



*Facultad
de
Ciencias*

**DESARROLLO DE UN SISTEMA
RECOMENDADOR PARA GRUPOS EN
ENTORNOS DE COMPUTACIÓN UBICUA**
(Development of a recommender system for
groups in ubiquitous computing environments)

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Sergio Moreno Barceló

Director: Rafael Duque Medina

Noviembre - 2017

Agradecimientos

Quiero agradecer profundamente a todas las personas que me han ayudado en el desarrollo de este proyecto y principalmente al director del mismo Rafael Duque, tanto por la dedicación que me ha prestado a lo largo de todo el proyecto como por el aprendizaje tan importante que me han supuesto sus comentarios sobre el mismo.

También quisiera agradecer de una forma muy especial tanto a mi familia, que ha sabido estar y apoyarme en todo el proceso, como a mi amigo y compañero Alex, en quien he encontrado un gran referente a seguir para la ejecución del proyecto.

Resumen

El gran aumento de productos de ocio y servicios tecnológicos que ha habido en estos años, así como la gran cantidad de información disponible acerca de los mismos, se traduce en un aumento de opciones nuevas para el usuario dificultando su elección. Es por ello, por lo que, en nuestro sistema actual de vida, se hace necesario el uso de los sistemas recomendadores, los cuales tienen como finalidad facilitar la elección al usuario tanto ofreciendo la mejor opción de entre tantas posibilidades como proporcionando un ahorro de tiempo en la realización de la búsqueda.

Uno de los problemas que ocasiona el gran aumento de información es el que se plantea a la hora de realizar actividades en grupo, como podrían ser el hecho de ir a un restaurante, ir a ver una película, ir a una tienda, etc., no sólo por la cantidad de información a tener en cuenta sino también por la heterogeneidad que se puede plantear en las preferencias de cualquier grupo. Con el fin de resolver este problema se pretende crear una aplicación móvil con sistema operativo Android que funcione como sistema recomendador grupal, ayudando así a los grupos de usuarios a tomar la mejor decisión posible y en el menor tiempo posible, indicándoles, por ejemplo, desde qué película ver hasta qué museo visitar.

Para desarrollar esta aplicación se utilizará un algoritmo recomendador que se encargará de comparar las diferentes opciones con las preferencias de los usuarios del grupo. Además, se hará uso del posicionamiento GPS del usuario para realizar las recomendaciones en función de su cercanía a los distintos puntos de interés, facilitando así la interacción del usuario con la aplicación.

Palabras clave: sistema de recomendación grupal, aplicación Android, algoritmo recomendador, posicionamiento GPS.

Abstract

The great increase of leisure products and technological services that has been in these years, as well as the large amount of information available about them, means an increase in new options for the user, which makes it difficult for him to choose. That is why, in our current life system, it starts to be necessary to use the recommender systems, which have the purpose of facilitate the choice of the user by offering the best option from among many possibilities and providing a time saving in the realization of the search.

One of the problems that causes the great increase of the information, is the one that is posed at the time of realize groups activities, such as going to a restaurant, going to see a movie, going to a store, etc., not only by the amount of information to be taken into account but also by the heterogeneity that can be raised in the preferences of any group. In order to solve this problem, the intention is to create a mobile application with an Android operating system that works as a group recommender system. in order to solve this problem, the intention is to create a mobile application with an Android operating system that works as a group recommender system, helping the user groups to make the best possible decision in the shortest possible time and suggesting them, for example, from which movie to see to what museum to visit.

To develop this application, it will be needed to use a recommender algorithm that will compare the different options with the preferences of the group's users. In addition, the user's GPS positioning will be used to make the recommendations based on their proximity to the different points of interest, thus facilitating the user's interaction with the application.

Keywords: group recommender system, Android application, recommender algorithm, GPS positioning.

Índice

| | |
|---|------------|
| Agradecimientos | I |
| Resumen | II |
| Abstract | III |
| 1. Introducción | 1 |
| 1.1. Motivación | 1 |
| 1.2. Contexto tecnológico | 2 |
| 1.3. Objetivos | 3 |
| 1.4. Estructura del documento | 4 |
| 2. Material y métodos | 5 |
| 2.1. Herramientas y tecnologías | 5 |
| 2.1.1. MySQL | 5 |
| 2.1.2. phpMyAdmin | 5 |
| 2.1.3. MySQL Workbench 6.3 CE | 6 |
| 2.1.4. Android Studio | 6 |
| 2.1.5. Xampp | 7 |
| 2.1.6. Fake GPS location | 7 |
| 2.2. Dataset | 7 |
| 2.3. Metodología | 8 |
| 3. Análisis de requisitos y casos de uso | 9 |
| 3.1. Requisitos funcionales | 9 |
| 3.2. Requisitos no funcionales | 10 |
| 3.3. Casos de uso | 10 |
| 4. Diseño e implementación | 19 |
| 4.1. Diseño arquitectónico | 19 |
| 4.2. Diseño e implementación de la aplicación | 19 |
| 4.2.1. Capa de datos | 19 |
| 4.2.2. Capa de negocio | 20 |
| 4.2.3. Capa de presentación | 25 |
| 5. Pruebas Software | 30 |
| 5.1. Pruebas de un proyecto software | 30 |
| 5.1.1. Pruebas de unidad | 30 |
| 5.1.2. Pruebas de integración | 30 |
| 5.1.3. Pruebas de validación y sistema | 30 |

| | |
|--|-----------|
| 5.2. Pruebas realizadas en la aplicación | 30 |
| 5.2.1. Comprobación del punto de interés | 31 |
| 5.2.2. Comprobación del algoritmo recomendador | 31 |
| 6. Conclusiones y trabajos futuros | 34 |
| 6.1. Conclusiones | 34 |
| 6.2. Trabajos futuros | 36 |

Índice de figuras

| | | |
|-----|---|----|
| 1. | Logo de MySQL | 5 |
| 2. | Logo de phpMyAdmin | 5 |
| 3. | Logo de MySQL Workbench 6.3 CE | 6 |
| 4. | Logo de Android Studio | 6 |
| 5. | Logo de Xampp | 7 |
| 6. | Modelo de desarrollo iterativo e incremental | 8 |
| 7. | Casos de uso | 11 |
| 8. | Diseño de la base de datos | 20 |
| 9. | Código de acceso a la base de datos | 21 |
| 10. | Código del servicio encargado de calcular la posición Gps | 22 |
| 11. | Código de la clase recomendador | 23 |
| 12. | Diseño de la capa de presentación | 26 |
| 13. | Inicio de sesión | 26 |
| 14. | Registrarse | 26 |
| 15. | Pantalla principal | 27 |
| 16. | Listado de grupos | 28 |
| 17. | Nuevo grupo | 28 |
| 18. | Listado de películas | 28 |
| 19. | Añadir película | 28 |
| 20. | Listado de películas | 29 |
| 21. | Añadir película | 29 |
| 22. | Calificación de la película de id 1 | 32 |
| 23. | Películas con calificaciones bajas | 32 |
| 24. | Películas ofertadas por el punto de interés | 32 |
| 25. | Identificadores de las películas | 33 |
| 26. | Película recomendada | 33 |
| 27. | Importación del fichero ratings2.csv | 34 |

Índice de tablas

| | | |
|----|---|----|
| 1. | Requisitos funcionales. | 9 |
| 2. | Caso de uso: Registrarse | 12 |
| 3. | Caso de uso: Iniciar sesión | 13 |
| 4. | Caso de uso: Gestión grupos | 14 |
| 5. | Caso de uso: Crear grupos | 15 |
| 6. | Caso de uso: Unirse a un grupo | 16 |
| 7. | Caso de uso: Recomendar películas | 17 |
| 8. | Caso de uso: Ver lista de películas | 17 |
| 9. | Caso de uso: Añadir películas | 18 |

1. Introducción

Este capítulo analiza las motivaciones que justifican la realización de un proyecto encaminado a desarrollar un sistema recomendador para grupos de usuarios y el interés de este tipo de sistemas en el ámbito de la computación ubicua. El capítulo incluye cuatro secciones. La primera sección describe las motivaciones que sustentan el proyecto a desarrollar. La segunda sección contextualiza el papel de los sistemas recomendadores dentro del paradigma de la computación ubicua. La tercera sección detalla los objetivos concretos que se plantean alcanzar con la realización del proyecto. La cuarta sección muestra la estructura global del resto de esta memoria.

1.1. Motivación

La actual proliferación en el uso de sistemas interactivos para diversas finalidades (laborales, ocio, comunicación, etc.) hace recomendable que el usuario tenga a su alcance mecanismos que le orienten sobre cómo seleccionar las aplicaciones, servicios o productos que mejor se adapten a sus necesidades o preferencias. Esta situación es consecuencia de que la gran masificación de productos y servicios tecnológicos que el usuario tiene a su alcance puede ser un obstáculo a la hora de seleccionar de forma efectiva y eficiente aquellos que más se adecuan a sus intereses.

