

Índice

CAPÍTULO 1	3
PREFACIO	3
1.1 Introducción al proyecto	4
1.2 Propósito	6
1.3 Objetivos	6
CAPÍTULO 2	9
ANTECEDENTES	9
2.1 Sistemas de Recomendación	10
2.1.1 Clasificación de los Sistemas de Recomendación.....	11
2.1.2 Utilidad de los Sistemas de Recomendación.....	16
2.1.3 Realimentación en Sistemas de Recomendación.....	17
2.1.4 Elección del conjunto de datos para Sistemas de Recomendación.....	18
2.1.5 Ejemplos reales de Sistemas de Recomendación.....	19
2.2 Sistemas de Recomendación Colaborativos	22
2.2.1 Introducción a los Sistemas de Recomendación Colaborativos.....	22
2.2.2 Categorías de los algoritmos de Filtrado Colaborativo.....	24
2.2.3 Ejemplos reales de Sistemas de Recomendación Colaborativos.....	25
CAPÍTULO 3	35
PROYECTO	35
3.1 Revisión de Algoritmos de Filtrado Colaborativo	37
3.1.1 Notación.....	37
3.1.2 Algoritmo K-nn.....	38
3.1.3 Medidas de similaridad.....	40
3.2 Algoritmos de predicción basados en ítem	43
3.2.1 Algoritmo Básico de Filtrado Colaborativo.....	43
3.2.2 Mejoras sobre los Algoritmos de Filtrado Colaborativo.....	45
3.3 Estudio Comparativo	50
3.3.1 Pruebas sobre Algoritmos Básicos.....	55
3.3.2 Pruebas con Algoritmos de Filtrado Colaborativos Mejorados.....	77
3.3.3 Comparativa Final.....	99
CAPÍTULO 4	101
SISTEMA DE RECOMENDACIÓN COLABORATIVO MOVIESRECOMMENDER II	101
4.1 Especificación de Requerimientos	103
4.1.1 Requerimientos funcionales.....	104
4.1.2 Requerimientos no funcionales.....	107
4.2 Análisis del Sistema	110
4.2.1 Casos de uso.....	111
4.2.2 Escenarios.....	123
4.3 Diseño del Sistema	125
4.3.1 Diseño de los datos.....	125
4.3.2 Diseño de la interfaz.....	132
4.4 Implementación	136
4.4.1. Tipo de arquitectura de la aplicación.....	136
4.4.2. Lenguajes de programación utilizados.....	137
4.4.3 Herramienta de desarrollo.....	139
4.4.4 Actualización del algoritmo de filtrado.....	139
4.4.5 Instalación en el servidor y funcionamiento de MoviesRecommender II.....	140
CAPÍTULO 5	141
CONCLUSIONES	141

ANEXO I	145
MANUAL DE INSTALACIÓN DEL SERVIDOR	145
Material necesario	146
Paso 1: Instalar Apache	146
Paso 2: Instalar PHP	155
Paso 3: Configurar Apache y PHP	155
Paso 4: Descomprimir archivos	157
Paso 5: Conectar base de datos.....	158
ANEXO II	163
MANUAL DE USUARIO	163
Usuario no registrado	165
Usuario registrado	169
BIBLIOGRAFÍA	179
1. Bibliografía específica sobre Sistemas de Recomendación Colaborativos.....	180
2. Bibliografía general para la realización del proyecto	182

CAPÍTULO 1.

PREFACIO.

1.1 Introducción al proyecto

El número de empresas que ofrecen sus productos o servicios en Internet se multiplica cada año. Esta realidad indiscutible ha hecho que se produzca una tremenda competencia cada vez más encarnizada para asegurar su supervivencia. Debido a esta competencia, las empresas deben ofertar una serie de servicios diferenciadores que les permitan no sólo mantener su clientela sino atraer nuevos clientes desde los competidores.

Una de las estrategias más interesantes para conseguir esto es la de desarrollar servicios de marketing personalizado basados en **sistemas de recomendación**. Estos sistemas se encargan de suministrar a los usuarios información personalizada y diferenciada sobre determinados productos y/o servicios que pueden ser de interés para ellos. Es decir, se encargan de guiar a un usuario mediante recomendaciones en la búsqueda de aquellos servicios o productos que puedan ser más atractivos para él, modificando el proceso de navegación y búsqueda. Esto es sin duda una gran ventaja para los clientes, que encontrarán lo que necesitan de una forma más rápida, cómoda y fácil dentro de las enormes bases de datos que ofertan las tiendas electrónicas en Internet y además descubrirán nuevos productos o servicios que le puedan ser atractivos, que de otra manera les hubiese sido mucho más difícil o incluso imposible de encontrar.

Existen diversos tipos de sistemas de recomendación, siendo los dos más importantes y utilizados los siguientes:

1. Sistemas de recomendación basados en contenido

Se basan en la similitud entre objetos, es decir, predicen que para un usuario serán de interés aquellos objetos muy parecidos en su contenido con aquellos que ya sabemos que han sido de su agrado [1].

2. Sistemas de recomendación colaborativos

Son más cercanos a la forma de pensar de los seres humanos que los basados en contenido. Son aquellos en los que las recomendaciones se realizan basándose solamente en los términos de similitud entre los gustos de los usuarios [1].

El objeto de este proyecto es el estudio y desarrollo de un **Sistema de Recomendación Colaborativo**. Se podría decir que el funcionamiento de este tipo de sistemas de

recomendación sigue los siguientes pasos [13]:

1. El sistema guarda un perfil de cada usuario que consiste en las evaluaciones del mismo sobre objetos conocidos por él y que pertenezcan a la base de datos de productos a recomendar.
2. Se mide el grado de similitud entre los diferentes usuarios del sistema en base a sus perfiles y se crean grupos de usuarios con características afines.
3. El sistema utilizará toda la información obtenida en los pasos anteriores para realizar las recomendaciones. A cada usuario se le recomendarán objetos que no haya evaluado y que hayan sido evaluados de manera positiva por los miembros de su grupo afín.

Por lo tanto, este tipo de sistema no tiene en consideración el contenido y las características de los productos que se recomiendan sino que sean del gusto de usuarios con gustos semejantes al usuario que demanda el usuario de este sistema.

Una vez comentados brevemente la situación actual del mercado de Internet y las características de los novedosos y útiles sistemas de recomendación podemos pasar a introducir brevemente en qué consiste el proyecto que se documenta en esta memoria.

En este proyecto se pretende realizar un sistema de recomendación con filtrado colaborativo que se aplicará al ámbito de la recomendación en tiendas de alquiler o venta de películas o videoclubs. Para ello utilizaremos los datos que proporciona la web <http://www.grouplens.org>, que se han utilizado de base para realizar estudios de investigación en el ámbito de los sistemas de recomendación colaborativos.

Estos datos están en formato texto por lo que nuestro primer paso será formatearlos hasta convertirlos a un formato que nos sea de utilidad. Nosotros hemos elegido un formato de base de datos **Microsoft Access** aunque hubiera resultado igualmente válido para nuestro propósito cualquier otro formato de base de datos manejable mediante sentencias **SQL** como MySQL o similares.

Una vez hemos dado el formato a nuestra base de datos realizamos un estudio

comparativo de distintos tipos de algoritmos básicos de filtrado colaborativo, así como de mejoras propuestas en la literatura sobre los mismos, para observar cuál o cuales de las propuestas existentes nos proporcionarán mejores resultados sobre los datos que tenemos para el sistema que pretendemos realizar.

Una vez realizado este estudio comparativo elegiremos el mejor de estos algoritmos y lo utilizaremos para implementar un sistema de recomendación con una arquitectura cliente/servidor y con una interfaz web que lleva por nombre **MoviesRecommender II** y de la que se propondrá una versión prototipal o beta.

Por último se expondrán una serie de conclusiones tanto sobre los sistemas de recomendación en general como sobre el desarrollado para este proyecto y adjuntaremos diversos manuales e información que consideramos útil para el correcto entendimiento y funcionamiento de los modelos y aplicaciones desarrolladas en este proyecto. Finalmente, para aquellos lectores que estén interesados en el tema, se incluirá una amplia bibliografía.

1.2 Propósito

Hacer un estudio comparativo sobre algoritmos de filtrado colaborativo y mejoras aplicadas a los mismos. De esta manera, atendiendo a los resultados obtenidos de dicho estudio crear un sistema de recomendación colaborativo basado en técnicas de inteligencia artificial para la gestión del alquiler de películas, que utilice el algoritmo que muestre un mejor comportamiento del estudio realizado.

1.3 Objetivos

1. Búsqueda y revisión bibliográfica.
2. Crear una base de datos de películas disponibles a partir de los datos en <http://www.grouplens.org> para poder desarrollar correctamente las fases siguientes.
3. Fijar un algoritmo de filtrado colaborativo básico y posibles mejoras para su posterior implementación y estudio en procesos de recomendación sobre productos de la base

de datos obtenida en 2.

4. Desarrollo de un proceso de extracción de conocimiento que permita predecir las valoraciones de un usuario con arreglo a las hechas por los miembros más similares al mismo.
5. Estudio de la efectividad de los distintos algoritmos de filtrado colaborativo y de los modelos de predicción mediante varios procesos de *hold out*.
6. Implementación de un sistema de recomendación basado en los resultados obtenidos en los puntos anteriores con una arquitectura cliente/servidor y una web amigable que permita al usuario interactuar con facilidad con el sistema.

CAPÍTULO 2.

ANTECEDENTES.

2.1 Sistemas de Recomendación

El ser humano necesita información para tomar decisiones de cualquier tipo pero muchas veces se encuentra con que la información que tiene disponible es demasiado amplia o inconexa, tiene una sobrecarga de información y no es capaz de extraer la que es verdaderamente relevante. Es necesario filtrar esta información relevante diseminada en grandes volúmenes de información y en esta situación es donde entran con fuerza para ayudar al usuario los sistemas de recomendación.

Una definición formal de sistema de recomendación es la siguiente: *se trata de aquel sistema que tiene como principal tarea seleccionar ciertos objetos de acuerdo a los requerimientos del usuario* [7].

Otra definición de un sistema de recomendación podría ser: *el sistema que utiliza las opiniones de los usuarios de una comunidad para ayudar a usuarios de esa comunidad a encontrar contenidos de su gusto entre un conjunto sobrecargado de posibles elecciones* [7].

Pero los sistemas de recomendación no se basan en ningún modelo novedoso sino que lo que hacen es algo que existe desde que el ser humano tiene consciencia e inteligencia: pedir consejo o recomendación a expertos en la materia o seguir a aquellos individuos que tienen gustos similares al del usuario, o bien seleccionar objetos que tienen características similares a objetos que le hayan gustado anteriormente o que se parecen al que inicialmente buscaba. Estas dos tendencias milenarias han dado lugar a las dos ramas principales (aunque no las únicas) de los sistemas de recomendación: los colaborativos y los basados en contenido, respectivamente.

Tanto los sistemas de recomendación colaborativos como los basados en contenido necesitan una enorme cantidad de información sobre usuarios y objetos para poder realizar unas recomendaciones de calidad [6]. Debido a esta circunstancia han surgido otros tipos de sistemas de recomendación que pueden trabajar y ofrecer recomendaciones de calidad sin necesitar una cantidad de información tan grande, como los sistemas de recomendación basados en conocimiento. También existen otros sistemas de recomendación: demográficos y los basados en utilidad que explicaremos en el apartado siguiente. Además, en los últimos tiempos se han desarrollado sistemas de recomendación híbridos, los cuales recogen los mejores aspectos de dos o más tipos de sistemas de recomendación para conseguir unos

resultados todavía mejores a la hora de realizar sus recomendaciones.

2.1.1 Clasificación de los Sistemas de Recomendación

Los sistemas de recomendación se clasifican atendiendo a su funcionamiento, dando lugar a varios tipos de sistemas de recomendación:

1. Sistemas de Recomendación Basados en Contenido

Un sistema de recomendación basado en contenido es aquel en el cual las recomendaciones son realizadas basándose en un perfil creado a partir del análisis del contenido de los objetos que el mismo usuario ha comprado, utilizado o visitado en el pasado [1].

En otras palabras: extraen características de los objetos no conocidos aún por el usuario actual y las comparan con el perfil del mismo para predecir sus preferencias sobre tales objetos. Lo que se pretende es recomendar objetos muy similares en su contenido a objetos que ya sabemos que son del agrado del usuario en cuestión, o sea, los que forman parte de su perfil.

El filtrado basado en contenido era el más extendido de los sistemas de recomendación hasta la explosión definitiva del filtrado colaborativo pero tiene un claro y, en muchos casos, grave problema como es la **sobre-especialización**. Esta sobre-especialización se da al reducir las recomendaciones a unos contenidos muy similares sin tener en cuenta la posible arbitrariedad de los gustos e intereses de los usuarios. Otro problema de los sistemas de recomendación basados en contenido es que de un objeto sólo se puede conocer una información parcial, normalmente textual, mientras que la información contextual, visual o semántica es más difícil de conocer y por lo tanto se pierden conexiones entre objetos similares de manera menos obvia.

Se han intentado múltiples soluciones para estos problemas como la incorporación de una cierta aleatoriedad a las búsquedas, la indexación semántica latente (LSI) de la información textual [5] o las medidas de similitud basadas en ontologías pero sin duda la mejor solución a tales problemas es que exista una buena retro-alimentación entre el sistema y sus usuarios.

2. Sistemas de Recomendación Colaborativos

Los sistemas de recomendación basados en un filtrado colaborativo son aquellos en los que las recomendaciones se realizan basándose solamente en los términos de similitud entre los usuarios [1]. Es decir, los sistemas colaborativos recomiendan objetos que son del gusto de otros usuarios de intereses similares.

Para la realización de un buen sistema de recomendación colaborativo (es decir, un sistema que ofrezca recomendaciones de calidad) es necesario utilizar un buen algoritmo de filtrado colaborativo. Estos algoritmos se pueden encuadrar dentro de dos categorías: los algoritmos basados en memoria o usuario y los basados en modelos o ítem.

Conforme la utilización de estos sistemas de recomendación se ha ido popularizando se han ido descubriendo una serie de problemas como son la escasez, la escalabilidad y el problema del ítem nuevo [4]. Multitud de estudios y experimentos se han llevado a cabo en los últimos tiempos con la intención de minimizar estos problemas.

Este tipo de sistemas de recomendación son el eje sobre el que gira este proyecto por lo que se merecen un estudio más detallado en un epígrafe posterior.

3. Sistemas de Recomendación Basados en Conocimiento

Los sistemas de recomendación basados en conocimiento realizan inferencia entre las necesidades y preferencias de cada usuario para sugerir recomendaciones [6].

A diferencia de otros sistemas de recomendación, los basados en conocimiento no dependen de grandes cantidades de información sobre objetos puntuados (basados en contenido) y usuarios particulares (colaborativos) sino que lo único que necesitan es tener un conocimiento general sobre el conjunto de objetos y un conocimiento informal de las necesidades del usuario.

El principal problema de estos sistemas de recomendación es que aunque no requieren mucha información sí que requieren un gran esfuerzo humano para realizar las recomendaciones mediante todo tipo de heurísticas de inferencia.

4. Sistemas de Recomendación Demográficos

Los sistemas de recomendación demográficos tienen como objetivo clasificar al usuario en función de sus características demográficas, realizando a continuación las recomendaciones basándose en clases demográficas. Un primer ejemplo de este tipo de recomendación lo constituía Grundy [14], que era un sistema que recomendaba libros basándose en la información personal que se almacenaba en el sistema a través de un dialogo interactivo. Se buscaba la correspondencia entre las respuestas de los usuarios en este dialogo y una biblioteca de estereotipos de usuario, que había sido compilada de manera manual. Otros sistemas de recomendación mas recientes también hacen uso de este tipo de técnicas; por ejemplo, en [15], se usan grupos demográficos para llevar a cabo una investigación de marketing que permite sugerir una serie de productos y servicios; la clasificación del usuario en un determinado grupo demográfico se realiza mediante una pequeña encuesta. En otros sistemas, se utilizan métodos de aprendizaje en máquinas para clasificar a los usuarios basándose en sus datos demográficos [16].

La representación de la información demográfica en un modelo de usuario puede variar considerablemente; así, Grundy [14] usaba características de los usuarios que se anotaban manualmente con unos determinados intervalos de confianza, y ahora sin embargo existen técnicas demográficas que realizan “correlaciones persona a persona”, de manera similar a como lo hace el filtrado colaborativo pero con distintos datos. El beneficio de la aproximación demográfica radica en que puede no necesitar un histórico de datos de usuario, contrariamente al caso del filtrado colaborativo, y como hemos visto, a las técnicas basadas en contenido.

En cuanto a problemas y requisitos en este tipo de sistemas de recomendación merece la pena comentar que es difícil recoger los datos demográficos necesarios, porque las personas son reticentes a dar la información personal a un sistema, además es un sistema no anónimo, por lo que conlleva problemas de privacidad. Se necesita investigación estadística y/o social para saber cómo traducir los grupos culturales de la persona a las necesidades informativas

5. Sistemas de Recomendación Basados en Utilidad

Los recomendadores basados en utilidad recomiendan utilizando el cálculo de la utilidad de cada uno de los servicios para el usuario. Evidentemente, el problema clave a resolver aquí es

cómo crear una función que defina la utilidad para cada usuario y que después pueda ser empleada de manera adecuada para la recomendación [17]. El beneficio de las recomendaciones basadas en utilidad viene del hecho de que puede tener en cuenta para el cálculo de la utilidad algunas características que no están estrictamente relacionadas con los servicios ofrecidos, como por ejemplo, la confianza en el vendedor o la disponibilidad del producto, siendo posible llegar a soluciones de compromiso, por ejemplo entre precio y plazo de entrega para un usuario que tiene una necesidad inmediata.

Para los sistemas de recomendación basados en utilidad también existen una serie de inconvenientes tales como que requieren que el usuario defina la función de utilidad que debe ser satisfecha, puesto que requieren que el usuario tenga en cuenta todas las cualidades del dominio. Podemos añadir que estos son sistemas estáticos y que no pueden aprender o mejorar sus recomendaciones como pueden hacer otros sistemas. Tampoco pueden adaptarse al usuario individual o a los dominios cambiantes

6. Híbridos

Todos los sistemas de recomendación vistos hasta ahora tienen sus puntos fuertes y sus talones de Aquiles por lo que es lógico pensar en intentar maximizar sus bonanzas y minimizar sus puntos débiles mediante la hibridación de dos o más técnicas de recomendación.

Los sistemas híbridos entre los basados en contenido y los colaborativos guardan las preferencias del usuario y las combinan con los objetos más relevantes para realizar las recomendaciones [6].

También existen los sistemas híbridos entre los basados en conocimiento y los colaborativos, los basados en contenido y los basados en conocimiento e incluso entre los colaborativos y las redes sociales.

Comparación entre las técnicas de recomendación

Todas las técnicas de recomendación tienen sus puntos fuertes y débiles. De entre todos los problemas, podemos destacar quizás como el más importante el problema de la falta de

datos cuando se empieza a utilizar el sistema, conocido en la literatura como el problema del ramp-up [18]. Este término realmente se refiere a dos problemas diferentes, aunque relacionados:

- **Nuevos usuarios:** debido a que las recomendaciones son el resultado de la comparación entre el usuario objetivo y otros usuarios, basándonos solamente en anteriores interacciones de los usuarios con el sistema, un usuario del que se disponen pocos datos resulta difícil de clasificar, y por tanto, de recomendar.
- **Nuevos productos:** análogamente, un nuevo producto del que no se dispone todavía de suficientes datos de acceso de los usuarios a éste, puede ser complicado de recomendar.

En la Tabla 2.1 se resumen las ventajas e inconvenientes de cada una de las cinco técnicas de recomendación explicadas. La notación seguida en esta tabla es la siguiente:

- **A:** Capacidad para identificar usuarios similares que acceden a servicios de diferente tipo; en otras palabras, capacidad para identificar usuarios similares a pesar de que puedan parecer heterogéneos analizando estrictamente los accesos a productos que realizan.
- **B:** No necesita conocimiento del campo que tratan los servicios.
- **C:** Adaptabilidad, es decir, capacidad de mejorar con el tiempo.
- **D:** El sistema es capaz de autorealimentarse.
- **E:** No presenta problemas de ramp-up.
- **F:** Sensibilidad a cambios en las preferencias.
- **G:** Puede incluir características que no tienen que ver, estrictamente, con el producto a recomendar.
- **H:** Capacidad de relacionar las necesidades de los usuarios con los productos a recomendar.
- **I:** Problema de ramp-up para nuevos usuarios.
- **J:** Problema de ramp-up para nuevos servicios.
- **K:** No funciona bien para usuarios cuyo comportamiento pueda considerarse como no nítido, en el sentido de que pertenece a dos grupos de usuarios distintos, cada uno de los cuales define un claro comportamiento.
- **L:** Depende de disponer de un histórico de datos grande.
- **M:** Problema de la “estabilidad-plasticidad” que consiste en adaptarse ante los datos relevantes y mantenerse estable ante datos irrelevantes.
- **N:** Debe obtener información demográfica.
- **O:** El usuario debe introducir una función de utilidad.

- **P:** No posee capacidad de aprendizaje.
- **Q:** Se necesita conocimiento de cualquier tipo (por parte del producto, funcional o por parte del usuario).

Como se puede observar en la Tabla 2.1, las técnicas colaborativas y demográficas son las únicas que tienen la capacidad para identificar y, por tanto, recomendar con éxito a usuarios heterogéneos; en otras palabras, son capaces de encontrar similitudes entre usuarios más allá de que accedan, o no, a los mismos productos. Las técnicas basadas en conocimiento también pueden tener esta capacidad, aunque a expensas de poseer el conocimiento adecuado para ello.

TÉCNICA	VENTAJAS	INCONVENIENTES
Colaborativa	A, B, C, D	I, J, K, L, M
Basada en contenido	B, C, D	I, L, M
Demográfica	A, B, C	I, K, L, M, N
Basada en utilidad	E, F, G	O, P
Basada en conocimiento	E, F, G, H	P, Q

Tabla 2.1 Comparativa de las diferentes técnicas de recomendación.

2.1.2 Utilidad de los Sistemas de Recomendación

Los sistemas de recomendación resultan de vital importancia para el marketing personalizado ya que reducen el tiempo de búsqueda de los productos, consiguen una mayor efectividad en las búsquedas y, por lo tanto, una mayor satisfacción en los clientes.

Para lograr estos objetivos, todos los sistemas de recomendación llevan a cabo dos tareas [8]:

- **Predecir:** los sistemas de recomendación predicen una serie de objetos, servicios o productos en los que un usuario o cliente particular podría estar interesado.

- **Recomendar los N-mejores objetos:** los sistemas de recomendación identifican los N objetos en los que el usuario estaría más interesado.

2.1.3 Realimentación en Sistemas de Recomendación

Un sistema de recomendación no debe ser una entidad estática sino evolucionar en el tiempo en cuanto a la calidad de sus recomendaciones y pronósticos en base a la experiencia y nueva información adquiridas. Para conseguir este objetivo se utilizan mecanismos de realimentación entre el sistema y los gustos de los usuarios. Existen dos tipos de mecanismos de realimentación: los implícitos y los explícitos.

1. Realimentación implícita

Un mecanismo de realimentación implícito es aquel que proporciona información al sistema de recomendación acerca de los gustos de los usuarios sin que éstos sean conscientes de esta situación. Por lo tanto este tipo de realimentación no es directa sino que se realiza mediante procesos de data mining usando diversos tipos de medidas como pueden ser el tiempo de visualización del objeto, el número de veces que el objeto es solicitado, etc.

Esta realimentación implícita tiene el problema de depender en demasía del contexto y de ser excesivamente hipotética (podemos suponer que solicitar la visualización de un objeto muchas veces indica un especial interés por parte del usuario pero no tiene porque ser de esa manera) por lo que no resulta ser la más apropiada para todas las situaciones de recomendación.

2. Realimentación explícita

Un mecanismo de realimentación explícito es aquel basado en la acción directa por parte del usuario para indicar que objetos determinados del sistema son de su interés. Esta interacción directa se puede realizar mediante votaciones numéricas o, más sencillo aún, que el usuario diga si el objeto es o no de su agrado.

Este mecanismo tampoco se encuentra exento de problemas como pueden ser la

voluntariedad del cliente o el tiempo consumido.

2.1.4 Elección del conjunto de datos para Sistemas de Recomendación

La elección de un adecuado conjunto de datos es algo primordial para evaluar la calidad de un sistema de recomendación. Para que esta elección resulte la adecuada hay que contestar a las siguientes dos preguntas [7]:

1. ¿Análisis online u offline?

Es importante decidir como se va a trabajar sobre los datos: si de manera online u offline. En el análisis offline se usa una técnica o algoritmo para predecir ciertos valores retenidos de un conjunto de datos y los resultados son analizados mediante una o varias métricas de error. Este análisis offline tiene la ventaja de ser rápido y económico pero también tiene dos desventajas importantes: el problema de la escasez de datos y el problema de obtener sólo como resultado la bondad de la predicción.

Por contra, el análisis online permite obtener otros resultados como son la actuación de los usuarios participantes, su satisfacción o su participación. En su defecto es más lento y caro que el análisis offline.

2. ¿Datos reales o sintetizados?

Otra elección importante es elegir entre un conjunto de datos reales (recopilados de usuarios reales sobre objetos reales) o un conjunto de datos sintetizados (creados específicamente para el sistema de recomendación, sin ninguna base real). Los datos sintéticos son más fáciles de crear que los reales ya que no hay que realizar encuestas ni otros métodos para conseguirlos del mundo real pero sin embargo es recomendable usar estos últimos y sólo utilizar los sintetizados en las primeras fases de desarrollo del sistema, siendo sustituidos por los reales cuando haya un número importante de éstos recopilados.

2.1.5 Ejemplos reales de Sistemas de Recomendación

En este epígrafe vamos a hacer un repaso a distintos sistemas de recomendación (salvo los colaborativos que se verán en su propio epígrafe) que se pueden encontrar en el mercado.

a) Amazon (<http://www.amazon.com>)



Figura 2.1. *Página personalizada en Amazon*

La poderosa empresa norteamericana de comercio electrónico que empezó siendo una librería online es un ejemplo paradigmático de sistema de recomendación híbrido que **mezcla los enfoques basado en contenido y colaborativo**. El sistema guarda las preferencias del usuario activo y las combina con objetos relevantes para generar recomendaciones (las ya celebres páginas "*People who bought this item... also bought these items...*" como la de la figura que acompaña este párrafo).

b) IMDB Recommendation Center (<http://spanish.imdb.com/Sections/Recommendations>)

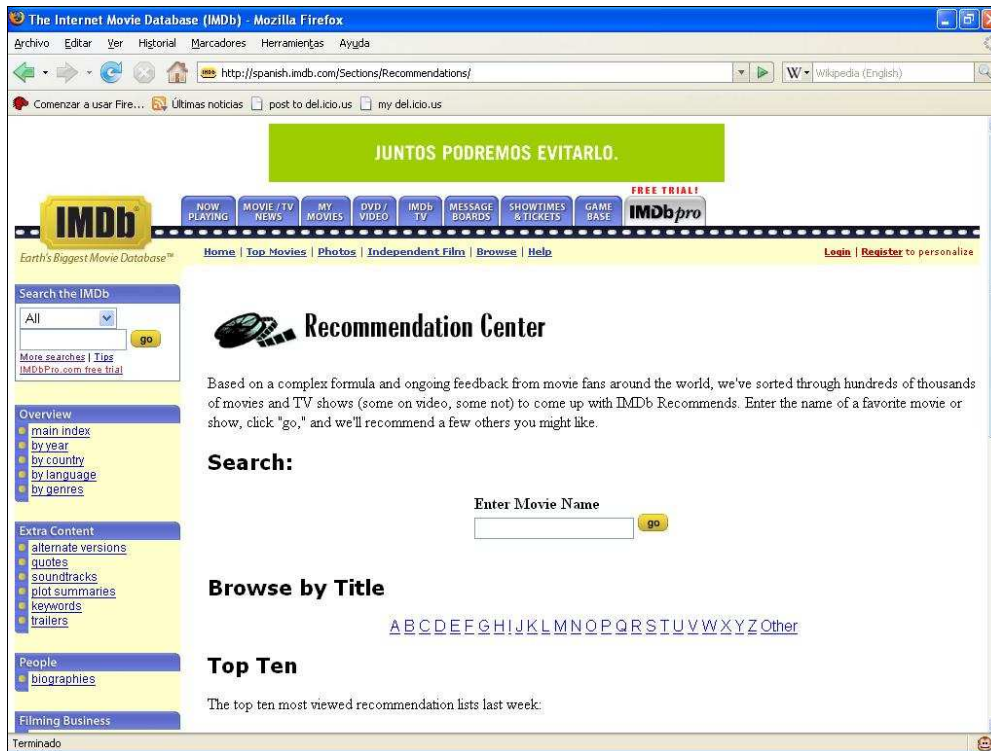


Figura 2.2. Interfaz del Centro de Recomendación de IMDB

La mayor base de datos de cine y televisión del mundo y uno de los sitios web más populares y visitados de todo Internet también ofrece a sus usuarios un sistema de recomendación: se llama **Recommendation Center** y esta **basado en contenido**. El usuario introduce la película o show televisivo que más le guste y el sistema le ofrece una lista con diez recomendaciones. Como método de realimentación o feedback con el sistema, el usuario puede señalar las recomendaciones con las que no este de acuerdo y proponer recomendaciones nuevas con lo que el algoritmo se va depurando con la interacción del usuario. De todas formas no es el servicio más exitoso que ofrece **IMDB** y eso se debe principalmente a que no es un sistema de recomendación especialmente bueno y acertado.

c) Entrée Chicago

Entrée Chicago (Guo; 2006) es un sistema de recomendación basado en conocimiento desarrollado por el Intelligent Information Laboratory (Infolab) de la Universidad de Chicago que ayuda al usuario a decidir entre más de 700 restaurantes de la ciudad de Chicago según sus restaurantes preferidos de otras ciudades norteamericanas. Además de ofrecer recomendaciones dispone de reviews de todos los restaurantes y de mapas para llegar a ellos.

d) One Llama! (<http://www.onellama.com/>)

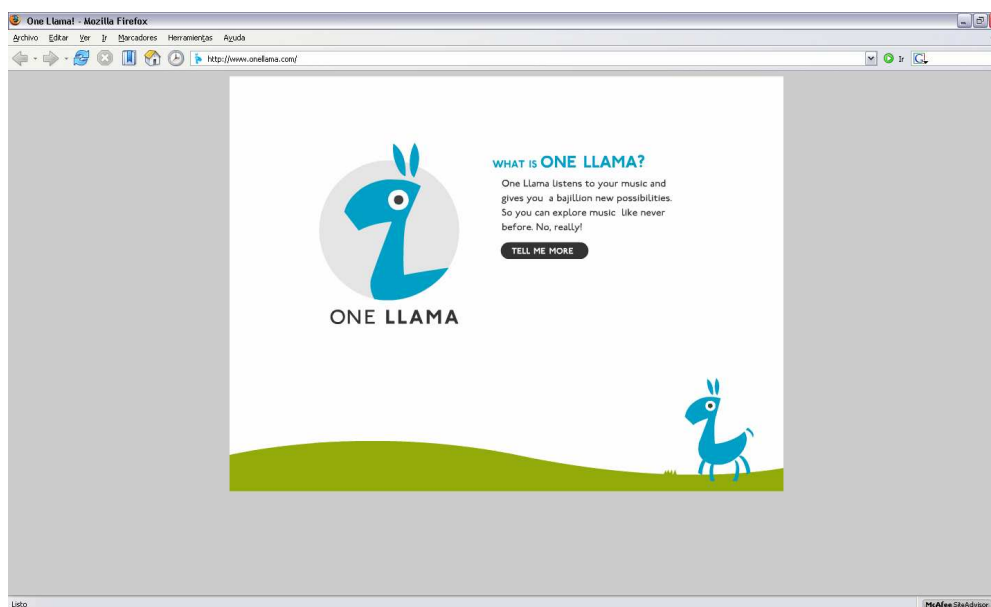


Figura 2.3. *Página de inicio de One Llama*

Un investigador del National Center for Supercomputing Applications, llamado Tcheng, trabajó en el desarrollo de este software que permite analizar música y categorizarla. Utiliza la tecnología 'oído artificial'. Este sistema ayuda a los usuarios a ordenar sus colecciones de música y también recomienda nuevas canciones que seguramente son de su agrado.

El punto fuerte de este sistema es que procesa cada canción, crea un conjunto de datos para cada una mediante técnicas de **minería de datos** y determina canciones con características similares y del mismo estilo musical.

2.2 Sistemas de Recomendación Colaborativos

Debido a que en este proyecto nos proponemos hacer un estudio de ciertos algoritmos de filtrado colaborativo para implementar posteriormente un sistema de recomendación colaborativo vamos a hacer una revisión más profunda y detallada sobre este tipo de sistemas de recomendación.

2.2.1 Introducción a los Sistemas de Recomendación Colaborativos

Un sistema de recomendación colaborativo es aquel en el que las recomendaciones se realizan basándose solamente en los términos de similitud entre los usuarios [1]. Es decir, los sistemas colaborativos recomiendan objetos que son del gusto de otros usuarios de intereses similares en vez de recomendar objetos similares a los que le gustaban en un pasado al usuario activo como sucedía con los basados en contenido. Se podría decir que este tipo de sistemas de recomendación se basan en el concepto del *boca a boca* entre sus usuarios para realizar sus recomendaciones.

La base teórica de los sistemas de recomendación colaborativos es bastante sencilla: se forman grupos de usuarios más cercanos, es decir, aquellos cuyos perfiles son más parecidos y a un usuario de un grupo se le recomiendan objetos que él no tenga puntuados pero que tengan unas puntuaciones positivas por parte del resto de usuarios de ese grupo, es decir, de los más similares a él.

