new ArrayList<NameValuePair>();
    try{
        httpClient=new DefaultHttpClient();
        httpPost= new HttpPost("http://192.168.1.4/likeDislike.php");
        pairs.add(new BasicNameValuePair("userId",
Login.getIdUser().toString()));
        pairs.add(new BasicNameValuePair("textId",
Integer.toString(perObject.getTextId())));
        pairs.add(new BasicNameValuePair("like", like));

        httpPost.setEntity(new UrlEncodedFormEntity(pairs));
        HttpResponse response = httpClient.execute(httpPost);
    }catch(Exception e){
        Log.e("log_tag", "Error en la conexion "+e.toString());
    }
}

```

Figura 2.19 Enviar *feedback* del usuario a la base de datos, código.

En la Figura 2.19 se definen los datos a mandar mediante POST al script PHP cuando se pulsa el botón de “Me gusta” o “No me gusta”. Falta entonces definir cómo será el script PHP, cosa que se puede visualizar en la Figura 2.20.

```
//se inserta el feedback del usuario en la base de datos, un 1 o un 0 en función de si la
//información le ha sido de utilidad o no
```

```
$query = "UPDATE rel_atributos_usuario SET meGusta=".$_POST['like']." WHERE
id_usuario=".$_POST['userId']." AND id_texto=".$_POST['textId'];
```

Figura 2.20 Consulta para insertar el *feedback* del usuario en la base de datos.

Finalmente, la Figura 2.21 muestra cómo se visualiza el menú en la aplicación.



Figura 2.21 Menú para poder registrar el *feedback* del usuario.

CAPÍTULO 3. APRENDIZAJE AUTOMÁTICO

3.1. ¿Qué es aprendizaje automático?

El aprendizaje automático, también conocido como *datamining* (minería de datos), es una tecnología que permite analizar datos y devolver información útil para el usuario basándose en patrones o correlaciones de los diferentes campos analizados.

En este proyecto, se pretende que la aplicación sirva para ayudar a recordar las tareas del día a día a personas con un leve grado de alzhéimer. Por lo tanto, existirá un filtro que dependerá del propio usuario y de la información contextual del marcador (lugar, compañía, día, hora y si la información le ha sido de utilidad o no).

La Figura 3.1 muestra un ejemplo de cómo funciona el aprendizaje automático.



Figura 3.1 Aprendizaje automático por similitud.

3.2. ¿Qué es Weka?

Weka es un conjunto de librerías Java compuesto por diversos algoritmos de aprendizaje automático. Ha sido desarrollado por la universidad de Waikato, en Nueva Zelanda y es un software libre distribuido bajo la licencia de GNU-GPL.

Permite importar datos para realizar un pre-procesado de los mismos y una vez importados permite aplicar algoritmos de regresión para predecir un resultado.

Weka trabaja con ficheros de formato .arff (Attribute-Relation File Format), que tienen una estructura dividida en tres partes:

- **Cabecera:** en donde se define el nombre de la relación (de tipo String).

@relation <nombre-de-la-relación>

- **Declaración de los atributos:** se declaran los atributos que contendrá el archivo y de qué tipo son.

```
@attribute <nombre-del-atributo> <tipo>
```

Donde <nombre-del-atributo> es de tipo String, y en el caso de contener espacios, es necesario introducirlo entre comillas. Los tipos de atributos que acepta Weka son: NUMERIC (para números reales), INTEGER (para números enteros), DATE (para fechas), STRING (para cadenas de texto) y ENUMERADO (para definir los distintos valores que puede tener el atributo). A continuación se muestra un ejemplo del tipo ENUMERADO, ya que será el que se usará en la aplicación desarrollada.

```
@attribute dia {lunes, martes, miercoles, jueves, viernes, sabado, domingo}
```

- **Datos:** se declaran los datos separando los diferentes valores de los atributos por comas y con saltos de línea entre las diferentes relaciones.