Los últimos avances tecnológicos también han traído consigo nuevas formas de interacción con los sistemas informáticos por parte de los usuarios. Mientras que tradicionalmente el ordenador de sobremesa era el dispositivo central que facilitaba la interacción con el usuario, actualmente los nuevos dispositivos móviles y las conexiones inalámbricas permiten soluciones de movilidad que garantizan al usuario acceder a los servicios en cualquier momento y lugar y estar en comunicación constante con otros usuarios. Esta circunstancia frecuentemente hace necesario que el usuario demande una recomendación sobre los servicios o productos más adecuados a sus características y ubicación (por ejemplo tiendas, restaurantes o museos cercanos y del gusto del usuario) e incluso que esta recomendación se adecúe también a otros usuarios cuando el contexto así lo demanda (por ejemplo, grupo de turistas que hacen un recorrido por la ciudad y demandan recomendaciones por parte de un dispositivos móvil que sean del gusto de todos ellos).

1.2. Contexto tecnológico

Los sistemas recomendadores son herramientas software que se encargan de recomendar o sugerir productos concretos o ítems, basándose en las preferencias de los usuarios a los que se les quiere realizar la recomendación. [1,2]. La gran proliferación en el uso de dispositivos móviles y la constante utilización de aplicaciones software (servicios de mensajería, comercio electrónico, etc.) hace que sea factible disponer de repositorios con información descriptiva de las preferencias de los usuarios. Actualmente estos sistemas recomendadores tienen uno de sus principales campos de aplicación en los entornos de computación ubicua. En 1988, Mark Weiser acuñó el término computación ubicua para definir un nuevo paradigma de interacción donde los recursos computacionales se distribuyen en el entorno hasta que se hacen indistinguibles y proveen información y servicios para los usuarios cuando y donde lo deseen [3]. Así el usuario está rodeado de dispositivos (smartphones, tarjetas RFID/NFC, paneles electrónicos, etc.) embebidos en algunos casos en objetos cotidianos y que hacen uso de conexiones inalámbricas para ofrecer los servicios al usuario que mejor se adapten a sus necesidades en cada momento. El papel de los sistemas recomendadores dentro del paradigma de computación ubicua está encaminado a reconocer el contexto en el que se encuentra el usuario en cada momento y proporcionar una recomendación que se adecúe al usuario y al entorno en el que se encuentra.

Se pretende por tanto desarrollar un sistema recomendador de carácter grupal, que actúe en los diversos entornos de computación ubicua llevando en todo momento un registro de la posición GPS de los usuarios. Esto facilitará en gran medida generar las distintas recomendaciones a los usuarios en función de su proximidad a diferentes puntos de interés. Que la aplicación sea de carácter grupal y no individual supone una mayor dificultad a la hora de realizar las recomendaciones, ya que implica la existencia de nuevos factores a tener en cuenta a la hora de realizar una recomendación, tales como el tamaño o la heterogeneidad de los grupos.

El presente sistema estará enfocado a proporcionar recomendaciones según los intereses de los usuarios, aunque en este proyecto se ha decidido enfocar el mismo al ámbito del ocio, al considerarse que es éste en el que más impacto tendría, dada la importancia que hoy en día tiene el ocio en nuestra sociedad. Por tanto, se debe aclarar que será capaz de recomendar desde qué película ir a ver al cine hasta a qué restaurante ir a comer, entre otras. Este sistema recomendador se desarrollará como una aplicación para entornos móviles con sistema operativo Android en la que los usuarios

deberán suscribirse a aquellos servicios de recomendación en los que estén interesados. Como la idea del proyecto es realizar un sistema recomendador grupal dentro del ámbito del ocio, se pretende hacer uso de un algoritmo que sea aplicable a todos los servicios dentro de dicho ámbito. Dado que el sistema será el mismo para todos los servicios se realizará inicialmente sólo para uno de ellos permitiéndonos así comprobar su correcto funcionamiento, siendo el elegido para este fin el servicio de la recomendación de películas, puesto que es un apartado lo suficientemente amplio como para llegar a una gran cantidad de público, objetivo éste también muy importante a considerar en cualquier aplicación.

Para que la aplicación pueda funcionar de manera correcta es importante reconocer tanto la necesidad de que los datos reflejen las características y preferencias del usuario de una forma fidedigna, como la necesidad de disponer de una base de datos suficientemente amplia y actualizada de lo que se pretende recomendar.

1.3. Objetivos

El objetivo principal de este proyecto es la implementación de una aplicación para dispositivos móviles con sistema operativo Android que utilice un algoritmo capaz de recomendar a un grupo de usuarios, de acuerdo a la ubicación donde se encuentren, la oferta u ofertas de ocio a las que estén suscritos, siendo necesario para ello cumplir los siguientes puntos.

- Desarrollar un algoritmo capaz de filtrar la información en función de los gustos o necesidades de un grupo de usuarios de manera que pueda realizar recomendaciones acertadas.
- Hacer uso de la geolocalización para adaptar las recomendaciones a la posición de los usuarios.
- Diseñar una base de datos que permita almacenar tanto la información de los usuarios como la información sobre aquello que se pretende recomendar.
- Diseñar una interfaz gráfica que le permita al usuario interactuar de manera intuitiva y atractiva con la aplicación.

1.4. Estructura del documento

Esta memoria contiene cinco capítulos más que reflejan toda la información relacionada con el proyecto, siendo estos capítulos los siguientes:

Capítulo 2: Material y métodos. Aquí se describe el dataset, las herramientas y tecnologías y la metodología que se ha utilizado para el desarrollo del proyecto.

Capítulo 3: Análisis de requisitos y casos de uso. En este capítulo se describen los requisitos funcionales y no funcionales del proyecto desarrollado así como los casos de uso de los que dispone la aplicación móvil.

Capítulo 4: Diseño e implementación. Aquí se expondrá el diseño arquitectónico y su implementación en la aplicación.

Capítulo 5: Pruebas. En este último capítulo se explicarán las pruebas realizadas en la aplicación para comprobar su funcionamiento.

Capítulo 6: Conclusiones y trabajos futuros. En este capítulo se explicarán las conclusiones extraídas del desarrollo del proyecto y los posibles trabajos a realizar en el futuro.

2. Material y métodos

Este capítulo se divide en 3 subcapítulos. En el primero se explicarán tanto las herramientas como las tecnologías utilizadas para desarrollar el proyecto. En el segundo subcapítulo se explicará tanto la procedencia como el contenido del dataset utilizado para realizar las pruebas del sistema recomendador. Por último, en el tercer subcapítulo se tratará la metodología utilizada para llevar a cabo el desarrollo del proyecto.

2.1. Herramientas y tecnologías

Para el desarrollo de la aplicación móvil se ha hecho uso de distintas herramientas y lenguajes de programación y a continuación se va a explicar cuál ha sido su función para el desarrollo del proyecto.

2.1.1. MySQL



MySQL (Figura 1) [4]: Es el sistema de gestión de base de datos relacionales más popular, lo desarrolla, distribuye y soporta MySQL AB. Debido a que es el más popular existe más información sobre él, facilitando así el trabajo desde otras herramientas.

Figura 1: Logo de MySQL

2.1.2. phpMyAdmin



phpMyAdmin (Figura 2) [5]: Es una herramienta de software libre desarrollada en PHP, destinada a la administración de MySQL a través de la web.

Figura 2: Logo de phpMyAdmin

En este proyecto se ha utilizado phpMyAdmin para crear la estructura de todas las tablas de la base de datos. Además este gestor de base de datos permite ver, modificar, añadir o eliminar datos de manera muy sencilla, por lo que se ha utilizado para modificar algunos aspectos de la base de datos del dataset. También se ha

utilizado esta herramienta para ejecutar consultas SQL de prueba que posteriormente se han usado en la aplicación móvil.

2.1.3. MySQL Workbench 6.3 CE



Figura 3: Logo de MySQL Workbench 6.3 CE

MySQL Workbench (Figura 3) [6]: Proporciona a los administradores de bases de datos y a los desarrolladores un entorno de herramientas integrado para diseño y modelado de bases de datos, desarrollo de SQL, administración de base de datos y migración de la base de datos.

Esta herramienta se ha utilizado para crear las relaciones entre las tablas. Además, utilizando MySQL Workbench se puede hacer uso de la ingeniería inversa para generar el modelo de la base de datos a modo de esquema, permitiéndonos ver de esta forma todas las relaciones existentes en la base de datos.

2.1.4. Android Studio



Figura 4: Logo de Android Studio

Android Studio (Figura 4) [7]: IDE oficial para Android que nos permite crear apps para dispositivos Android, así como la edición, depuración y compilación de código. La aplicación ha sido totalmente desarrollada en este entorno haciendo uso de Java y XML.