Shardaham y Maes [13] distinguieron tres fases fundamentales en el funcionamiento de los sistemas de recomendación colaborativos:

1. El sistema guarda un perfil de cada usuario con sus evaluaciones sobre objetos conocidos por él y que pertenecen a la base de datos sobre la que se trabaje.
2. Se mide el grado de similitud entre los diferentes usuarios del sistema en base a sus perfiles y se crean grupos de usuarios con características afines.
3. El sistema utiliza toda la información obtenida en los dos pasos anteriores para realizar

las recomendaciones. A cada usuario le recomendará objetos que no haya evaluado previamente y que hayan sido evaluados de manera positiva por los miembros del grupo en el que esté incluido.

Al igual que los basados en contenido, los sistemas de recomendación colaborativos tienen una serie de problemas que se han ido detectando conforme se han hecho más populares y más utilizados, como son los problemas de: la escasez, la escalabilidad y del ítem nuevo [4]. Comentemos las principales características de cada uno de estos problemas:

- **Escasez**

Los sistemas de recomendación colaborativos necesitan una gran cantidad de datos, muchos usuarios puntuando muchos ítems similares para así poder calcular los grupos de vecinos y, en base a ellos, realizar las recomendaciones. Si en nuestra base de datos tenemos pocos usuarios o pocas puntuaciones por parte de cada usuario, nuestra matriz de puntuaciones será muy escasa y los cálculos de vecindad, predicción y recomendación no pueden ser realizados con la suficiente seguridad y exactitud obteniendo por lo tanto unas recomendaciones de baja calidad.

- **Escalabilidad**

Los sistemas de recomendación colaborativos usan por norma general algoritmos de cálculo de los k vecinos más cercanos (knn, K-nearest neighbors) para obtener la similaridad entre usuarios. Estos algoritmos son costosos computacionalmente y su coste crece linealmente cuanto mayor sea el número de usuarios y de ítems por lo que con bases de datos con millones de elementos, al aumentar el número de datos, el sistema sufrirá graves problemas de escalabilidad.

- **Problema del ítem nuevo**

En los sistemas de recomendación colaborativos los ítems nuevos, que tienen muy pocas o, incluso, ninguna puntuación no van a ser recomendados prácticamente nunca. De la misma forma, los nuevos usuarios en el sistema recibirán muy pobres predicciones debido a que ellos han puntuado muy pocos ítems y se hace difícil encuadrarlos en algún grupo de vecinos. Estos dos hechos nos hacen ver que estos

sistemas de recomendación requieren un cierto tiempo antes de empezar a hacer predicciones y recomendaciones ciertamente relevantes y acertadas.

Una gran cantidad de experimentos, estudios e investigaciones se han realizado en los últimos años para encontrar técnicas que reduzcan el impacto de estos problemas en los sistemas de recomendación con filtrado colaborativo.

Para reducir el problema de la escasez se han intentado la utilización de puntuaciones implícitas [10], la correlación entre ítems [12] y el filtrado híbrido. Mientras que para tratar de mejorar la escalabilidad se han propuesto la reducción de la dimensionalidad [11] y aproximaciones basadas en modelos. Finalmente, estudios han demostrado que las técnicas de web mining como los árboles de decisión son útiles para paliar el problema de los ítems y usuarios nuevos [3].

2.2.2 Categorías de los algoritmos de Filtrado Colaborativo

Para obtener un sistema de recomendación colaborativo de calidad es necesario elegir un buen algoritmo de filtrado colaborativo. Estos algoritmos pueden dividirse en dos categorías:

- **Algoritmos basados en memoria o basados en usuario**

Estos algoritmos utilizan la base de datos completa para generar una predicción. El funcionamiento de estos algoritmos es el siguiente: se utilizan técnicas estadísticas para encontrar un conjunto de vecinos al usuario activo y posteriormente se utilizan una serie de algoritmos que combinan las preferencias de esta vecindad para realizar las predicciones y recomendaciones.

Estos algoritmos basados en usuario son muy populares y exitosos en la práctica pero son también los que con más ferocidad sufren los problemas de escasez y escalabilidad vistos anteriormente por lo que se hizo necesaria la aparición de otro tipo de algoritmos como los que vienen a continuación.

- Algoritmos basados en modelos o basados en ítem

Estos algoritmos proporcionan recomendaciones de ítems desarrollando primero un modelo (ya sea mediante redes bayesianas, clustering o modelos basados en reglas) de las puntuaciones de los usuarios sobre los ítems.

No se utilizan técnicas estadísticas sino una aproximación probabilística que calcula el valor esperado de una predicción del usuario dados sus puntuaciones sobre otros ítems. Es decir, estos algoritmos miran en el conjunto de ítems que el usuario activo ha puntuado o evaluado y calcula como de similar son estas puntuaciones con respecto al ítem activo con el fin de realizar una predicción para el mismo.

No obstante también hay que puntualizar, que no todos los algoritmos basados en ítem han de desarrollar obligatoriamente un modelo de las puntuaciones de los usuarios.

Por tanto, en los sistemas de recomendación colaborativos basados en ítem (sin desarrollar modelo alguno) son en los que centrarán el proyecto que se detalla en esta memoria por lo que veremos con más profundidad estos algoritmos y sus principales aspectos cuando entremos a analizar en profundidad el proyecto realizado.

2.2.3 Ejemplos reales de Sistemas de Recomendación Colaborativos

En este epígrafe vamos a hacer un repaso a distintos sistemas de recomendación colaborativos que se pueden encontrar en el mercado actualmente o que han sido de vital importancia para el desarrollo, arraigo y auge de los mismos.

a) Tapestry

El pionero. Tapestry, un proyecto de Xerox PARC, está considerado como el primer sistema de recomendación que implementaba filtrado colaborativo. Tapestry permitía a sus usuarios encontrar documentos basados en comentarios hechos previamente por otros usuarios. Al ser un experimento pionero surgieron muchos problemas ya que sólo funcionaba

correctamente con pequeños grupos de personas y eran necesarias consultas de palabras específicas para obtener resultados lo que dificultaba en gran medida el propósito último del filtrado colaborativo. También tenía otras carencias como la falta de privacidad. A pesar de todo fue un sistema que resultó crucial para el posterior fulgurante crecimiento de los sistemas de recomendación colaborativos.

b) Zagat Survey (<http://www.zagat.com>)



Figura 2.4. Interfaz de zagat.com

Zagat Survey es una empresa americana fundada en 1979 que se dedica a la edición de todo tipo de guías de restaurantes, hoteles, clubes o tiendas de distintas ciudades de los Estados Unidos y Canadá. En zagat.com los usuarios registrados pueden votar distintos aspectos (hasta 30) del local referido y, además, introducir pequeños comentarios con su experiencia. En base a estas votaciones los responsables de la empresa asignan sus puntuaciones en sus guías anuales y hacen recomendaciones individuales a sus usuarios a través de su web.

c) FilmAffinity (<http://www.filmaffinity.com>)



Figura 2.5. Página principal de filmaffinity.com

FilmAffinity es un proyecto español de gran proyección internacional que desde 2002 se encarga de recibir puntuaciones de todo tipo de películas y dar recomendaciones a sus usuarios. Su funcionamiento es sencillo: el nuevo usuario puede empezar a votar las películas que haya visto con la ayuda de unos tours dirigidos que atemperan de manera inteligente los problemas de escasez y nuevo ítem; posteriormente el sistema calcula las almas gemelas de este nuevo usuario y le recomienda las películas favoritas de estas almas gemelas que no haya votado el usuario. También ofrece la posibilidad de puntuar series de televisión, de ver la información de tus almas gemelas y contactar con ellas, realizar críticas y comentarios sobre las películas y series vistas y consultar los diferentes tops por géneros o décadas o generales.

d) MovieLens (<http://movielens.umn.edu>)

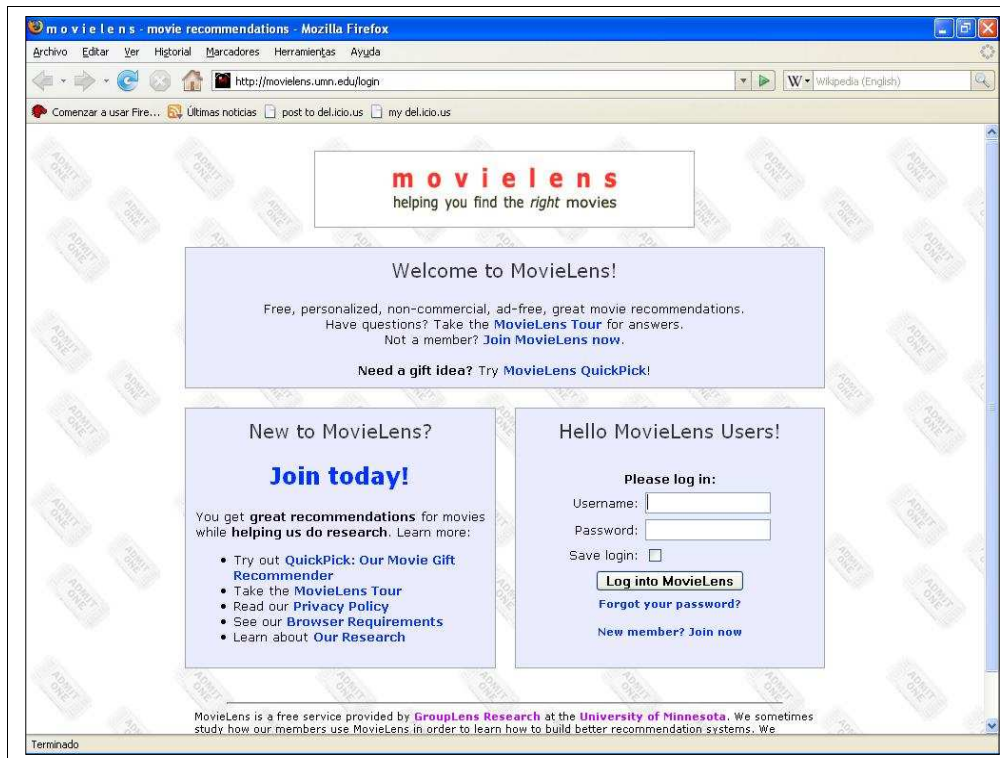


Figura 2.6. *Interfaz de MovieLens*

MovieLens es un sistema de recomendación de películas online basado en filtrado colaborativo. Desarrollado por el GroupLens Research de la **Universidad de Minnesota** (<http://www.grouplens.org>), recolecta puntuaciones sobre películas de sus usuarios y en base a esos datos agrupa los usuarios de similares gustos. Atendiendo a las puntuaciones de todos los usuarios dentro de un grupo se intenta predecir para cada usuario individual su opinión sobre películas que todavía no ha visto.

En un principio utilizaba algoritmos basados en usuario para realizar sus predicciones y recomendaciones pero desde hace un tiempo emplea algoritmos basados en ítem porque dan mejores resultados.

Los datos sobre sus usuarios y sus puntuaciones son privados pero los investigadores de GroupLens mantienen como públicas dos ejemplos de 100000 y un millón de puntuaciones respectivamente. Estos ejemplos se pueden descargar desde la propia página web (<http://www.grouplens.org>) siendo el de 100000 puntuaciones el usado en este proyecto como conjunto de datos de partida.

e) Last.fm (<http://www.last.fm>)

Figura 2.7. Página de recomendaciones de last.fm

La popularidad adquirida en los últimos tiempos por los sistemas de recomendación colaborativos ha propiciado que se haya ampliado el rango de aplicación de los mismos. Una nueva generación de sistemas de recomendación colaborativos ha surgido en el ámbito de la radio musical vía Internet.

last.fm es un sistema de recomendación musical además de una red social y una radio vía Internet. Cada nuevo usuario va creando su propio perfil de manera muy sencilla: el usuario puede escuchar las radios personalizadas (canciones favoritas) del resto de usuarios y decidir si les gustan (pasan a formar parte del perfil del propio usuario) o si por el contrario no quiere volverlas a escuchar. Con este perfil y gracias a un algoritmo de filtrado colaborativo el sistema va cerrando el cerco sobre los usuarios con gustos más afines con el usuario activo (los denominados vecinos) y recomendando grupos y artistas que no se encuentran en su perfil pero que sí son favoritos dentro de su vecinos. Estas acciones son continuas por lo que la calidad de las recomendaciones se refina de manera ciertamente importante cuando se es un usuario veterano dentro la aplicación.

Sin duda, last.fm y otras aplicaciones de similares características, están siendo una de las grandes revoluciones de los últimos tiempos en Internet gracias a su marcado carácter social, a su amplio catalogo musical (last.fm contaba con más de 100.000 canciones a noviembre de 2006) y a la calidad de sus recomendaciones.

f) Where to Cycle (<http://www.wheretocycle.com/>)

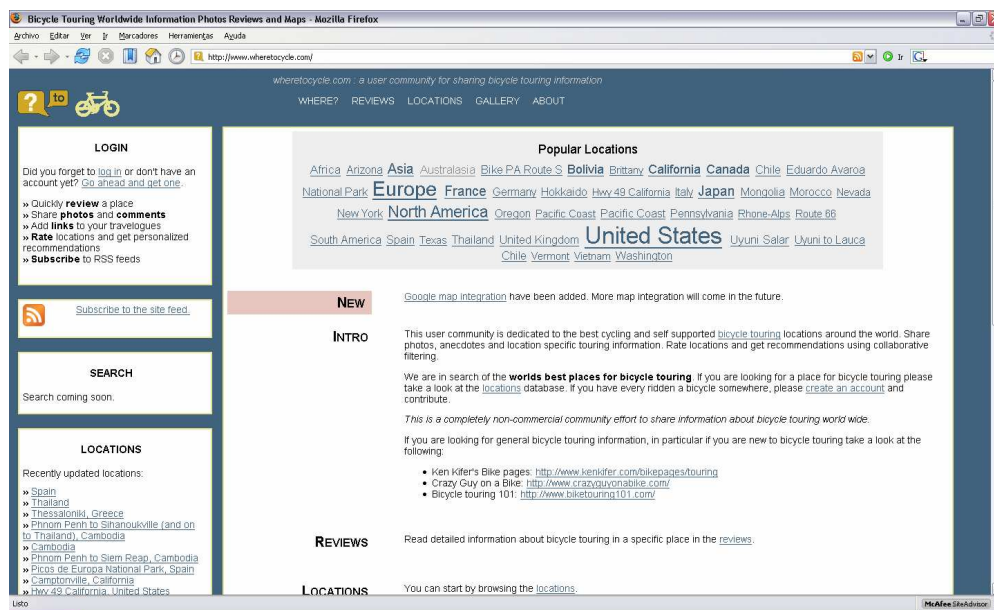
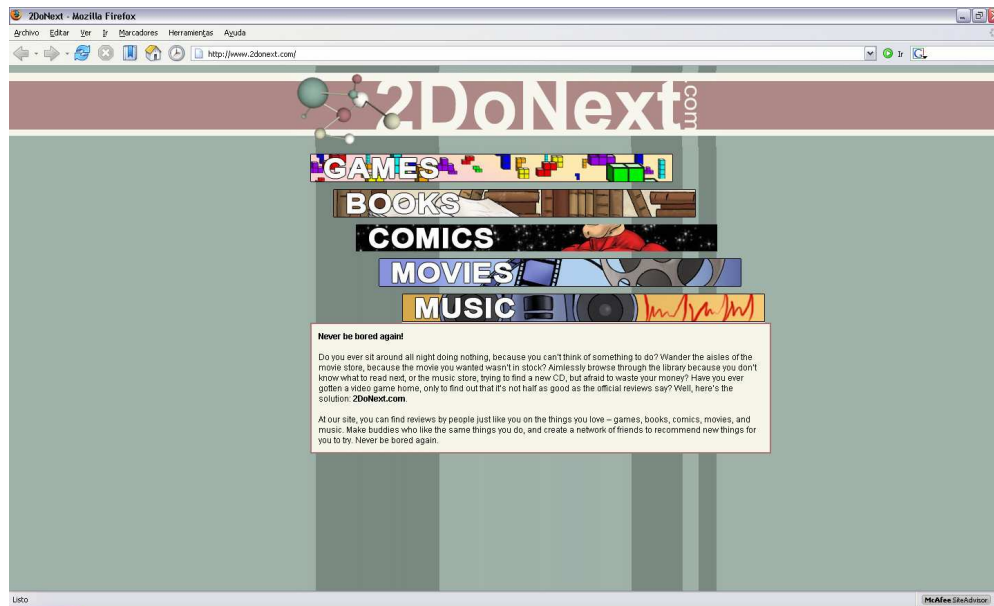


Figura 2.8. *Página de recomendaciones de wheretocycle*

Esta comunidad de usuarios se dedica a compartir información sobre viajes realizados en bicicleta por todo el mundo. Se dedica a fines no comerciales e intentan localizar los mejores lugares para viajar en bicicleta. Permite realizar puntuaciones sobre los lugares visitados por el usuario así como realizar las recomendaciones más adecuadas usando **filtrado colaborativo**.

g) 2DoNext (<http://www.2donext.com/>)Figura 2.9. *Página de recomendaciones de 2donext*

Después de años de trabajo, Mike se decidió a colgar su sitio web. El sitio web se llama 2DoNext, la idea de este sitio es permitir a la gente puntuar películas, videojuegos, libros, comics y música (cada uno por separado). También ofrece la posibilidad de compartir experiencias con familiares, amigos y todo tipo de personas que han puntuados ítems similares.

Como no, el sitio web incluye un sistema basado en **filtrado colaborativo** que recomendará otros ítems de acuerdo con lo que el propio usuario ha estado puntuando. El objetivo es permitir conectarse a los usuarios con otros usuarios y puntuar sus hobbies, de tal forma que se les pueda recomendar acerca de qué es lo próximo que podrían hacer para entretenerse.

h) Pure Video (<http://purevideo.com/>)

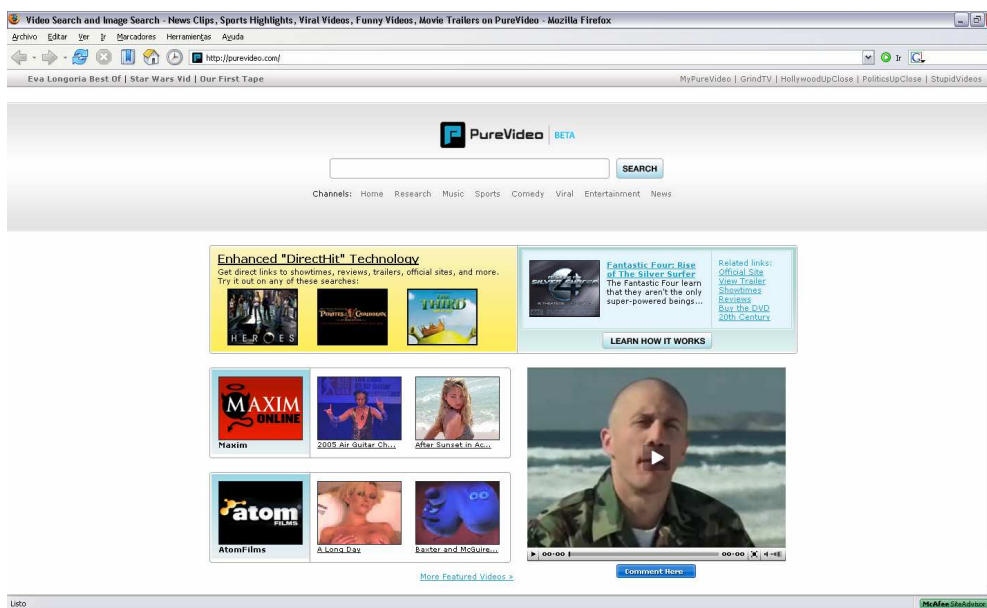


Figura 2.10. *Página principal de Pure Video*

Basada en El Segundo, CA, PureVideo Networks, Inc. es una compañía dedicada a las nuevas tecnologías que permite descubrir y promocionar videos online. Fue fundada por Softbank Capital, en Mayo de 2005, como consecuencia de la revolución de los videos online generados por los propios usuarios.

Con los videos online como un medio al alza en el mundo de la web 2.0, son muchos los sitios web de videos en el Mercado. El motor de búsqueda PureVideo, fue recientemente puesto en marcha para intentar ayudar y organizar los espacios de videos online. En este caso el **filtrado colaborativo** y el perfil de usuario deberían avanzar bastante de manera que el usuario sea capaz de acceder a su lista de archivos multimedia desde cualquier sitio, o buscar nuevos archivos de una manera fácil.

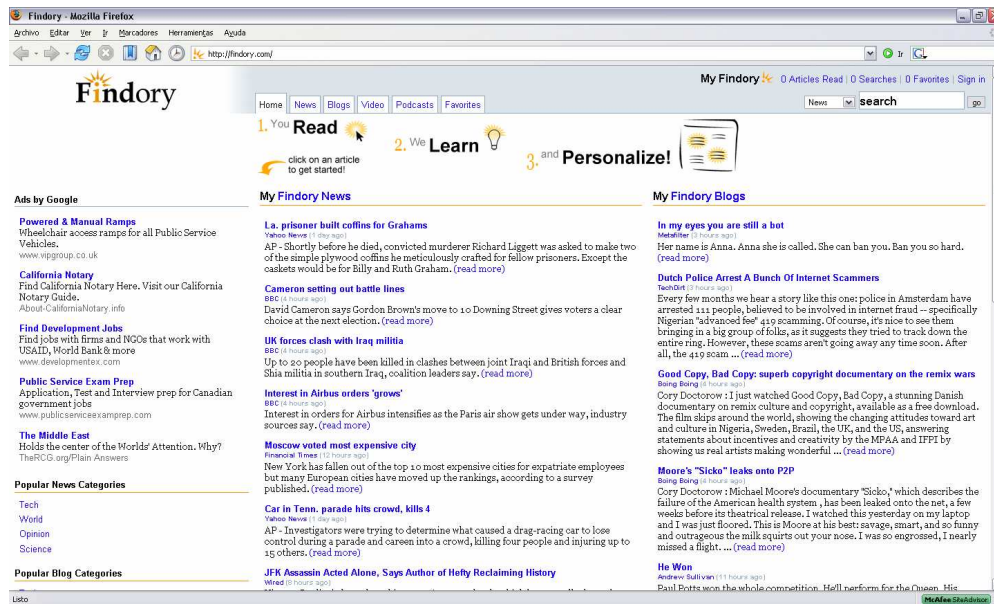
i) Findory (<http://findory.com/>)

Figura 2.11. Página de inicio de findory

Findory es un sitio de noticias de actualidad que emplea **filtrado colaborativo** para poder mostrarle al usuario noticias de su interés. Fue desarrollado por Greg Linden en Seattle, proveniente de Amazon (el rey de los sitios de recomendaciones).

j) Silver Egg (<http://www.silveregg.co.jp/en/>)

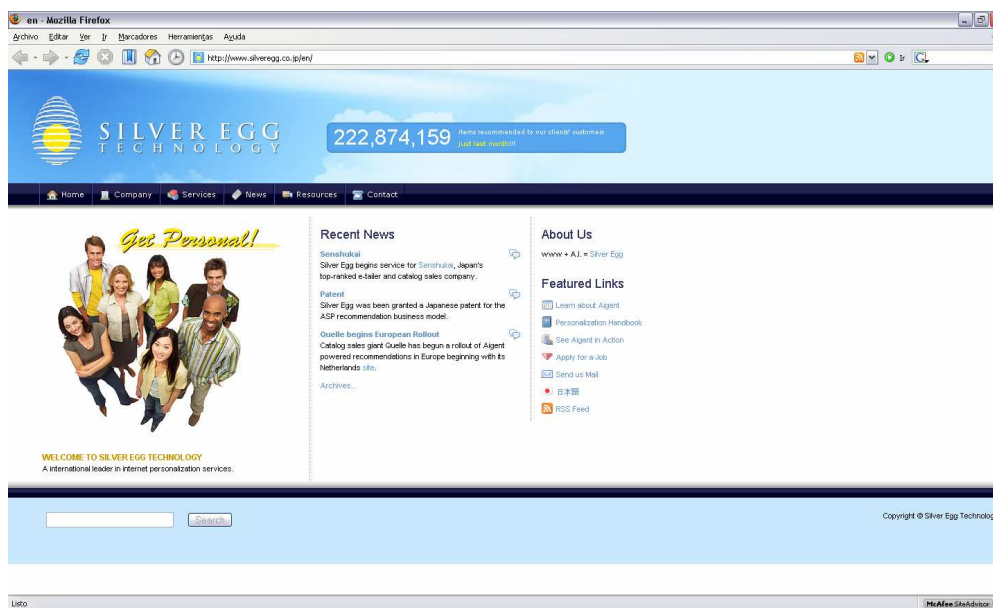


Figura 2.12. *Página de inicio de Silver Egg*

Silver Egg proporciona servicios web comerciales que funcionan con tecnología de Inteligencia Artificial. El servicio más importante es Aigent, un servicio en ASP empleado en las páginas web líderes de comercio electrónico para realizar recomendaciones de productos personalizadas. Utiliza **filtrado colaborativo basado en redes bayesianas**, para elegir los productos que se van a recomendar. Adaptando continuamente las recomendaciones a los cambios de compra del usuario.

Otros sistemas de recomendación colaborativos (tanto de ámbito comercial como no comercial) que pueden ser de interés para que el lector compruebe el funcionamiento y utilidad de los mismos son los siguientes: Pandora (<http://www.pandora.com>), Live365 (<http://www.live365.com>), TiVo (<http://www.tivo.com>), aNobii (<http://www.anobii.com/anobi>) y otros muchos como los que pueden ser consultados en la siguiente dirección ([http://en.wikipedia.org/wiki/Collaborative filtering](http://en.wikipedia.org/wiki/Collaborative_filtering)).

CAPÍTULO 3.

PROYECTO.

Una vez presentado el proyecto a grandes rasgos con su propósito y objetivos, y realizada la introducción teórica a los sistemas de recomendación en general y a los sistemas de recomendación colaborativos en particular, llega el momento de pasar a detallar el desarrollo del proyecto que se ha realizado.

Este proyecto consta de cuatro fases bien diferenciadas:

1. La primera fase consiste en una introducción y desarrollo de los distintos tipos de algoritmos de filtrado colaborativo básicos basados en ítem, y un diseño de pruebas que aplica dichos algoritmos a la base de datos movieranks [25].
2. La segunda fase se basa en implementar una serie de mejoras, propuestas en la literatura, aplicadas a los algoritmos de filtrado colaborativo básicos sobre nuestra base de datos. También se realizará un diseño de pruebas en el que quedarán reflejados los resultados obtenidos.
3. La tercera fase consiste en realizar un estudio comparativo entre las distintas pruebas realizadas en los puntos 1 y 2, con el fin de obtener el algoritmo de filtrado colaborativo que mejores resultados obtenga, lo cual indicará que es el que mejor se adapta a la base de datos tratada.
4. La cuarta fase consiste en implementar un prototipo de un sistema de recomendación basado en una arquitectura cliente/servidor con interfaz web implementando el algoritmo de filtrado colaborativo que resulte en el apartado 3.

En este capítulo abordamos las tres primeras fases y en el siguiente capítulo se presenta el sistema de recomendación, objeto de la última fase.

3.1 Revisión de Algoritmos de Filtrado Colaborativo

En este apartado vamos a encargarnos de detallar pormenorizadamente el desarrollo de los distintos algoritmos de filtrado colaborativo sobre la base de datos que nos ocupa.

En la introducción teórica se presentaban los sistemas de recomendación colaborativos. El primer paso para su realización es formar grupos con los usuarios o ítems de la base de datos más similares entre sí. Para formar estos grupos se aplicarán distintas medidas de similitud y un algoritmo de clasificación K-nn.

Una vez creados estos grupos y evaluado el algoritmo K-nn mediante la técnica *hold-out* se estudiará el comportamiento de distintos algoritmos de predicción.

3.1.1 Notación

Antes de empezar a enumerar las numerosas fórmulas, expresiones y algoritmos que se van a detallar en este estudio resulta conveniente dejar clara la notación que se va a emplear para que no se produzcan equívocos entre los lectores:

- Un usuario será aquel elemento representado por $u_i \in U = \{u_1, \dots, u_n\}$
- Un ítem será aquel elemento representado por $i_i \in I = \{i_1, \dots, i_m\}$
- La similitud entre dos ítems i_j, i_k se representará como $s(i_j, i_k)$
- Una evaluación o puntuación de un usuario u_i sobre un ítem i_j se representará como r_{u_i, i_j}
- Finalmente, una predicción sobre el ítem i_j del usuario u_i se representará como p_{u_i, i_j}

3.1.2 Algoritmo K-nn

Un paso imprescindible para la realización de un sistema de recomendación colaborativo de calidad es la formación de grupos de usuarios (si es un sistema de recomendación colaborativo basado en memoria) o de ítems (si es basado en modelos, como es el caso de este proyecto) de características similares. Esta actividad se puede ver como la primera parte de un **problema de clasificación** de la base de datos y existen multitud de técnicas y algoritmos, llamados clasificadores, para resolverlo. Uno de los clasificadores de mayor difusión y mejores prestaciones es el **algoritmo K-nn**, que es el que se va a utilizar en este proyecto para formar los grupos de ítems más similares para cada uno de los ítems de la base de datos.

El algoritmo K-nn (k nearest neighbors, k vecinos más cercanos) es un tipo de clasificador basado en instancias. Estos clasificadores, que trabajan directamente sobre los datos sin construir ningún tipo de modelo sobre la base de datos, están basados en aprendizaje por analogía encuadrándose dentro del paradigma perezoso del aprendizaje. Este paradigma se caracteriza por:

- El trabajo se retrasa lo máximo posible.
- No se construye ningún modelo, el modelo es la propia base de datos sobre la que se trabaja.
- Se trabaja cuando llega un nuevo objeto a clasificar: se buscan los casos más parecidos y la clasificación se construye en función de la clase a la que dichos casos pertenezcan.

a) Funcionamiento del algoritmo K-nn:

Siendo i el objeto a clasificar debemos seleccionar los k objetos con $K = \{i_1, \dots, i_k\}$ tal que no existe ningún ejemplo i' fuera de K con $s(i, i') < s(i, i_j), j = 1, \dots, k$.

Una vez encontrados los k -vecinos se puede proceder a la clasificación de dos formas distintas:

- Voto por la mayoría: se clasifica el nuevo objeto en la clase mayoritaria entre los objetos de K .
- Voto compensado según la distancia: se clasifica el objeto según su distancia ponderada con el resto de objetos de K .

b) Descripción algorítmica

1. Se separan los datos en dos conjuntos disjuntos: entrenamiento (E) y test (T)
2. Llega un nuevo objeto i_a a clasificar
3. Se obtienen los k objetos i_1, \dots, i_k del conjunto E mas cercanos a i_a
4. Se clasifica el objeto i_a de una de las dos maneras siguientes:
 1. Voto por la mayoría
 2. Voto ponderado según la distancia

c) Principales características

- Es un algoritmo robusto frente al ruido cuando el valor de k es moderado ($k > 1$).
- Es bastante eficaz cuando el número de clases posibles es alto y cuando los datos son heterogéneos o difusos.
- Tiene una complejidad temporal de $O(dn^2)$ siendo $O(d)$ la complejidad de la métrica de similitud utilizada.
- El hecho de no utilizar modelos sino la base de datos al completo provoca que sea

ineficiente en memoria.

- Es válido tanto para clasificación como para predicción numérica.

Para este estudio se utilizará un algoritmo K-nn para seleccionar los k ítems más similares para cada uno de los ítems que componen nuestra base de datos. Lo que variará de una prueba a otra será la métrica de medida o similaridad empleada para calcularlo y el tamaño de los conjuntos de test y entrenamiento utilizados para su evaluación *hold-out*.

3.1.3 Medidas de similaridad

Un paso previo crucial para el filtrado colaborativo basado en ítem es el cálculo de la similaridad entre los distintos ítems y la selección de los k más similares. El concepto de similaridad se puede representar de muy diversas formas de entre las cuales hemos elegido para este proyecto la siguiente:

$s(x, y): U \times U \rightarrow [-1, 1]$, midiendo el grado de similaridad entre x e y siendo mayor cuanto más cerca de 1 se encuentre.

La similaridad cumple con las propiedades:

- Reflexiva: ($s(x, x) = 1$)
- Simétrica: ($s(x, y) = s(y, x)$).

Por lo tanto para calcular los ítems más similares a uno dado x , con una medida de similaridad entre ítems $s(x, y): U \times U \rightarrow [-1, 1]$, $y \in U - x$, seleccionamos los k ítems y tal que la función sea máxima.

Es conveniente aclarar que, en este proyecto, para calcular la similaridad entre dos ítems x e y , sólo se tendrán en cuenta a aquellos usuarios que hayan evaluado a ambos ítems y no siendo tomados en consideración el resto.

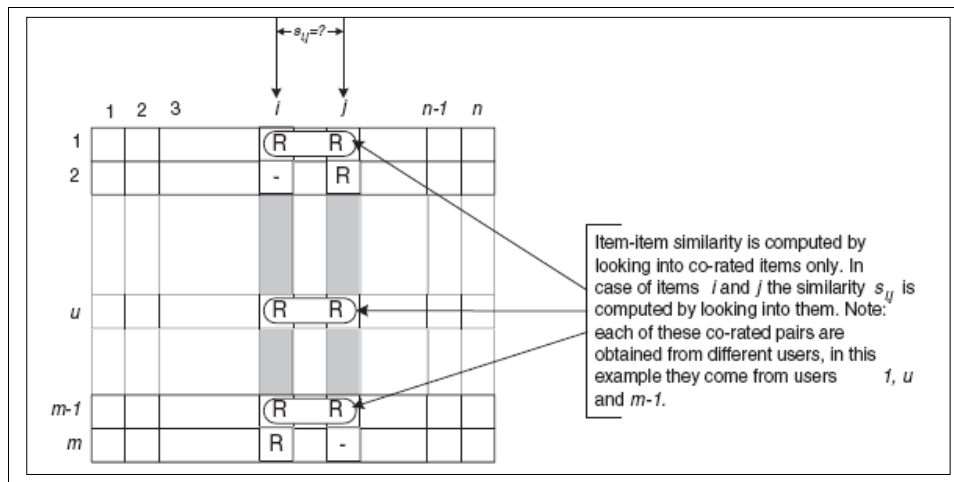


Figura 3.1.1 Cálculo de similaridad basado sólo en ítems co-evaluados

Existen multitud de funciones que nos pueden dar una medida de la similaridad entre dos elementos pero nosotros hemos elegido para este proyecto implementar solo las tres siguientes:

1. Coeficiente Coseno

En este método se supone que dos ítems x e y son vectores en el espacio y la similaridad entre ellos vendrá dada por el coseno que formen sus ángulos. La expresión para su cálculo es la siguiente:

$$s(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n (x_i)^2 \sum_{i=1}^n (y_i)^2}}$$

Siendo x_i el valor del objeto x para el usuario i , y_i el valor del objeto y para el usuario i y n el número de usuarios que han evaluado tanto x como y .