```
@data
```

```
manana,lunes,dentro,solo,yes,'tomar las pastillas'  
manana,lunes,fuera,novio,no,'desayunar'
```

3.3. Implementación del aprendizaje automático en la aplicación Android realizada

Para poder implementar el aprendizaje automático primero de todo ha sido necesario importar la librería Weka en el eclipse. Con el objetivo de hacer más fluida la aplicación se ha decidido utilizar una librería Weka, disponible para Android y desarrollada por la Universidad de Castilla-La Mancha [13].

Seguidamente ha sido necesario modificar el diagrama de base de datos creado inicialmente para poder incluir la información contextual. El diagrama de base de datos final es el que se observa en la Figura 3.2.

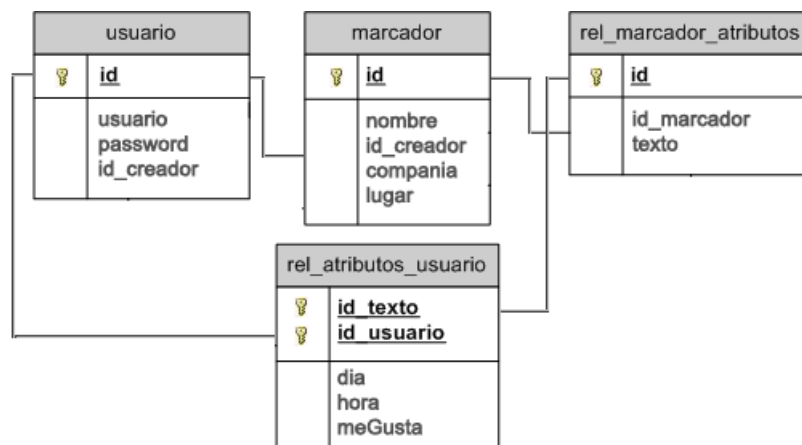


Figura 3.2 Diagrama de base de datos.

A continuación se ha empezado a implementar el aprendizaje automático mediante Weka para la aplicación desarrollada. Primero de todo ha sido necesario obtener los datos almacenados en la base de datos para un determinado usuario. Estos datos servirán como entrenamiento para predecir. Esto se ha realizado llevando a cabo la consulta siguiente a la base de datos:

```
$query = "SELECT r2.hora, r2.dia, m.lugar, m.compania, r2.meGusta as gusta, r.texto as alerta FROM rel_marcador_atributos r LEFT JOIN marcador m ON m.id=r.id_marcador LEFT JOIN rel_atributos_usuario r2 ON r.id=r2.id_texto LEFT JOIN usuario u ON r2.id_usuario = u.id WHERE u.usuario='".$$_POST['user']."'";
```

Una vez los datos han sido obtenidos, se crea un fichero .arff con éstos para que Weka pueda utilizarlos. La Figura 3.3 muestra como se realiza este proceso.

```
//se define el nombre del archivo y se crea para su edición.
```

```
FileOutputStream fOut = context.openFileOutput("remember_training.arff", context.MODE_WORLD_READABLE);
```

```
OutputStreamWriter osw = new OutputStreamWriter(fOut);
```

```
//parte 1. Cabecera
```

```
TRAININGSTRING="@relation 'remember-data'\n";
```

```
//parte 2. Declaración de atributos
```

```
TRAININGSTRING+="@attribute hora {madrugada, manana, mediodia, tarde, anochece, noche}\n";
```

```
TRAININGSTRING+="@attribute dia {lunes, martes, miercoles, jueves, viernes, sabado, domingo}\n";
```

```
TRAININGSTRING+="@attribute lugar {dentro, fuera}\n";
```

```
TRAININGSTRING+="@attribute compania {solo, multitud, novio}\n";
```

```
TRAININGSTRING+="@attribute gusta {yes, no}\n";
```

```
TRAININGSTRING+="@attribute alerta {'tomar las pastillas', desayunar, 'lavar los dientes', comer, merendar, bailar, 'ir al medico', 'escuchar musica', cenar, ducha}\n";
```

```
TRAININGSTRING+="\n";
```

```
//parte 3. Datos
```

```
TRAININGSTRING+="@data\n";
```

```
try{
```

```
JSONArray jArray = new JSONArray(result);
```

```
JSONObject json_data=null;
```

```
for(int i=0;i<jArray.length();i++){
```

```
    json_data = jArray.getJSONObject(i);
```

```
    String hora = json_data.getString("hora");
```

```
    String dia = json_data.getString("dia");
```

```
    String lugar = json_data.getString("lugar");
```

```
    String compania = json_data.getString("compania");
```

```
    Integer gusta = json_data.getInt("gusta");
```

```
    String gusta2 = "";
```

```
    if(gusta==1) gusta2 = "yes";
```

```

        else gusta2 = "no";
        String alerta = json_data.getString("alerta");