Además, esta herramienta nos permite ejecutar directamente en nuestro teléfono móvil la aplicación para comprobar el funcionamiento de la misma. También cuenta con un emulador que nos permite probar la aplicación en diferentes modelos de móvil y en diferentes versiones de Android.

2.1.5. Xampp



Xampp (Figura 5) [8]: Es una distribución de Apache que contiene entre otros componentes MySQL con su gestor phpMyAdmin, el cual ha sido utilizado para generar las tablas de la base de datos de la aplicación.

Figura 5: Logo de Xampp

La versión de Xampp que se ha utilizado ha sido la 1.8.1, versión antigua del programa. Esto se debe a que la librería utilizada en Android Studio para acceder a la base de datos no permitía dicho acceso utilizando la última versión del programa.

2.1.6. Fake GPS location

Fake GPS location [9]: Es una aplicación para teléfonos móviles que nos permite simular que estamos en cualquier posición del mundo. Se ha usado esta aplicación para poder comprobar el posicionamiento GPS, pudiendo posicionarnos directamente en cualquier punto de interés existente.

2.2. Dataset

En este apartado se va a explicar tanto el origen como los componentes del dataset utilizado para comprobar el correcto funcionamiento de la aplicación. Este dataset ha sido obtenido de MovieLens, web de recomendación de películas que cuenta con cientos de miles de usuarios registrados. Esta web ha sido desarrollada por Grouplens Research, un laboratorio de investigación del departamento de informática e ingeniería de Minnesota en 1997.

Este DataSet contiene 20.000.263 puntuaciones de películas y 27.278 películas. El fichero que contiene las películas, contiene además del título de las mismas, el género o géneros a los que pertenece dicha película. Las puntuaciones de las películas han sido creadas por 138.493 usuarios entre el 9 de enero de 1995 y el 31 de marzo de 2015. Además de esto, otro de los ficheros del dataset contiene los links de las películas a páginas especializadas como <http://www.imdb.com/> y <https://www.themoviedb.org/>.

Por último, el dataset contiene dos ficheros con etiquetas siendo estas etiquetas palabras o conjuntos de palabras claves sobre las películas, por ejemplo acción, sangre, política, prisión. . . Uno de los ficheros contiene 1128

etiquetas, que son fijas para todas las películas, es decir, cada película tiene asignadas las mismas 1128 etiquetas. El otro fichero se encarga de relacionar estas etiquetas con las películas y de asignarles un valor numérico que indica la relevancia de la etiqueta con la película.

2.3. Metodología

Es necesario establecer una metodología adecuada que permita controlar la evolución del proyecto. La metodología es el marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo del proyecto [11]. Para el desarrollo de este proyecto se ha elegido seguir una metodología iterativa e incremental (Figura 6), que consiste en descomponerlo en pequeños módulos, donde cada uno de éstos aporta una funcionalidad distinta al sistema.

Esta metodología permite tener una versión operativa del proyecto desde una etapa temprana de su desarrollo, y partiendo de esta versión ir agregándole las nuevas funcionalidades. Debido a que con cada iteración se van cumplimentando diferentes requisitos se reduce el número de errores, y éstos solo afectarán a la nueva funcionalidad y no a toda la aplicación. Esta metodología le aporta al proyecto una mayor facilidad de cambios durante el proceso de desarrollo.

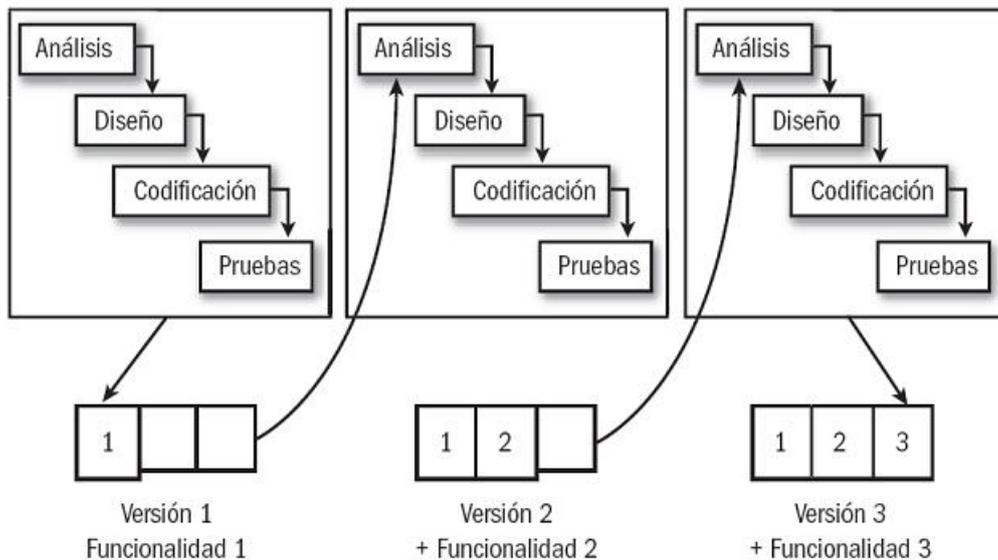


Figura 6: Modelo de desarrollo iterativo e incremental

3. Análisis de requisitos y casos de uso

Este capítulo contiene tres subcapítulos, el primero contiene cuáles son cada una de las funcionalidades de la propia aplicación. El segundo subcapítulo cuáles son aquellas características que debe poseer la aplicación y el tercer subcapítulo contiene todos los casos de uso de la aplicación detallados.

3.1. Requisitos funcionales

A través de los siguientes requisitos funcionales se especifica qué es lo que puede hacer, tanto el sistema como los tipos de usuarios existentes.

| Identificador | Descripción |
|---------------|--|
| RF01 | Cualquier usuario podrá registrarse en la aplicación rellenando todos los campos del formulario de registro. |
| RF02 | El sistema solo permitirá el acceso a usuarios registrados. Los usuarios deberán introducir su nombre de usuario y contraseña para acceder al sistema. |
| RF03 | Cualquier usuario registrado podrá acceder al listado de sus películas y calificaciones, así como agregar a la lista nuevas películas calificadas. |
| RF04 | Cualquier usuario registrado podrá acceder a la gestión de grupos, pudiendo añadir o unirse a un nuevo grupo, y visualizar el listado de aquellos grupos a los que pertenezca. |
| RF05 | El sistema comprobará la posición de todos los usuarios que estén conectados y que tengan activado el posicionamiento GPS del móvil. |
| RF06 | El sistema comparará la posición de los usuarios con la de los diferentes puntos de interés cada vez que la posición de los usuarios cambie. |
| RF07 | El sistema mostrará la película recomendada a los usuarios del grupo que se encuentren dentro del punto de interés. |

Tabla 1: Requisitos funcionales.

3.2. Requisitos no funcionales

En este segundo subcapítulo se especifican las características que posee el sistema, como pueden ser los parámetros de seguridad que cumple, la eficiencia que se busca, la usabilidad y la disponibilidad.

Seguridad: El sistema asegura el cumplimiento de la ley orgánica de protección de datos, basada en el artículo 18 de la constitución española que dice: “La Ley limitará el uso de la informática para garantizar el honor y la intimidad personal y familiar de los ciudadanos y el pleno ejercicio de sus derechos”

Eficiencia: El sistema consigue la eficiencia gracias al algoritmo empleado, realizando la recomendación de la manera más acertada en el menor tiempo posible, dependiendo éste del tamaño del grupo y de las películas vistas por cada usuario.

Usabilidad: La aplicación posee una interfaz sencilla que pretende que la aplicación pueda ser usada por cualquier persona, tenga o no experiencia en el uso de aplicaciones móviles.

Disponibilidad: La aplicación funcionará las 24 horas del día, pero dado que funciona por GPS no será posible realizar una recomendación si el usuario no se encuentra en el punto de interés debido a que esté cerrado.

3.3. Casos de uso

A continuación se van a exponer todos los casos de uso de los que dispone la aplicación. Tal y como muestra la siguiente figura, la aplicación sólo admite dos actores, el usuario no registrado cuya única acción posible es registrarse, y el usuario registrado, el cual tiene acceso al resto de casos de uso expuestos y no hay más roles dado que la figura de un administrador no es necesaria para la aplicación, cualquier cambio a los usuarios se realizará directamente desde la base de datos a la cual no tienen acceso los usuarios, a excepción de las propias consultas que realiza la aplicación a sus datos de usuario.