2. Distancia Euclídea

Una forma de calcular la similaridad entre dos objetos puede ser calcular la distancia entre los mismos ya que cuanto menor sea esa distancia mayor será la similaridad. Una distancia cumple con las siguientes propiedades:

1. $d(x, y) = 0 \Leftrightarrow x = y$
2. $d(x, y) = d(y, x)$ (Propiedad simétrica)
3. $d(x, z) \leq d(x, y) + d(y, z)$

Existen múltiples funciones para el cálculo de distancias y una de las más importantes y utilizadas es la distancia euclídea cuya expresión genérica es la siguiente:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Pero esta distancia plantea el inconveniente de que el valor resultante no está restringido a un intervalo entre $[0, 1]$ que es el intervalo en el que hemos comentado que vamos a representar la similaridad. Para remediar este inconveniente recurrimos a la normalización de la fórmula anterior, que quedaría de la siguiente manera:

$$d_n(x, y) = \frac{\sqrt{\sum_{i=1}^n (x_i - y_i)^2}}{\sqrt{\sum_{i=1}^n A_i^2}}$$

Siendo A_i la amplitud del dominio del usuario i , la cual se calcula como: $A_i = b_i - a_i$ donde b y a son los límites superior e inferior respectivamente de tal dominio. Este dominio del usuario, en nuestro proyecto, es el comprendido por el intervalo de enteros entre 1 y 5, ambos inclusive.

La similaridad será por tanto: $s(x, y) = 1 - d_n(x, y)$

3. Coeficiente de Correlación de Pearson

Este coeficiente de correlación de Pearson apareció por primera vez en la literatura dentro del contexto del proyecto **GroupLens** donde se empleaba como base para la asignación de pesos. Este coeficiente es un índice que mide la relación lineal entre dos variables cuantitativas, siendo independiente de la escala de medidas de dichas variables y estando acotado dentro del intervalo $[-1, 1]$. La expresión que permite calcular este coeficiente es la

siguiente:

$$s(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Siendo \bar{x} e \bar{y} la media de todos los valores de x e y respectivamente.

3.2 Algoritmos de predicción basados en ítem

Una vez calculado el conjunto de los ítems más similares para cada uno de los ítems de la base de datos mediante alguna de las medidas de similaridad vistas anteriormente, llega el momento del paso más crucial en el filtrado colaborativo: elegir la técnica o algoritmo adecuado para realizar la predicción.

3.2.1 Algoritmo Básico de Filtrado Colaborativo

No existen algoritmos mejores o peores sino que existen algoritmos que se ajustan mejor o peor al conjunto de datos [7]. Esto se debe a que muchos de los algoritmos de filtrado colaborativo han sido diseñados para un conjunto de datos específico. Si este conjunto de datos específico tiene muchos más usuarios que ítems puede resultar inapropiada su ejecución sobre conjuntos de datos donde se tienen más ítems que usuarios y viceversa.

En este proyecto hemos tomado en consideración dos de los múltiples algoritmos que existen para resolver esta cuestión y vamos a comprobar cual se ajusta mejor a nuestra base de datos:

1. item average + adjustment

Esta técnica presupone que una predicción para un usuario concreto sobre un ítem es igual al valor medio de ese ítem más un ajuste que viene a ser la suma ponderada de las

evaluaciones hechas por el usuario y su similaridad con el ítem activo [9]. La expresión para dicha técnica es la siguiente:

$$p^{u_a, i_a} = \bar{r}_{i_a} + \frac{\sum_{h=1}^n s(i_a, i_h) (r_{u_a, i_h} - \bar{r}_{u_a})}{\sum_{h=1}^n |s(i_a, i_h)|}$$

Siendo u_a el usuario activo, i_a el ítem cuyo valor se quiere predecir y \bar{r}_{i_a} y \bar{r}_{u_a} las puntuaciones medias del usuario y el ítem, respectivamente. Estas medias se calculan de la siguiente manera:

$$\bar{r}_{i_a} = \frac{\sum_{h=1}^m r_{u_h, i_a}}{m} \quad \bar{r}_{u_a} = \frac{\sum_{h=1}^n r_{u_a, i_h}}{n}$$

Donde n es el número de ítems que el usuario activo ha puntuado y m es el número de usuarios que han puntuado el ítem a predecir.

Existen dos enfoques diferenciados para afrontar esta técnica atendiendo al número de valores seleccionados para realizar la predicción:

- **Todos menos 1** -> se conocen todas las evaluaciones que ha hecho el usuario salvo la que se quiere predecir.
- **Dados n** -> solo se conocen n evaluaciones del total que ha hecho el usuario. Esta n suele ser un número potencia de 2.

2. weighted sum

Este método calcula la predicción de un ítem i por parte del usuario activo u_a como la suma de las evaluaciones del usuario u_a sobre ítems similares a i . Cada una de estas evaluaciones esta ponderada por la correspondiente similaridad $s(i, j)$ entre los ítems i y j [12]. Podemos denotar esta técnica de la siguiente manera:

$$p(u_a, i_a) = \frac{\sum_{h=1}^k s(i_a, i_h) * r_{u_a, i_h}}{\sum_{h=1}^k |s(i_a, i_h)|}$$

Indicando k los k ítems más similares al ítem i_a .

Básicamente, esta técnica intenta captar como evalúa el usuario activo a ítems similares al que se quiere predecir. Es necesario ponderar estas evaluaciones con la similaridad para asegurarnos de que la predicción entra dentro del rango previamente definido.

3.2.2 Mejoras sobre los Algoritmos de Filtrado Colaborativo

Sobre los modelos iniciales se han propuesto una serie de mejoras [2] sobre algoritmos de filtrado colaborativo para comprobar como funcionan dichas mejoras sobre la base de datos sobre la que queremos trabajar, y cuales son las distintas variaciones de sus parámetros a los largo de este proyecto. Las distintas mejoras de algoritmos de filtrado colaborativo que tendremos en cuenta se basarán en el cálculo de predicción, para lo cual a continuación se desarrollará en qué consiste cada una.

1. *Voto por Defecto*

El algoritmo del voto por defecto surge de la necesidad de que cuando hay relativamente pocos votos realizados por el usuario activo (u_a) o por el usuario del emparejamiento (u_i), el algoritmo de correlación de Pearson no obtiene buenos resultados ya que éste sólo tiene en cuenta los votos realizados por los dos usuarios:

$$s(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Esta propuesta de mejora se basa en el Algoritmo de Correlación de Pearson, explicado en el apartado 3.1.3 de esta memoria. Ahora bien, si proponemos un valor por defecto, d , para los ítems que han sido votados sólo por uno de los usuarios, ya sí se puede realizar el

emparejamiento entre los dos usuarios de mayor número de valoraciones. Como sugerencia para la valoración de d , existen dos posibilidades, utilizar un valor neutral o utilizar algún valor cercano a valoraciones negativas.

También se propone utilizar ese mismo valor por defecto d para un número k de ítems que ninguno de los dos usuarios haya votado. Este número adicional de ítems se utiliza para potenciar la similaridad entre los dos usuarios. Para nuestro estudio, emplearemos la siguiente ecuación:

$$s(a,i) = \frac{(n+k)(\sum_j v_{a,j}v_{i,j} + kd^2) - (\sum_j v_{a,j} + kd)(\sum_j v_{i,j} + kd)}{\sqrt{((n+k)(\sum_j v_{a,j}^2 + kd^2) - (\sum_j v_{a,j} + kd)^2)((n+k)(\sum_j v_{i,j}^2 + kd^2) - (\sum_j v_{i,j} + kd)^2)}}$$

Siendo n , los ítems votados por al menos uno de los dos usuarios, y los sumatorios en j vienen referidos a los ítems que el usuario activo y el del emparejamiento han votado.

Para comprender este algoritmo correctamente, realizaremos una explicación gráfica utilizando un ejemplo:

- Paso 1. Tomamos como referencia los votos que han realizado los usuarios entre m ítems posibles (los votos quedan marcados con la letra R).

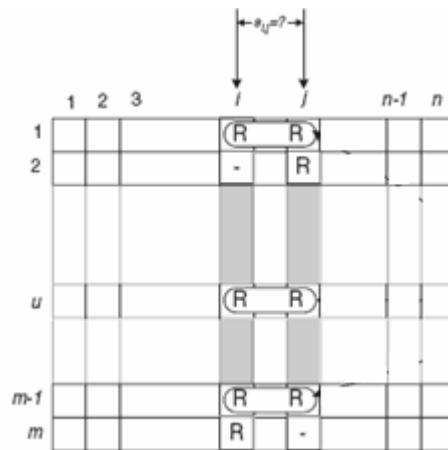


Figura 3.2.1 Puntuaciones iniciales de los usuarios

- Paso 2. Para los ítems que han puntuado uno de los dos usuarios, proponemos puntuar con el valor por defecto d en el usuario correspondiente (**R**).

	1	2	3	...	<i>i</i>	<i>j</i>	...	<i>n-1</i>	<i>n</i>
1					R	R			
2					R	R			
...									
<i>u</i>					R	R			
...									
<i>m-1</i>					R	R			
<i>m</i>					R	R			

Figura 3.2.2 Puntuaciones de los usuarios con el voto por defecto

Ahora estamos en disposición de aplicar la ecuación propuesta para el cálculo de la similitud mediante el voto por defecto.

Para nuestro caso los valores escogidos son los siguientes:

- d: se trata del valor asignado a los votos por defecto, en nuestro caso tomaremos la valoración neutra 3.
- k: número adicional de ítems que potencia la similitud entre los usuarios, 30.

2. Frecuencia Inversa de Usuario

La idea de esta mejora se basa en que el usuario que ha puntuado muchos ítems, no es muy útil para el cálculo de la similitud como lo son los usuarios que han puntuado menos ítems, debido a que un usuario que puntúa todos los ítems tendrá una similitud alta con todos los usuarios con los que lo emparejemos.

Para llevar a cabo este fin, calcularemos un parámetro f_j como $f_j = \log \frac{n}{n_j}$ donde n_j es el número de ítems que ha votado el usuario j y n es el número total de ítems en la base de datos. De este modo, si el usuario vota todos los ítems, f_j será 0, y en el caso en el que el usuario no puntúe ningún ítem, f_j será 1.

Para aplicar la frecuencia inversa de usuario, de entre los métodos propuestos en la literatura, nosotros utilizaremos una modificación del algoritmo de coeficiente coseno, explicado en el punto 3.1.3:

$$s(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n (x_i)^2 \sum_{i=1}^n (y_i)^2}}$$

Esta modificación consiste en multiplicar dicha ecuación por el factor f_j propuesto

$$s(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n (x_i)^2 \sum_{i=1}^n (y_i)^2}} \cdot f_j$$

de manera que los usuarios que tienen la f_j más alta, serán los que menos ítems han puntuado, por tanto quedará disminuída la similaridad cuando al número de ítems puntuados (n_j) por el usuario sea alto.

3. Frecuencia Directa de Usuario

Por intuición, hemos pensado una variante de la Frecuencia Inversa de Usuario. Hemos explicado que ésta consiste en considerar como mejores a aquellos usuarios que han puntuado un menor número de ítems. Sin embargo, en el caso particular tratado en este proyecto acerca del Sistema de Recomendación de Alquiler de Películas y según las características de la base de datos usada, podemos pensar que los usuarios que puntúan un número muy grande de películas (ítems) no se pueden penalizar y sí potenciar ya que estos usuarios tienen un mayor grado de conocimiento de la materia y son muy fiables en sus puntuaciones. La experiencia adquirida por estos usuarios nos puede ser de gran ayuda para realizar nuestro cálculo de similaridad.

Análogamente a la frecuencia inversa de usuario, utilizaremos una modificación del algoritmo de coeficiente coseno, pero esta vez multiplicada por un nuevo factor f_j , calculado de

la siguiente manera, $f_j = 10^{\frac{n_j}{n}}$, donde n_j es el número de ítems que ha votado el usuario j y n es el número total de ítems en la base de datos.

Quedando la ecuación utilizada en este caso de la siguiente manera:

$$s(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n (x_i)^2 \sum_{i=1}^n (y_i)^2}} \cdot f_j$$

4. Amplificación de Casos

La Amplificación de Casos se refiere a una modificación en la similaridad a la hora de realizar el cálculo de la predicción. Esta transformación potenciará similitudes cercanas y penalizará similitudes lejanas.

Entre los posibles algoritmos de cálculo de similaridad utilizados, para poder aplicar esta modificación, y según hemos podido investigar entre los artículos de la literatura seguida, tendremos que aplicar el Coeficiente Coseno.

El algoritmo de predicción que ha de utilizarse se basa en el item average + adjustment explicado en el apartado 3.2.1 de esta memoria, el cual presupone que una predicción para un usuario concreto sobre un ítem es igual al valor medio de ese ítem más un ajuste que es la suma ponderada de las evaluaciones hechas por el usuario y su similaridad con el ítem activo:

$$pu_{a,i_a} = \bar{ri}_a + \frac{\sum_{h=1}^n s(i_a, i_h)(ru_{a,i_h} - \bar{ru}_a)}{\sum_{h=1}^n |s(i_a, i_h)|}$$

En cuanto a las transformaciones realizadas a esta fórmula para aplicar esta mejora, son las siguientes:

- Si la similaridad es mayor o igual que cero:
$$pu_{a,i_a} = \bar{ri}_a + \frac{\sum_{h=1}^n s(i_a, i_h)^\rho (ru_{a,i_h} - \bar{ru}_a)}{\sum_{h=1}^n |s(i_a, i_h)|}$$

- Si la similaridad es menor que cero:
$$pu_{a,i_a} = \bar{ri}_a + \frac{\sum_{h=1}^n (-(-s(i_a, i_h)^\rho))(ru_{a,i_h} - \bar{ru}_a)}{\sum_{h=1}^n |s(i_a, i_h)|}$$

El valor que hemos utilizado en nuestros experimentos para el parámetro ρ , es el indicado en la literatura estudiada, y será de 2.5.

3.3 Estudio Comparativo

A lo largo de este apartado, se realizarán las pruebas necesarias sobre los algoritmos básicos de filtrado colaborativo y sobre sus mejoras para obtener los mejores resultados. Previo a dicho estudio realizaremos una descripción de las especificaciones, del conjunto de datos y de las métricas de evaluación que son comunes a los dos estudios.

1. Especificaciones software y hardware

Resulta fundamental especificar los requerimientos tanto hardware como software con las que se van a realizar las pruebas ya que la variación de los mismos podría producir unos resultados sustancialmente diferentes a los que se han obtenido en este proyecto.

Las pruebas se han realizado en distintos PC's de sobremesa con distintas características de microprocesador y de memoria RAM sobre un sistema operativo **Windows XP Professional Service Pack 2**. El uso de un equipo u otro repercutirá sólo en el tiempo en el que se obtendrán los resultados de las pruebas. Por tanto los valores calculados son independientes del equipo empleado.

El lenguaje de programación empleado para la implementación de estas pruebas ha sido **Java** mediante el entorno de desarrollo **NetBeans 5.0** y el sistema gestor de bases de datos empleado para la comunicación con el conjunto de datos de prueba ha sido **Microsoft Access** a través de **ODBC**.

2. Conjunto de datos

El conjunto de datos utilizado para evaluar los experimentos realizados en este proyecto es un ejemplo de la base de datos **MovieLens** (disponible en <http://www.grouplens.org>) formado por 943 usuarios, 1682 películas y 100000 puntuaciones habiendo puntuado cada uno de esos 943 usuarios un mínimo de 20 películas y habiendo sido puntuada cada película al menos una vez.

Esta base de datos se encuentra en un formato textual poco manejable y eficiente por lo que la hemos transformado a un formato de base de datos más aconsejable para su

tratamiento por parte de un módulo desarrollado en Java a través de ODBC como es **MS Access**. La implementación en Java del módulo de transformación o formateo se encuentra disponible en el **Anexo IV**.

Una vez realizada esta transformación o formateo de la base de datos a un formato más adecuado a nuestros propósitos se ha construido una base de datos en MS Access llamada **movieranks** que contiene las tres tablas siguientes:

USUARIOS: una tabla de 943 filas en la que cada una de estas filas está compuesta por los siguientes campos:

- ID_USER: entero. Llave primaria. Identificador numérico y unívoco del usuario.
- EDAD: entero. Edad del usuario.
- GENERO: cadena de 1 carácter. Género del usuario (M si es hombre, F si es mujer).
- PROFESION: cadena de 25 caracteres. Profesión del usuario.
- COD_POSTAL: cadena de 5 caracteres. Código postal del usuario.
- NUM_PUNTUACIONES: entero. Campo calculable. Número de películas puntuadas por el usuario.



Figura 3.3.1 Vista Diseño de la tabla USUARIOS

PELICULAS: tabla de 1682 filas con los siguientes campos cada una:

- ID_MOVIE: entero. Llave primaria. Identificador numérico y unívoco de la película.
- TITULO: cadena de 81 caracteres. Título de la película.
- FECHA: fecha. Fecha de estreno de la película.
- IMDB_URL: cadena de 134 caracteres. Enlace a la entrada en IMDB de la película.
- DESCONOCIDO: booleano. Verdadero si no se conoce el género de la película.
- ACCION: booleano. Verdadero si la película es de acción.
- AVENTURAS: booleano. Verdadero si la película es de aventuras.
- ANIMACION: booleano. Verdadero si la película es de animación.
- INFANTIL: booleano. Verdadero si es una película infantil.
- COMEDIA: booleano. Verdadero si la película es una comedia.
- CRIMEN: booleano. Verdadero si es una película de crimen.
- DOCUMENTAL: booleano. Verdadero si la película es un documental.
- DRAMA: booleano. Verdadero si la película es un drama.
- FANTASIA: booleano. Verdadero si la película es de fantasía.
- NEGRO: booleano. Verdadero si la película es de género negro.
- TERROR: booleano. Verdadero si la película es de terror.
- MUSICAL: booleano. Verdadero si la película es un musical.
- MISTERIO: booleano. Verdadero si la película es de misterio.
- ROMANTICO: booleano. Verdadero si la película es romántica.
- CIENCIA-FICCION: booleano. Verdadero si la película es de ciencia-ficción.
- THRILLER: booleano. Verdadero si la película es un thriller.
- GUERRA: booleano. Verdadero si la película es de guerra.
- WESTERN: booleano. Verdadero si la película es un western.

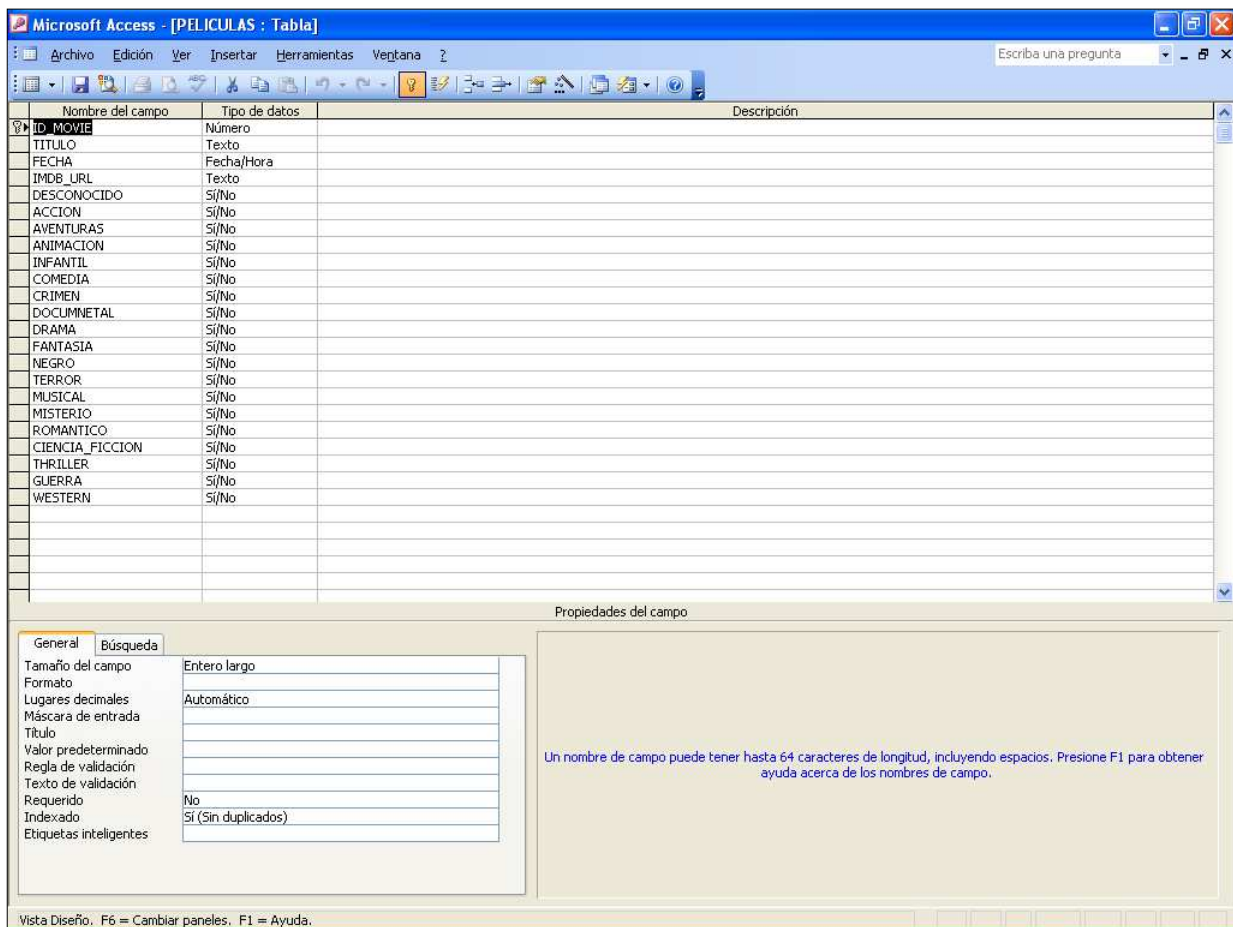


Figura 3.3.2. Vista Diseño de la tabla PELICULAS

PUNTUACIONES: tabla con 100000 filas con los siguientes campos cada una:

- ID_USER: entero. Llave primaria. Llave foránea. Identificador del usuario.
- ID_MOVIE: entero. Llave primaria. Llave foránea. Identificador de la película.
- RATING: byte. Puntuación (1, 2, 3, 4 o 5) del usuario sobre la película.

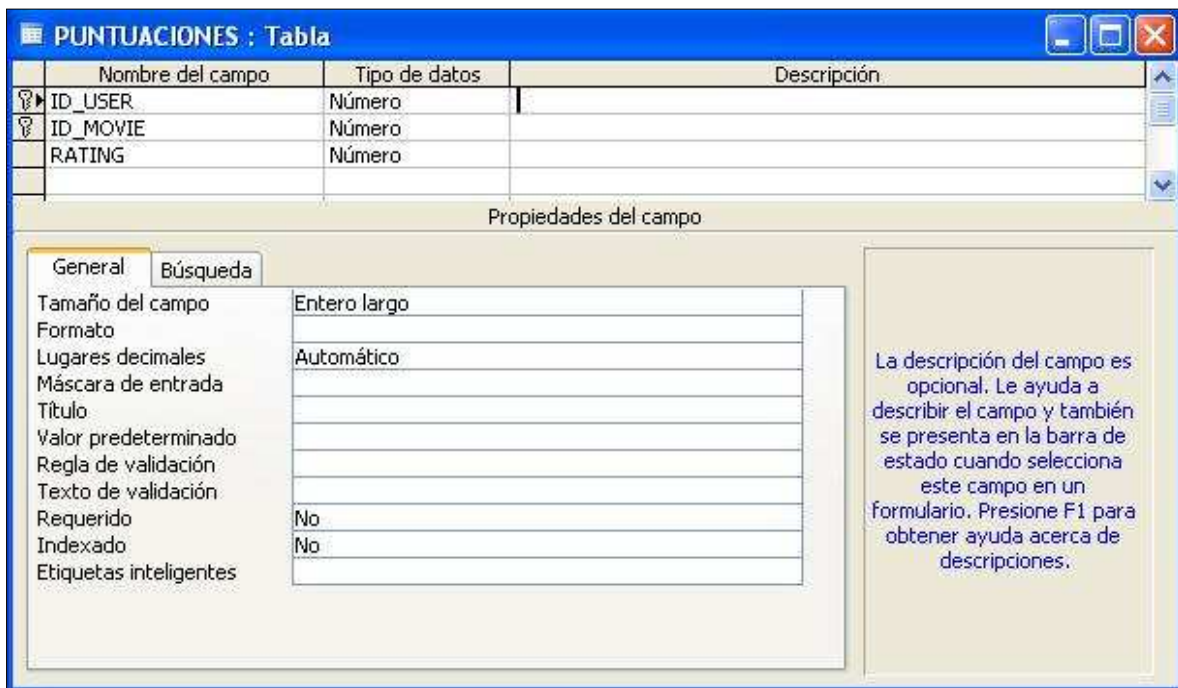


Figura 3.3.3. Vista Diseño de la tabla PUNTUACIONES

Por otra parte, en la base de datos **movieranks** existe otra tabla auxiliar llamada **ALQUILADAS** que guarda las películas alquiladas pero todavía no puntuadas de cada usuario. Esta tabla no tiene incidencia en este estudio sino para el posterior desarrollo del sistema de recomendación colaborativo por lo que será allí donde se detalle adecuadamente.

3. Métricas de evaluación

Para evaluar la bondad de los resultados de las pruebas realizadas existen multitud de métricas. Dentro de estas métricas, un tipo muy importante son las **métricas de precisión**, de las cuales existen, principalmente dos clases: las métricas de precisión estadística y las métricas de precisión de apoyo a la decisión [9].

- **Métricas de precisión estadística:** son aquellas que evalúan la precisión de un sistema de recomendación comparando las predicciones numéricas con las puntuaciones reales para cada ítem que tenga tanto puntuación como predicción. Algunas de estas métricas son el MAE (Mean Absolute Error), el RMSE (Root Mean Squared Error) o la correlación.

- **Métricas de precisión de apoyo a la decisión:** son aquellas que evalúan como de efectivamente las predicciones ayudan a los usuarios a seleccionar ítems adecuados. Algunas de las métricas de esta clase son la puntuación inversa, la sensibilidad ROC (Receiver Operating Characteristic) o la sensibilidad PRC (Precision Recall Curve).

Sin duda, elegir unas buenas métricas resulta fundamental. Para este proyecto hemos elegido dos cuyos resultados se pueden complementar bastante bien.

La primera de estas métricas es una métrica de precisión estadística llamada **MAE** (Mean Absolut Error) que es, con diferencia, la métrica de este tipo más utilizada. El MAE es una medida en valor absoluto de la desviación entre las puntuaciones reales (r) y sus predicciones (p) cuya expresión es la siguiente:

$$MAE = \frac{\sum_{h=1}^n |p_h - r_h|}{n}$$

Cuanto menor sea este MAE, que estará acotado por la amplitud del dominio de las puntuaciones, más exactas serán las predicciones permitiendo unas mejores recomendaciones.

La segunda métrica que vamos a utilizar es una métrica temporal ya que de nada nos sirve que el algoritmo de predicción proporcione predicciones muy exactas si el coste en tiempo para ofrecerlas es demasiado alto. Calcularemos el tiempo de ejecución de cada algoritmo en milisegundos (ms) siendo el mejor de estos algoritmos el que consiga ofrecer unos mejores predicciones en un menor tiempo.

3.3.1 Pruebas sobre Algoritmos Básicos

Una vez presentados los algoritmos de clasificación, las medidas de similaridad y las técnicas de predicción que se van a utilizar se puede pasar al diseño de las pruebas para comparar combinaciones de distintos valores de estos parámetros y decidir cual de ellas ofrece unos mejores resultados. En tal diseño de pruebas vamos a detallar las

especificaciones software y hardware en las que se van a realizar las pruebas, el conjunto de datos sobre el que se van a realizar y las distintas métricas de evaluación que se van a utilizar para analizar los resultados obtenidos.

3.3.1.1 Implementación de Pruebas Básicas

Como ya se ha comentado en varias ocasiones este capítulo de la memoria está dedicado al estudio comparativo de distintos algoritmos de filtrado colaborativo mediante la evaluación y análisis de los resultados de diversas pruebas propuestas. Pues bien, en este apartado se va a detallar el proceso de implementación de dichas pruebas.

El primer paso consiste en dividir el conjunto de datos detallado en el apartado anterior en dos conjuntos disjuntos independientes de usuarios: uno será el denominado **conjunto de entrenamiento** mientras que el otro, aconsejablemente más pequeño, será el denominado **conjunto de test**. A este enfoque se le conoce como una técnica de evaluación **hold-out** y suele emplearse para bases de datos relativamente grandes como la que se utiliza en este proyecto.

Sobre el conjunto de entrenamiento se aplicará el algoritmo **k-nn** para calcular el conjunto de vecinos más similares para cada uno de los ítems. Esta similaridad será calculada mediante uno de las tres **medidas de similaridad** presentadas anteriormente en esta memoria.

Este conjunto de k-vecinos es indispensable para el **cálculo de predicciones**, un cálculo que se realizará para los usuarios contenidos en el conjunto de test mediante la utilización de uno de los **algoritmos de predicción** ya estudiados.

Finalmente obtendremos una medida de la precisión de estas predicciones gracias a la medida de precisión estadística **MAE** (Mean Absolut Error) y una medida del **tiempo empleado** para realizarlas.

3.3.1.2 Evaluación de Resultados

Se han realizado 6 pruebas diferentes, ejecutándose 20 iteraciones de cada una de ellas. Para cada una de estas pruebas se han variado los valores de uno o más de los siguientes

parámetros:

- **Porcentaje Entrenamiento/ Test:** indica el porcentaje de la base de datos que se usará como conjunto de entrenamiento en tanto por uno siendo el conjunto de test el resto de la base de datos.
- **Número de vecinos:** proporciona el valor k del algoritmo **k-nn** para la construcción del conjunto de k ítems más similares para cada ítem.
- **Medida de similitud:** indica cual de las tres medidas de similitud consideradas (*distancia euclídea, coeficiente coseno o coeficiente de correlación de Pearson*) se ha utilizado.
- **Algoritmo de predicción:** indica cual de los dos algoritmos de predicción considerados se ha utilizado. Siempre que se utilice el algoritmo item+adjustment se emplearán los dos enfoques vistos previamente: *todos menos 1* (identificado como **TM1**) y *dados n* con los valores 2, 4 y 8 para dicha n (identificados respectivamente como **D2, D4, D8**).

Los valores de estos parámetros para cada una de las pruebas son los siguientes:

	PRUEBA 1	PRUEBA 2	PRUEBA 3	PRUEBA 4	PRUEBA 5	PRUEBA 6
ENT/TEST	0.8/0.2	0.8/0.2	0.6/0.4	0.8/0.2	0.8/0.2	0.8/0.2
Nº VEC.	20	20	20	20	10	40
MED. SIMILAR.	Coeficiente Coseno	Coeficiente Coseno	Coeficiente Coseno	Correlación de Pearson	Coeficiente Coseno	Coeficiente Coseno
ALG. PRED.	item+adjust.	weighted sum	item+adjust.	item+adjust.	item+adjust.	item+adjust.

Tabla 3.3.1. Valores de los parámetros de las distintas pruebas

Los resultados presentados en las siguientes tablas y gráficos se corresponden con los valores medios de las métricas de evaluación MAE y temporal para cada una de las ejecuciones ya comentadas.