        TRAININGSTRING+=hora+","+dia+","+lugar+","+compania+","+gusta2+
        ,+" "+alerta+"\n";
    }

    //se insertan todos los datos en el archivo
    osw.write(TRAININGSTRING);

}catch(JSONException e1){
    return "No existen datos.";
}catch (ParseException e1){
    e1.printStackTrace();
}

//se guardan los cambios realizados en el archivo y se cierra para terminar su edición.
osw.flush();
osw.close();

```

Figura 3.3 Creación del fichero .arff utilizado para entrenar Weka.

Posteriormente se obtiene la información contextual en función del marcador reconocido por la aplicación y se guardan en otro archivo .arff.

La consulta realizada a la base de datos para obtener dicha información es la siguiente:

```
$query = "SELECT lugar, compania FROM marcador WHERE
id=".$_POST['markerId'];
```

El archivo creado es similar al anterior pero solo incluye un dato, ya que se quiere obtener la predicción más útil para el usuario en función del marcador utilizado. En el caso de los datos a predecir por Weka, ya que estos son desconocidos, se introducirán con el símbolo '?' tal y como se muestra en la siguiente línea.

```
TESTSTRING+=hora_actual2+","+dia_actual2+","+lugar_actual+","+compania_actual+
",?,?,?\n";
```

A continuación es necesario que Weka lea los datos de ambos archivos, se defina qué clasificador se utilizará para procesar los datos así como también se defina qué campo va a ser el que se quiera predecir (en este caso, el campo utilizado es el 5, que es el de alerta y se define mediante el término '*class*').

En este proyecto, el clasificador utilizado es el IBk, que es un clasificador del tipo "*Nearest-neighbour*", el cual utiliza la distancia para determinar que instancia entrenada es la más cercana a la instancia dada para predecir.

Lo comentado anteriormente, se puede observar en la Figura 3.4.

```
FileInputStream fln = context.openFileInput("remember_training.arff");
FileInputStream fln2 = context.openFileInput("remember_test.arff");

Reader isr = new InputStreamReader(fln);
Instances train = new Instances(new BufferedReader(isr));

Reader isr2 = new InputStreamReader(fln2);
Instances test = new Instances(new BufferedReader(isr2));

//clasificador utilizado para el entrenamiento
Classifier cls = new IBk(1);
train.setClassIndex(5);
test.setClassIndex(5);
cls.buildClassifier(train);