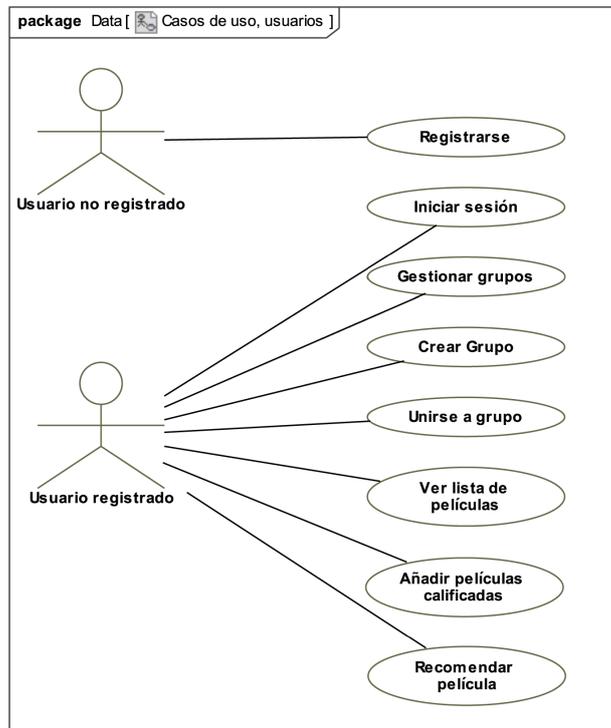


Figura 7: Casos de uso

Ahora pasamos a detallar más específicamente cada uno de los casos de usos que se muestran en el diagrama anterior.

| | |
|----------------------|--|
| Id + Nombre | CS01-Registrar Usuario. |
| Actor principal | Usuario no registrado. |
| Actor secundario | |
| Descripción | El sistema permitirá añadir un nuevo usuario a la base de datos. |
| Evento de activación | El usuario selecciona la opción “registrarse aquí”. |
| Precondición | El usuario no tiene que estar registrado con anterioridad. |
| Garantías si éxito | El usuario quedará registrado en el sistema y sus datos guardados. |
| Garantías Mínimas | La confidencialidad e integridad de los datos del usuario. |
| Escenario Principal | <ol style="list-style-type: none"> 1. El sistema mostrará el formulario con los campos nombre, correo electrónico, usuario y contraseña. 2. El usuario rellena los campos del formulario. 3. El usuario indica al sistema que procese el formulario. 4. El sistema comprueba que haya acceso a internet. 5. El sistema comprueba que todos los campos obligatorios han sido rellenos. 6. El sistema comprueba que el usuario no se encuentre registrado en el sistema. 7. El sistema guarda los datos del usuario en la base de datos. 8. El sistema informa al usuario de que ha sido registrado correctamente. |
| Extensiones | <ol style="list-style-type: none"> 4a. El usuario no tiene acceso a internet. <ol style="list-style-type: none"> a.1 El sistema le pide al usuario que compruebe la conexión a internet. 5a. El usuario no introduce todos los campos obligatorios. <ol style="list-style-type: none"> a.1 El sistema le indica los campos que le faltan por rellenar. 6a. El campo usuario se encuentran en el sistema <ol style="list-style-type: none"> a.1 El sistema le indica que el usuario está registrado. |
| Comentarios | |

Tabla 2: Caso de uso: Registrarse

| | |
|----------------------|--|
| Id + Nombre | CS02-Iniciar sesión. |
| Actor principal | Usuario registrado. |
| Actor secundario | |
| Descripción | El sistema dará acceso a la aplicación al usuario. |
| Evento de activación | El usuario selecciona la opción iniciar sesión. |
| Precondición | El usuario tiene que estar registrado en el sistema. |
| Garantías si éxito | El usuario accederá a la pantalla inicial de la aplicación. |
| Garantías Mínimas | La confidencialidad e integridad de los datos del usuario. |
| Escenario Principal | <ol style="list-style-type: none"> 1. El sistema mostrará el formulario de inicio de sesión con los campos usuario y contraseña. 2. El usuario rellena los campos del formulario. 3. El usuario indica al sistema que procese el formulario. 4. El sistema comprueba que haya acceso a internet. 5. El sistema comprueba que todos los campos obligatorios han sido rellenados. 6. El sistema comprueba que los datos correspondan a un usuario. 7. El sistema informa al usuario que se ha iniciado sesión con éxito. 8. El sistema carga la pantalla de inicio de la aplicación. |
| Extensiones | <ol style="list-style-type: none"> 4a. El usuario no tiene acceso a internet. <ol style="list-style-type: none"> a.1 El sistema le pide al usuario que compruebe la conexión a internet. 5a. El usuario no introduce todos los campos obligatorios. <ol style="list-style-type: none"> a.1 El sistema le indica los campos que le faltan por rellenar. 6a. El campo usuario no se encuentran en el sistema. <ol style="list-style-type: none"> a.1 El sistema le indica que el usuario no existe. |
| Comentarios | |

Tabla 3: Caso de uso: Iniciar sesión

| | |
|----------------------|--|
| Id + Nombre | CS03-Gestión grupos. |
| Actor principal | Usuario registrado. |
| Actor secundario | |
| Descripción | El sistema mostrará la pantalla con el listado de grupos a los que pertenece el usuario. |
| Evento de activación | El usuario selecciona la opción “gestionar grupos”. |
| Precondición | El usuario tiene que haber iniciado sesión en el sistema. |
| Garantías si éxito | El sistema mostrará una pantalla con los grupos a los que pertenece el usuario. |
| Garantías Mínimas | |
| Escenario Principal | 1. El sistema muestra una lista de los grupos a los que pertenece el usuario. |
| Extensiones | |
| Comentarios | |

Tabla 4: Caso de uso: Gestión grupos

| | |
|----------------------|--|
| Id + Nombre | CS04-Crear grupos. |
| Actor principal | Usuario registrado. |
| Actor secundario | |
| Descripción | El sistema mostrará el formulario crear grupo. |
| Evento de activación | El usuario selecciona la opción “crear grupo”. |
| Precondición | El usuario tiene que haber iniciado sesión en el sistema. |
| Garantías si éxito | El sistema creará un nuevo grupo el cual se mostrará en la lista de grupos. |
| Garantías Mínimas | |
| Escenario Principal | <ol style="list-style-type: none"> 1. El sistema mostrará un formulario para introducir el nombre del nuevo grupo. 2. El usuario rellena el campo nombre del formulario. 3. El usuario indica al sistema que procese el formulario. 4. El sistema comprueba que haya acceso a internet. 5. El sistema comprueba que el nombre introducido no esté repetido. 6. El sistema añade el nuevo grupo al usuario. |
| Extensiones | <ol style="list-style-type: none"> 4a. El usuario no tiene acceso a internet. <ol style="list-style-type: none"> a.1 El sistema le pide al usuario que compruebe la conexión a internet. |
| Comentarios | |

Tabla 5: Caso de uso: Crear grupos

| | |
|----------------------|--|
| Id + Nombre | CS05-Unirse a un grupo. |
| Actor principal | Usuario registrado. |
| Actor secundario | |
| Descripción | El sistema mostrará el formulario crear grupos. |
| Evento de activación | El usuario selecciona la opción “unirse a grupo”. |
| Precondición | El usuario tiene que haber iniciado sesión en el sistema. |
| Garantías si éxito | El sistema añadirá al usuario al grupo indicado, mostrándolo en las lista de grupos. |
| Garantías Mínimas | |
| Escenario Principal | <ol style="list-style-type: none"> 1. El sistema mostrará un formulario para introducir el nombre del grupo al que desea unirse. 2. El usuario rellena el campo nombre del formulario. 3. El usuario indica al sistema que procese el formulario. 4. El sistema comprueba que haya acceso a internet. 5. El sistema comprueba que exista el nombre del grupo. 6. El sistema añade al usuario al grupo introducido. |
| Extensiones | <ol style="list-style-type: none"> 4a. El usuario no tiene acceso a internet. <ol style="list-style-type: none"> a.1 El sistema le pide al usuario que compruebe la conexión a internet. |
| Comentarios | |

Tabla 6: Caso de uso: Unirse a un grupo

| | |
|----------------------|--|
| Id + Nombre | CS06-Recomendar películas. |
| Actor principal | Usuario registrado. |
| Actor secundario | |
| Descripción | El sistema mostrará la película recomendada para el grupo con el que ha asistido al punto de interés. |
| Evento de activación | La posición GPS del usuario coincide con la de un punto de interés. |
| Precondición | El usuario tiene que haber iniciado sesión en el sistema. El usuario tiene que tener activado el GPS del móvil. |
| Garantías si éxito | El sistema mostrará la película recomendada al grupo. |
| Garantías Mínimas | |
| Escenario Principal | 1. El sistema comparará la posición GPS con la de los puntos de interés. 2. El sistema ejecutará el algoritmo recomendador. 3. El sistema mostrará la película más recomendada al usuario. |
| Extensiones | |
| Comentarios | La recomendación será más precisa en función de la heterogeneidad de los miembros del grupo. |