3.3.1.3 Resultados de las Pruebas

PRUEBA 1

Iteración	MAE D2	Tiempo D2	MAE D4	Tiempo D4
Iteración 1	0,9091844	6,095238	0,9091844	6,4179893
Iteración 2	0,860809	5,5079365	0,8863211	6,047619
Iteración 3	0,8792336	5,714286	0,9336889	5,2486773
Iteración 4	0,8274456	6,3650794	0,87336	5,878307
Iteración 5	0,86159664	5,9312167	0,8818423	5,9312167
Iteración 6	0,88274693	6,6243386	0,8983355	6,571429
Iteración 7	0,87925327	5,724868	0,91216934	5,4497356
Iteración 8	0,8573806	6,148148	0,8834329	6,148148
Iteración 9	0,893788	5,6666665	0,9013497	5,820106
Iteración 10	0,87137294	5,5079365	0,88979936	5,2486773
Iteración 11	0,87945104	6,4550266	0,9008881	6,7777777
Iteración 12	0,8927211	4,978836	0,9204125	5,6719575
Iteración 13	0,8852441	6,3597884	0,91919035	5,3492064
Iteración 14	0,85976565	5,7830687	0,88290125	5,4497356
Iteración 15	0,87416464	5,4074073	0,916549	5,2433863
Iteración 16	0,88925016	4,724868	0,92068464	5,296296
Iteración 17	0,86703396	6,84127	0,8916349	5,724868
Iteración 18	0,85441107	5,5185184	0,8953181	5,5132275
Iteración 19	0,88827103	6,4074073	0,90811175	6,835979
Iteración 20	0,9076144	5,301587	0,92021275	5,5132275
Valores Medios	0,87603691	5,8531746	0,90226934	5,8068783

Tabla 3.3.2.a Resultados de Prueba 1

Iteración	MAE D8	Tiempo D8	MAE TM1	Tiempo TM1
Iteración 1	0,9091844	6,724868	0,8399574	0,052910052
Iteración 2	0,91598016	5,0846562	0,7961574	1,1640211
Iteración 3	0,9691272	5,031746	0,8235353	0,6878307
Iteración 4	0,9062245	5,555553	0,7579644	0,74603176
Iteración 5	0,91135806	4,6084657	0,9014391	0,7407407
Iteración 6	0,91679156	7,047619	0,8413623	2,1693122
Iteración 7	0,9460321	5,7777777	0,8255099	0,6878307
Iteración 8	0,91131616	5,6137567	0,847643	1,1111112
Iteración 9	0,91731286	5,571429	0,9025	0,52910054
Iteración 10	0,9165067	5,555553	0,8256069	0,74603176
Iteración 11	0,9398318	6,4232802	0,8389686	1,4285715
Iteración 12	0,94389373	4,878307	0,82146186	0,84656084
Iteración 13	0,9437338	5,878307	0,8818918	0,74603176
Iteración 14	0,91171217	5,4074073	0,9714764	1,1640211
Iteración 15	0,93262964	4,825397	0,8545881	0,7407407
Iteración 16	0,9670446	5,3439155	0,79279286	0,52910054
Iteración 17	0,9212311	5,3439155	0,84542274	0,52910054
Iteración 18	0,91266114	5,026455	0,9055007	0,31746033
Iteración 19	0,94462746	6,94709	0,8242213	0,8994709
Iteración 20	0,9413919	5,1904764	0,83496475	0,47619048
Valores Medios	0,92892955	5,591799	0,84664824	0,81560847

Tabla 3.3.2.b Resultados de Prueba 1

Como una tabla de tal tamaño puede ser poco clarificadora a la hora de extraer cualquier tipo de conclusión, se van a presentar los datos en dos gráficos de líneas (uno para cada una de las métricas de evaluación) para comprobar su comportamiento a través de las iteraciones:

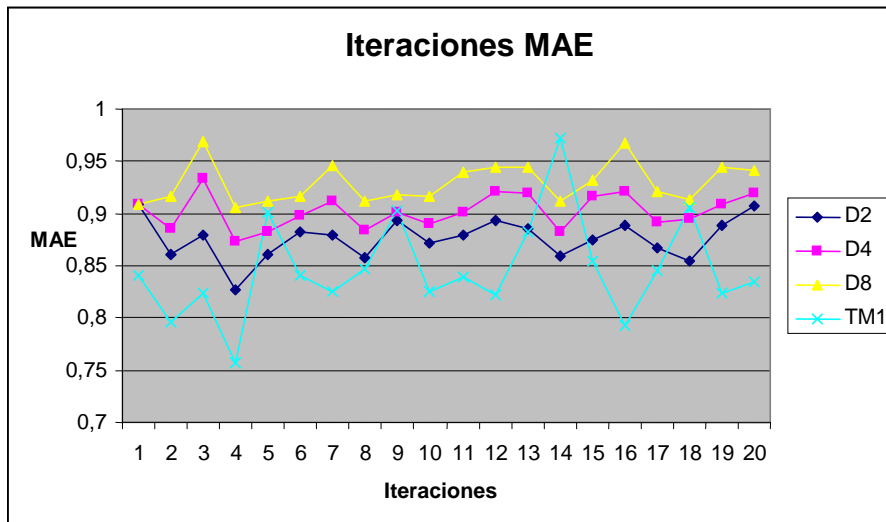


Figura 3.3.4. Gráfico de MAE de Prueba 1

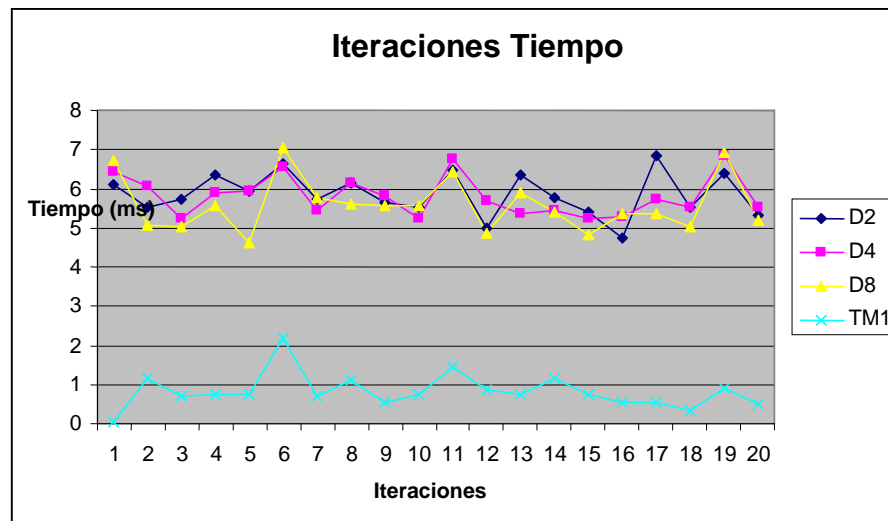


Figura 3.3.5 Gráfico de Tiempo de Prueba 1

Los gráficos anteriores ayudan a comprender el comportamiento de cada uno de los enfoques a lo largo de las ejecuciones pero no dejan claro cual de ellos es el mejor.

Para resolver esta situación se han calculado los valores medios de cada enfoque para las dos métricas empleadas y se presentan en los siguientes gráficos de barras:

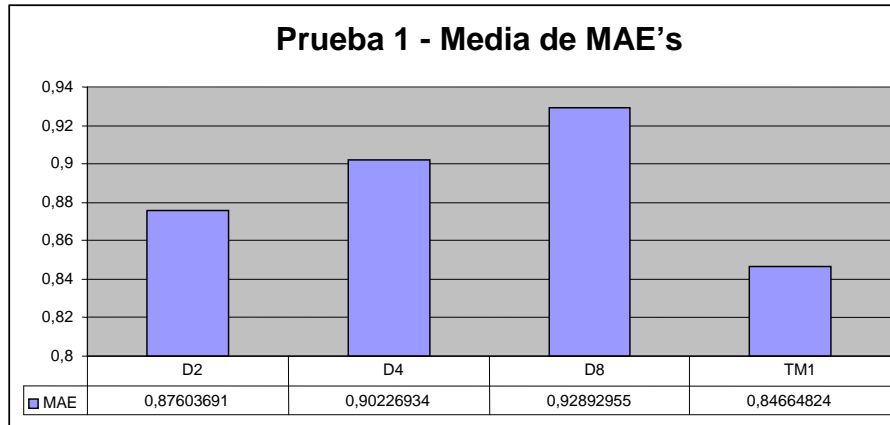


Figura 3.3.6. Media MAE de Prueba 1

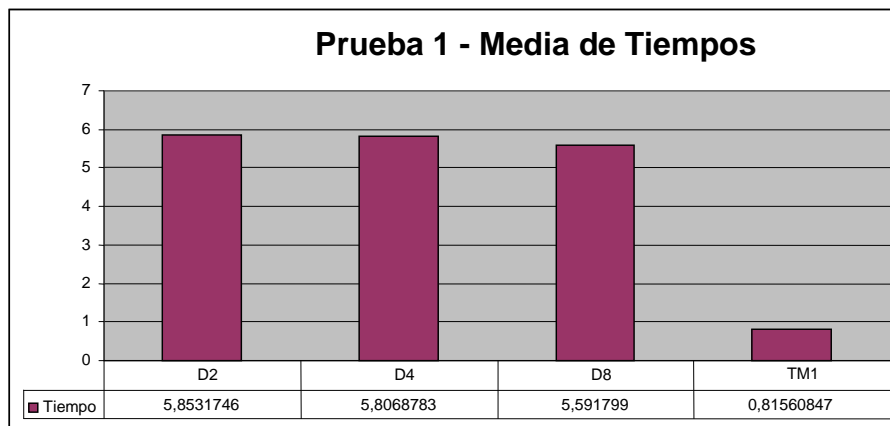


Figura 3.3.7. Media de Tiempo de Prueba 1

A la vista de estos dos gráficos se observa que el enfoque **Todos Menos 1** obtiene unos resultados mejores en términos de tiempo que los demás enfoques y, aunque con unos márgenes mucho más igualados, también obtiene unos mejores resultados en cuanto a precisión de la predicción. Esta vez, las dos métricas no se contradicen y señalan al mismo enfoque como el mejor. Por lo tanto, a la hora de elegir qué prueba se ha saldado con mejores resultados, tomaremos en consideración el enfoque **Todos Menos 1**.

PRUEBA 2

Iteración	MAE	Tiempo
Iteración 1	0,7471493	0,052910052
Iteración 2	0,76753604	0,052910052
Iteración 3	0,7380677	0,052910052
Iteración 4	0,758815	0
Iteración 5	0,6601054	0
Iteración 6	0,72152346	0,052910052
Iteración 7	0,75190395	0,052910052
Iteración 8	0,80621403	0,052910052
Iteración 9	0,6928996	0
Iteración 10	0,67935616	0
Iteración 11	0,7982117	0,52910054
Iteración 12	0,7453146	0
Iteración 13	0,6918417	0,105820104
Iteración 14	0,7857819	0,05820106
Iteración 15	0,7051473	0
Iteración 16	0,7463787	0
Iteración 17	0,67296684	0
Iteración 18	0,77536833	0
Iteración 19	0,76340055	0
Iteración 20	0,6837682	0
Valores Medios	0,73458752	0,050529101

Tabla 3.3.3. Resultados de Prueba 2

Una vez obtenida esta tabla se van a extraer los datos en diversos gráficos que mostrarán el comportamiento del algoritmo a lo largo de las ejecuciones y el valor medio de las métricas MAE y temporal para el mismo:

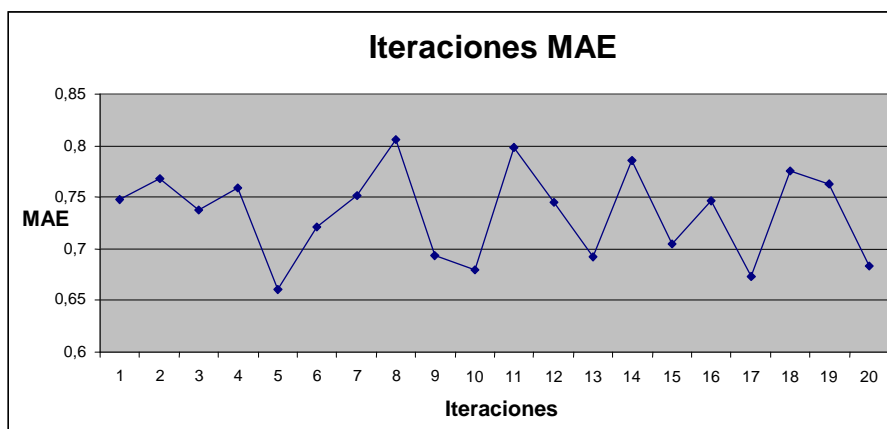


Figura 3.3.8. Gráfico de MAE de Prueba 2

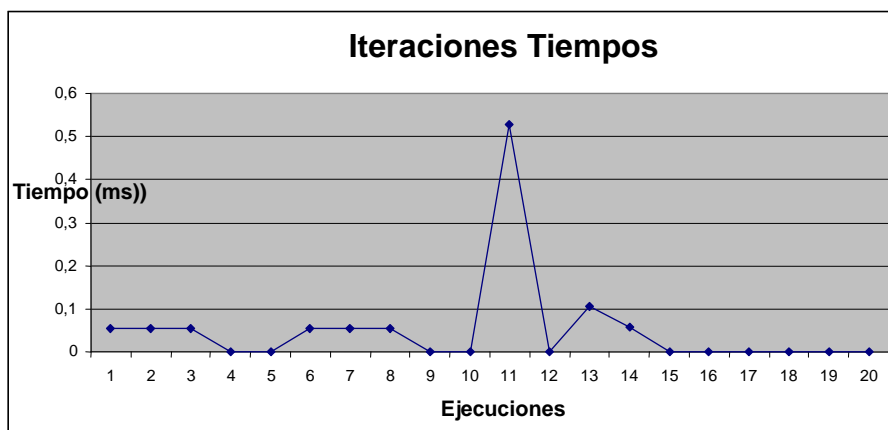


Figura 3.3.9. Gráfico de Tiempo de Prueba 2

Se observa que esta prueba (en la que se tienen los mismos parámetros que para **Prueba 1** salvo el algoritmo de predicción) los resultados son bastante buenos, sobre todo en el tema de tiempos donde se puede decir que las ejecuciones se realizan de forma instantánea. Sin duda tiene muchas opciones de ser la prueba con mejores resultados.

PRUEBA 3

Iteración	MAE D2	Tiempo D2	MAE D4	Tiempo D4
Iteración 1	0,89330035	5,4668436	0,9152147	5,363395
Iteración 2	0,8856924	6,1352787	0,9203394	5,97878
Iteración 3	0,87468016	5,156499	0,906191	5,2864723
Iteración 4	0,87846494	6,376658	0,8963687	6,6896553
Iteración 5	0,8706919	6,769231	0,8935741	6,888594
Iteración 6	0,8560503	6,259947	0,88483757	6,535809
Iteración 7	0,8834645	6,424403	0,9113812	6,32626
Iteración 8	0,8848531	5,7161803	0,896048	5,840849
Iteración 9	0,8686024	6,005305	0,899244	6,2519894
Iteración 10	0,90519404	6,4323606	0,94469315	6,427056
Iteración 11	0,8729384	5,4960213	0,9134295	5,8488064
Iteración 12	0,88148755	5,4668436	0,91291827	5,525199
Iteración 13	0,8999509	6,4084883	0,9231452	5,6525197
Iteración 14	0,8960156	4,888594	0,9334678	5,474801
Iteración 15	0,90800893	5,7612734	0,9257292	5,204244
Iteración 16	0,8512323	6,7480106	0,88089776	6,1909814
Iteración 17	0,88107586	5,31565	0,91245776	5,2572947
Iteración 18	0,8971592	5,498674	0,9191528	5,949602
Iteración 19	0,8740259	5,973475	0,89465946	5,5305037
Iteración 20	0,881792	6,4005303	0,9054176	6,244032
Valores Medios	0,88223404	5,9350133	0,90945836	5,9233422

Tabla 3.3.4.a Resultados de Prueba 3

Iteración	MAE D8	Tiempo D8	MAE TM1	Tiempo TM1
Iteración 1	0,9312285	4,8647213	0,7891631	0,85411143
Iteración 2	0,9207101	6,2175064	0,879499	0,4774536
Iteración 3	0,9188787	5,498674	0,82645065	0,6366048
Iteración 4	0,92968816	6,69496	0,87647164	1,2493368
Iteración 5	0,9066327	6,29443	0,80260146	0,928382
Iteración 6	0,9089134	6,509284	0,8005821	0,90716183
Iteración 7	0,94587654	7,0928383	0,8450893	0,928382
Iteración 8	0,9216531	5,4164457	0,81564826	0,6366048
Iteración 9	0,9222548	6,4190984	0,81764555	1,0079576
Iteración 10	0,9581149	6,3448277	0,87638116	0,6366048
Iteración 11	0,93857694	5,5782495	0,8751021	0,7161804
Iteración 12	0,94721955	5,5331564	0,8973061	1,0079576
Iteración 13	0,9404347	6,474801	0,8463662	0,7692308
Iteración 14	0,9508041	5,2572947	0,8534515	0,5835544
Iteración 15	0,944099	5,132626	0,8708212	0,7161804
Iteración 16	0,9048089	6,2413793	0,87350345	2,0477455
Iteración 17	0,91588527	5,2519894	0,825637	0,5039788
Iteración 18	0,9335893	5,204244	0,84777	0,3183024
Iteración 19	0,9238785	6,2122016	0,81878644	0,4244032
Iteración 20	0,9201863	5,657825	0,8358831	0,7161804
Valores Medios	0,92917167	5,8948276	0,84370797	0,80331568

Tabla 3.3.4.b Resultados de Prueba 3

Al igual que para las pruebas anteriores se va a proceder a extraer los datos de esta tabla a distintos gráficos para, de esta manera, poder conocer mejor su comportamiento y determinar cual de los enfoques obtiene unos mejores resultados:

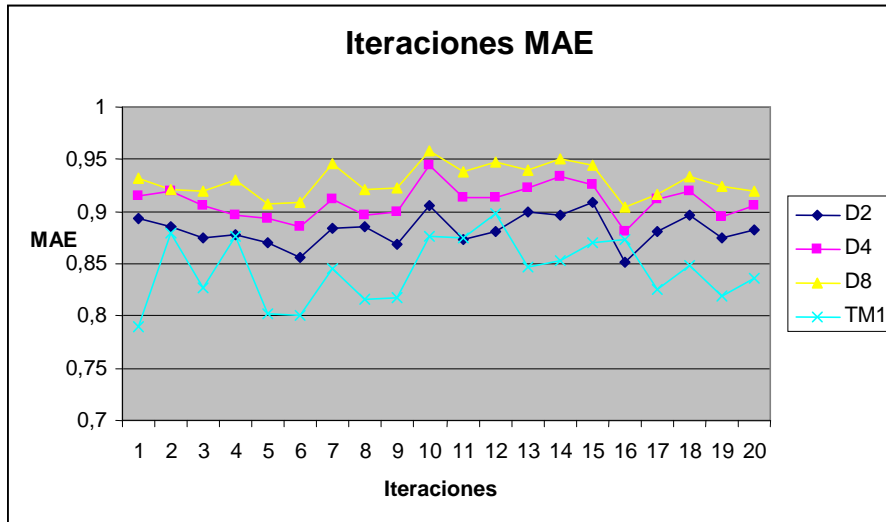


Figura 3.3.10. Gráfico de MAE de Prueba 3

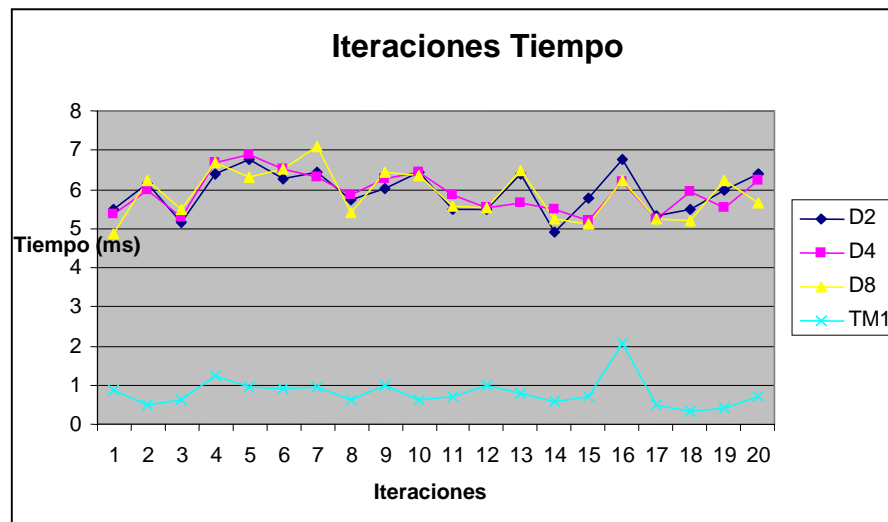


Figura 3.3.11. Gráfico de Tiempo de Prueba 3

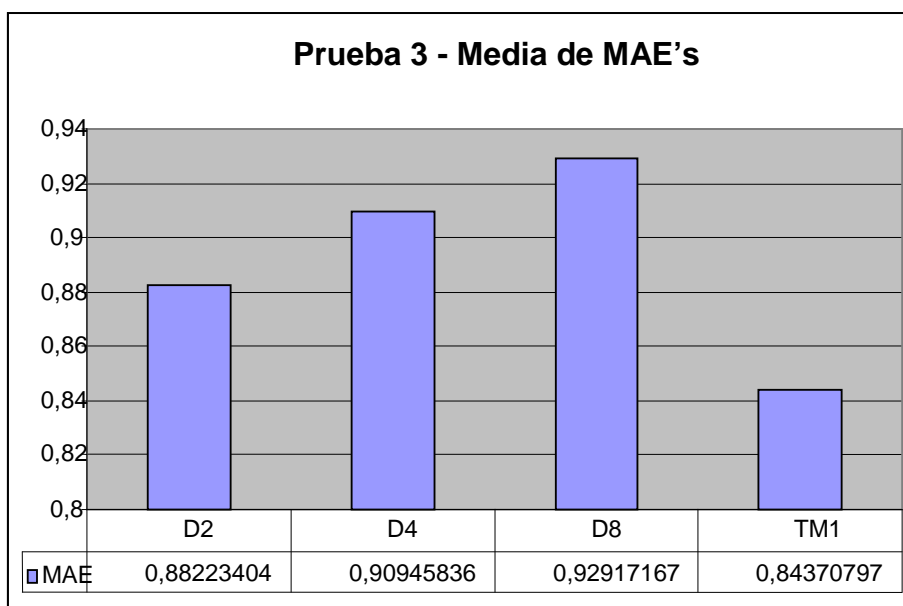


Figura 3.3.12. Media MAE de Prueba 3

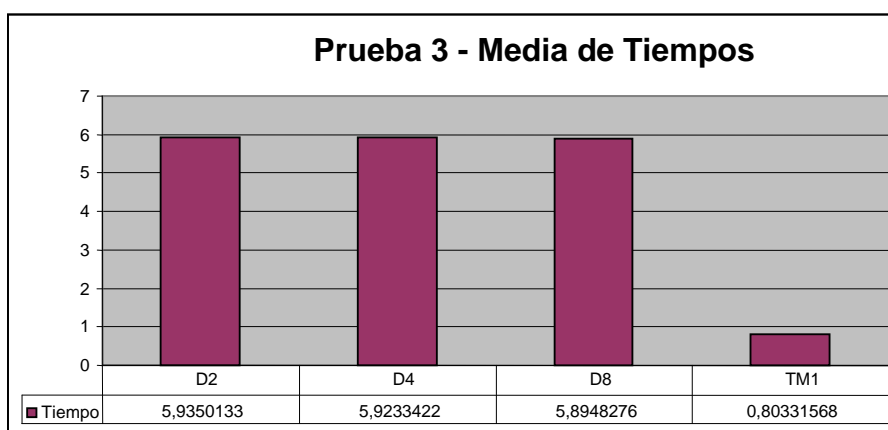


Figura 3.3.13. Media de Tiempo de Prueba 3

Con respecto a **Prueba 1** se ha variado un aspecto importante: el tamaño de los conjuntos de prueba y test. Ahora, el tamaño del conjunto de prueba es bastante menor con lo que es posible que se ha resentido la calidad del algoritmo **K-nn** y, por ende, el de la predicción. De todas formas esto se comprobará en la comparativa final para la cual para esta prueba se tomará en consideración el enfoque **Todos Menos 1** ya que resulta el que mejor resultados consigue tanto en tiempos como en precisión.

PRUEBA 4

Iteración	MAE D2	Tiempo D2	MAE D4	Tiempo D4
Iteración 1	0,8365449	6,6719575	0,8567084	6,4074073
Iteración 2	0,85205007	6,2063494	0,8568071	6,042328
Iteración 3	0,82911175	5,4021163	0,8424223	5,883598
Iteración 4	0,8712737	5,7777777	0,8861082	4,9312167
Iteración 5	0,81840605	5,7777777	0,8287324	5,714286
Iteración 6	0,8534792	5,6666665	0,8649634	5,031746
Iteración 7	0,86439955	6,296296	0,87317735	6,4550266
Iteración 8	0,8608898	6,142857	0,8669227	6,571429
Iteración 9	0,84827	5,989418	0,86184424	6,095238
Iteración 10	0,8364576	5,5132275	0,84677404	5,883598
Iteración 11	0,8581736	5,4603176	0,87718195	5,132275
Iteración 12	0,8331444	5,2433863	0,85204023	5,1957674
Iteración 13	0,8444746	6,2010584	0,8468639	5,830688
Iteración 14	0,86578757	6,6772485	0,8760373	5,989418
Iteración 15	0,8239494	5,6190476	0,8368945	6,2539682
Iteración 16	0,8278694	5,5079365	0,8446043	5,3650794
Iteración 17	0,8302364	5,6772485	0,8516222	4,814815
Iteración 18	0,8256317	5,4603176	0,8331051	4,820106
Iteración 19	0,85536635	5,291005	0,875424	4,867725
Iteración 20	0,8306874	6,73545	0,85026306	6,2486773
Valores Medios	0,84331017	5,865873	0,85642483	5,6767196

Tabla 3.3.5.a Resultados de Prueba 4

Iteración	MAE D8	Tiempo D8	MAE TM1	Tiempo TM1
Iteración 1	0,8780762	6,5185184	0,8607042	0,95238096
Iteración 2	0,8824493	6,1957674	0,91682166	0,47619048
Iteración 3	0,8596457	5,994709	0,86716366	0,26455027
Iteración 4	0,90868163	5,6666665	0,8683027	0,63492066
Iteración 5	0,85602224	5,883598	0,94376343	1,3227513
Iteración 6	0,8837512	4,5608463	0,9014174	0,64021164
Iteración 7	0,90054226	5,7883596	0,7685422	0,47619048
Iteración 8	0,8894639	5,4550266	0,8442674	0,21164021
Iteración 9	0,87639487	5,296296	0,7600607	0,31746033
Iteración 10	0,87195504	4,873016	0,8615474	0,31746033
Iteración 11	0,88356274	5,7830687	0,9248854	0,63492066
Iteración 12	0,86891615	5,3492064	0,9355303	0,105820104
Iteración 13	0,87992805	6,571429	0,8092929	0,26455027
Iteración 14	0,8990298	5,984127	0,89334834	0,42328042
Iteración 15	0,85647357	5,132275	0,912916	0,4814815
Iteración 16	0,86261165	4,5502644	0,8746802	0,52910054
Iteración 17	0,88166857	6,037037	0,8394401	0,26455027
Iteración 18	0,8530259	5,031746	0,8833269	0,63492066
Iteración 19	0,87770987	5,4708996	0,9561469	0,52910054
Iteración 20	0,8605016	6,2486773	0,8289949	0,47619048
Valores Medios	0,87652051	5,6195767	0,87255763	0,497883605

Tabla 3.3.5.b Resultados de Prueba 4

Al igual que para las pruebas anteriores se va a representar los datos de esta tabla a distintos gráficos para, de esta manera, poder conocer mejor su comportamiento y determinar cuál de los enfoques obtiene unos mejores resultados:

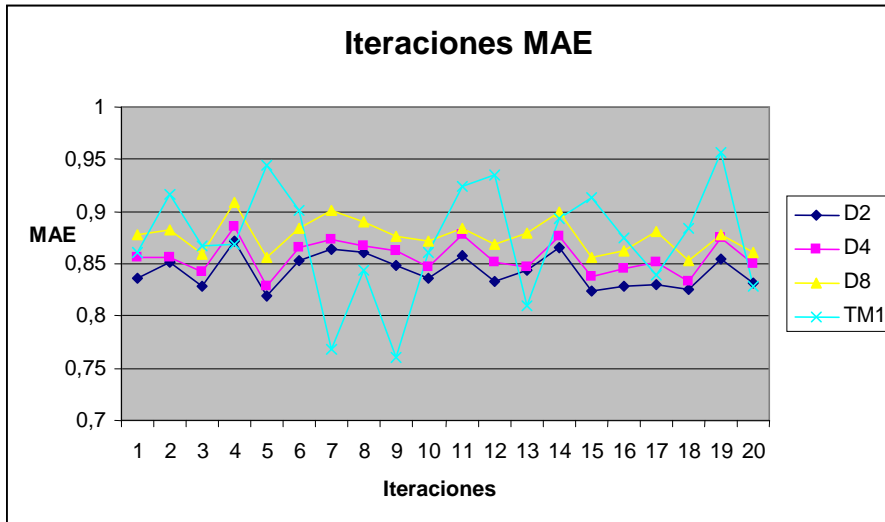


Figura 3.3.14. Gráfico de MAE de Prueba 4

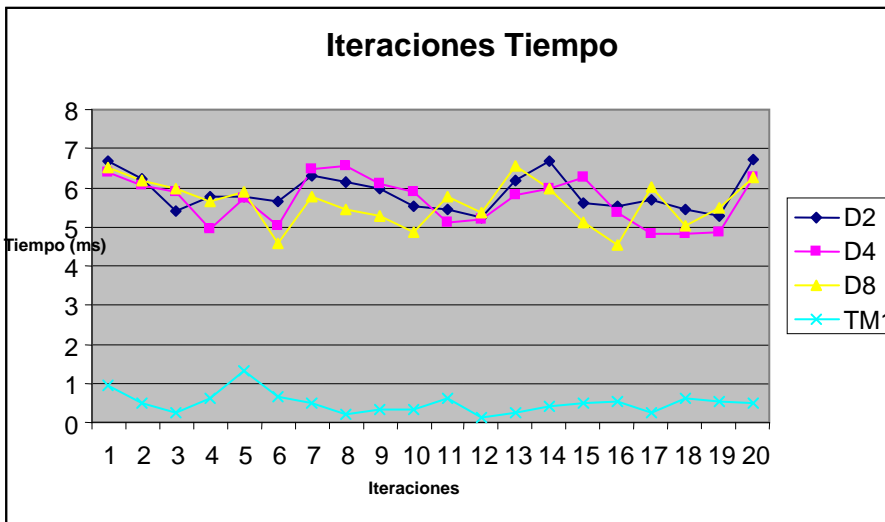


Figura 3.3.15. Gráfico de Tiempo de Prueba 4

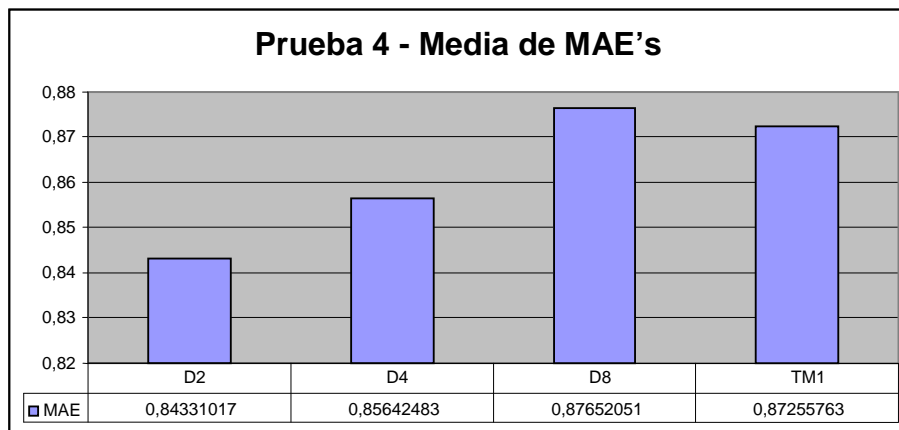


Figura 3.3.16. Media MAE de Prueba 4

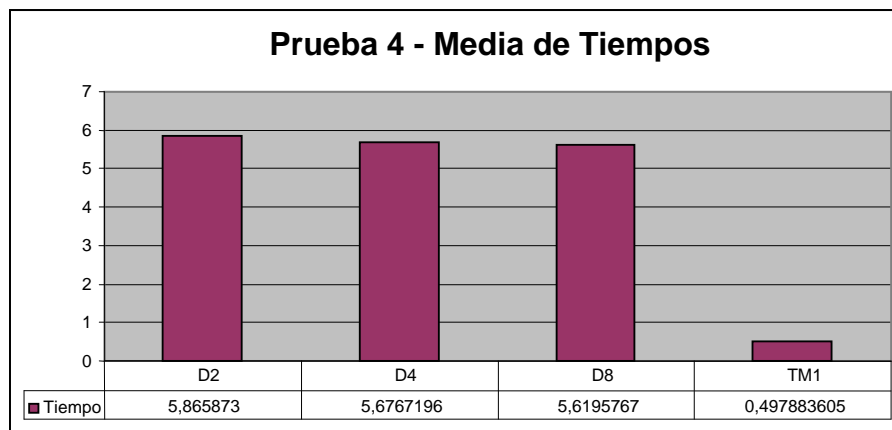


Figura 3.3.17. Media de Tiempo de Prueba 4

Para esta prueba se ha optado por implementar como métrica de similaridad el coeficiente de **correlación de Pearson** manteniendo el resto de parámetros como en Prueba 1 y nos encontramos por primera vez con que las métricas de evaluación difieren en sus resultados: en cuanto al tiempo el enfoque Todos Menos 1 resulta claramente el mejor pero, sin embargo, en cuanto a la precisión MAE, el enfoque Datos 2 obtiene un valor medio mejor mientras que el Todos Menos 1 obtiene la peor media de todos los enfoques y su comportamiento sufre muchos altibajos.

Ante esta situación, se hace indispensable el seguir una norma pre-establecida para decidir cual de los dos enfoques elegir y se ha decidido que esta norma sea la siguiente: *se elegirá siempre el enfoque que obtenga unas mejores predicciones salvo que estas predicciones las obtenga en unos tiempos mucho mayores, del orden de las varias decenas de milisegundos,*

que el resto.

Es siguiendo esta norma como se decide que sea el enfoque **Dados 2** el que se tenga en consideración para la comparativa final entre pruebas.