//predicciones realizadas por Weka
double pred = cls.classifyInstance(test.instance(0));
```

Figura 3.4 Código Weka en Java utilizado para predecir resultados.

CAPÍTULO 4. EVALUACIÓN EXPERIMENTAL

En este capítulo se detallan los requisitos funcionales y la usabilidad de la aplicación creada. También se describen los componentes hardware y software utilizados para llevar a cabo las pruebas de funcionamiento de la aplicación.

4.1. Escenario de pruebas

El escenario de pruebas consiste en la utilización de una red Wifi doméstica en la cual se ha conectado el cliente y el servidor.

4.1.1. Hardware

Las pruebas de funcionamiento de la aplicación se han llevado a cabo usando un portátil como servidor y un teléfono móvil como cliente. El portátil utilizado es un Dell XPS L502X con un procesador Intel Core i5 a 2'4GHz, 4 GB de RAM y un adaptador de red Intel con conectividad Wifi. El dispositivo móvil es una HTC Desire, también conectada vía Wifi.

4.1.2. Software

El sistema operativo del portátil es Windows 7; para poder desarrollar la aplicación ha sido necesario instalar eclipse con el plugin de Android y WampServer. El móvil utiliza el sistema operativo Android en su versión 2.2.

Tanto el portátil como el móvil han sido configurados con IPs privadas de la subred 192.168.1.X siendo la IP asignada al portátil la 192.168.1.4 y la del teléfono móvil la 192.168.1.5.

4.1.3. Configuración

Para llevar a cabo estas pruebas se han utilizado tres marcadores distintos con diferente información contextual: lugar (dentro o fuera de casa) y compañía (sólo o acompañado). Además, las líneas de entrenamiento iniciales de Weka eran cincuenta, puesto que no todos los días y horas habían sido entrenados.

4.2. Requisitos

4.2.1 Requisitos funcionales

La aplicación permite realizar diversas funciones, que son mostradas a continuación:

- ✓ Control de acceso de usuarios.
- ✓ Detección de marcadores.
- ✓ Devolución de una respuesta fiable.
- ✓ Registro del *feedback* del usuario.

Todas estas funciones han sido verificadas en el entorno descrito. Es necesario comentar que la correcta detección de los marcadores es muy sensible a la luz y además, si existen dos marcadores prácticamente idénticos, su detección puede ser errónea. Para ello, conviene que los marcadores sean lo más distintos posible.

La devolución de una respuesta fiable depende del número de líneas de entrenamiento de Weka, es decir de la cantidad de datos guardados con valoraciones del usuario y utilizados para predecir el resultado, las cuales eran inicialmente cincuenta. Cabe comentar que al final de la realización del proyecto este número se incrementó hasta los setenta. Se ha detectado que cuanto mayor es el número de líneas de entrenamiento mayor es la fiabilidad de la respuesta, siendo ésta no fiable si el número de líneas de entrenamiento para una determinada combinación de día y hora es inferior a cinco. Asimismo, el *feedback* del usuario es tenido en cuenta a la hora de predecir el resultado mediante Weka, ya que de este modo es más probable que la respuesta devuelta sea fiable.

Finalmente, mencionar que la aplicación es muy sensible a la calidad de la red. Si ésta es mala, la aplicación funcionará pero habrá indicios de que la conexión no funciona correctamente. Por ejemplo, del mismo modo que en una comunicación en *streaming*, se podrán apreciar cortes mientras la aplicación espera respuesta del servidor, si ésta se demora demasiado.

4.3. Usabilidad

Existen una gran variedad de métodos de evaluación aplicables a cualquier sistema y/o aplicación para medir su usabilidad. En este trabajo se han utilizado como referencia los cuestionarios de IBM [14] que permiten medir la satisfacción del usuario teniendo en cuenta la facilidad de uso del sistema. Estos cuestionarios recopilan datos objetivos (por ejemplo el tiempo de

realización de una tarea) y subjetivos (como la percepción que el usuario tiene sobre la facilidad de uso).

Se ha tomado como referencia el cuestionario *The After-Scenario Questionnaire* (ASQ) de IBM en donde se realiza un corto cuestionario a los usuarios para evaluar su satisfacción una vez usada la aplicación. Las preguntas abordan importantes componentes como: la facilidad de realización de una tarea (en este caso, de obtener una respuesta fiable al visualizar un marcador), el tiempo para completar una tarea y la adecuación de información de apoyo (que en esta aplicación no se ha incluido debido a que el usuario sólo debe autenticarse y posteriormente enfocar con la cámara del móvil a un marcador para obtener una respuesta y no se ha considerado necesario). En este cuestionario los ítems se miden en una escala de siete puntos en donde en los extremos se encuentran los términos “Totalmente de acuerdo” para la 1 y “Totalmente en desacuerdo” para la 7. Además de estos tres componentes se han incluido varios componentes más procedentes del cuestionario *The Computer System Usability Questionnaire* (CSUQ) que permiten saber cómo se siente el usuario al utilizar la aplicación.