Tabla 7: Caso de uso: Recomendar películas

| | |
|----------------------|---|
| Id + Nombre | CS07-Ver lista de películas. |
| Actor principal | Usuario registrado. |
| Actor secundario | |
| Descripción | El sistema mostrará la lista de películas vistas por el usuario junto a su calificación. |
| Evento de activación | El usuario selecciona la opción “ver películas”. |
| Precondición | El usuario tiene que haber iniciado sesión en el sistema. |
| Garantías si éxito | El sistema mostrará la lista de las películas. |
| Garantías Mínimas | |
| Escenario Principal | 1. El sistema mostrará la lista de las películas vistas por el usuario junto a su calificación. |
| Comentarios | |

Tabla 8: Caso de uso: Ver lista de películas

| | |
|----------------------|--|
| Id + Nombre | CS08-añadir películas. |
| Actor principal | Usuario registrado. |
| Actor secundario | |
| Descripción | El sistema mostrará el formulario añadir película. |
| Evento de activación | El usuario selecciona la opción “añadir películas”. |
| Precondición | El usuario tiene que haber iniciado sesión en el sistema. |
| Garantías si éxito | El sistema añadirá la película a la lista de películas vistas por el usuario. |
| Garantías Mínimas | |
| Escenario Principal | <ol style="list-style-type: none"> 1. El sistema muestra un formulario para agregar película calificada, con los campos título y calificación. 2. El usuario rellena los campos del formulario. 3. El usuario indica al sistema que procese el formulario. 4. El sistema comprueba que haya acceso a internet. 5. El sistema comprueba que el nombre introducido no esté repetido. 6. El sistema informa al usuario que se ha añadido la película con éxito. 7. El sistema carga la pantalla con la lista de las películas vistas por el usuario. |
| Comentarios | |

Tabla 9: Caso de uso: Añadir películas

4. Diseño e implementación

Este capítulo se divide en dos subcapítulos. En el primero se define y detalla el funcionamiento del diseño arquitectónico y en el segundo se relaciona este diseño con el desarrollo del proyecto.

4.1. Diseño arquitectónico

La arquitectura en la que se basa la aplicación se conoce como arquitectura de tres capas que consiste en dividir la aplicación en tres partes interdependientes, en la que cada una de ellas cumple una función diferente dentro del sistema. Estas tres capas son:

Capa de datos: Es en esta capa donde se encuentra toda la estructura de la base de datos a la que se conecta la aplicación.

Capa de presentación: Hace referencia a la interfaz de usuario, siendo ésta por tanto la capa que se encarga de la comunicación con el usuario.

Capa de negocio: Esta capa es la lógica de la aplicación, es decir, todo el código que le da una funcionalidad a la interfaz de usuario de la capa de presentación y el código que conecta con la base de datos. Esta capa por tanto hace de enlace entre las otras dos.

4.2. Diseño e implementación de la aplicación

En este subcapítulo se hablará del diseño y la implementación de cada una de las capas de la aplicación desarrollada.

4.2.1. Capa de datos

En esta capa se encuentra la estructura de la base de datos que ha sido generada a partir del dataset de Movielens. La tabla principal de la base de datos es la tabla **users**, que contiene toda la información de los usuarios registrados en la aplicación. Esta información corresponde a los datos personales del usuario, como su nombre, email, nombre de usuario y contraseña. En esta tabla también se guarda la posición GPS del usuario, almacenada en función de la latitud y la longitud. Cada usuario tiene además una lista de películas calificadas en la tabla **ratings**, y una lista de grupos a los que pertenece en la tabla **groupsusers**.

Por otra parte, la tabla **movies** contiene la información de todas las películas existentes en el sistema y además se relaciona con otras tablas como son **links** y **genomenscores**. La primera contiene los enlaces de las películas a páginas web especializadas y la segunda las etiquetas fijas de las películas. Por último, la tabla **puntosdeintereses** contiene la información acerca de los diferentes puntos de interés registrados en la aplicación, accediendo desde ésta a la tabla **moviessitios**, la cual contiene las películas ofertadas por cada punto de interés.

Las tablas mencionadas en esta capa de datos son las imprescindibles para que el algoritmo recomendador de la capa de negocio pueda realizar la recomendación. La siguiente figura muestra todas las tablas de la base de datos

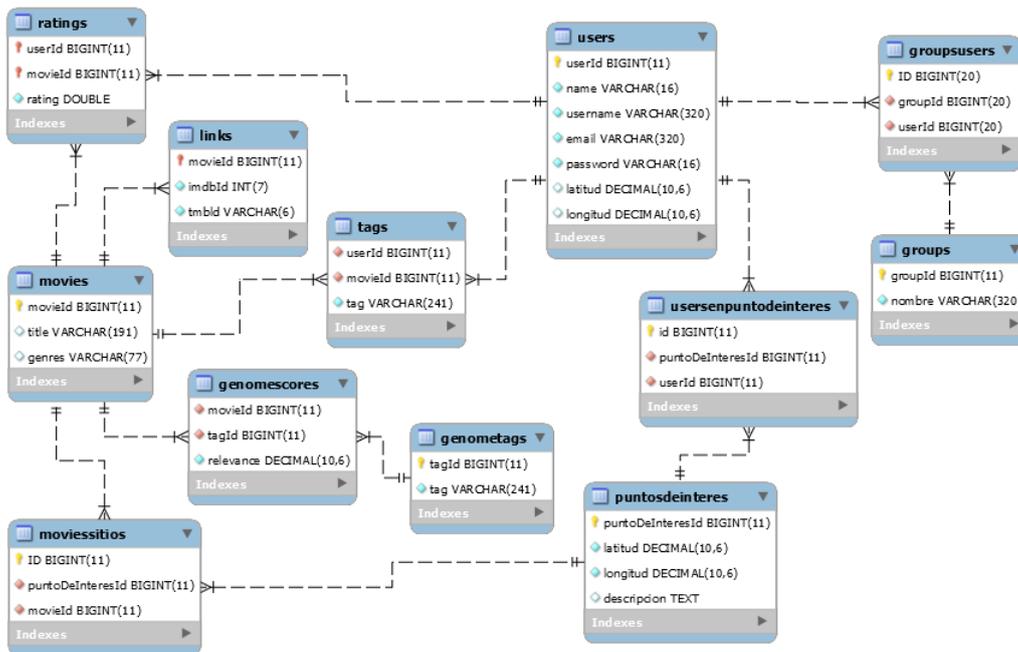


Figura 8: Diseño de la base de datos

4.2.2. Capa de negocio

En este apartado se expondrán las funcionalidades más importantes de la aplicación como son la conexión a la base de datos, el posicionamiento GPS y el algoritmo recomendador.

4.2.2.1. Conexión con la base de datos

En primer lugar, la conexión a la base de datos se hace a través de la clase `ConnectionClass`, como se muestra en la figura 9. Para acceder a la base de datos es necesario indicarle el nombre de la base de datos, el host, el usuario y la contraseña de la misma. Esta clase hace uso de la librería `MySQL Connector/J` que es la api oficial de java para el acceso a base de datos de `MySQL`.

```
public class ConnectionClass implements Serializable{
    String classs = "com.mysql.jdbc.Driver";

    String url = "jdbc:mysql://192.168.1.33/tfg";
    String un = "ride";
    String password = "789";

    @SuppressWarnings("NewApi")
    public Connection CONN() {
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
            .permitAll().build();
        StrictMode.setThreadPolicy(policy);
        Connection conn = null;
        String ConnURL = null;
        try {
            Class.forName(classs);
            conn = DriverManager.getConnection(url, un, password);
            conn = DriverManager.getConnection(ConnURL);
        } catch (SQLException se) {
            Log.e("ERROR", se.getMessage());
        } catch (ClassNotFoundException e) {
            Log.e("ERROR", e.getMessage());
        } catch (Exception e) {
            Log.e("ERROR", e.getMessage());
        }
        return conn;
    }
}
```

Figura 9: Código de acceso a la base de datos

4.2.2.2. Posicionamiento GPS

Para esta parte se ha utilizado un servicio, que es un componente de la aplicación que no requiere de ninguna interfaz, ya que el usuario no interac-

túa con ella [12], por lo que ésta se ejecuta en segundo plano incluso aunque el usuario no esté interactuando directamente con la aplicación.