PRUEBA 5

Iteración	MAE D2	Tiempo D2	MAE D4	Tiempo D4
Iteración 1	0,86696905	4,825397	0,8971425	5,984127
Iteración 2	0,845517	4,3439155	0,86411136	5,6243386
Iteración 3	0,8430198	5,0846562	0,85652953	4,984127
Iteración 4	0,82992387	5,883598	0,8434733	5,185185
Iteración 5	0,85896873	5,2433863	0,8873888	4,9259257
Iteración 6	0,84033525	5,2380953	0,8577663	4,5502644
Iteración 7	0,84013665	5,1904764	0,8524127	5,6666665
Iteración 8	0,85781723	5,031746	0,8730222	5,4074073
Iteración 9	0,84813744	6,142857	0,89451057	5,1904764
Iteración 10	0,8629659	4,714286	0,8885409	5,4126983
Iteración 11	0,85142136	5,089947	0,86582416	4,7671957
Iteración 12	0,83779997	5,4074073	0,8591151	5,2486773
Iteración 13	0,87188923	4,9206347	0,89007473	4,7671957
Iteración 14	0,86474746	5,137566	0,8874667	5,4603176
Iteración 15	0,8865382	4,6137567	0,90608037	5,1957674
Iteración 16	0,84064746	4,867725	0,85337394	5,2539682
Iteración 17	0,85982704	5,296296	0,88465124	5,4708996
Iteración 18	0,84678316	5,5608463	0,85996443	5,3492064
Iteración 19	0,85465544	5,1957674	0,86832386	4,6613755
Iteración 20	0,8792419	5,291005	0,8904219	4,7671957
Valores Medios	0,85436711	5,1539683	0,87400973	5,1936508

Figura 3.3.6.a Resultados de Prueba 5

Iteración	MAE D8	Tiempo D8	MAE TM1	Tiempo TM1
Iteración 1	0,9141342	3,7089946	0,9072662	2,5978837
Iteración 2	0,8763314	3,973545	0,8406394	3,074074
Iteración 3	0,8997373	4,6031747	0,9131363	2,2275133
Iteración 4	0,85594904	5,137566	0,76707053	0,31746033
Iteración 5	0,90763444	4,6719575	0,8459521	1,0582011
Iteración 6	0,88309723	5,095238	0,87210196	3,8730159
Iteración 7	0,88226414	4,2380953	0,9171906	1,3280423
Iteración 8	0,9017307	4,5502644	0,922505	1,3333334
Iteración 9	0,90415	4,978836	0,93261915	0,8518519
Iteración 10	0,9142797	4,4973545	0,89275736	2,867725
Iteración 11	0,88185364	4,714286	0,9269985	2,3280423
Iteración 12	0,8734808	4,708995	0,85118383	3,8201058
Iteración 13	0,9353956	4,3544974	0,8598805	1,3227513
Iteración 14	0,90103316	4,978836	0,88086855	0,5873016
Iteración 15	0,91948307	5,132275	0,80939966	2,9629629
Iteración 16	0,90312403	5,3544974	0,8320645	3,915344
Iteración 17	0,90902203	4,867725	0,92881316	2,6984127
Iteración 18	0,88195854	4,3968253	0,7908301	2,4444444
Iteración 19	0,90531075	5,031746	0,8606862	2,6560845
Iteración 20	0,923175	4,4126983	0,9083502	1,7460318
Valores Medios	0,89865724	4,6703704	0,87301569	2,20052911

Figura 3.3.6.b Resultados de Prueba 5

Al igual que para las pruebas anteriores se va a proceder a extraer los datos de esta tabla a distintos gráficos para, de esta manera, poder conocer mejor su comportamiento y determinar cual de los enfoques obtiene unos mejores resultados:

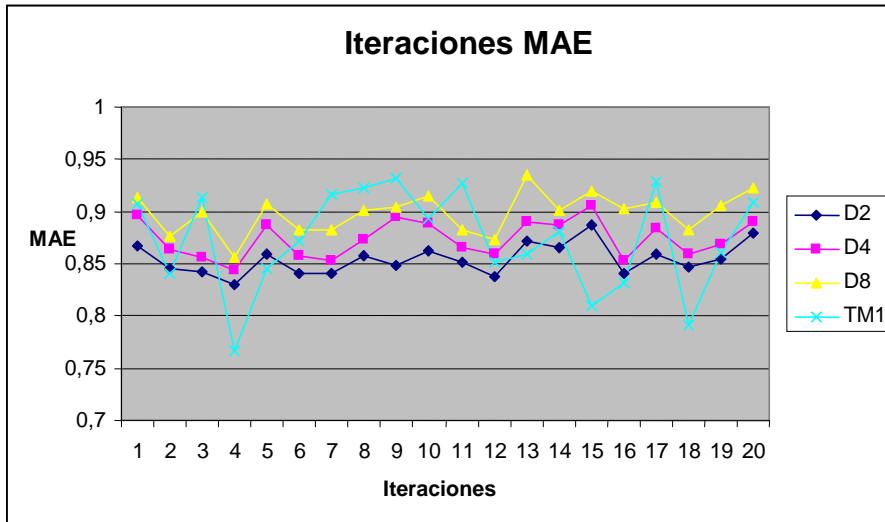


Figura 3.3.18. Gráfico de MAE de Prueba 5

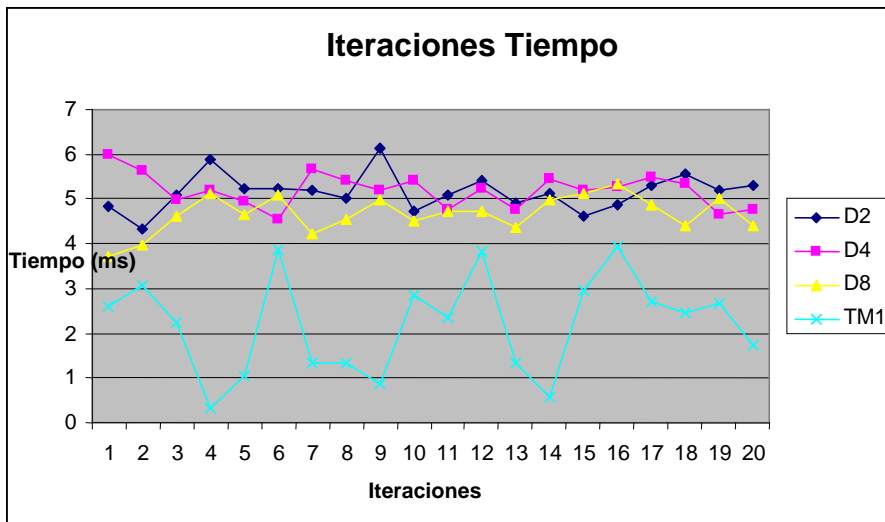


Figura 3.3.19. Gráfico de Tiempo de Prueba 5

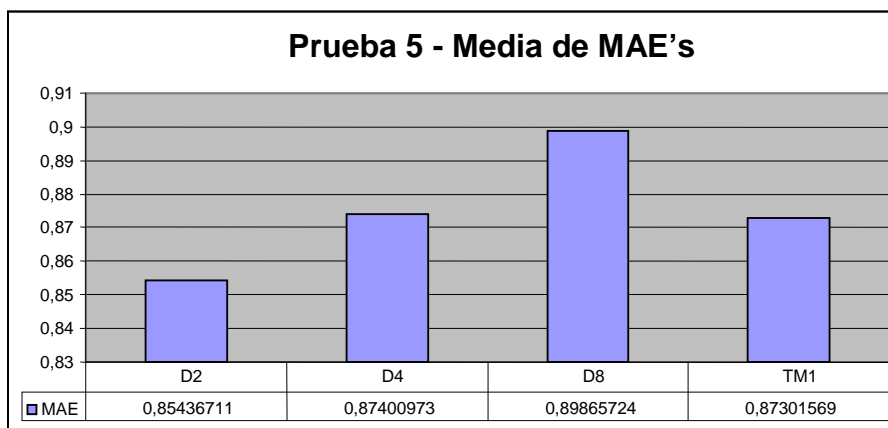


Figura 3.3.20. Media MAE de Prueba 5

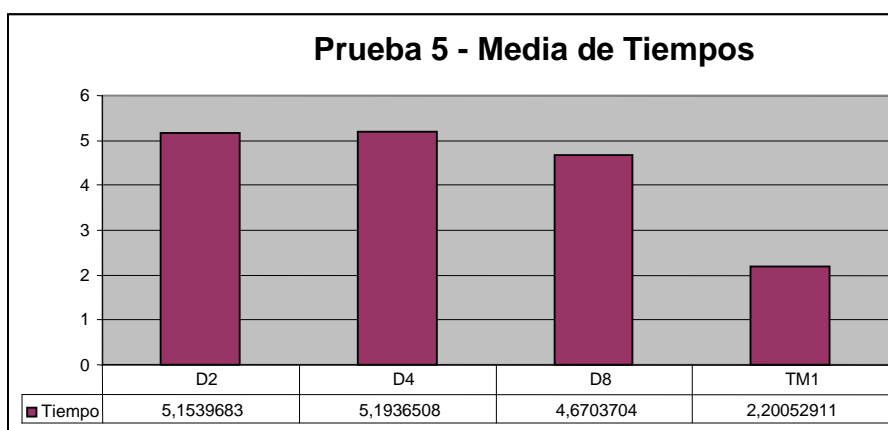


Figura 3.3.21. Media de Tiempo de Prueba 5

Tanto para esta prueba como para la siguiente hemos variado el **número de vecinos** que se calcularán para cada uno de los ítems del conjunto de entrenamiento para comprobar como afecta el tamaño de este conjunto de vecinos en la predicción. En esta prueba se ha pasado de 20 a 10 vecinos y nos encontramos con que las dos métricas de evaluación vuelven a diferir: el enfoque Datos 2 realiza mejores y más constantes predicciones mientras que Todos Menos 1 obtiene los mejores tiempos de ejecución. Como en la prueba anterior antepone la precisión al tiempo ya que las diferencias en milisegundos son inapreciables y tomamos en consideración para la comparativa final al enfoque **Datos 2**.

PRUEBA 6

Iteración	MAE D2	Tiempo D2	MAE D4	Tiempo D4
Iteración 1	0,90194535	5,6137567	0,9330488	5,132275
Iteración 2	0,9283263	4,6666665	0,93042356	4,296296
Iteración 3	0,91019446	4,878307	0,9588665	5,1957674
Iteración 4	0,9023815	5,6243386	0,92250836	5,137566
Iteración 5	0,9200539	5,4021163	0,96138096	5,4603176
Iteración 6	0,92652965	4,285714	0,9868059	5,719577
Iteración 7	0,91061544	5,0793653	0,9536638	5,5608463
Iteración 8	0,92507315	5,2433863	0,9536549	5,4074073
Iteración 9	0,9170559	4,7724867	0,9275307	5,4550266
Iteración 10	0,96115494	5,0846562	0,97623384	5,5079365
Iteración 11	0,9487743	4,4444447	0,9399986	3,8148148
Iteración 12	0,9007503	5,0846562	0,9519072	5,4074073
Iteración 13	0,928964	5,1904764	0,95691574	5,1957674
Iteración 14	0,89666265	5,5079365	0,9058138	4,142857
Iteración 15	0,87876725	4,978836	0,9087125	5,0846562
Iteración 16	0,90710205	5,0846562	0,9318752	4,724868
Iteración 17	0,88921064	4,9259257	0,9527363	4,5132275
Iteración 18	0,92218	3,925926	0,9311507	4,5608463
Iteración 19	0,8537299	5,7777777	0,9024189	5,5026455
Iteración 20	0,88576114	6	0,92661893	4,5026455
Valores Medios	0,91076164	5,0785715	0,94061326	5,0161376

Tabla 3.3.7.a Resultados de Prueba 6

Iteración	MAE D8	Tiempo D8	MAE TM1	Tiempo TM1
Iteración 1	0,9224521	5,148148	0,8785218	2,3862433
Iteración 2	0,93445486	4,2380953	0,8509358	2,8042328
Iteración 3	0,9601047	5,3492064	0,78445894	2,862434
Iteración 4	0,9398476	5,5079365	0,8758615	2,910053
Iteración 5	0,9507927	5,291005	0,83165884	5,5132275
Iteración 6	0,966225	5,148148	0,8477021	2,3862433
Iteración 7	0,9663543	5,2063494	0,7904137	3,021164
Iteración 8	0,97372794	5,301587	0,83304375	4,5026455
Iteración 9	0,95398974	4,984127	0,876602	2,8042328
Iteración 10	0,9649713	4,724868	0,8774581	3,1216931
Iteración 11	0,95247054	4,984127	0,8377185	3,132275
Iteración 12	0,96031195	5,3544974	0,799156	3,1746032
Iteración 13	0,95505124	5,3439155	0,82683885	3,2910054
Iteración 14	0,91789865	5,0846562	0,8056894	3,7619047
Iteración 15	0,9308502	5,301587	0,85651004	2,9682539
Iteración 16	0,9627555	5,1904764	0,81148046	3,068783
Iteración 17	0,9284814	4,5555553	0,8445794	2,6455026
Iteración 18	0,969939	4,5555553	0,8567844	2,8042328
Iteración 19	0,92815894	5,6243386	0,7830478	2,915344
Iteración 20	0,9053871	4,9206347	0,918258	2,915344
Valores Medios	0,94721124	5,0907407	0,83933597	3,1494709

Tabla 3.3.7.b Resultados de Prueba 6

Representando los datos de esta tabla en distintos gráficos nos ayudan a conocer mejor su comportamiento y determinar cual de los enfoques obtiene unos mejores resultados:

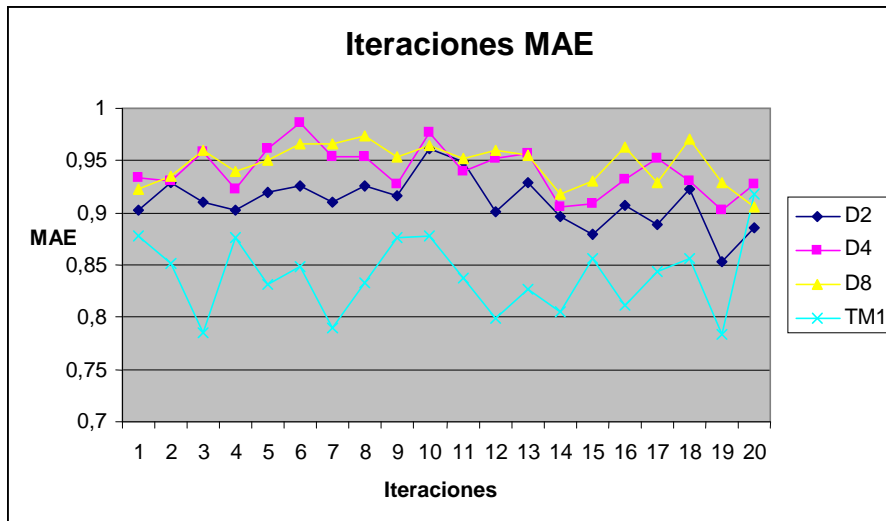


Figura 3.3.22. Gráfico de MAE de Prueba 6

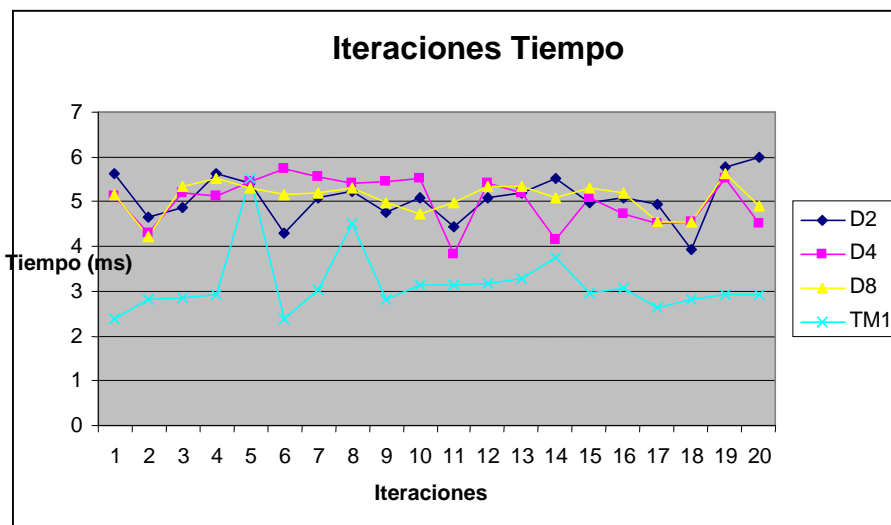


Figura 3.3.23. Gráfico de Tiempo de Prueba 6

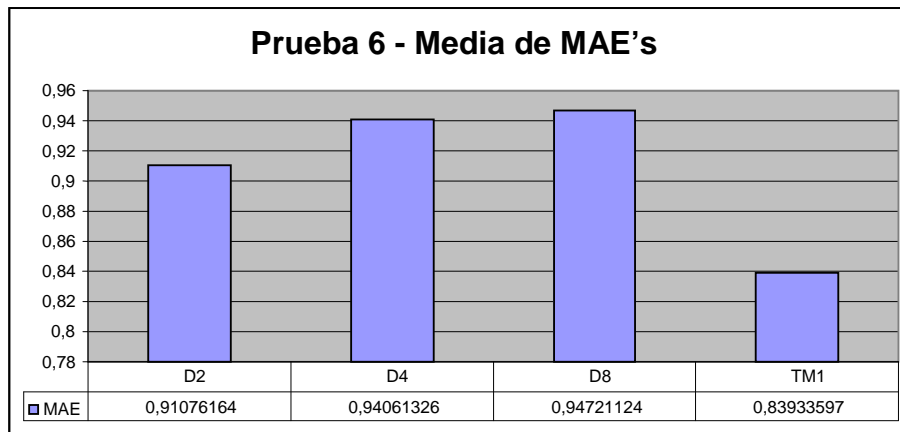


Figura 3.3.24. Media MAE de Prueba 6

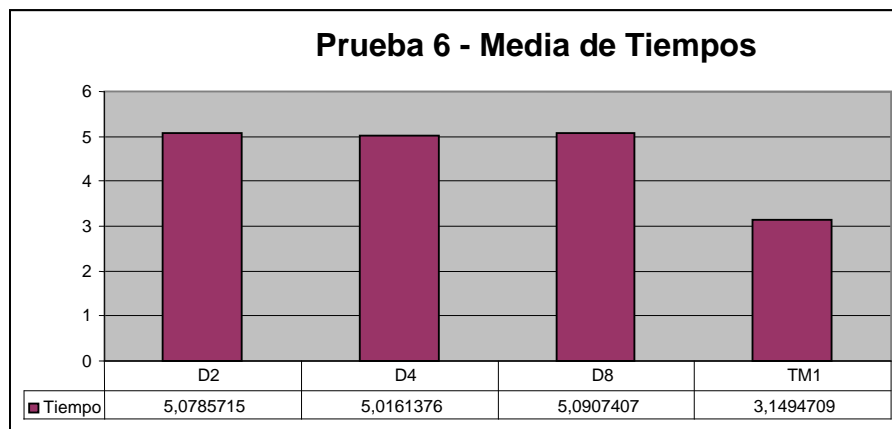


Figura 3.3.25. Media de Tiempo de Prueba 6

Para esta prueba se ha aumentado el **número de vecinos** de 20 a 40 y esta vez las dos métricas de evaluación coinciden en que, aunque por márgenes estrechos, el enfoque **Todos Menos 1** realiza las mejores predicciones en el mejor tiempo por lo que este enfoque será el que se tome en consideración para la comparativa final entre las distintas pruebas.

3.3.1.4 Comparativa entre las Pruebas

Una vez realizadas las pruebas y analizados sus resultados llega el momento de comparar estos resultados y determinar cual prueba los ha obtenido mejores para la posterior implementación de un sistema de recomendación colaborativo con los valores de los parámetros de dicha prueba.

Como en las pruebas que han implementado el algoritmo de predicción **item average + adjustment** se presentan distintos resultados según el enfoque seguido vamos a recordar cual de estos enfoques es el tomado en consideración para cada prueba:

PRUEBA	ENFOQUE
PRUEBA 1	Todos Menos 1
PRUEBA 3	Todos Menos 1
PRUEBA 4	Dados 2
PRUEBA 5	Dados 2
PRUEBA 6	Todos Menos 1

Tabla 3.3.8. Enfoque elegido para cada prueba

En el siguiente gráfico de barras se muestra con claridad los resultados comparados de todas las pruebas para la métrica MAE:

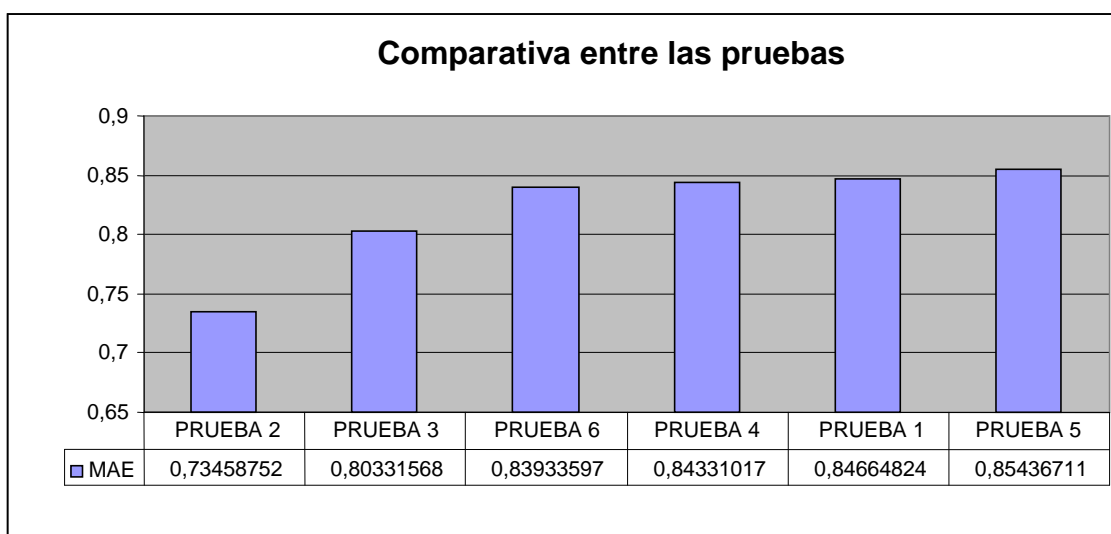


Figura 3.1.26. Comparativa de todas las pruebas

A la vista del gráfico queda claro que tanto para la métrica de precisión MAE como para la métrica temporal es la **Prueba 2** la que obtiene mejores resultados utilizando un algoritmo de filtrado colaborativo básico con los siguientes parámetros:

	PRUEBA2
ENT/TEST	0.8/0.2
Nº VEC.	20
MED. SIMILAR.	Coeficiente Coseno
ALG. PRED.	weighted sum

Tabla 3.3.9. Valores del Sistema de Recomendación Colaborativo

3.3.2 Pruebas con Algoritmos de Filtrado Colaborativos Mejorados

Según la literatura estudiada [2], proponemos unas modificaciones, para los algoritmos presentados en 3.2.2, que pueden mejorar dichos resultados.

Describiremos dichas mejoras en este punto y en el siguiente apartado comentaremos su eficiencia realizando un estudio comparativo entre todas las ejecuciones realizadas.

A la vista de los resultados de las pruebas del punto 3.3.1, emplearemos los parámetros para los cuales se han obtenido los mejores resultados. Estos son:

- **Porcentaje Entrenamiento/ Test:** 0.8 / 0.2.
- **Número de vecinos (k-nn):** 20.

Mediante la utilización de dichos valores, se han realizado 7 pruebas diferentes con los algoritmos de mejora propuestos. Ejecutándose igualmente durante 20 iteraciones cada una de ellas.

Para cada una de estas pruebas se han variado los valores de uno o más de los siguientes parámetros:

- **Algoritmo de Mejora para el cálculo de la Medida de similaridad:** indica cual de los tres algoritmos de mejora considerados se ha utilizado (*voto por defecto, frecuencia inversa de usuario o frecuencia directa de usuario*).

- **Algoritmo de predicción:** indica cual de los dos algoritmos de predicción considerados se ha utilizado. Siempre que se utilice el algoritmo *item+adjustment* se emplearán los dos enfoques vistos previamente: *todos menos 1* (identificado como **TM1**) y *dados n* con los valores 2, 4 y 8 para dicha n (identificados respectivamente como **D2, D4, D8**).
- Para la aplicación del algoritmo de mejora de Amplificación de Casos, como se ha comentado en apartados anteriores, se utilizará como medida de similaridad el *coeficiente coseno* y como algoritmo de predicción una modificación del *item+adjustment*.

Los valores de estos parámetros para cada una de las pruebas son los siguientes:

	ALG. MEJORA		ALG. PRED.
PRUEBA MEJORA 1	Voto por Defecto	d=3 k'=30	item+adjust.
PRUEBA MEJORA 2	Voto por Defecto	d=3 k'=30	weighted sum
PRUEBA MEJORA 3	Frecuencia Inversa		item+adjust.
PRUEBA MEJORA 4	Frecuencia Inversa		weighted sum
PRUEBA MEJORA 5	Frecuencia Directa		item+adjust.
PRUEBA MEJORA 6	Frecuencia Directa		weighted sum
PRUEBA MEJORA 7	Amplificación de Casos		item+adjust.
Siendo k' el nº de votos por defecto adicionales			

Tabla 3.3.10. Valores de los parámetros de las distintas pruebas para las mejoras

Los resultados presentados en las siguientes tablas y gráficos se corresponden con los valores medios de las métricas de evaluación MAE y temporal para cada una de las ejecuciones utilizadas en el estudio de los Algoritmos de Filtrado Colaborativo básicos.

3.3.2.1 Resultados de las Mejoras

- PRUEBA MEJORA 1. Voto por defecto con predicción item+adjustment

Ejecución	MAE D2	Tiempo D2	MAE D4	Tiempo D4
Iteración 1	0,8439539	4,6084657	0,8567988	4,7671957
Iteración 2	0,8463602	4,4550266	0,8701862	4,7724867
Iteración 3	0,8471604	4,3968253	0,8539911	5,185185
Iteración 4	0,83285224	4,6666665	0,85506845	4,3439155
Iteración 5	0,8513231	5,3492064	0,85956657	4,7671957
Iteración 6	0,84610915	5,2433863	0,8529315	4,5026455
Iteración 7	0,83623904	4,285714	0,84563434	4,285714
Iteración 8	0,80206203	4,6137567	0,8172926	4,6666665
Iteración 9	0,82985514	4,9259257	0,83420616	4,7671957
Iteración 10	0,8357056	5,026455	0,84890306	4,6719575
Iteración 11	0,8506868	4,0846562	0,8609881	4,285714
Iteración 12	0,8461248	4,820106	0,8564458	4,867725
Iteración 13	0,842059	4,6137567	0,85527956	4,5555553
Iteración 14	0,84957314	4,719577	0,8555855	4,126984
Iteración 15	0,8423892	5,148148	0,85791814	5,724868
Iteración 16	0,8384843	4,4021163	0,84839964	4,296296
Iteración 17	0,8502215	4,3439155	0,86436146	4,820106
Iteración 18	0,8343463	4,026455	0,8436609	4,7724867
Iteración 19	0,8761768	5,571429	0,87744623	5,2433863
Iteración 20	0,82905304	4,873016	0,831476	5,1904764
Valores Medios	0,841536784	4,708730195	0,852307006	4,730687775

Tabla 3.3.11.a Resultados de mejora Prueba 1

Ejecución	MAE D8	Tiempo D8	MAE TM1	Tiempo TM1
Iteración 1	0,87805283	5,2010584	0,80993634	0,63492066
Iteración 2	0,8728155	4,3915343	0,9087805	0,47619048
Iteración 3	0,8747977	3,7566137	0,8617176	0,75132275
Iteración 4	0,8591766	4,5555553	0,933258	0,31746033
Iteración 5	0,8920643	3,9682539	0,92272353	0,6878307
Iteración 6	0,88007545	4,4497356	0,9144269	0,31746033
Iteración 7	0,86605805	4,095238	0,8368861	0,42328042
Iteración 8	0,8275704	5,1904764	0,91608673	0,37037036
Iteración 9	0,86457956	4,825397	0,8855721	0,7989418
Iteración 10	0,8748973	4,291005	0,88923395	0,58201057
Iteración 11	0,90217847	4,5555553	0,89402854	0,42328042
Iteración 12	0,8715338	4,941799	0,83108294	0,42328042
Iteración 13	0,8817712	4,4497356	0,89302105	0,7407407
Iteración 14	0,8800728	4,984127	0,8734524	0,42328042
Iteración 15	0,872191	4,4973545	0,8093984	0,8994709
Iteración 16	0,8771987	3,862434	0,8733646	0,42328042
Iteración 17	0,8804032	4,9259257	0,86733234	0,47619048
Iteración 18	0,8597065	5,037037	0,9083487	0,37037036
Iteración 19	0,9151206	4,978836	0,9320296	0,95238096
Iteración 20	0,8564315	4,5026455	0,9499113	0,64021164
Valores Medios	0,874334773	4,57301586	0,885529581	0,556613756

Tabla 3.3.11.b Resultados de mejora Prueba 1

Para poder extraer cualquier conclusión acerca de los valores representados en la tabla, emplearemos unos gráficos de líneas (uno para cada métrica de evaluación empleada) para observar su comportamiento en las distintas ejecuciones.

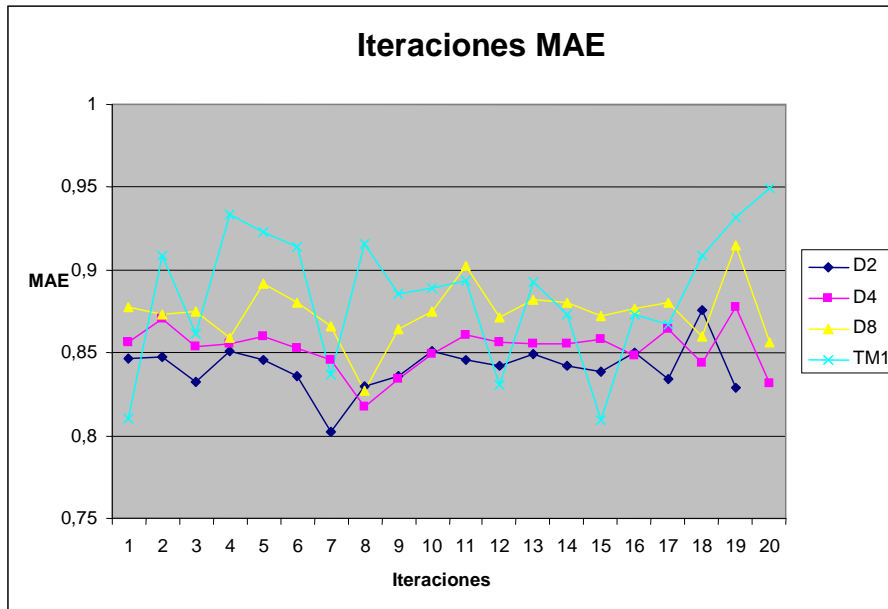


Figura 3.3.27 Gráfico de mejora de MAE de Prueba 1

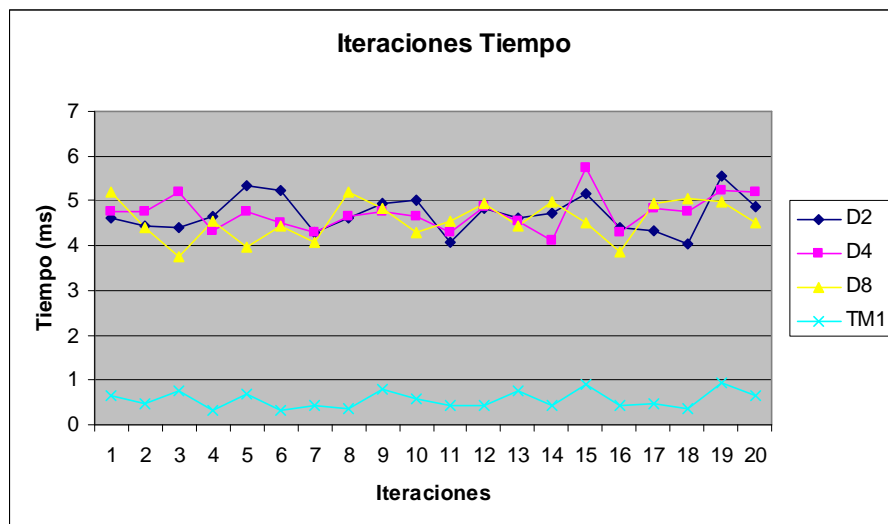


Figura 3.3.28. Gráfico de mejora de Tiempo de Prueba 1

Los gráficos anteriores intentan clarificar el comportamiento de cada uno de los enfoques en las ejecuciones pero no muestran cual de ellos es el mejor.

En los siguientes gráficos se presentan los valores medios de cada enfoque:

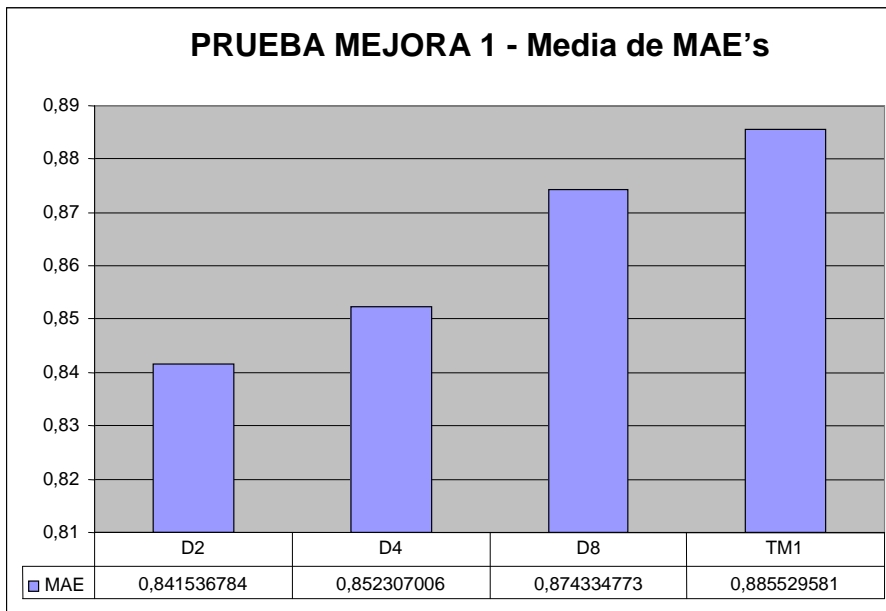


Figura 3.3.29. Media de mejora de MAE de Prueba 1

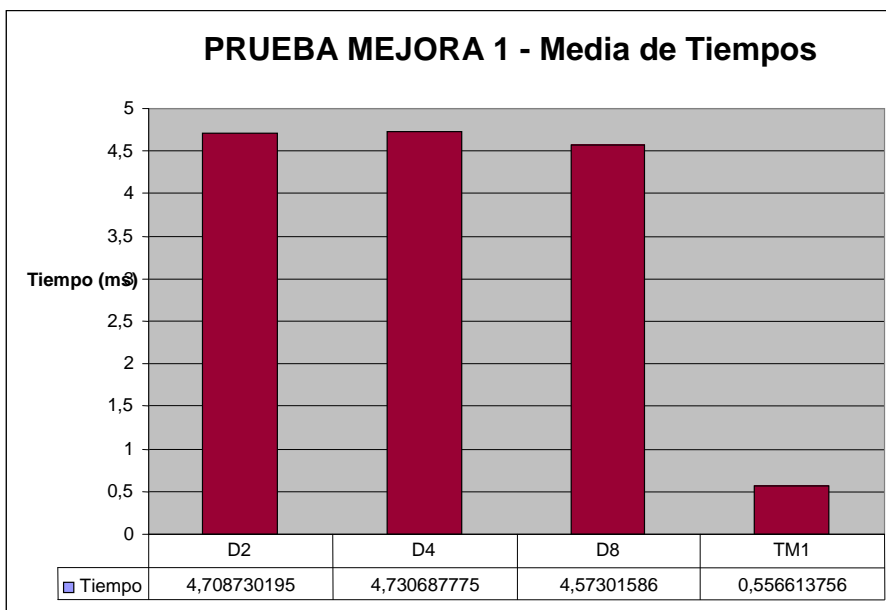


Figura 3.3.30 Media de mejora de Tiempo de Prueba 1

Nos encontramos el caso particular en el que las métricas de evaluación difieren en sus resultados; esto es, en cuanto al tiempo el enfoque Todos Menos 1 resulta como el mejor, pero, en cuanto a la precisión MAE, el enfoque Datos 2 obtiene un valor medio mejor.