Para la evaluación de la aplicación implementada se han utilizado tres marcadores distintos. Los usuarios han testeado la aplicación realizando los siguientes pasos:

1. Introducir un usuario erróneo.
2. Introducir un usuario correcto.
3. Enfocar con la cámara a un marcador.
4. Esperar hasta obtener una respuesta.
5. Introducir un *feedback* sobre si la información ofrecida le ha sido de utilidad o no.

Los puntos 3, 4 y 5 se han repetido tantas veces como se ha deseado.

4.3.1. Usuarios avanzados

Algunos comentarios destacados de los usuarios son:

“Se perciben problemas de lentitud debido al funcionamiento mediante threads.”

“Sería útil añadir cierta información de ayuda, para saber cómo funciona la aplicación.”

“Porqué no mostrar el texto en 3D? Sería más visual.”

“Esta aplicación es muy interesante. Puede ayudar a muchas personas en su día a día.”

De todo lo que estos usuarios han comentado se extraen las siguientes conclusiones. Android no está optimizado para aplicaciones con realidad

aumentada debido a que su funcionamiento es mediante threads y comporta problemas de intercomunicación entre los mismos. Todos interactúan en la misma pantalla; mientras existe un thread que se encarga de mostrar lo que capta la cámara existen otros múltiples realizando operaciones en segundo plano. A todo esto hay que sumarle que se han añadido varios threads para interactuar con la pantalla, como por ejemplo en el caso de los toasts que aparecen por pantalla. Por todo esto se pueden llegar a apreciar ralentizaciones de la aplicación, cosa que se corrige con los nuevos terminales de doble núcleo o cuatro núcleos.

Se podría mejorar la fluidez de la aplicación, se podría añadir información de ayuda acerca de la aplicación para facilitar su uso e incluso en vez de aparecer una etiqueta con el texto sería más visual que apareciera un texto en tres dimensiones. Cabe mencionar que para realizar estas mejoras propuestas serían necesarios móviles más potentes y no todo el mundo posee un móvil de altas prestaciones.

4.3.2. Usuarios normales

Para conocer el grado de satisfacción del usuario en función de la usabilidad de la aplicación implementada, se ha creado un cuestionario de satisfacción. Dicho cuestionario es anónimo y ha sido contestado por un total de doce personas que han probado la aplicación con el objetivo principal de poder utilizar los resultados obtenidos para mejorar la misma de cara a una posible línea de trabajo futura. A continuación se muestra el cuestionario creado y en color rojo se resalta el valor medio de las calificaciones obtenidas.

Rodee la calificación que usted otorga a cada una de las siguientes afirmaciones.

Estoy satisfecho con lo fácil que es utilizar la aplicación.

Totalmente de acuerdo	1	2	3	4	5	6	7	Totalmente en desacuerdo
-----------------------	---	---	----------	---	---	---	---	--------------------------

Sería necesaria alguna información de apoyo para saber cómo funciona la aplicación.

Totalmente de acuerdo	1	2	3	4	5	6	7	Totalmente en desacuerdo
-----------------------	----------	---	---	---	---	---	---	--------------------------

Utilizar la aplicación es simple.

Totalmente de acuerdo	1	2	3	4	5	6	7	Totalmente en desacuerdo
-----------------------	---	----------	---	---	---	---	---	--------------------------

Estoy satisfecho con el tiempo de espera necesario para obtener información.

Totalmente de acuerdo	1	2	3	4	5	6	7	Totalmente en desacuerdo
-----------------------	---	----------	---	---	---	---	---	--------------------------

Me siento cómodo con la aplicación.

Totalmente de acuerdo	1	2	3	4	5	6	7	Totalmente en desacuerdo
-----------------------	---	----------	---	---	---	---	---	--------------------------

La información ofrecida es efectiva y me ayuda a realizar las tareas del día a día.

Totalmente de acuerdo	1	2	3	4	5	6	7	Totalmente en desacuerdo
-----------------------	---	---	----------	---	---	---	---	--------------------------

Fue fácil aprender a usar la aplicación.

Totalmente de acuerdo	1	2	3	4	5	6	7	Totalmente en desacuerdo
-----------------------	---	----------	---	---	---	---	---	--------------------------

CAPÍTULO 5. CONCLUSIONES Y LÍNEAS FUTURAS

5.1. Impacto medioambiental