Para poder calcular la posición GPS de los usuarios, se ha utilizado la API `Android.location`, la cual permite obtener sus posiciones en función de la latitud y la longitud.

```
private class LocationListener implements android.location.LocationListener
{
    Location mLastLocation;

    public LocationListener(String provider)
    {
        Log.e(TAG, "LocationListener " + provider);
        mLastLocation = new Location(provider);
    }

    @Override
    public void onLocationChanged(Location location)
    {
        Log.e(TAG, "onLocationChanged: " + location);

        latitudStr = location.toString().substring(location.toString()
            .lastIndexOf("gps ") + 4, location.toString().lastIndexOf(",") - 3);
        longitudStr = location.toString().substring(location.toString()
            .lastIndexOf(latitudStr) + latitudStr.length() + 1, 32);

        latitudStr = latitudStr.replaceAll(",", ".");
        latitud = Double.parseDouble(latitudStr);

        longitudStr = longitudStr.replaceAll(",", ".");
        longitud = Double.parseDouble(longitudStr);
    }
}
```

Figura 10: Código del servicio encargado de calcular la posición GPS

4.2.2.3. Algoritmo Recomendador

Cuando la posición calculada por la clase anterior coincide con la de un punto de interés se activa la clase del recomendador, clase que se encargará de utilizar el algoritmo diseñado para obtener la película recomendada. Esta recomendación se le mostrará al usuario a través de un `webview`, componente de Android que permite visualizar webs.

```

ArrayList<UsuarioPelículas> películasDelGrupo = new ArrayList<>();

películasDelGrupo = obtenPelículasGrupo(con, userActual);

ArrayList<Integer> películasCineId = películasSitio(con, puntoDeInterésId);

//vectorizamos cada película en función de su género y sus etiquetas
mapa = vectorizarPelículas(películasDelGrupo, películasCineId, con, userActual,
    puntoDeInterésId);

int eleccion = 0;

double base = 1.0;

eleccion = algoritmoRecomendador(mapa, películasDelGrupo, películasCineId );

int imdbId = generaurl(con, eleccion);

WebSettings webSettings = webView.getSettings();
webSettings.setJavaScriptEnabled(true);
webView.setWebViewClient(new WebViewClient());
String url = "http://www.imdb.com/title/tt0"+imdbId;
webView.loadUrl(url);

```

Figura 11: Código de la clase recomendador

Para poder realizar la recomendación de las películas a los grupos de usuarios se ha utilizado un algoritmo recomendador que funciona siguiendo los siete pasos que se explican a continuación.

Primer paso: Al entrar al punto de interés se activa la recomendación y lo primero que se hace es buscar los miembros del grupo a los que pertenece dicho usuario.

Segundo paso: Una vez obtenidos los miembros del grupo, se obtienen las películas vistas por los usuarios cuya calificación dada sea igual o mayor que 4, siendo la calificación más alta posible el 5. Se obtienen únicamente las películas con calificación alta, dado que el algoritmo lo que hará tras realizar los demás pasos, es obtener la película de las ofertadas por el punto de interés que más se asemeje a las películas con las mejores calificaciones dadas por los usuarios del grupo.

Tercer paso: Se obtienen las películas ofertadas por el punto de interés en el que se encuentran los usuarios del grupo.

Cuarto Paso: Una vez obtenidas todas las películas con las que se va a trabajar, el siguiente paso es vectorizarlas. Esto se hace en función de las características de la película, que en este caso, son los posibles géneros y las etiquetas fijas que tienen las películas. Dado que hay 18 posibles géneros y cada película tiene 1128 etiquetas, las películas, pasarán a ser vectores de 1146 elementos. Los primeros 18 elementos corresponderán al género y tomarán los valores 0 o 1 en función de si la película es de dicho género o no. El resto de elementos tomará un valor numérico decimal comprendido entre 0 y 1, el cual indicará la relación de la etiqueta con la película, siendo más próximo a 1 cuando la etiqueta tiene más relevancia en la película y más próximo a 0 cuando la etiqueta tiene menos relevancia con el contenido de la película.

Quinto paso: Tras obtener los vectores de cada película, el siguiente paso es calcular la distancia entre las películas de los usuarios y las del cine pudiendo obtener así cuál es la película ofertada por el punto de interés que más concuerda con las preferencias de los usuarios del grupo. Para calcular la distancia entre estos vectores lo que se hace es utilizar la similitud del coseno, es decir, utilizar la siguiente formula:

$$\vec{v}_1 = (a, b, c) \quad (1)$$

$$\vec{v}_2 = (e, f, g) \quad (2)$$

$$similitud(\vec{v}_1, \vec{v}_2) = \frac{a * e + b * f + c * g}{\sqrt{a^2 + b^2 + c^2} * \sqrt{e^2 + f^2 + g^2}} \quad (3)$$

Esta fórmula permite obtener, por tanto, la distancia entre dos vectores. Cada usuario tendrá un Array del tamaño de las películas ofertadas por el punto de interés, de manera que almacena la distancia que hay entre las películas vistas por cada usuario y las películas ofertadas por el punto de interés. Y para poder entenderlo se ha propuesto un ejemplo en el que se cuenta con un grupo de tres usuarios, y 4 películas ofertadas por el punto de interés, quedando por tanto, 3 arrays de 4 elementos cada uno.

Usuario 1: [0.34, 0.23, 0.65, 0.12]

Usuario 2: [0.21, 0.54, 0.11, 0.34]

Usuario 3: [0.23, 0.41, 0.82, 0.43]

Sexto Paso: Una vez obtenidos estos arrays, el siguiente paso es ordenarlos de menor a mayor y cambiar su valor por el de la id de la película ofertada a la que pertenecen. Suponiendo que las id son 1,2,3 y 4, el ejemplo anterior quedaría de la siguiente forma.

Usuario 1: [4, 2, 1, 3]
Usuario 2: [3, 1, 4, 2]
Usuario 3: [1, 2, 4, 3]

De esta manera se refleja el orden de preferencia de los usuarios ordenado de izquierda a derecha, siendo la que más encaja con el usuario 1 la película 4 y la que menos la película 3.

Séptimo Paso: En este último paso se obtendrá la película más recomendada para los 3 usuarios del ejemplo. Para ello se les asignará un peso a cada película en función de la posición, tomando la película de más a la izquierda del array, es decir, la más recomendable para cada usuario un valor de 4, y la de más a la derecha, es decir, la menos recomendable, un valor de 1. Estos pesos se han establecido en función de la cantidad de películas ofertadas por el cine. En el caso de este ejemplo, la película de id 1 tiene un peso de 9, la película de id 2 tiene un peso 7, la película de id 3 tiene un peso de 6 y la película de id 4 tiene un peso de 8. Por tanto, la película que se le recomendará a los usuarios será la película de id 1.

4.2.3. Capa de presentación

En este apartado se va a explicar primero, el diseño de la interfaz de la aplicación y después se explicará y mostrará la interfaz de usuario utilizada en la aplicación.

El diagrama siguiente muestra el mapa de navegación de los usuarios en la aplicación, es decir, muestra qué pantallas están conectadas entre sí y el recorrido que tendremos que hacer para llegar de una pantalla a otra.

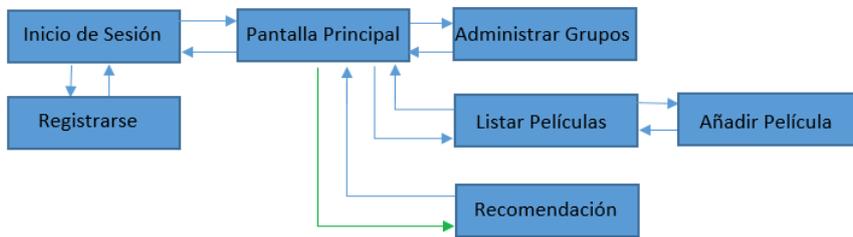


Figura 12: Diseño de la capa de presentación

Las flechas azules indican el recorrido que puede hacer el usuario por voluntad propia, mientras que la flecha verde representa el cambio de pantalla sin intervención directa del usuario. Por lo tanto, esta flecha verde se debe a que la recomendación se activa en función de la posición del usuario, sin que éste interactúe con la aplicación.

La primera pantalla que se muestra al ejecutar la aplicación es el inicio de sesión (Figura 13), donde el usuario tendrá que introducir su nombre de usuario y contraseña para poder acceder al sistema. En caso de no estar registrado, el usuario podrá registrarse seleccionando la opción: “Registrarse Aquí”. Esto abrirá la pantalla de registro en la que el usuario deberá introducir su nombre, dirección de correo electrónico, usuario y contraseña para poder registrarse, tal y como se muestra en la figura 14.



Figura 13: Inicio de sesión



Figura 14: Registrarse

Una vez el usuario registrado inicia sesión, se abre la pantalla principal de la aplicación (figura 15), en la que se muestra el perfil personal del usuario y

los servicios de recomendación a los que está suscrito. Tal y como se mencionó anteriormente sólo es funcional uno de estos servicios de recomendación, que es el de la recomendación de las películas.