En este caso, seguimos la siguiente norma para decidir cual de los dos enfoques elegir: se elegirá siempre el enfoque que obtenga unas mejores predicciones salvo que estas

predicciones las obtenga en unos tiempos mucho mayores, del orden de las varias decenas de milisegundos, que el resto.

Por este motivo, se decide que sea el enfoque Datos 2 el que se tenga en consideración para la comparativa final entre pruebas.

- PRUEBA MEJORA 2. Voto por defecto con predicción weighted sum

Ejecución	MAE	Tiempo
Iteración 1	0,6360849	0,07936508
Iteración 2	0,54767364	0
Iteración 3	0,63002366	0
Iteración 4	0,5605799	0,08465608
Iteración 5	0,5680847	0
Iteración 6	0,59977	0,07936508
Iteración 7	0,52810663	0
Iteración 8	0,57786494	0
Iteración 9	0,5800374	0,08465608
Iteración 10	0,5646129	0
Iteración 11	0,65660083	0
Iteración 12	0,60495853	0
Iteración 13	0,5650807	0
Iteración 14	0,622029	0
Iteración 15	0,6253748	0
Iteración 16	0,61428803	0
Iteración 17	0,65351546	0,16402116
Iteración 18	0,6093413	0,08465608
Iteración 19	0,54230404	0
Iteración 20	0,63946915	0
Valores Medios	0,596290026	0,028835978

Tabla 3.3.12 Resultados de mejora Prueba 2

Para poder extraer conclusiones de estos valores, los presentaremos en gráficos de líneas para observar el comportamiento del algoritmo a lo largo de las ejecuciones:

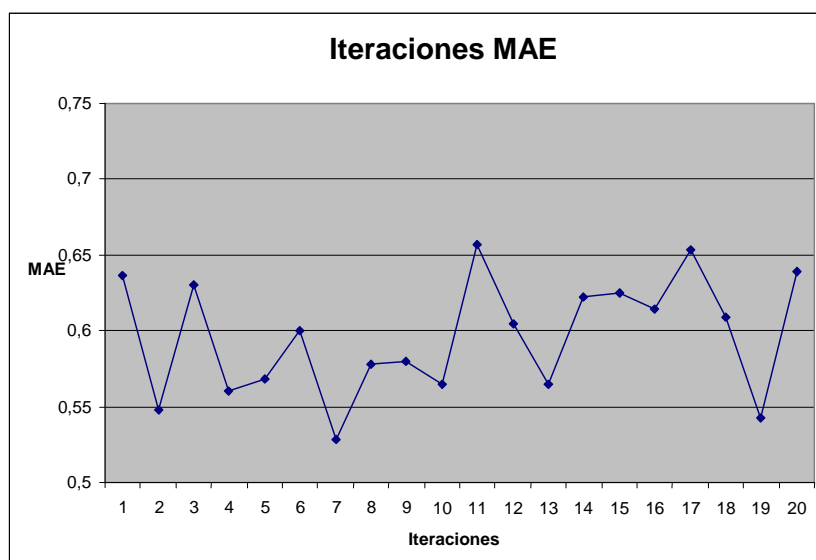


Figura 3.3.31. Gráfico de mejora de MAE de Prueba 2



Figura 3.3.32. Gráfico de mejora de Tiempo de Prueba 2

Se observa en esta prueba que los resultados son extraordinariamente buenos, sobre todo teniendo en cuenta la métrica de predicción, pero este resultado se obtiene debido a que en el momento de hacer la predicción se produce un sobreajuste en los cálculos.

Según la literatura, el algoritmo de mejora del Voto por Defecto se basa en la fórmula de la correlación de Pearson, y sólo es aplicable el algoritmo item+adjustment como algoritmo de predicción.

Por lo tanto, como en la Prueba 2, aplicamos el algoritmo de predicción weighted sum, no tendremos en cuenta los resultados obtenidos.

- PRUEBA MEJORA 3. Frecuencia inversa de usuario con predicción item+adjustment

Ejecución	MAE D2	Tiempo D2	MAE D4	Tiempo D4
Iteración 1	0,881631	5,4391537	0,89545685	5,285714
Iteración 2	0,8668762	12,412699	0,89326113	5,4708996
Iteración 3	0,8806711	7,3756614	0,90276146	4,973545
Iteración 4	0,8630956	4,962963	0,9100613	4,941799
Iteración 5	0,87741685	3,7936509	0,9196162	6,116402
Iteración 6	0,86559826	4,851852	0,9002362	5,05291
Iteración 7	0,8870966	4,296296	0,93154204	4,100529
Iteración 8	0,89775515	5,4656086	0,92961127	3,7195768
Iteración 9	0,87363213	5,4497356	0,8798769	5,5291004
Iteración 10	0,8875504	5,5820107	0,92414033	6,3492064
Iteración 11	0,8681796	5,6190476	0,9079258	4,698413
Iteración 12	0,86784774	5,148148	0,8923499	4,6613755
Iteración 13	0,8829574	4,4814816	0,89925563	4,9259257
Iteración 14	0,8575847	5,2222223	0,89051163	6,878307
Iteración 15	0,8835827	4,8095236	0,90752107	5,6349206
Iteración 16	0,8968245	4,4708996	0,9262909	4,3809524
Iteración 17	0,8782003	5,3492064	0,8826397	5,6455026
Iteración 18	0,86337876	4,994709	0,90594643	5,4021163
Iteración 19	0,85089445	5,84127	0,9039263	5,296296
Iteración 20	0,88397133	4,941799	0,90511745	4,169312
Valores Medios	0,875737239	5,5253969	0,905402425	5,161640165

Tabla 3.3.13.a Resultados de mejora Prueba 3

Ejecución	MAE D8	Tiempo D8	MAE TM1	Tiempo TM1
Iteración 1	0,93179256	4,566138	0,829561	4,5502644
Iteración 2	0,9085175	8,677249	0,8670033	8,259259
Iteración 3	0,92181534	4,042328	0,86500716	1,8148148
Iteración 4	0,91247165	5,5449734	0,8636712	1,6613756
Iteración 5	0,9061917	4,6190476	0,8691112	1,9153439
Iteración 6	0,9222593	5,708995	0,9246098	1,6613756
Iteración 7	0,9328504	4,6402116	0,8102588	2,4074075
Iteración 8	0,9492673	5,269841	0,9029924	3,3862433
Iteración 9	0,93221015	5,058201	0,8315409	2,6507936
Iteración 10	0,9540993	4,9206347	0,85291696	0,41798943
Iteración 11	0,9049247	5,973545	0,8790135	2,9682539
Iteración 12	0,91649634	5,3756614	0,8820604	3,084656
Iteración 13	0,9338557	4,708995	0,8377479	1,9100529
Iteración 14	0,8945964	5,846561	0,8397461	1,3280423
Iteración 15	0,93253016	5,87831	0,87922484	7,1904764
Iteración 16	0,96581644	5,185185	0,90215933	4,883598
Iteración 17	0,89840645	5,3544974	0,9538273	3,7354498
Iteración 18	0,92189884	5,15873	0,95392454	2,153439
Iteración 19	0,91155803	6,3809524	0,7837871	1,1587301
Iteración 20	0,93410367	4,126984	0,9643792	1,7407408
Valores Medios	0,924283097	5,351852025	0,874627147	2,943915317

Tabla 3.3.13.b Resultados de mejora Prueba 3

Al igual que en las pruebas anteriores, visualizaremos los datos de estas tablas en distintos gráficos para determinar el enfoque con el que se obtienen unos mejores resultados:

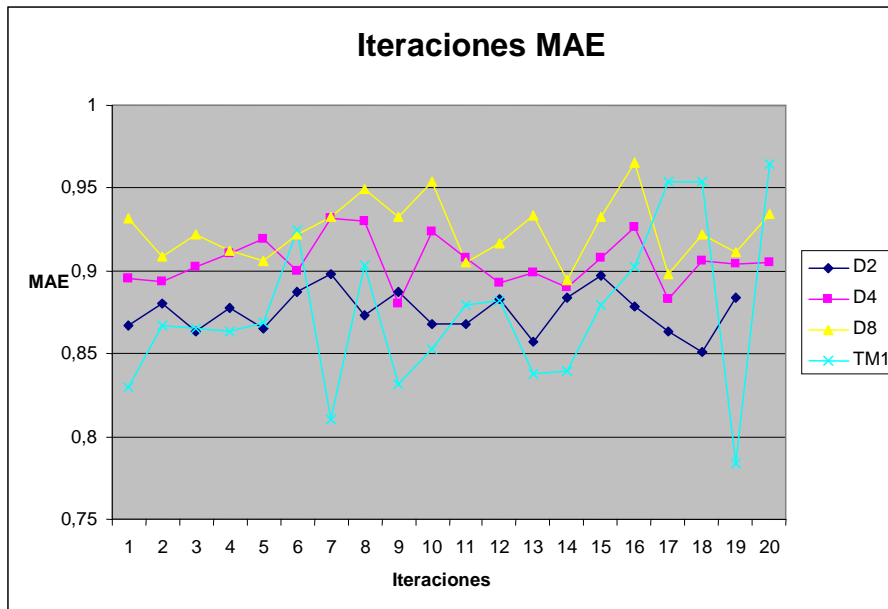


Figura 3.3.33. Gráfico de mejora de MAE de Prueba 3

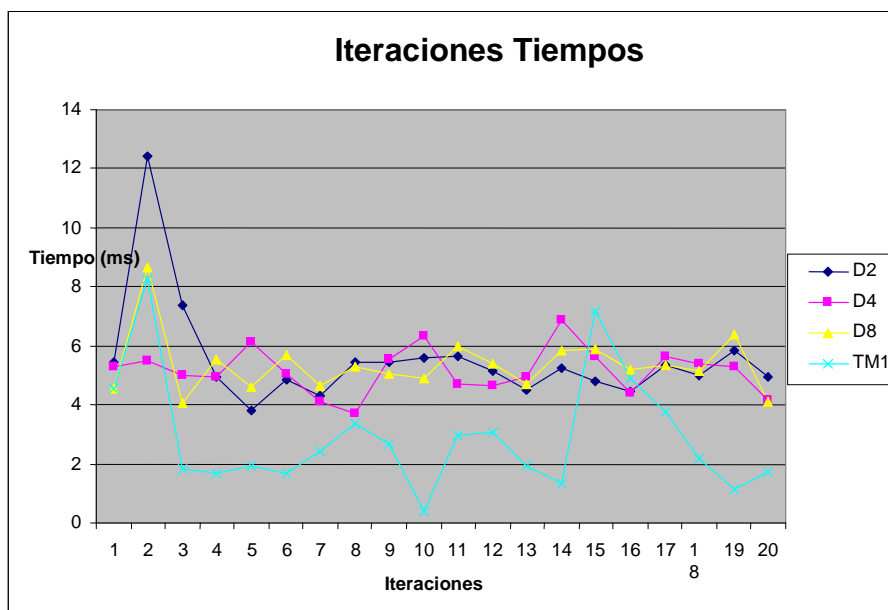


Figura 3.3.34. Gráfico de mejora de Tiempo de Prueba 3

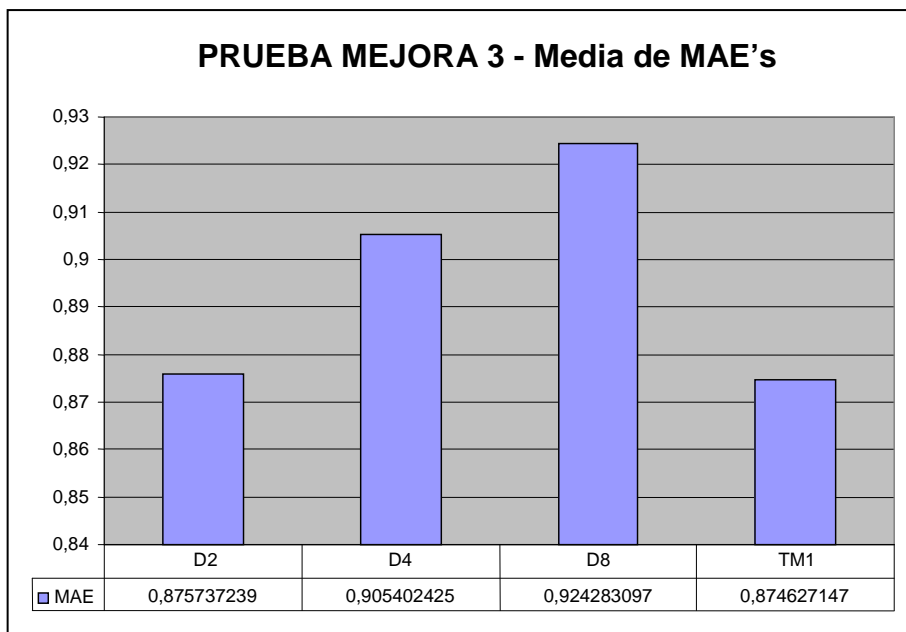


Figura 3.3.35. Media de mejora de MAE de Prueba 3

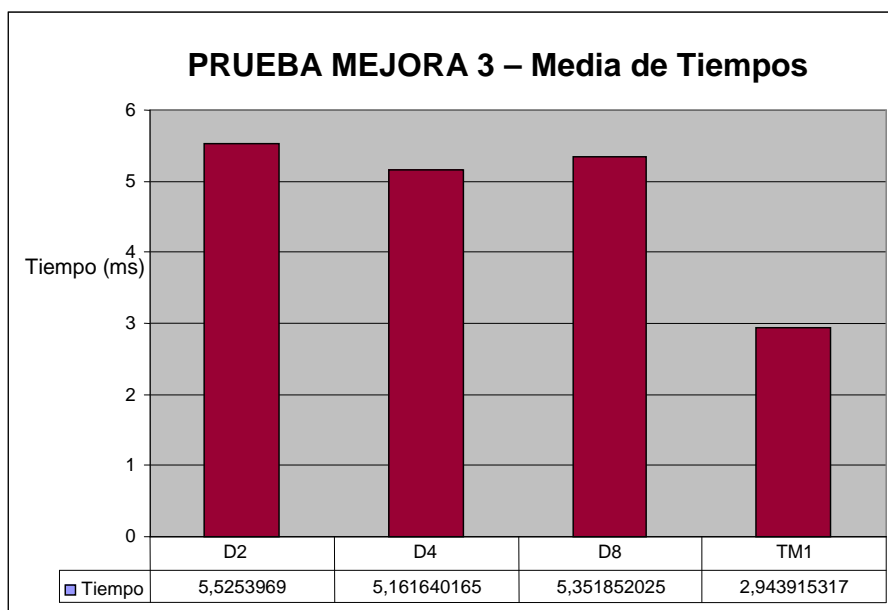


Figura 3.3.36. Media de mejora de Tiempo de Prueba 3

Teniendo en cuenta lo observado en estas dos figuras, se observa que el enfoque Todos Menos 1 obtiene unos resultados mejores en términos de tiempo que los demás enfoques, y también obtiene mejores resultados en cuanto a la precisión de la predicción. Por lo tanto, elegiremos el enfoque Todos Menos 1 como el mejor para esta prueba.

- PRUEBA MEJORA 4. Frecuencia inversa de usuario con predicción weighted sum

Ejecución	MAE	Tiempo
Iteración 1	0,66991645	0
Iteración 2	0,8559589	0
Iteración 3	0,76364315	0
Iteración 4	0,79109603	0
Iteración 5	0,79646444	0
Iteración 6	0,774174	0,08465608
Iteración 7	0,7321634	0,08465608
Iteración 8	0,7562713	0
Iteración 9	0,7754037	0,07936508
Iteración 10	0,7840665	0,08465608
Iteración 11	0,75685924	0,08465608
Iteración 12	0,696147	0
Iteración 13	0,80631876	0,07936508
Iteración 14	0,7983384	0
Iteración 15	0,78154063	0,07936508
Iteración 16	0,71347654	0
Iteración 17	0,7418092	0
Iteración 18	0,7385313	0
Iteración 19	0,8165882	0
Iteración 20	0,76229036	0
Valores Medios	0,76552875	0,028835978

Tabla 3.3.14 Resultados de mejora Prueba 4

Mostraremos los valores de esta tabla utilizando gráficos de líneas para observar el comportamiento del algoritmo a lo largo de las ejecuciones:

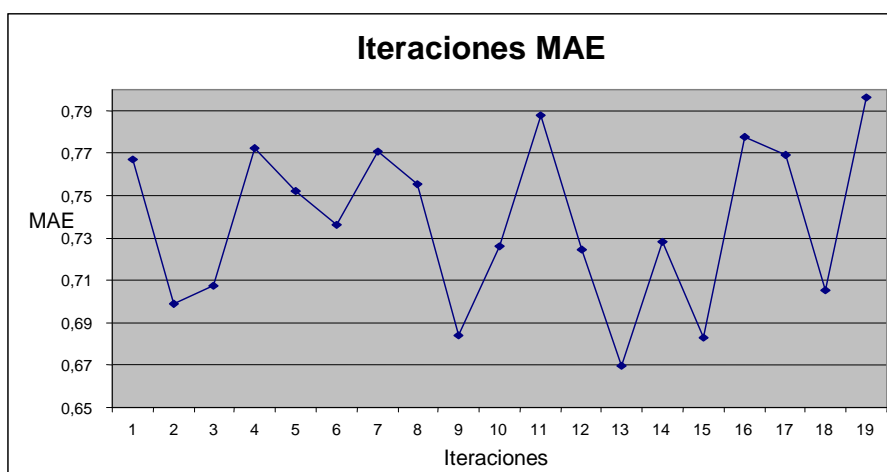


Figura 3.3.37. Gráfico de mejora de MAE de Prueba 4



Figura 3.3.38. Gráfico de mejora de Tiempo de Prueba 4

Se observa que aplicando esta prueba se obtienen resultados bastante buenos, sobretodo atendiendo al tiempo.

PRUEBA MEJORA 5. Frecuencia directa de usuario con predicción item+adjustment

Ejecución	MAE D2	Tiempo D2	MAE D4	Tiempo D4
Iteración 1	0,8793263	3,5767195	0,90204835	3,3862433
Iteración 2	0,86664015	2,5555556	0,9013001	2,6455026
Iteración 3	0,86841327	4,7671957	0,88789725	2,5608466
Iteración 4	0,8613186	3,2804232	0,8814157	3,1798942
Iteración 5	0,8926722	3,132275	0,9299248	2,9047618
Iteración 6	0,87731385	3,8994708	0,90273225	3,153439
Iteración 7	0,8738112	3,222223	0,90924907	3,3862433
Iteración 8	0,9000387	2,978836	0,93828017	3,142857
Iteración 9	0,891516	3,068783	0,93157345	2,4814816
Iteración 10	0,83668035	3,989418	0,86856604	3,1587303
Iteración 11	0,8532735	3,5291004	0,8847433	3,6878307
Iteración 12	0,819292	2,8201058	0,8576162	4,0793653
Iteración 13	0,8837141	3,4761906	0,91340804	3,4126985
Iteración 14	0,8629735	3,2328043	0,9017612	2,6613758
Iteración 15	0,88380235	3,7089946	0,9225241	3,222223
Iteración 16	0,9242413	3,0423281	0,9229463	3,1587303
Iteración 17	0,8771293	3,7354498	0,9358876	3,978836
Iteración 18	0,8836822	3,4814816	0,9050946	2,8253968
Iteración 19	0,8532774	3,7248678	0,8662143	2,8730159
Iteración 20	0,8687164	3,222223	0,8898325	4,301587
Valores Medios	0,872891634	3,4222222	0,902650766	3,210052915

Tabla 3.3.15.a Resultados de mejora Prueba 5

Ejecución	MAE D8	Tiempo D8	MAE TM1	Tiempo TM1
Iteración 1	0,9237944	3,3915343	0,87749577	0,07936508
Iteración 2	0,9137266	3,6931217	0,8506098	0,16402116
Iteración 3	0,91653425	2,8201058	0,8222092	0,16402116
Iteración 4	0,9157177	2,9682539	0,78521496	0,08465608
Iteración 5	0,94096464	3,4814816	0,8757336	0,25396827
Iteración 6	0,9173307	3,5608466	0,8936032	0,24867725
Iteración 7	0,9466304	2,5502646	0,8709781	0,07936508
Iteración 8	0,96894616	3,3915343	0,90754884	0,08465608
Iteración 9	0,94956976	3,05291	0,8855067	0,16402116
Iteración 10	0,8667169	2,8730159	0,80008525	0
Iteración 11	0,9009445	3,6560845	0,7997774	0,16402116
Iteración 12	0,89781624	2,8095238	0,779867	0,08465608
Iteración 13	0,94136614	3,6084657	0,8267959	0,07936508
Iteración 14	0,91347015	3,1058202	0,8858531	0,08465608
Iteración 15	0,92910224	2,8835979	0,8765421	0,24867725
Iteración 16	0,95118207	3,9682539	0,89461327	0
Iteración 17	0,9271212	3,6402116	0,84858066	0,24867725
Iteración 18	0,9400336	2,7037036	0,82366985	0,24338624
Iteración 19	0,903574	2,8201058	0,76693106	0,08465608
Iteración 20	0,9109189	4,291005	0,8168593	0,08465608
Valores Medios	0,923773028	3,263492035	0,844423753	0,132275131

Tabla 3.3.15.b Resultados de mejora Prueba 5

Para poder extraer conclusiones de estos valores, los presentaremos en gráficos de líneas para observar el comportamiento del algoritmo a lo largo de las ejecuciones:

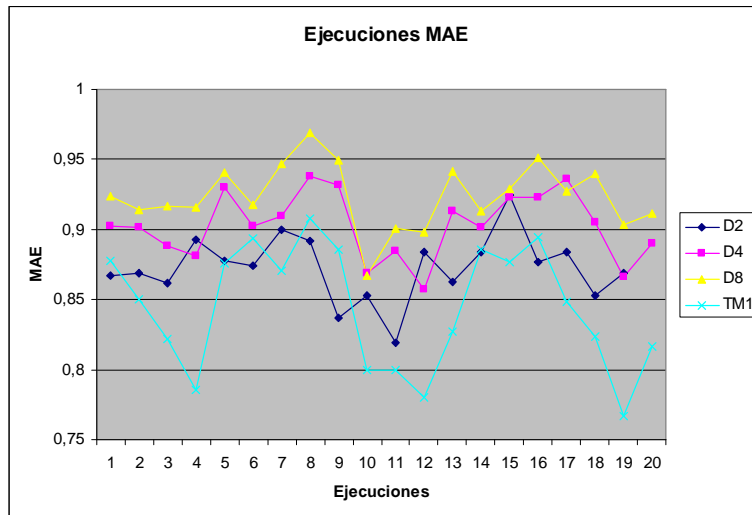


Figura 3.3.39. Gráfico de mejora de MAE de Prueba 5

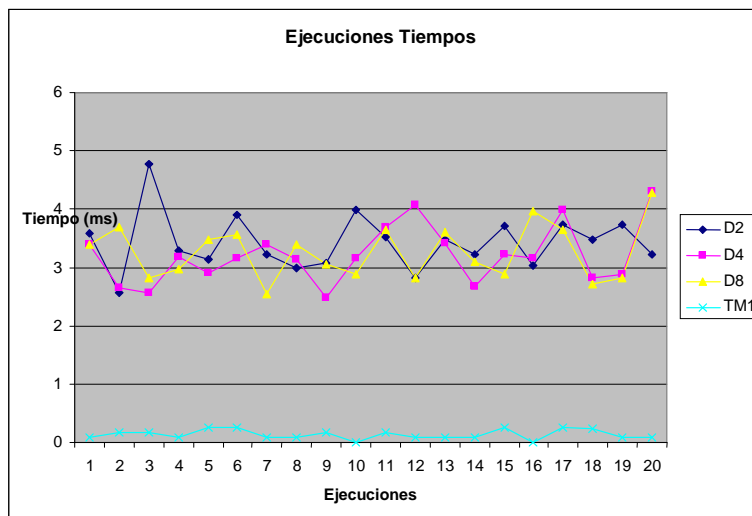


Figura 3.3.40. Gráfico de mejora de Tiempo de Prueba 5

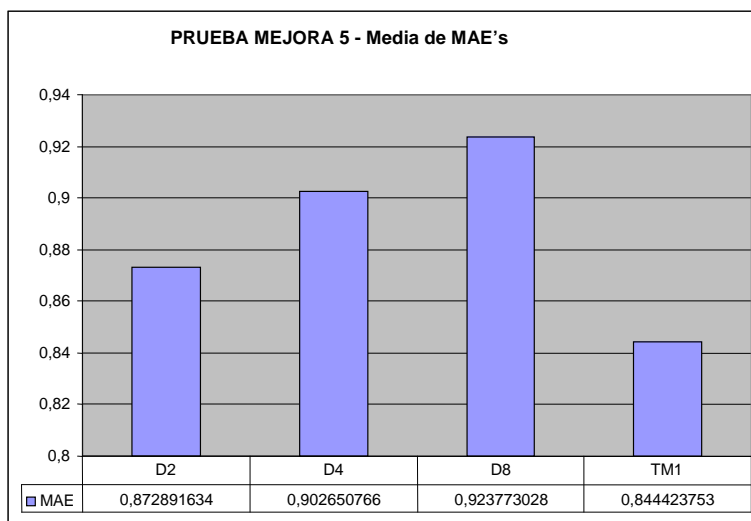


Figura 3.3.41. Media de mejora de MAE de Prueba 5

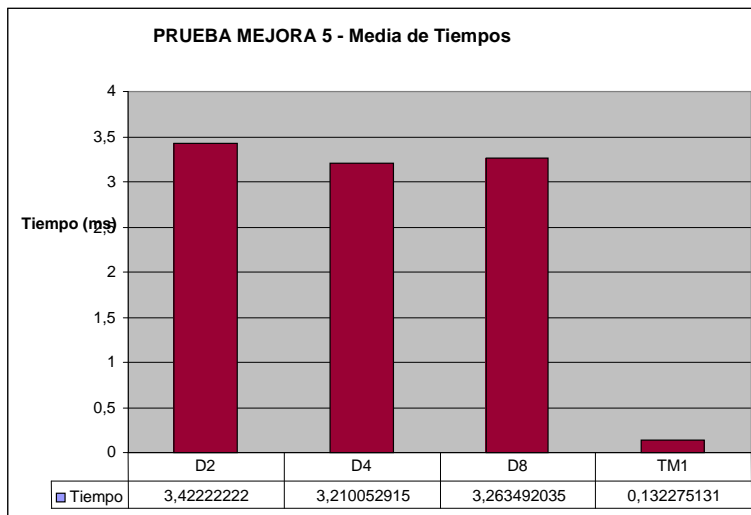


Figura 3.3.42. Media de mejora de Tiempo de Prueba 5

Para esta prueba las dos métricas de evaluación coinciden en que el enfoque Todos Menos 1 realiza las mejores predicciones en el mejor tiempo, por lo que se empleará éste para la comparativa final entre las distintas pruebas.

- **PRUEBA MEJORA 6.** Frecuencia directa de usuario con predicción weighted sum

Ejecución	MAE	Tiempo
Iteración 1	0,6685823	0
Iteración 2	0,76724833	0,052910052
Iteración 3	0,6988505	0
Iteración 4	0,7076199	0,052910052
Iteración 5	0,7722927	0,052910052
Iteración 6	0,7519638	0
Iteración 7	0,7359203	0
Iteración 8	0,77061164	0
Iteración 9	0,7553786	0
Iteración 10	0,6840608	0,052910052
Iteración 11	0,7262854	0
Iteración 12	0,78802145	0,052910052
Iteración 13	0,72444606	0,052910052
Iteración 14	0,6695868	0
Iteración 15	0,7284033	0
Iteración 16	0,68303174	0
Iteración 17	0,7774537	0,052910052
Iteración 18	0,76909214	0,052910052
Iteración 19	0,70543337	0
Iteración 20	0,7961722	0
Valores Medios	0,734022752	0,021164021

Tabla 3.3.16 Resultados de mejora Prueba 6

Mostraremos los valores de esta tabla utilizando gráficos de líneas para observar el comportamiento del algoritmo a lo largo de las ejecuciones:

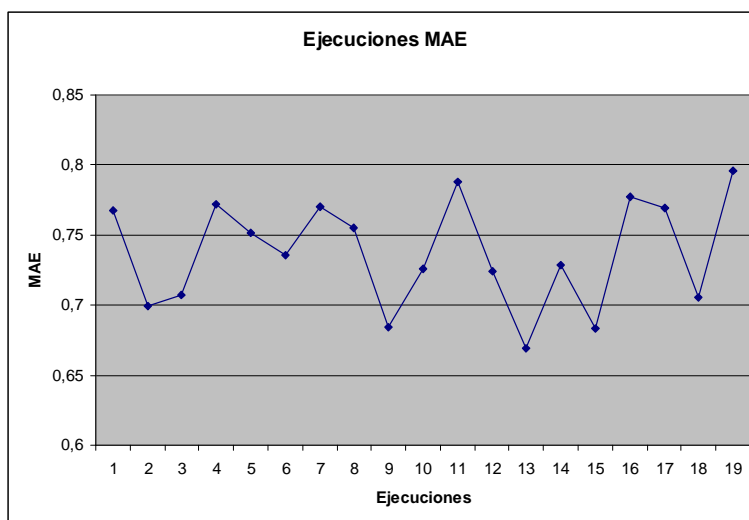


Figura 3.3.43. Gráfico de mejora de MAE de Prueba 6

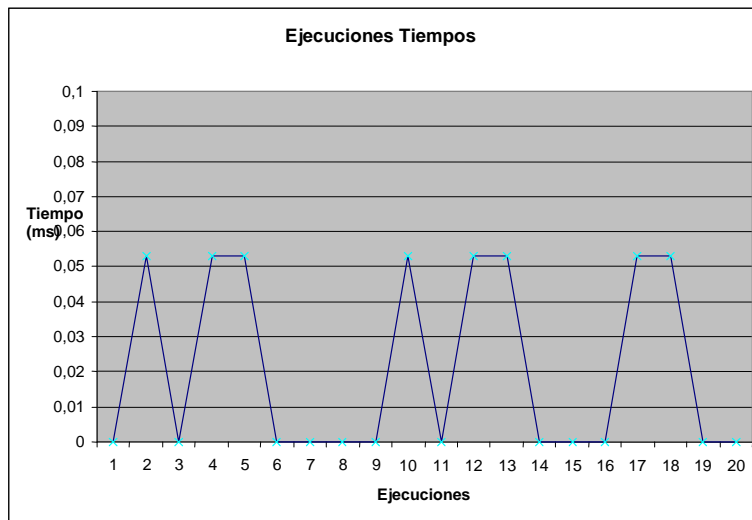


Figura 3.3.44. Gráfico de mejora de Tiempo de Prueba 6

Si aplicamos esta mejora que hemos implementado 'por intuición' para realizar las predicciones sobre nuestra base de datos, vemos que obtenemos muy buenos resultados, tanto en tiempo como en la métrica de predicción.