Este proyecto no tiene un impacto medioambiental elevado por estar basado en la implementación de una aplicación en Android. Se podría tener en consideración el impacto que producen sobre el medio ambiente los teléfonos móviles por ser difícilmente reciclables, incluir sustancias tóxicas para el entorno (como el PVC usado en el cableado de los cargadores, ya retirado por la mayoría de los fabricantes) y su consumo de energía. Este último factor también debe ser considerado en los demás elementos utilizados, como es el caso del ordenador y del router.

5.2. Análisis de los objetivos marcados

Una vez visto el resultado se puede decir que se han cumplido los objetivos del proyecto. Se ha implementado una aplicación en Android utilizando la librería AndAR para conseguir un sistema de realidad aumentada. Otra librería utilizada, Weka, ofrece un sistema capaz de aprender por sí mismo. También se ha creado la administración web a través de la cual configurar el acceso a la aplicación y la información acerca de los marcadores. Esto último es requisito para la librería Weka ya que necesita de un conjunto de líneas de entrenamiento iniciales para poder ofrecer una respuesta fiable. Finalmente, se ha realizado una evaluación experimental para detectar qué piensan los usuarios de la aplicación y cómo ésta podría mejorarse.

Cabe decir que los cuatro requisitos funcionales marcados como objetivos al inicio del proyecto también se han llevado a cabo. Estos son: control de acceso de usuarios, detección de marcadores, devolución de una respuesta fiable y registro del *feedback* del usuario. Hay que recordar que la detección de marcadores depende mucho de la luminosidad de la sala y si ésta no es similar a las condiciones que existían al crear las plantillas de los diferentes marcadores puede que éstos no sean detectados de forma correcta. Por eso se recomienda crear las plantillas en un sitio con buena luminosidad.

Al utilizar la librería AndAR existen varias restricciones en los marcadores: estos deben ser cuadrados y deben ser de color gris o negro, ya que esta librería basa su funcionamiento, como ya se ha explicado anteriormente, en información de color de los patrones.

Se han ido realizando introducciones a los diferentes conceptos utilizados, como realidad aumentada y aprendizaje automático, para que un usuario sin experiencia en este campo pueda entender sin problemas cómo funciona la aplicación.

5.3. Conclusiones del proyecto

La principal conclusión es que en este proyecto se ha podido demostrar la posibilidad de realizar una aplicación para ayudar a personas con problemas de memoria. Para poder conseguirlo ha sido necesario seguir un proceso durante la realización del proyecto, el cual se detalla a continuación.

El primer paso antes de empezar fue aprender cómo programar en Android y qué solución aportar al problema inicial.

Tras un tiempo de formación en un lenguaje de programación muy usado actualmente, se pasó a investigar qué librerías permitirían realizar la solución pensada. Aquí aparecieron AndAR y Weka, cuyo requisito fue la utilización de un servidor con base de datos. Para ello se decidió utilizar WAMP ya que esta herramienta se había usado durante la carrera y permitía usar PHP, lenguaje con el cual había trabajado durante algunos años.

Desde un principio, la idea del proyecto fue difícil de implementar pero interesante. Éste me permitía introducir en el mundo de Android y crear una aplicación para ayudar a otras personas.

Si usando un servidor sencillo y un dispositivo móvil se ha logrado realizar este proyecto, no es difícil pensar que Android tiene mucho potencial y sólo es necesario encontrar qué aplicación realizar.

Como recomendaciones para la realización de aplicaciones similares decir que ambas librerías, AndAR y Weka, han permitido desarrollar la idea planteada inicialmente teniendo mínimos conocimientos de los mismos. Siendo éstas herramientas con funcionalidades suficientes para las necesidades expuestas en este proyecto y no muy complicadas de utilizar. En este sentido pueden contener alguna limitación, como por ejemplo en el caso de AndAR, que al ser una librería sencilla no permite localización en interiores. Esto no ha sido inconveniente en este proyecto, dado que no se requería dicha funcionalidad debido a que cada marcador tenía vinculada cierta información contextual.

5.4. Conclusión personal

Ha sido muy interesante realizar este proyecto ya que he aprendido a programar en Android, tecnología en auge a día de hoy. Además del lenguaje de programación, el proyecto me ha permitido apreciar los diferentes procesos que ocurren desde que se enfoca con una cámara hasta que se superpone un objeto virtual, que es lo que se consigue mediante la realidad aumentada.