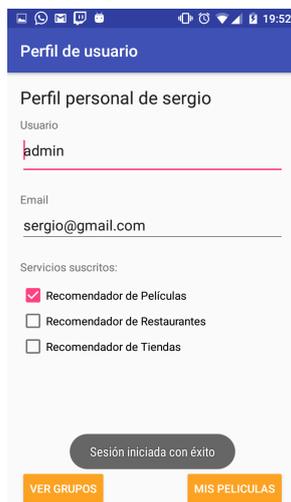


Figura 15: Pantalla principal

Al iniciar sesión, en segundo plano comienzan a cargarse datos, de forma que se agiliza la ejecución del resto de funcionalidades de la aplicación. Además de esta carga de datos, la aplicación comienza a calcular la posición GPS del usuario, que se va comparando con la de los diferentes puntos de interés. De esta forma, al entrar un usuario en uno de estos puntos de interés, la aplicación ejecutará el algoritmo recomendador. Esto sucede incluso aunque el usuario no esté utilizando el teléfono móvil, debido a que los cálculos de recomendación se ejecutan también en segundo plano y sin que el usuario tenga que intervenir, haciendo uso, por tanto, de la computación ubicua.

El algoritmo recomendador se encarga de realizar la recomendación a todos los usuarios del grupo que se encuentren en el punto de interés. Para que la aplicación pueda saber quiénes son los miembros del grupo, desde la pantalla inicial de la aplicación el usuario puede acceder a la gestión de grupos, en donde podrá crear o unirse a un grupo tal y como se muestra en las figuras 16 y 17.



Figura 16: Listado de grupos



Figura 17: Nuevo grupo

Para comprobar el funcionamiento de la aplicación se ha usado el dataset explicado anteriormente, pero dado que estos datos son únicamente para verificar dicho funcionamiento, la aplicación tiene que permitir a los usuarios crear una lista de películas vistas y calificadas. Para ello, los usuarios pueden acceder a la lista de sus películas desde la pantalla inicial. En esta pantalla se les permitirá a los usuarios visualizar la lista de sus películas, así como añadir nuevas películas calificadas a la lista, como puede observarse en las figuras siguientes.

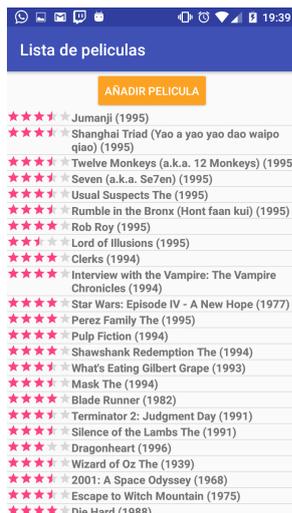


Figura 18: Listado de películas



Figura 19: Añadir película

Para que el usuario pueda añadir correctamente las películas, la aplicación le ayudará mostrando en la pantalla todas las películas que contengan el String escrito, por ejemplo, si el usuario escribe la palabra “Bat” se le mostrarán todas las películas que contengan dicho String (figura 20), facilitando así que el usuario agregue la película que desea a la lista sin errores.

La pantalla que se muestra tras ejecutarse el algoritmo recomendador corresponde a una vista de la página <http://www.imdb.com>, como se puede ver en la figura 21, la cual contiene información acerca de la película recomendada, que puede ser relevante para los usuarios.

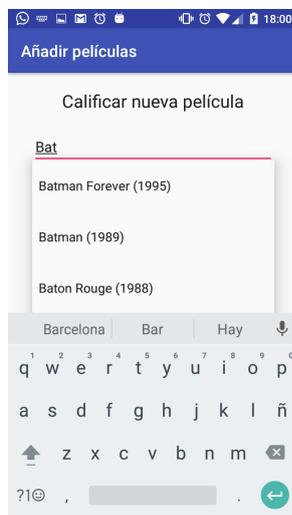


Figura 20: Listado de películas



Figura 21: Añadir película

5. Pruebas Software

En este capítulo se explica en primer lugar los tipos de pruebas existentes en un proyecto software y realizadas a lo largo del desarrollo de este proyecto, y en segundo lugar se explica más detalladamente las pruebas realizadas a las dos funcionalidades más importantes de la aplicación.

5.1. Pruebas de un proyecto software

Cualquier aplicación software requiere que se compruebe el funcionamiento correcto del mismo, para ello se explicará el método seguido en la comprobación de este sistema. Esta comprobación se ha basado principalmente en tres tipos que son primero la prueba de unidad, luego la prueba de integración y por último la prueba de validación y sistema [13].

5.1.1. Pruebas de unidad

Para entender qué son las pruebas de unidad es necesario primero indicar qué es una unidad. En la programación orientada a objetos esta unidad corresponde a una clase, por tanto las pruebas de unidad son aquellas que se hacen para comprobar el correcto funcionamiento de cada una de las clases.

5.1.2. Pruebas de integración

Las pruebas de integración verifican que los componentes que han sido comprobados anteriormente de forma unitaria funcionen también correctamente al unirse entre ellos, es decir, que la interrelación entre las distintas unidades funcione.

5.1.3. Pruebas de validación y sistema

Con estas pruebas se comprueba que el resultado final de la aplicación, una vez realizadas las pruebas de integración, sea el esperado tanto en las acciones visibles para el usuario como a nivel funcional.

5.2. Pruebas realizadas en la aplicación

Para comprobar el correcto funcionamiento de la aplicación ha sido necesario verificar el funcionamiento de las dos partes principales de la aplicación que son el posicionamiento GPS, para saber si el usuario se encuentra en un punto de interés y la recomendación realizada.

5.2.1. Comprobación del punto de interés

Para que la aplicación detecte que el usuario ha entrado en un punto de interés se le ha asignado al punto de interés un pequeño área, de manera que el usuario no tenga que estar justo en un punto sino muy cercano a él. Para comprobar el funcionamiento de este área, y ajustarlo lo mejor posible se ha hecho uso de una aplicación llamada Fake GPS location explicada anteriormente. De esta forma se ha logrado comprobar que al acercarse lo suficiente al punto de interés la aplicación inicia la recomendación.

5.2.2. Comprobación del algoritmo recomendador

Debido a la falta de usuarios reales para comprobar que la recomendación es acertada, se ha decidido realizar las pruebas en función de los datos disponibles. Para esto, se ha elegido extraer un tanto por ciento determinado de las películas vistas por los usuarios de la base de datos existente, que pasarán a formar parte de las películas ofertadas por los puntos de interés, simulando así un escenario real con el que comprobar que la recomendación realizada es la correcta. Si el resultado de las recomendaciones coincide con las películas que mayor puntuación tenían antes de ser extraídas, se considera que la recomendación es válida. Y puesto que en la comprobación realizada en este sistema ha sucedido lo expuesto damos por correcta la prueba.

A pesar de haberse considerado correcta la prueba realizada, no siempre se ha obtenido el resultado esperado y esto se debe a dos motivos. El primero es que no todos los usuarios tienen una cantidad suficiente de películas vistas como para que el algoritmo pueda recomendarles correctamente qué película ver. El segundo motivo es debido a que el hecho de que un usuario concreto haya calificado con una elevada puntuación todas las películas que ha visto de un mismo género, no significa que la siguiente película de ese mismo género tenga que recibir también una puntuación alta.

A continuación se muestra un ejemplo de las pruebas realizadas al algoritmo recomendador. En la siguiente imagen se observa como los usuarios de id 10, 11 y 12 han visto y calificado la película con id 1 de manera positiva.

| + Opciones | | | | userId | movioid | rating |
|--------------------------|--|--|--|--------|---------|--------|
| <input type="checkbox"/> | | | | 10 | 1 | 4 |
| <input type="checkbox"/> | | | | 11 | 1 | 4.5 |
| <input type="checkbox"/> | | | | 12 | 1 | 4 |

Figura 22: Calificación de la película de id 1

La siguiente imagen muestran una película de cada uno de los usuarios del grupo que ha sido calificada con peor nota que la mostrada anteriormente.

| + Opciones | | | | userId | movioid | rating |
|--------------------------|--|--|--|--------|---------|--------|
| <input type="checkbox"/> | | | | 10 | 1204 | 3 |
| <input type="checkbox"/> | | | | 11 | 70305 | 0.5 |
| <input type="checkbox"/> | | | | 12 | 5 | 2 |

Figura 23: Películas con calificaciones bajas

Tras conocer estos datos, se ha procedido a borrarlos de la base de datos con la finalidad de que estos simulen ser películas nuevas para los usuarios, esto implica que las películas de id 1, 5, 1204 y 70305, ya no se encuentran entre las películas vistas por los usuarios 10, 11 y 12. El siguiente paso ha sido introducir las películas eliminadas de la base de datos entre las ofertadas por el punto de interés, tal y como se muestra en la siguiente figura.

| + Opciones | | | | ID | puntoDeInteresId | movioid |
|--------------------------|--|--|--|----|------------------|---------|
| <input type="checkbox"/> | | | | 1 | | 1 |
| <input type="checkbox"/> | | | | 2 | | 5 |
| <input type="checkbox"/> | | | | 3 | | 1204 |
| <input type="checkbox"/> | | | | 4 | | 70305 |

Figura 24: Películas ofertadas por el punto de interés

Para dar por bueno el funcionamiento de la aplicación, el resultado de la recomendación realizada deberá ser la película de id 1, dado que antes de ser eliminada no solo era la película con más calificación dada, sino que además esta puntuación alta era dada por los 3 usuarios del grupo. Tal y como se puede ver en figura 25, la película recomendada es “Toy Story” que coincide con la película de id 1 como se ve en la figura 26.

| movieid | title | genres |
|---------|--------------------------|---|
| 1 | Toy Story (1995) | Adventure Animation Children Comedy Fantasy |
| 2 | Jumanji (1995) | Adventure Children Fantasy |
| 3 | Grumpier Old Men (1995) | Comedy Romance |
| 4 | Waiting to Exhale (1995) | Comedy Drama Romance |

Figura 25: Identificadores de las películas



Figura 26: Película recomendada

6. Conclusiones y trabajos futuros

En este capítulo se comentará en primer lugar, si se han conseguido realizar aquellos objetivos marcados al principio del proyecto, y en segundo lugar, los posibles trabajos futuros para el mismo.