- PRUEBA MEJORA 7. Amplificación de casos con predicción item+adjustment

Ejecución	MAE D2	Tiempo D2	MAE D4	Tiempo D4
Iteración 1	0,84049284	4,4444447	0,8538418	4,878307
Iteración 2	0,83506316	4,9259257	0,85103816	4,973545
Iteración 3	0,8349108	4,9312167	0,8416174	5,3544974
Iteración 4	0,8213799	4,5079365	0,84157217	4,6613755
Iteración 5	0,8505231	4,6190476	0,8709695	4,708995
Iteración 6	0,8459705	5,2063494	0,8518026	6,031746
Iteración 7	0,84468085	4,5502644	0,8532739	4,867725
Iteración 8	0,854756	4,719577	0,881641	4,6137567
Iteración 9	0,82885355	4,7777777	0,8472254	4,285714
Iteración 10	0,87034804	4,820106	0,8956675	4,5555553
Iteración 11	0,8360732	5,031746	0,84895235	5,031746
Iteración 12	0,85742575	4,137566	0,86161697	4,0740743
Iteración 13	0,8356772	5,301587	0,84947866	5,6137567
Iteración 14	0,8306814	5,4497356	0,8383381	5,4126983
Iteración 15	0,8216879	5,089947	0,8295823	5,3968253
Iteración 16	0,82698554	4,296296	0,84274846	4,825397
Iteración 17	0,85645735	4,978836	0,86814284	4,3915343
Iteración 18	0,83497	5,132275	0,86325043	5,2433863
Iteración 19	0,8129912	5,037037	0,83221984	4,291005
Iteración 20	0,8214389	4,984127	0,8398817	4,978836
Valores Medios	0,838068359	4,847089915	0,853143054	4,909523805

Tabla 3.3.17.a Resultados de mejora Prueba 7

Ejecución	MAE D8	Tiempo D8	MAE TM1	Tiempo TM1
Iteración 1	0,86372757	4,825397	0,9040571	0,42328042
Iteración 2	0,87478757	5,2010584	0,8745949	0,8994709
Iteración 3	0,8758614	4,4973545	0,9034898	0,58201057
Iteración 4	0,8653255	4,714286	0,91222775	0,52910054
Iteración 5	0,8876421	4,2380953	0,9265555	0,6878307
Iteración 6	0,8814987	4,973545	0,910075	0,9047619
Iteración 7	0,8805217	4,566138	0,8736918	0,58201057
Iteración 8	0,891088	4,7671957	0,8325007	0,7407407
Iteración 9	0,8699273	4,5026455	0,8432718	0,58201057
Iteración 10	0,921788	5,089947	0,8973962	0,52910054
Iteración 11	0,8667889	5,6190476	0,84248334	0,26455027
Iteración 12	0,89549726	4,825397	0,8970902	0,95238096
Iteración 13	0,8801081	4,714286	0,9142453	0,37037036
Iteración 14	0,8576022	4,708995	0,8961257	0,58201057
Iteración 15	0,8479409	4,7777777	0,88229513	0,47619048
Iteración 16	0,85346776	5,031746	1,0174098	0,37037036
Iteración 17	0,8881915	4,185185	0,8320251	0,7989418
Iteración 18	0,87623686	4,830688	0,93165696	0,6878307
Iteración 19	0,84287137	4,291005	0,9150779	0,47619048
Iteración 20	0,85081285	4,9259257	0,81865406	0,21164021
Valores Medios	0,873584277	4,76428577	0,891246202	0,58253968

Tabla 3.3.17.b Resultados de mejora Prueba 7

Al igual que en las pruebas anteriores, visualizaremos los datos de estas tablas en distintos gráficos para determinar el enfoque con el que se obtienen unos mejores resultados:

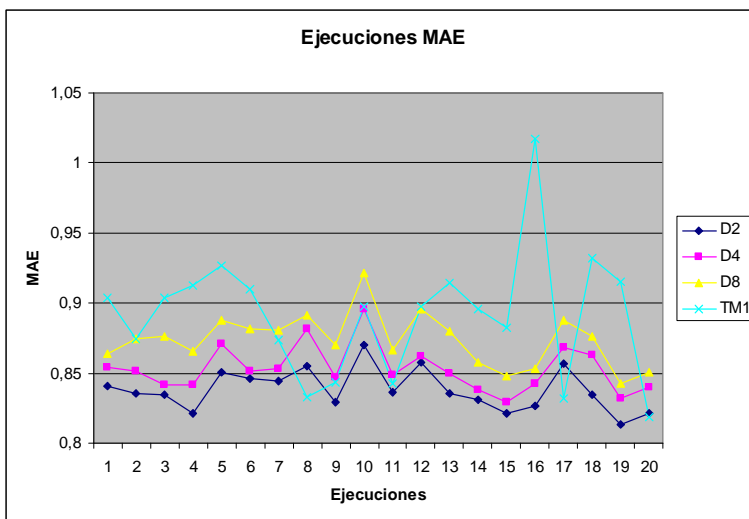


Figura 3.3.45. Gráfico de mejora de MAE de Prueba 7

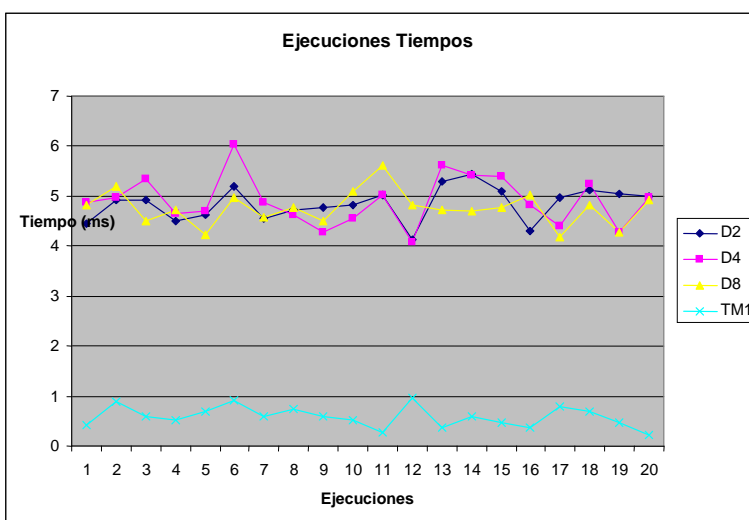


Figura 3.3.46. Gráfico de mejora de Tiempo de Prueba 7

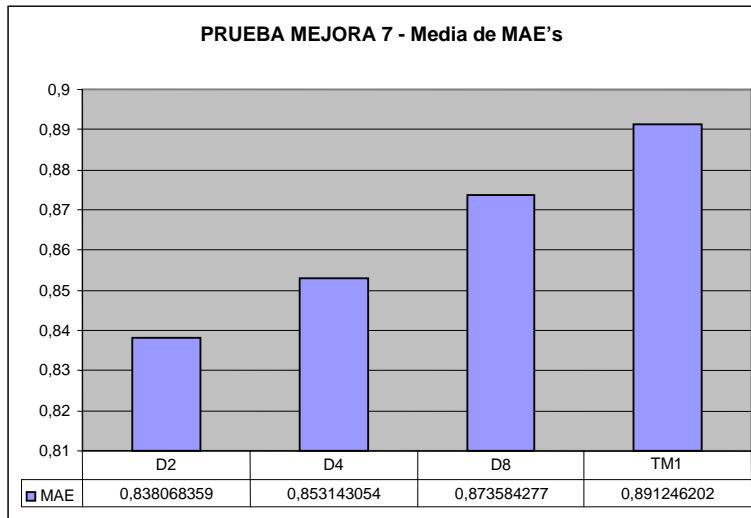


Figura 3.3.47. Media de mejora de MAE de Prueba 7

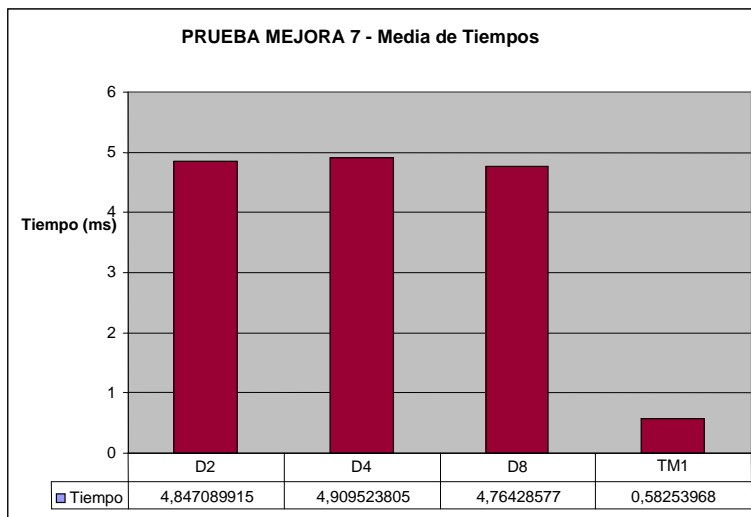


Figura 3.3.48. Media de mejora de Tiempo de Prueba 7

Nos encontramos ante una situación ya planteada con anterioridad en el que las métricas de evaluación difieren en sus resultados. Si atendemos al tiempo el enfoque Todos Menos 1 resulta como el mejor, pero, en cuanto a la precisión MAE, el enfoque Datos 2 obtiene un valor medio mejor.

Siguiendo la norma propuesta en la Prueba 1 de los algoritmos de mejora, nos decidimos por que sea el enfoque Datos 2 el que se tenga en consideración para la comparativa final entre pruebas.

3.3.2.2 Comparativa entre las Mejoras

Una vez realizadas las pruebas y analizados sus resultados comparamos dichos resultados y determinaremos la prueba con la que se han obtenido resultados mejores para la posterior implementación del sistema de recomendación colaborativo.

Como en las pruebas que han implementado el algoritmo de predicción **item average + adjustment** se presentan distintos resultados según el enfoque seguido vamos a recordar cual de estos enfoques es el tomado en consideración (es decir, el que mejores resultados haya obtenido) para cada prueba:

PRUEBA	ENFOQUE
PRUEBA MEJORA 1	Dados 2
PRUEBA MEJORA 3	Todos Menos 1
PRUEBA MEJORA 5	Todos Menos 1
PRUEBA MEJORA 7	Dados 2

Tabla 3.3.18. Enfoque elegido para cada prueba

Ahora bien, en la siguiente tabla se recogen los datos que se han obtenido en cada prueba en la que se ha aplicado la mejora correspondiente a los algoritmos de filtrado colaborativo. Los resultados atienden a las dos métricas tenidas en cuenta a lo largo de las iteraciones, MAE y tiempo:

PRUEBA	Media MAE	Media Tiempo
PRUEBA MEJORA 1	0,841536784	4,708730195
PRUEBA MEJORA 3	0,874627147	2,943915317
PRUEBA MEJORA 4	0,765552875	0,028835978
PRUEBA MEJORA 5	0,844423753	0,132275131
PRUEBA MEJORA 6	0,734022752	0,021164021
PRUEBA MEJORA 7	0,838068359	4,847089915

Tabla 3.3.19. Datos de Medias de las dos métricas en las pruebas realizadas

En el siguiente gráfico de barras se muestra con claridad los resultados comparados de todas las pruebas para la métrica MAE:

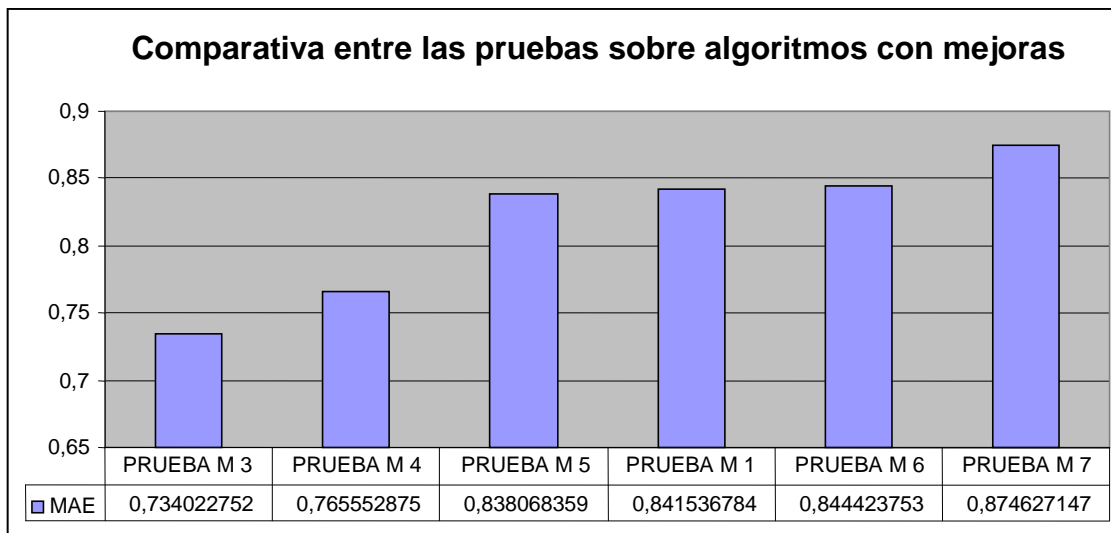


Figura 3.3.49. Comparativa entre las pruebas sobre algoritmos con mejoras

A la vista del gráfico, y con los datos de la tabla, queda claro que para la métrica de precisión MAE la **Prueba Mejora 6** es con la que se obtiene unos mejores resultados.

3.3.3 Comparativa Final

Teniendo en cuenta que en las primeras pruebas realizadas sobre algoritmos básicos de filtrado colaborativo para obtener los mejores parámetros llegamos a la conclusión de que el mejor resultado obtenido lo hacía con los siguientes parámetros:

ENT/TEST	0.8/0.2
Nº VEC.	20
MED. SIMILAR.	Coficiente Coseno
ALG. PRED.	weighted sum

Tabla 3.3.20. Parámetros del Sistema de Recomendación previo a las mejoras

Con dicha configuración se obtenían los resultados siguientes atendiendo a cada métrica estudiada:

Media MAE	Media Tiempo
0,734587523	0,050529101

Tabla 3.3.21. Valores del Sistema de Recomendación Colaborativo previo a las mejoras

A continuación presentamos una comparativa entre las pruebas realizadas sobre algoritmos básicos de filtrado colaborativo y las pruebas sobre algoritmos con mejoras, para que de manera gráfica se observen los resultados de cada prueba:

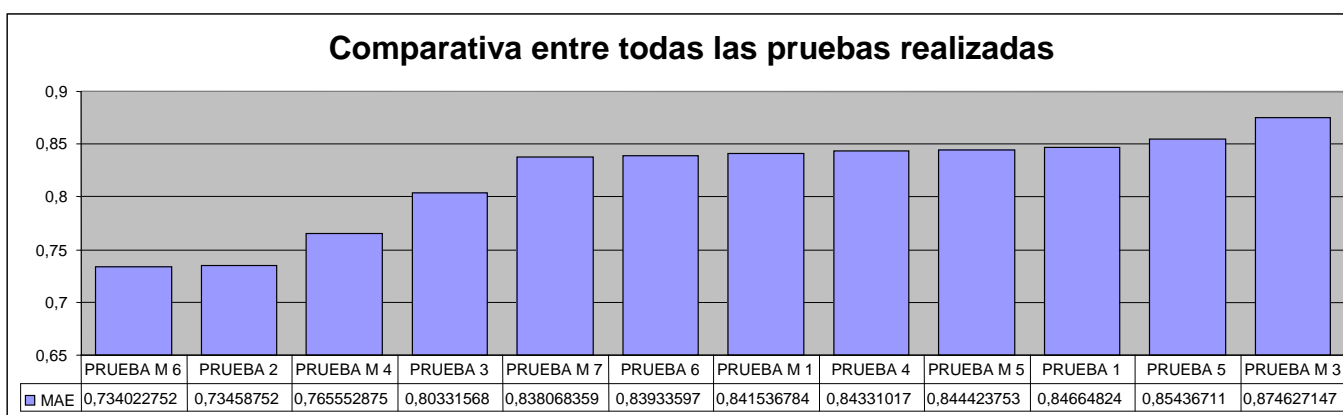


Figura 3.3.50. Comparativa de todas las pruebas

Vemos que con el algoritmo de mejora de la Prueba Mejora 6 se obtiene una ligera mejoría en la predicción. Por lo tanto, en la siguiente parte de este proyecto se implementará el prototipo de un sistema de recomendación colaborativo basado en una arquitectura cliente/servidor que utilizará un algoritmo de filtrado colaborativo formado por los siguientes valores de los parámetros:

ENT/TEST	0.8/0.2
Nº VEC.	20
MED. SIMILAR.	Frecuencia Directa
ALG. PRED.	weighted sum

Tabla 3.3.22. Valores del Sistema de Recomendación Colaborativo con las mejoras

CAPÍTULO 4.

SISTEMA DE RECOMENDACIÓN COLABORATIVO

MOVIESRECOMMENDER II

Una vez realizado el estudio comparativo tanto de los algoritmos de filtrado colaborativo como de las mejoras propuestas que conformaba la primera parte de este proyecto, en este apartado se va a detallar el desarrollo del prototipo de un sistema de recomendación colaborativo basado en una arquitectura cliente/servidor con interfaz web implementando el mejor de los algoritmos previamente estudiados.

Por lo tanto, esta segunda parte es un proyecto de desarrollo software y, como tal, para su desarrollo deben seguirse las actividades de la **Ingeniería del Software**. No existe una definición única y estandarizada para la Ingeniería del Software pero las dos que se presentan a continuación pueden resultar perfectamente válidas para este cometido:

- Ingeniería del Software es la construcción de software de calidad con un presupuesto limitado y un plazo de entrega en contextos de cambio continuo.
- Ingeniería del Software es el establecimiento y uso de principios y métodos firmes de ingeniería para obtener software económico que sea fiable y funcione de manera eficiente en máquinas reales.

Las actividades que conforman la Ingeniería del Software son las siguientes:

- **Especificación de Requerimientos:** se obtienen el propósito del sistema y las propiedades y restricciones del mismo.
- **Análisis del Sistema:** se obtiene un modelo del sistema correcto, completo, consistente, claro y verificable.
- **Diseño del Sistema:** se definen los objetivos del proyecto y las estrategias a seguir para conseguirlos.
- **Implementación:** se traduce el modelo a código fuente.

En los puntos siguientes se profundizará en cada una de estas actividades y en como se han llevado a cabo en el ámbito nuestro proyecto.

4.1 Especificación de Requerimientos

El primer paso en la Ingeniería del Software debe ser determinar el **propósito último del proyecto**, las propiedades que debe satisfacer y las restricciones a las que está sometido. Este es, sin duda, un paso de vital importancia dentro del desarrollo de un proyecto software ya que, sin conocer el propósito del proyecto y todas las limitaciones de diversa índole a las que debe hacer frente, difícilmente se podrá realizar una aplicación software que cumpla dicho propósito.

En un proyecto de ámbito comercial para una empresa real, para determinar el propósito del mismo se recurre a una serie de estudios como pueden ser entrevistas con los clientes, encuestas con posibles usuarios, estudios de la situación actual del sistema o estudios de viabilidad. En nuestro caso no nos encontramos ante un proyecto comercial sino ante uno académico por lo que el propósito es conocido desde el mismo momento de la concepción del mismo:

El desarrollo de un sistema de recomendación colaborativo para el alquiler de películas basado en una arquitectura cliente/servidor con interfaz web implementando el mejor algoritmo de filtrado colaborativo de los estudiados previamente.

Habiendo determinado el propósito último del proyecto, el siguiente paso consiste en especificar los requerimientos del mismo. Los **requerimientos de un proyecto software** son el conjunto de propiedades y/o restricciones definidas con total precisión, que dicho proyecto software debe satisfacer. Existen dos tipos bien diferenciados de tales requerimientos:

- **Requerimientos funcionales:** aquellos que se refieren específicamente al funcionamiento de la aplicación o sistema.
- **Requerimientos no funcionales:** aquellos no referidos al funcionamiento estricto sino a otros factores externos.

En los dos siguientes subapartados se pasarán a definir cuales son estos requerimientos (tanto funcionales como no funcionales) para el proyecto del que se ocupa esta memoria. Sin embargo, estas definiciones sólo serán previas ya que en la actividad de análisis del sistema, donde se crearán los casos de uso y sus escenarios, se descubrirán nuevas necesidades que

no son observables en esta primera actividad y que permitirán refinar completamente estos requerimientos.

4.1.1 Requerimientos funcionales

Los requerimientos funcionales de un sistema software son aquellos que se encargan de describir las funcionalidades que el sistema debe proporcionar a los usuarios del mismo para cumplir sus expectativas.

Normalmente, estos requerimientos se obtendrían de la interacción con el cliente mediante diversas entrevistas y/o encuestas. En nuestro caso, al tratarse de un proyecto académico, nos encontramos ante la situación de la no existencia de cliente alguno por lo que la información sobre las funcionalidades que debe disponer el sistema se ha obtenido investigando otros sistemas de recomendación colaborativos que ya se encuentran en el mercado, muy especialmente aquellos de reconocido éxito y gran número de usuarios.

En base a estas investigaciones realizadas se ha llegado a la conclusión de que las funcionalidades que el usuario potencial espera de un sistema de recomendación como el nuestro son las siguientes:

- Permitir el registro de nuevos usuarios.
- Recibir recomendaciones de objetos que no ha probado y pueden ser de su gusto, siempre que haya puntuado al menos 20 objetos.
- Puntuar los objetos que ha probado.
- Poder cambiar las puntuaciones de objetos ya puntuados.
- Consultar sus datos personales.
- Modificar su datos en caso de que cambien o sean erróneos.
- Disponer de una lista con todas las objetos disponibles en el sistema y toda la

información posible sobre los mismos.

- Disponer de mecanismos de ayuda.

Una vez definidas cuales son las funcionalidades que los usuarios reclaman a un sistema de recomendación, se hace necesario caracterizar de una manera más formal y concreta como va a responder a estas funcionalidades, dentro del ámbito del alquiler de películas, nuestro sistema:

1) Permitir el registro de nuevos usuarios.

El sistema debe de permitir el ingreso de nuevos usuarios en la base de datos.

2) Recibir recomendaciones de nuevas películas

El sistema debe proporcionar al usuario, basándose en las películas ya vistas por el mismo y sus puntuaciones, una lista con las películas que no ha visto o alquilado todavía y que pueden ser más de su agrado. Señalar que en el caso de los nuevos usuarios, éstos deben realizar un número mínimo de 20 puntuaciones para obtener recomendaciones.

3) Puntuar películas

El sistema debe permitir al usuario que puntúe películas, tanto aquellas que vaya viendo como aquellas que ya han sido puntuadas pero que, por alguna razón, el usuario quiera volver a puntuar. Además, debe actualizar la base de datos con esta información.

4) Visualizar datos personales

El sistema debe proporcionar al usuario la posibilidad de visualizar sus datos personales.

5) Modificar datos personales

El sistema debe permitir al usuario modificar sus datos personales y actualizar esta información en la base de datos.

6) Visualizar todas las películas

El sistema debe proporcionar la posibilidad de visualizar un listado con todas las

películas disponibles en la base de datos al usuario.

7) Visualizar sólo las películas ya puntuadas

El sistema debe proporcionar al usuario la posibilidad de visualizar un listado con las películas que ya ha puntuado y cuales son dichas puntuaciones.

8) Visualizar sólo las películas alquiladas pero no puntuadas

El sistema debe proporcionar al usuario la posibilidad de visualizar un listado con las películas que ha alquilado pero que todavía no ha puntuado.

9) Consultar ayuda

El sistema debe proporcionar al usuario algún medio de ayuda para que pueda conocer perfectamente el manejo de la aplicación o resolver cualquier duda puntual que pueda tener

Estas son las funcionalidades que debe proporcionar nuestro sistema de recomendación teniendo en cuenta de que en este proyecto vamos a desarrollar una versión prototipal del mismo. En una versión final del sistema, este debería satisfacer otras funcionalidades, como podrían ser:

- Recálculo instantáneo de las recomendaciones después de una actualización de la base de datos.
- Posibilidad de incorporar nuevas películas a la base de datos.
- Tours guiados de puntuaciones para que los nuevos usuarios puedan obtener un perfil amplio de manera rápida y sencilla.
- Posibilidad de realizar comentarios textuales los usuarios sobre las distintas películas.

4.1.2 Requerimientos no funcionales

Los requerimientos no funcionales son aquellos que restringen los requerimientos funcionales. Son tan importantes como los propios requerimientos funcionales y pueden incluso a llegar a ser críticos para la aceptación del sistema. Estos requerimientos normalmente especifican propiedades del sistema o del producto en si (plataforma, velocidad, rendimiento...) y del diseño de la interfaz gráfica con el usuario además de todas las restricciones impuestas por la organización (políticas de empresa, estándares, legalidad vigente...).

Al no ser este un proyecto comercial para ninguna organización o empresa real, no debemos someternos a restricciones organizacionales. Por lo tanto, los requerimientos no funcionales que se deben obtener y analizar son los referentes a las necesidades hardware y software de los equipos informáticos para que estos proporcionen al usuario las funcionalidades requeridas de forma eficiente y los referentes a la interfaz gráfica entre la aplicación y el usuario.

A) Requerimientos del equipo informático

Al hablar de los requerimientos del equipo informático y debido a que el marco del desarrollo de la aplicación es una arquitectura cliente/servidor, debemos diferenciar los requerimientos de equipo que necesita el servidor y los que necesita el cliente.

Las necesidades de **equipo informático del cliente** son muy simples ya que tan solo le hace falta un computador conectado a Internet (preferiblemente de banda ancha) y tener instalado un navegador capacitado para visualizar de forma correcta la aplicación (se recomienda Firefox u Opera pero podría ser válido cualquier otro).

Los requerimientos del **equipo informático del servidor**, el cual se aconseja que sea un equipo dedicado, son más amplios y se dividen en dos tipos: los requerimientos de hardware y los requerimientos software.

1) Hardware

- *Velocidad*: el equipo debe ser lo suficientemente rápido como para ejecutar la

aplicación en el menor tiempo posible y con la mayor fiabilidad. Cualquier microprocesador actual es capaz de cumplir con esta labor.

- *Memoria:* el equipo debe disponer de la suficiente memoria RAM libre para realizar las operaciones que se soliciten entre la aplicación y la base de datos.
- *Almacenamiento:* el equipo que haga la labor de servidor debe tener una capacidad de almacenamiento suficiente para almacenar la base de datos con la que trabaja la aplicación y permitir con holgura las transacciones entre el servidor y la base de datos.
- *Tarjeta gráfica:* las tarjetas gráficas de las que disponen los equipos informáticos actuales son de gran potencia por lo que es inútil establecer ningún requerimiento en este aspecto.
- *Monitor:* el monitor debe soportar una resolución de 1024x768 y superiores.
- *Conexión a Internet:* el servidor debe encargarse de que la aplicación sea accesible a través de Internet para todos sus usuarios por lo que es indispensable que se encuentre conectado a Internet a través de banda ancha las 24 horas del día.

2) Software

- *Sistema Operativo:* el servidor de la aplicación trabaja sobre un sistema operativo Windows XP Professional Service Pack 2.
- *Navegador:* la aplicación debe poder ser visualizada desde cualquier navegador web actual aunque se recomienda el uso de Firefox en sus últimas versiones.
- *Sistema Gestor de Bases de Datos:* la aplicación trabaja con una base de datos MS Access 2003.
- El resto del software necesario (servidor Http Apache, lenguaje de

programación PHP...) será proporcionado al administrador de la aplicación, el cual dispone de un manual para su instalación en el **Anexo I**.

B) Requerimientos de la interfaz

Los requerimientos de la interfaz gráfica entre la aplicación y el usuario están íntimamente ligados a la **usabilidad** y sus principios. La usabilidad se puede definir de varias formas [17]:

- Usabilidad se define coloquialmente como facilidad de uso, ya sea de una página web, una aplicación informática o cualquier otro sistema que interactúe con un usuario.
- La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.
- Usabilidad es la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico.

A partir de estas tres definiciones se pueden obtener los principios básicos de la usabilidad, los cuales se asociarán a los requerimientos no funcionales que deberá cumplir la interfaz gráfica:

- **Facilidad de aprendizaje:** se refiere a la facilidad con la que nuevos usuarios pueden tener una interacción efectiva. Depende de los siguiente factores:
 - *Predecibilidad:* una vez conocida la aplicación, se debe saber en cada momento a que estado se pasará en función de la tarea que se realice.
 - *Síntesis:* los cambios de estado tras una acción deben ser fácilmente captados.
 - *Generalización:* las tareas semejantes se resuelven de modo parecido.
 - *Familiaridad:* el aspecto de la interfaz tiene que resultar conocido y familiar para el usuario.

- *Consistencia*: siempre se han de seguir una misma serie de pasos para realizar una tarea determinada.
- **Flexibilidad**: relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.
- **Robustez**: es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos o, también, la capacidad del sistema para tolerar fallos. Está relacionada con los siguientes factores:
 - *Observación*: el usuario debe poder observar el estado del sistema sin que esta observación repercuta de forma negativa en él.
 - *Recuperación de información*: la aplicación debe poder deshacer alguna operación y permitir volver a un estado anterior.
 - *Tiempo de respuesta*: es el tiempo necesario para que el sistema pueda mostrar los cambios realizados por el usuario.

4.2 Análisis del Sistema

Una vez conocido el propósito del proyecto software, las propiedades que debe cumplir y las restricciones a las que debe someterse, llega el momento de **analizar el sistema** y crear un **modelo** del mismo que sea correcto, completo, consistente, claro y verificable. Para conseguir ésto se crearán y definirán casos de uso en base a los requerimientos previamente obtenidos y se describirán ciertos escenarios de acción de dichos casos de uso.

4.2.1 Casos de uso

Un **caso de uso** representa una clase de funcionalidad dada por el sistema como un flujo de eventos. También se puede definir como la representación de una situación o tarea de interacción de un usuario con la aplicación.

Los casos de uso describen como se realiza una tarea de manera exacta y constan de los siguientes elementos:

- Nombre único e unívoco
- Actores participantes
- Condiciones de entrada
- Flujo de eventos
- Condiciones de salida
- Requerimientos especiales

Por lo tanto, es necesario determinar cuales son los actores participantes en cada uno de los casos de uso. Un **actor** modela una entidad externa que se comunica con el sistema, es decir, es un tipo de usuario del sistema. Un actor, al igual que un caso de uso, debe tener un nombre único y puede tener una descripción asociada.

En nuestro sistema contamos con los tres actores siguientes:

- **Cliente:** se trata del usuario tipo de la aplicación, el que la va a utilizar para recibir recomendaciones, puntuar películas y demás.
- **Administrador:** se trata del responsable de la aplicación, el que se encarga de actualizar el algoritmo de filtrado colaborativo empleado en base a las nuevas puntuaciones que van realizando los usuarios del sistema.
- **BBDD:** se trata de la base de datos que proporciona los datos a la aplicación.

Una vez definidos cuales van a ser los actores del sistema, es el momento de crear los distintos casos de uso. A la hora de realizar esta acción es importante que cada uno de los requerimientos funcionales ya definidos aparezca en al menos uno de los casos de uso aunque, por otra parte, puede haber casos de uso nuevos, en los que no aparezca ninguno de

los requerimientos, ya que estamos en una fase de refinamiento del sistema donde queremos construir un modelo detallado del mismo.

Un paso previo a la creación y descripción de los distintos casos de uso es la obtención de los diversos **diagramas de casos de uso** de nuestro sistema, al que vamos a llamar **MoviesRecommender II**.

El primero es un *diagrama frontera*, es decir, un diagrama que describe completamente la funcionalidad de un sistema:

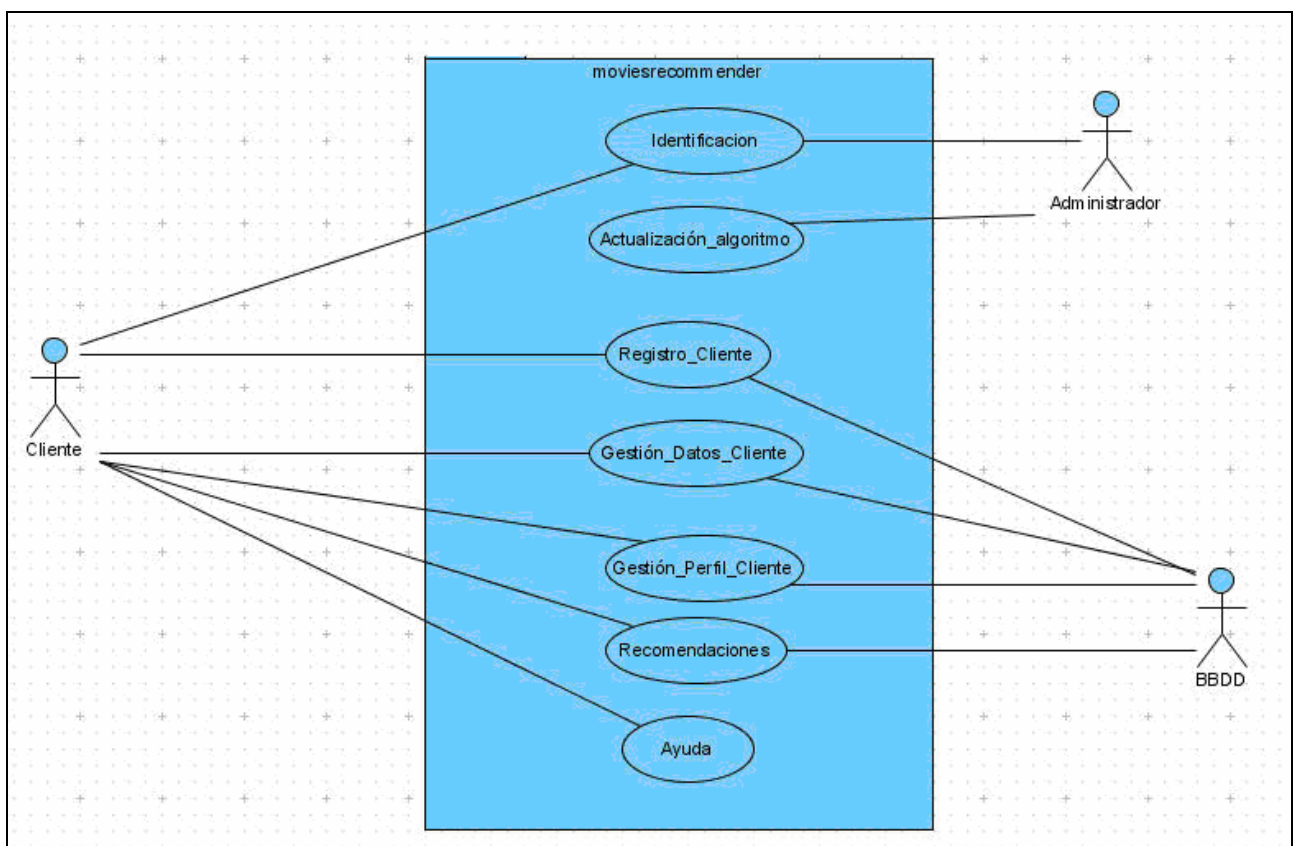


Figura 3.2.1. Diagrama frontera de MoviesRecommender II

Los casos de uso mostrados en un diagrama frontera pueden ser lo suficientemente exactos o, por el contrario, pueden ser concretados con un mayor detalle. A la hora de detallar un caso de uso se pueden emplear dos tipos de relaciones:

- *<<extend>>*: es una relación cuya dirección es hacia el caso de uso a detallar que representa comportamientos excepcionales del caso de uso.