Cabe decir que ha habido una mayor dificultad en el aprendizaje del lenguaje de programación Android porque al crear dicha aplicación se partía de una base ya hecha y había detalles que no era posible cambiar, cosa que ha dificultado el añadir las distintas funciones implementadas.

Este proyecto me ha permitido utilizar varios conocimientos a nivel de base de datos que había aprendido a lo largo de la carrera y en el mundo laboral. Juntando el área desconocida de Android y un servidor con base de datos y PHP se ha logrado llegar al objetivo de crear una aplicación que permita ayudar a recordar las tareas del día a día a personas con dificultades de memoria.

5.5. Líneas futuras

Tras este proyecto se podría ampliar la aplicación implementada para que no solamente se prediga la próxima acción a realizar sino que prediga las distintas acciones a hacer a lo largo del día y éstas se muestren como en una especie de calendario. También se podría vincular la aplicación al *google calendar* e insertar automáticamente una alarma para cada una de las distintas acciones a realizar. Hay que tener en cuenta que estas acciones podrían cambiar dependiendo de la información contextual, por lo que esta interacción entre cliente y servidor añadiría carga al sistema.

Por otra parte, también se podría ampliar la detección de marcadores para que estos no tengan que tener un borde cuadrado negro sino que sea capaz de detectar marcadores con formas y/o colores distintos.

Finalmente se podría pasar el procesamiento de datos de Weka al servidor, pero si el aumento de la cantidad de usuarios es muy elevado se tendría que ampliar el número de servidores.

BIBLIOGRAFÍA

- [1] “Proyecto OnTheBus”, disponible en Internet: <<http://www.onthebus-project.com/>> [última fecha de acceso: Junio 2012]
- [2] “*Mobile accessibility for Android*”, disponible en Internet: <<http://www.codefactory.es/en/products.asp?id=415>> [última fecha de acceso: Junio 2012]
- [3] Ronald Azuma. “*A survey of augmented reality*”, disponible en Internet: <<http://www.cs.unc.edu/~azuma/ARpresence.pdf>> [última fecha de acceso: Junio 2012]
- [4] “*AndAR - Android Augmented Reality*”, disponible en Internet: <<http://code.google.com/p/andar/>> [última fecha de acceso: Junio 2012]
- [5] “*How to build applications based on AndAR*”, disponible en Internet: <<http://code.google.com/p/andar/wiki/HowToBuildApplicationsBasedOnAndAR>> [última fecha de acceso: Junio 2012]
- [6] “*Eclipse software*”, disponible en Internet: <<http://www.eclipse.org/downloads/>> [última fecha de acceso: Junio 2012]
- [7] “*ADT plugin*”, disponible en Internet: <<http://developer.android.com/sdk/eclipse-adt.html>> [última fecha de acceso: Junio 2012]
- [8] “*AndAR Sample Project*”, disponible en Internet: <<http://andar.googlecode.com/files/AndARSampleProject.zip>> [última fecha de acceso: Junio 2012]
- [9] “*mk_patt para Windows*”, disponible en Internet: <<http://sourceforge.net/projects/artoolkit/files/artoolkit/2.72.1/ARToolKit-2.72.1-bin-win32.zip/download>> [última fecha de acceso: Junio 2012]
- [10] “*código fuente de mk_patt*”, disponible en Internet: <<http://sourceforge.net/projects/artoolkit/files/artoolkit/2.72.1/ARToolKit-2.72.1.tgz/download>> [última fecha de acceso: Junio 2012]
- [11] “*glut32.dll*”, disponible en Internet: <<http://user.xmission.com/~nate/glut.html>> [última fecha de acceso: Junio 2012]
- [12] “*WampServer*”, disponible en Internet: <<http://www.wampserver.com/en/presentation.php>> [última fecha de acceso: Junio 2012]
- [13] “*Weka for Android*”, disponible en Internet: <<https://github.com/rjmarsan/Weka-for-Android>> [última fecha de acceso: Junio 2012]

- [14] James R. Lewis, “*IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use*”, Human Factors Group Boca Raton, FL.
- [15] “*Data mining with WEKA, Part 1: Introduction and regression*”, <<http://www.ibm.com/developerworks/opensource/library/os-weka1/index.html>> [última fecha de acceso: Junio 2012]
- [16] “*Data mining with WEKA, Part 1: Introduction and regression*”, <<http://www.ibm.com/developerworks/opensource/library/os-weka1/index.html>> [última fecha de acceso: Junio 2012]
- [17] “*Data mining with WEKA, Part 3: Nearest Neighbor and server-side library*”, <<http://www.ibm.com/developerworks/opensource/library/os-weka1/index.html>> [última fecha de acceso: Junio 2012]
- [18] “*Use Weka in your Java code*”, disponible en Internet: <<http://weka.wikispaces.com/Use+WEKA+in+your+Java+code>> [última fecha de acceso: Junio 2012]
- [19] Clase Handler, disponible en Internet: <<http://developer.android.com/reference/android/os/Handler.html>> [última fecha de acceso: Junio 2012]