6.1. Conclusiones

Para extraer las conclusiones finales del trabajo, conviene recordar cuáles eran aquellos objetivos que se habían planteado al comienzo del proyecto. El objetivo principal era crear una aplicación móvil, que haciendo uso de un algoritmo recomendador fuera capaz de recomendar a los usuarios en función de la posición en la que se encuentren.

Para lograr este objetivo, el primer paso realizado fue cumplir el subobjetivo de crear una base de datos que contuviera toda la información necesaria para la aplicación. La creación de esta base de datos resultó ser más difícil de lo que se esperaba en un principio, debido a que el dataset utilizado para crear la misma contenía dos ficheros de más de 0,5GB, llegando a ocupar, por tanto, más de 1GB en total. El gran tamaño del dataset ha supuesto un problema dado que la mayoría de servidores de hosting online gratuitos no ofrecen más de 1GB de espacio, por lo que se tuvo que descartar la opción de utilizarlos. Por este motivo se decidió crear la base de datos haciendo uso de XAMPP y de su gestor de MySQL llamado phpMyAdmin. Este gestor resultó tener limitaciones para importar ficheros grandes, por lo que, pese a modificar ficheros propios de la configuración del gestor phpMyAdmin, no se lograron importar los ficheros. Finalmente se consiguió importar los ficheros haciendo uso de la consola de Windows mediante la cual se puede acceder al servidor MySQL de XAMPP.

```
mysql> load data infile 'C:\\ratings2.csv' into table ratings fields terminated by ','
Query OK, 20000263 rows affected, 65535 warnings (7 hours 21 min 59.77 sec)
Records: 20000263 Deleted: 0 Skipped: 0 Warnings: 0
```

Figura 27: Importación del fichero ratings2.csv

Tras crear esta base de datos, se comenzó a desarrollar la aplicación Android haciendo uso de la herramienta Android Studio, mediante la cual se ha conseguido alcanzar el resto de subobjetivos propuestos, siendo uno de estos subobjetivos el calcular la posición GPS de los usuarios. Para cumplir este subobjetivo se hace uso de la API Android.Location de Android, la cual nos permite acceder a los datos de posicionamiento del teléfono móvil. Además,

para que la aplicación fuera capaz de calcular la posición mientras se realizan otras tareas, se ha hecho uso de servicios que, como se explicó anteriormente, son componentes de Android que permiten que estos cálculos se realicen en segundo plano y sin necesidad de una interfaz.

Otro de los subobjetivos que se planteó al comienzo del proyecto, es el de realizar una interfaz gráfica intuitiva y atractiva para los usuarios. Esta interfaz se ha diseñado utilizando Android Studio, herramienta que ha permitido crear fácilmente la interfaz dado que permite visualizarla directamente tal y como quedaría en el teléfono móvil. Se considera que la aplicación es atractiva, dado que se ha usado una temática de colores, tipografía de letra y tamaño de la misma que no dificulta la lectura de los elementos que componen la interfaz. Y se considera intuitiva dado que no contiene ningún elemento que requiera de una explicación anterior para poder ser utilizado, además de informar en todo momento al usuario en caso de error. El informar al usuario se hace a través de mensajes temporales que le indican cuál es el error cometido, por ejemplo, en caso de que el usuario esté introduciendo mal el nombre de usuario o la contraseña para acceder a la aplicación, se le mostrará un mensaje que le indicará el motivo por el que no puede acceder al sistema.

El último de los subobjetivos era desarrollar un algoritmo recomendador, objetivo que se ha logrado cumplir tal y como se demuestra en el capítulo 6 en el que se comentan las pruebas realizadas para verificar el correcto funcionamiento del algoritmo.

Para finalizar y a modo de conclusión personal, el realizar este proyecto me ha ayudado a la hora de formarme como ingeniero informático, debido principalmente a que, con este proyecto, he podido unificar los conocimientos adquiridos a lo largo de la carrera en diferentes asignaturas. Además, he aprendido a diseñar un proyecto software completo, pasando por todas y cada una de las etapas de vida que tiene la realización de un proyecto de este nivel. Por último, y no menos importante, debido a la proyección que este conocimiento me puede suponer como profesional, me ha servido también este proyecto para introducirme en el mundo de las aplicaciones móviles a través del aprendizaje de la programación Android, de la que tenía un total desconocimiento antes de comenzar a realizar el proyecto.

6.2. Trabajos futuros

Esta aplicación se presenta como un algoritmo recomendador enfocado al ámbito del ocio, pero tal y como se comenta al principio de la memoria, la aplicación se ha focalizado en la recomendación de películas, por lo que uno de los trabajos futuros para este proyecto es el de ampliar la cantidad de servicios de recomendación que ofrece la aplicación.

Otra de las mejoras que sería interesante realizarle al proyecto, consistiría en trasladar el sistema de recomendación a un servidor fuera de la aplicación móvil. Esto se traduce en una mejora de la eficiencia de la aplicación, debido a que pasaría a ser el servidor el que ejecute el algoritmo recomendador puesto que tiene mayor capacidad de cómputo que un teléfono móvil. Este cambio, a su vez, podría ser necesario para realizar la ampliación de los servicios de recomendación que se ha comentado, dado que dicha ampliación podría ralentizar considerablemente el funcionamiento de la aplicación.

Además, también sería muy interesante desarrollar esta aplicación para dispositivos móviles con otros sistemas operativos como iOS, pudiendo así ampliar el número de usuarios al que va destinada la aplicación.

Referencias

- [1] Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In Recommender systems handbook (pp. 1-35). springer US.
- [2] Velez-Langs, O., and Santos, C. (2006). Sistemas recomendadores: Un enfoque desde los algoritmos genéticos. *Industrial data*, 9(1).
- [3] Jiménez, A. R., Seco, F., Prieto, C., and Roa, J. (2005). Tecnologías sensoriales de localización para entornos inteligentes. In I Congreso español de informática-Simposio de Computación Ubicua e Inteligencia Ambiental, UCAMI2005 (pp. 75-86).
- [4] MySQL. (2017). Mysql.com. Consultado 19 oct. 2017. Disponible en <https://www.mysql.com/>
- [5] contributors, p. (2017). phpMyAdmin. phpMyAdmin. Consultado 19 oct. 2017. Disponible en <https://www.phpmyadmin.net>
- [6] Download MySQL Workbench. (2017). Dev.mysql.com. Consultado 19 oct. 2017. Disponible en <https://dev.mysql.com/downloads/workbench/>
- [7] Download Android Studio and SDK Tools | Android Studio. (2017). Developer.android.com. Consultado 19 oct. 2017. Disponible en <https://developer.android.com/studio/index.html>
- [8] XAMPP Installers and Downloads for Apache Friends. (2017). Apachefriends.org. Consultado 19 oct. 2017. Disponible en <https://www.apachefriends.org/es/index.html>
- [9] Lexa, (2016). Fake GPS location (2.0.6) [Aplicación Móvil]. Recuperado de <https://play.google.com/store/apps/details?id=com.lexa.fakegps>
- [10] MovieLens. (2017). Movielen.org. Consultado 21 oct. 2017. Disponible en <https://movielens.org>
- [11] Roa, B. (2017). Desarrollo en Cascada Vs. Desarrollo Iterativo e Incremental. *Iscocom.blogspot.com.es*. Consultado 21 oct. 2017. Disponible en <http://iscocom.blogspot.com.es/2013/08/desarrollo-en-cascada-vs-desarrollo.html>
- [12] Service. (2017). Consultado 18 oct. 2017. Disponible en <https://developer.android.com/reference/android/app/Service.html>
- [13] Pressman, R. S., and Troya, J. M. (1988). *Ingeniería del software*.