- `<<include>>`: es una relación cuya dirección es contraria a la de la relación `<<extend>>` que representa un comportamiento común del caso de uso

En nuestro caso nos encontramos con que los casos de uso **Registro_Cliente**, **Gestión_Datos_Cliente** y **Gestión_Perfil_Cliente** requieren ser detallados en más profundidad:

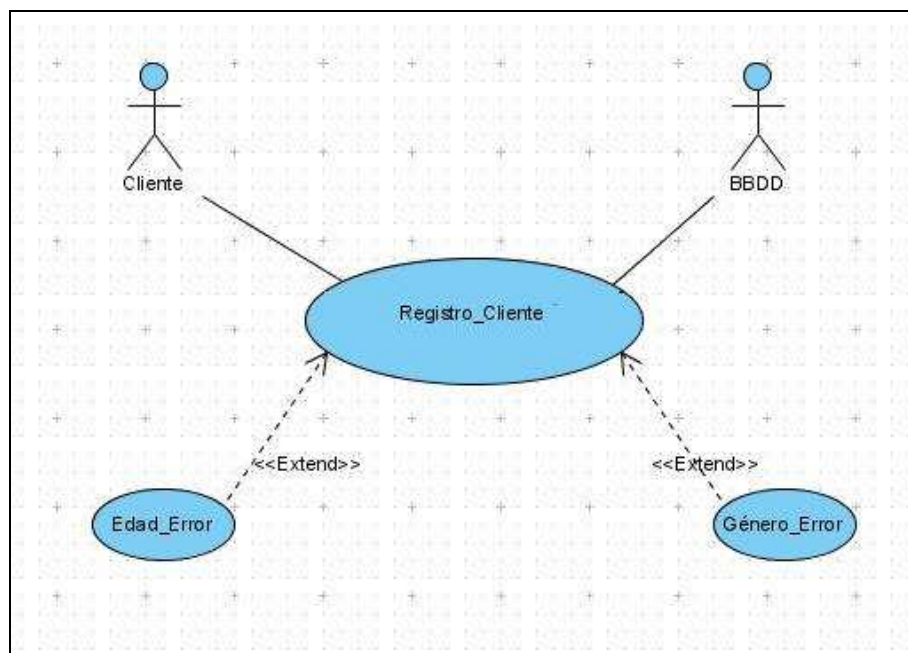


Figura 3.2.2. Diagrama del caso de uso Registro_Cliente

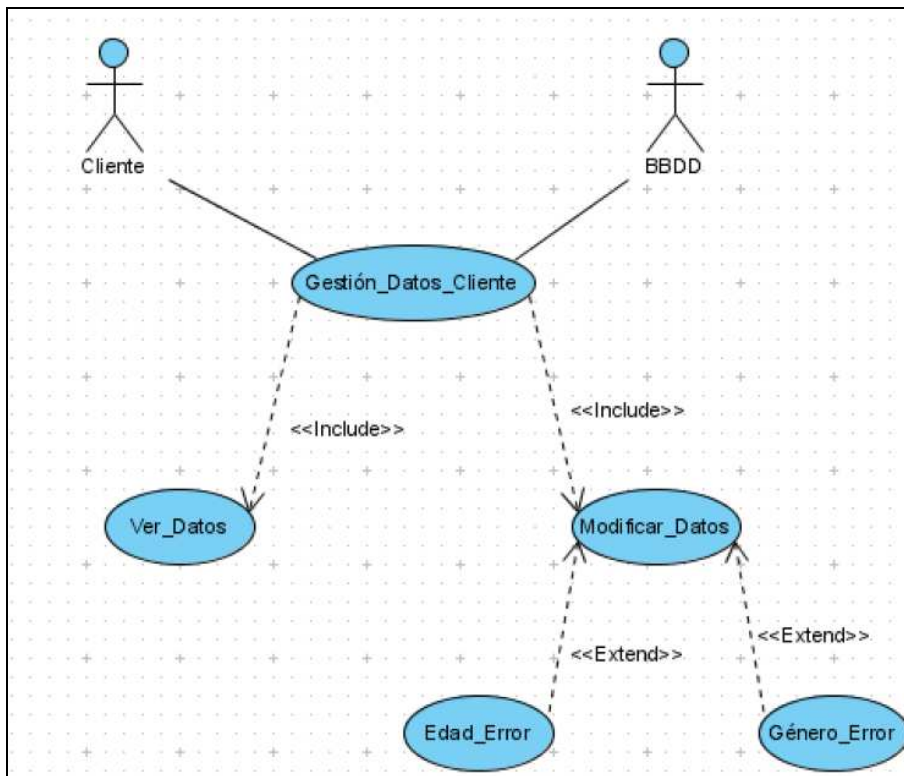


Figura 3.2.2. Diagrama del caso de uso Gestión_Datos_Cliente

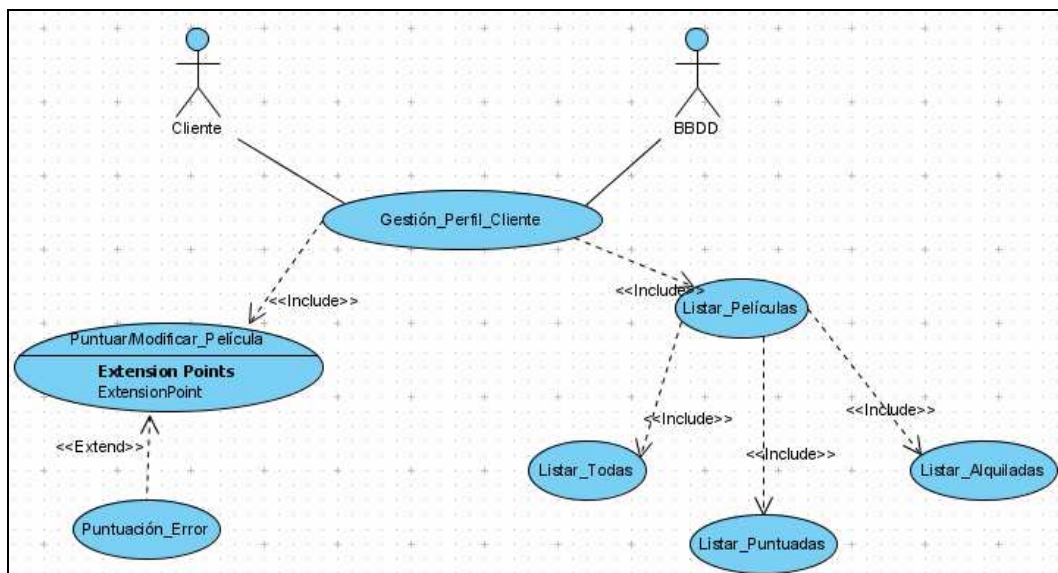


Figura 3.2.3. Diagrama del caso de uso Gestión_Perfil_Cliente

A continuación, se describen detalladamente cada uno de los casos de uso mostrados en las figuras anteriores:

➤ **Caso de Uso 1: Identificación**

Actores participantes: Cliente y Administrador

Condiciones de entrada: Que existan cuentas de usuario en la aplicación, y que el usuario se haya registrado con anterioridad.

Flujo de eventos:

1. El usuario (Cliente o Administrador) inicia la aplicación.
2. El sistema muestra un formulario de entrada.
3. El usuario introduce su identificador.
4. El sistema comprueba que el identificador es válido (E-1).
5. Según el identificador sea de:
 - 5.1. Cliente, entonces el usuario entra al sistema y este le muestra el *menú principal*.
 - 5.2. Administrador, entonces el usuario entra al sistema y este le muestra el *menú de administrador*.

Condiciones de salida: La contraseña ha sido comprobada.

Excepciones:

E-1:El identificador introducido por el usuario no es válido. El sistema informa al usuario de dicha situación. El usuario puede intentar introducir un identificador válido de Cliente o Administrador o salir del caso de uso.

➤ **Caso de Uso 2: Actualización algoritmo**

Actores participantes: Administrador

Condiciones de entrada: Administrador identificado correctamente.

Flujo de eventos:

1. El sistema muestra el menú de administrador.
2. Administrador elige la opción *Actualizar Algoritmo de Filtrado* del menú.
3. El sistema lanza un script para actualizar el algoritmo (E-1).
4. Una vez terminada la actualización, el sistema muestra la fecha en la que se ha realizado la misma y un mensaje informativo para Administrador.

Condiciones de salida: la actualización se produce satisfactoriamente y se solicita salir del caso de uso.

Excepciones:

E-1: el script lanzado por el sistema ha fallado al actualizar el algoritmo. El sistema se lo comunica a Administrador. Administrador puede volver a intentar lanzar el script o salir del caso de uso.

➤ **Caso de Uso 3: Registro Cliente**

Actores participantes: Cliente y BBDD

Condiciones de entrada: Existen Cliente y BBDD.

Flujo de eventos principal:

1. Cliente elige la opción *Registro* de la página principal.
2. El sistema le muestra a Cliente un formulario de entrada de datos.
3. Cliente introduce su edad (E-2), su género (E-3), su profesión y su código postal en el formulario.
4. El sistema actualiza BBDD (E-1) con los nuevos datos.
5. El caso de uso se inicia de nuevo.

Condiciones de salida: Se solicita la salida del caso de uso.

Excepciones:

E-1: Ha habido un error al comunicarse el sistema con BBDD. El sistema informa a Cliente de dicha situación. El caso de uso se inicia de nuevo.

E-2: Cliente ha introducido una edad inválida. El sistema informa a Cliente de dicha situación. Usuario puede intentar introducir de nuevo una edad o salir del caso de uso.

E-3: Cliente ha introducido un género inválido. El sistema informa a Cliente de dicha situación. Usuario puede intentar introducir de nuevo un género o salir del caso de uso.

➤ **Caso de Uso 4: Gestión Datos Cliente**

Actores participantes: Cliente y BBDD

Condiciones de entrada: Existen Cliente y BBDD.

Flujo de eventos principal:

1. El sistema muestra el menú principal.
2. Cliente elige la opción *Datos Personales* del menú principal.
3. El sistema muestra un menú con tres opciones y le pide a Cliente que elija:
 - Si Cliente elige *Ver Datos Personales*, se realiza S-1.
 - Si Cliente elige *Modificar Datos Personales*, se realiza S-2.
 - Si Cliente elige *Salir*, se termina el caso de uso.

Subflujos de eventos:

S-1: Ver_Datos

- 1.1. El sistema se comunica con BBDD (E-1) para obtener los datos personales de Cliente.
- 1.2. El sistema muestra sus datos personales a Cliente.
- 1.3. El caso de uso se inicia de nuevo.

S-2: Modificar_Datos

- 2.1. El sistema le muestra a Cliente un formulario de entrada de datos.
- 2.2. Cliente introduce su edad (E-2), su género (E-3), su profesión y su código postal en el formulario.
- 2.3. El sistema actualiza BBDD (E-1) con los nuevos datos.
- 2.4. El caso de uso se inicia de nuevo.

Condiciones de salida: Se solicita la salida del caso de uso.

Excepciones:

E-1: Ha habido un error al comunicarse el sistema con BBDD. El sistema informa a Cliente de dicha situación. El caso de uso se inicia de nuevo.

E-2: Cliente ha introducido una edad inválida. El sistema informa a Cliente de dicha situación. Usuario puede intentar introducir de nuevo una edad o salir del caso de uso.

E-3: Cliente ha introducido un género inválido. El sistema informa a Cliente de dicha situación. Usuario puede intentar introducir de nuevo un género o salir del caso de uso.

➤ **Caso de Uso 5: Gestión Perfil Cliente**

Actores participantes: Cliente y BBDD

Condiciones de entrada: Existen Cliente y BBDD.

Flujo de eventos principal:

1. El sistema muestra el menú principal.
2. Cliente elige la opción *Perfil de Usuario* del menú principal.
3. El sistema muestra un menú con tres opciones y le pide a Cliente que elija:
 - Si Cliente elige *Puntuar/Modificar Película*, se realiza S-1.
 - Si Cliente elige *Listados*, se realiza S-2.
 - Si Cliente elige *Salir*, se termina el caso de uso.

Subflujos de eventos:

S-1: Puntuar_Película

- 1.1. El sistema se comunica con BBDD (E-1) para obtener el listado de las películas.
- 1.2. El sistema muestra a Cliente el listado anterior.
- 1.3. Cliente elige la película que desea puntuar o modificar la puntuación.
- 1.4. El sistema le muestra a Cliente un formulario de entrada de datos.
- 1.5. Cliente introduce la puntuación (E-2) que le quiere dar a la película en el formulario.
- 1.6. El sistema actualiza BBDD (E-1) con los nuevos datos.
- 1.7. El caso de uso se inicia de nuevo.

S-2: Listar_Películas

- 2.1. El sistema muestra un menú con cuatro opciones y le pide a Cliente que elija:
 - Si Cliente elige *Listar Todas*, entonces se realiza S-2-1.
 - Si Cliente elige *Listar Puntuadas*, entonces se realiza S-2-2.
 - Si Cliente elige *Listar Alquiladas*, entonces se realiza S-2-3.
 - Si Cliente elige *Salir*, entonces se inicia el caso de uso.

S-2-1: Listar_Todas

- 2.1.1. El sistema se comunica con BBDD (E-1) para obtener todas las películas.
- 2.1.2. El sistema muestra un listado con todas las películas a Cliente.
- 2.1.3. El caso de uso se inicia de nuevo.

S-2-2: Listar_Puntuadas

- 2.2.1. El sistema se comunica con BBDD (E-1) para obtener las películas ya puntuadas por Cliente.
- 2.2.2. El sistema muestra un listado con las películas obtenidas a Cliente.
- 2.2.3. El caso de uso se inicia de nuevo.

S-2-3: Listar_Alquiladas

- 2.3.1. El sistema se comunica con BBDD (E-1) para obtener todas las películas que ha alquilado Cliente pero todavía no ha puntuado.
- 2.3.2. El sistema muestra un listado con las películas obtenidas a Cliente.
- 2.3.3. El caso de uso se inicia de nuevo.

Condiciones de salida: Se solicita la salida del caso de uso.

Excepciones:

E-1: Ha habido un error al comunicarse el sistema con BBDD. El sistema informa a Cliente de dicha situación. El caso de uso se inicia de nuevo.

E-2: Cliente ha introducido una puntuación inválida. El sistema informa a Cliente de dicha situación. Cliente puede intentar introducir de nuevo una puntuación o salir del caso de uso.

➤ **Caso de Uso 6: Recomendaciones**

Actores participantes: Cliente y BBDD

Condiciones de entrada: Existen Cliente y BBDD. Cliente tiene al menos 20 puntuaciones de películas en su perfil.

Flujo de eventos:

1. El sistema muestra el menú principal.
2. Cliente elige la opción *Obtener Recomendaciones* del menú principal.
3. El sistema se comunica con BBDD (E-1) y obtiene las puntuaciones hechas por Cliente.
4. El sistema calcula las películas recomendadas para Cliente en base a sus puntuaciones previas utilizando el algoritmo de filtrado colaborativo implementado.
5. El sistema obtiene de BBDD (E-1) la información de las películas recomendadas.
6. El sistema muestra a Cliente una lista con las películas recomendadas.

Condiciones de salida: Cliente visualiza correctamente su lista de películas recomendadas y solicita salida del caso de uso.

Excepciones:

E-1: Que haya error al comunicarse el sistema con la base de datos. El sistema informa a Cliente de dicha situación. El caso de uso se inicia de nuevo.

➤ **Caso de Uso 7: Ayuda**

Actores participantes: Cliente

Condiciones de entrada: Cliente identificado correctamente.

Flujo de eventos:

1. Cliente elige *Ayuda* de la lista de opciones.
2. El sistema muestra el mecanismo de ayuda implementado a Cliente.

Condiciones de salida: Cliente visualiza correctamente la ayuda y solicita salida del caso de uso.

4.2.2 Escenarios

Un caso de uso es una representación abstracta, una abstracción, de una funcionalidad del sistema a realizar. La representación concreta de un caso de uso se realiza mediante la creación de uno o más **escenarios** que muestren todas las interacciones posibles entre el sistema y sus usuarios.

Un escenario esta formado por los siguientes elementos:

- Un nombre único y unívoco
- Una descripción
- Los actores participantes
- El flujo de eventos

Como se ha indicado, para cada caso de uso puede haber varios escenarios. Para nuestro proyecto se han creado y descrito una cantidad importante de casos de uso por lo que no vamos a definir todos los escenarios de cada uno de ellos sino que vamos a definir unos pocos que puedan servir como ejemplo de las principales funcionalidades que el sistema va a desarrollar: realizar puntuaciones y obtener recomendaciones.

<p>Nombre: PuntuarPeliGoldenEye</p> <p>Descripción: El usuario con identificador 89 quiere puntuar con un 4 la película de título GoldenEye</p> <p>Actores: Cliente89 y BBDD</p> <p>Flujo de eventos:</p> <ol style="list-style-type: none">1. El usuario entra al sistema.2. El sistema muestra el formulario de entrada.3. El usuario introduce 89 en el formulario.4. El sistema valida correctamente y el usuario entra a la aplicación como <i>Cliente89</i>.5. El sistema muestra el menú principal de la aplicación.6. El usuario elige la opción <i>Perfil de Usuario</i> del menú.7. El sistema muestra un menú con 3 opciones: <i>Puntuar Película, Listados</i> y <i>Salir</i>.8. El usuario elige la opción <i>Puntuar Películas</i>.9. El sistema se comunica con BBDD y obtiene todas las películas disponibles.10. El sistema muestra una lista con las películas obtenidas.11. El usuario encuentra en la lista la película <i>GoldenEye</i> y la elige.12. El sistema le muestra un formulario al usuario para que introduzca la puntuación que le va a conceder a la película <i>GoldenEye</i>.13. El usuario introduce en el formulario un 4.14. El sistema comprueba que 4 es una puntuación válida15. El sistema actualiza BBDD con la nueva puntuación.16. El sistema comunica a usuario que la operación se ha realizado con éxito
--

Figura 3.2.4. Escenario PuntuarPeliGoldenEye

<p>Nombre: ObtenerRecCliente345</p> <p>Descripción: El usuario con identificador 345 quiere recibir del sistema una lista de películas recomendadas</p> <p>Actores: Cliente345 y BBDD</p> <p>Flujo de eventos:</p> <ol style="list-style-type: none">1. El usuario entra al sistema.2. El sistema muestra el formulario de entrada.3. El usuario introduce 345 en el formulario.4. El sistema valida correctamente y el usuario entra a la aplicación como <i>Cliente345</i>.5. El sistema muestra el menú principal de la aplicación.6. El usuario elige la opción <i>Obtener Recomendaciones</i> del menú.7. El sistema se comunica con BBDD y obtiene las puntuaciones hechas por el usuario.8. El sistema calcula las películas recomendadas para el usuario en base a sus puntuaciones previas y utilizando el algoritmo de filtrado colaborativo.9. El sistema obtiene de BBDD información de las 10 películas recomendadas.10. El sistema muestra al usuario una lista con sus 10 películas recomendadas.

Figura 3.2.5. Escenario ObtenerRecCliente345

4.3 Diseño del Sistema

Sin duda, realizar de manera adecuada cada una de las actividades que conlleva la **Ingeniería del Software** es indispensable para la realización de un proyecto software de calidad. Por lo tanto, no se puede decir que ninguna de estas actividades sea más importante que otra. Sin embargo, si podemos decir que la actividad de diseño es la más delicada y la más laboriosa de llevar a cabo.

Es delicada porque si no se lleva a cabo correctamente se hace imposible el codificar, de manera correcta, en la actividad de implementación el modelo obtenido en el análisis del sistema, lo que puede repercutir en el desperdicio de todo el esfuerzo realizado durante las primeras actividades de la Ingeniería del Software.

Y es laboriosa porque las estrategias a seguir para conseguir que esta traducción entre modelo y código se lleve a cabo correctamente son muy diversas y bastante complejas.

Se puede decir, por tanto, que el **diseño del sistema** es la actividad de la Ingeniería del Software en la que se identifican los objetivos finales del sistema y se plantean las diversas estrategias para alcanzarlos en la actividad de implementación.

Sin embargo, el sistema no se suele diseñar de una sola vez sino que hay que diferenciar entre el diseño y estructura de los datos que se van a manejar y el diseño de la interfaz entre la aplicación y el usuario. Estas dos fases del diseño no se realizan de forma consecutiva una detrás de la otra sino que lo normal es realizarlas de manera concurrente y finalizarlas a la vez.

4.3.1 Diseño de los datos

La intención de esta fase del diseño software es determinar la estructura que poseen cada uno de los elementos de información del sistema, es decir, la estructura de los datos sobre los que va a trabajar. Estos elementos son:

- Las *películas*, de las que conocemos su nombre, su año de producción, su fecha de estreno, el/los géneros a los que se adscribe y la URL de su entrada en IMDB.

- Los *usuarios*, de los que conocemos su edad, si es hombre o mujer, a que se dedica y su código postal además de su identificador del sistema y el número de películas que ha puntuado.
- Las *puntuaciones*, de las que conocemos el usuario que las hace, las películas que las reciben y, obviamente, el valor numérico de las mismas.
- Las *películas alquiladas* por los usuarios pero todavía no puntuadas.

Una vez determinados cuales son los elementos de información del sistema, se deben obtener sus representaciones en forma de tablas de una base de datos. Para ello, se debe realizar primeramente un diseño conceptual de la base de datos para, posteriormente, obtener las tablas requeridas. Para realizar este diseño conceptual se utilizará el modelo Entidad-Relación.

Modelo Entidad-Relación

El modelo Entidad-Relación (también conocido por sus iniciales: E-R) es una técnica de modelado de datos que utiliza **diagramas entidad-relación**. No es la única técnica de modelado pero si es la más extendida y utilizada.

Un diagrama entidad-relación esta compuesto por tres tipos de elementos principales:

- **Entidades:** objetos (cosas, conceptos o personas) sobre los que se tiene información. Se representan mediante rectángulos etiquetados en su interior con un nombre. Una *instancia* es cualquier ejemplar concreto de una entidad.
- **Relaciones:** interdependencias entre uno o más entidades. Se representan mediante rombos etiquetados en su interior con un verbo. Si la relación es entre una entidad consigo mismo se denomina *reflexiva*, si es entre dos entidades se denomina *binaria*, *ternaria* si es entre tres y *múltiple* si es entre más (muy raro).
- **Atributos:** características propias de una entidad o relación. Se representan mediante elipses etiquetados en su interior con un nombre.

En los diagramas entidad-relación también hay que tener en cuenta otros aspectos como pueden ser:

- **Entidades débiles:** son aquellas que no se pueden identificar unívocamente solo con sus atributos, es decir, necesitan de estar relacionadas con otras entidades para existir. Se representan con dos rectángulos concéntricos de distinto tamaño con un nombre en el interior del más pequeño.
- **Cardinalidad de las relaciones:** existen tres tipos de cardinalidades de una relación según el número de instancias de cada entidad que involucren:
 - *Uno a uno:* una instancia de la entidad A se relaciona solamente con una instancia de la entidad B. (1:1)
 - *Uno a muchos:* cada instancia de la entidad A se relaciona con varias de la entidad B. (1:*)
 - *Muchos a muchos:* cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B. (*:*)
- **Claves:** cada entidad de un diagrama entidad-relación debe tener una clave, debe estar formada por uno o más de sus atributos.

Una vez conocidos los elementos que forman parte de un diagrama entidad-relación podemos empezar a desarrollar el **modelo entidad-relación**. Los pasos a seguir son los siguientes:

1. Convertir el enunciado del problema (o, como es nuestro caso, los elementos del sistema software) en un **Esquema Conceptual** del mismo.
2. Convertir este Esquema Conceptual (o EC) en uno más refinado conocido como **Esquema Conceptual Modificado** (ECM).
3. Obtener las tablas de la base de datos a partir del Esquema Conceptual Modificado.

1) Esquema Conceptual

Necesitamos convertir nuestros elementos del sistema en entidades o relaciones. De manera obvia se puede llegar a la conclusión de que *películas* y *usuarios* se convierten en las entidades *PELICULAS* y *USUARIOS* respectivamente y que *puntuaciones* se transforma en la relación *PUNTUAR* que une las dos entidades. Por su parte, *películas alquiladas* se transforma en la relación *ALQUILAR* entre *PELICULAS* y *USUARIOS*. Nuestro EC quedaría de la siguiente forma:

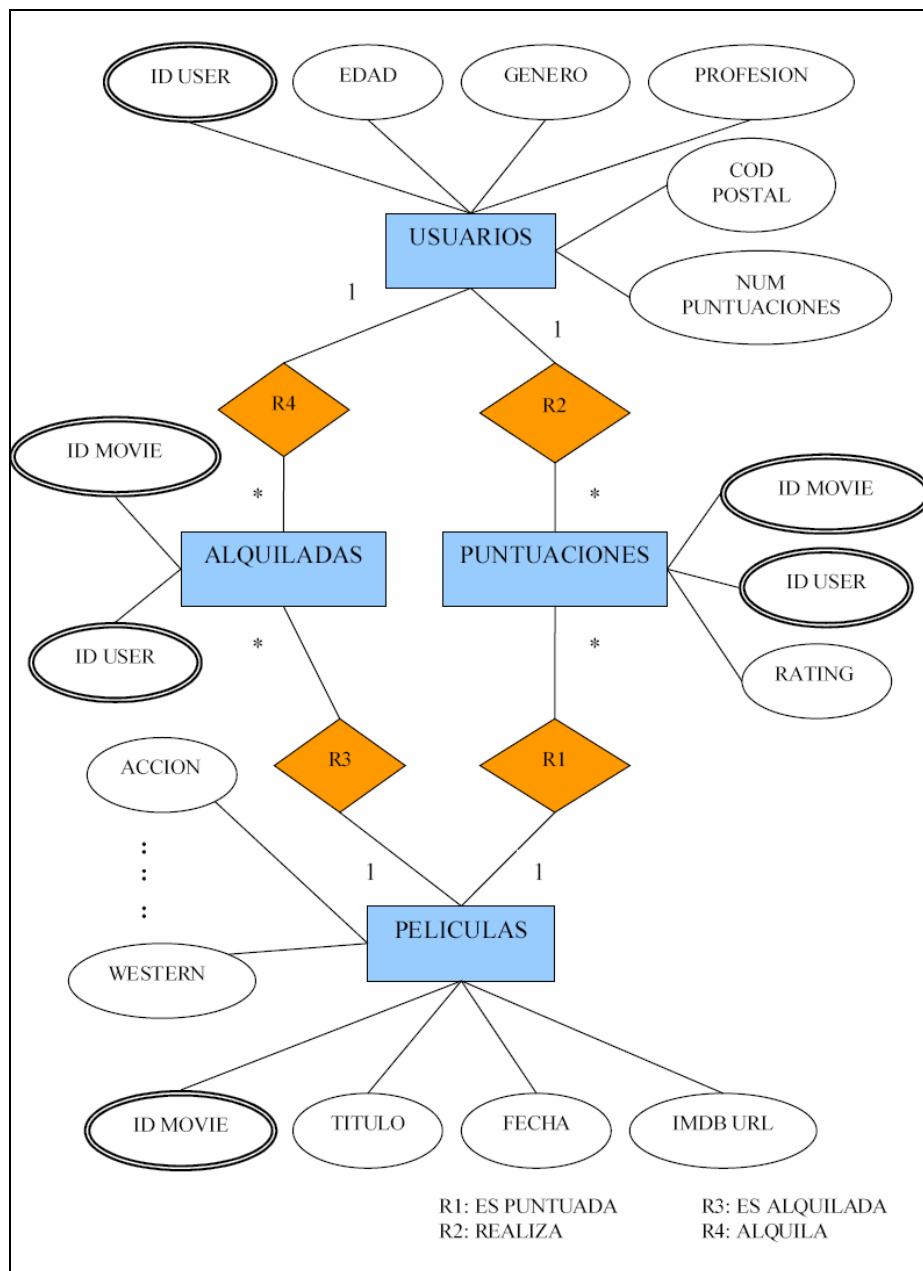


Figura 3.2.6. Esquema Conceptual

2) Esquema Conceptual Modificado

Para obtener el Esquema Conceptual Modificado debemos eliminar todas las entidades débiles, relaciones muchos a muchos y relaciones con atributos que haya en nuestro Esquema Conceptual. Por lo tanto, nuestro ECM queda como sigue:

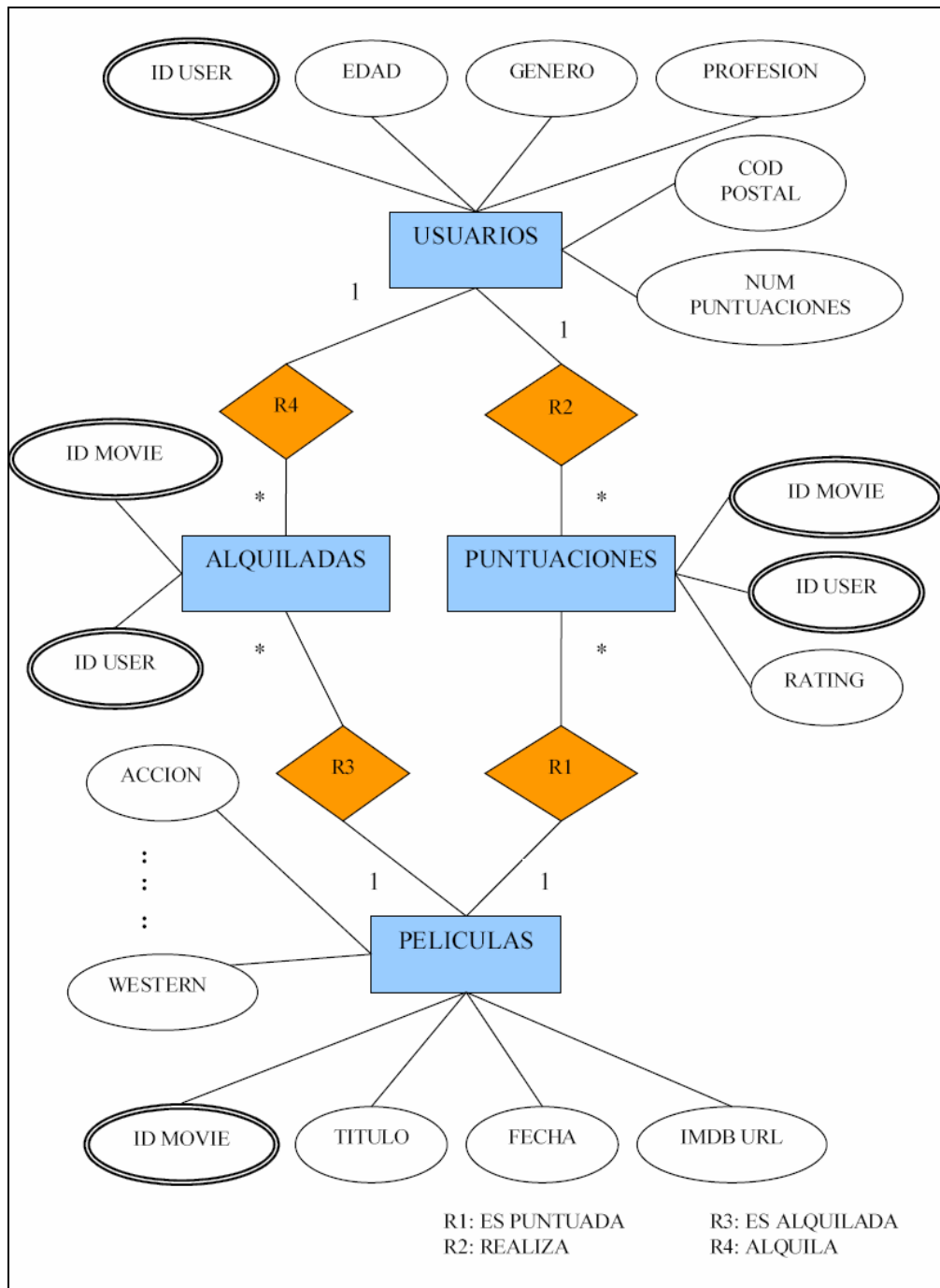


Figura 3.2.7. Esquema Conceptual Modificado

3) Tablas de la aplicación

A partir del ECM obtenido previamente podemos determinar las tablas de la base de datos, teniendo en cuenta que:

- Cada entidad del ECM se transforma en una tabla.
- Los atributos de una entidad se convierten en los campos de las tablas respectivas.

Por lo tanto, obtendremos las siguientes cuatro tablas: USUARIOS, PUNTUACIONES, PELICULAS y ALQUILADAS. A continuación se detallan cada una de estas tablas.

USUARIOS

Tabla que contiene la información sobre los usuarios de la aplicación. Esta formada por los siguientes campos:

CAMPO	TIPO	DESCRIPCIÓN	CLAVE
ID_USER	NUMBER(3)	Identificador unívoco de usuario	*
EDAD	NUMBER(3)	Edad del usuario	
GENERO	VARCHAR(1)	M si el usuario es hombre y F si es mujer	
PROFESION	VARCHAR(25)	Profesión del usuario	
COD_POSTAL	VARCHAR(5)	Código Postal del usuario	
NUM_PUNTUACIONES	NUMBER(3)	Número de películas que ha puntuado el usuario. Campo calculable.	

Tabla 3.2.1. Campos de la tabla USUARIOS

PELICULAS

Tabla que contiene la información de todas la películas de la aplicación. Esta formada por los siguientes campos:

CAMPO	TIPO	DESCRIPCIÓN	CLAVE
ID_MOVIE	NUMBER(4)	Identificador unívoco de la película	*
TITULO	VARCHAR(81)	Título de la película	
FECHA	DATE	Fecha de estreno de la película	
IMDB_URL	VARCHAR(134)	Dirección URL de la página en IMDB de la película	
DESCONOCIDO	BOOLEAN	La película es de género desconocido	
ACCION	BOOLEAN	La película es de acción	
AVENTURAS	BOOLEAN	La película es de aventuras	
ANIMACION	BOOLEAN	La película es de animación	
INFANTIL	BOOLEAN	La película es infantil	
COMEDIA	BOOLEAN	La película es de comedia	
CRIMEN	BOOLEAN	La película es de crimen	
DOCUMENTAL	BOOLEAN	La película es documental	
DRAMA	BOOLEAN	La película es de drama	
FANTASIA	BOOLEAN	La película es de fantasía	
NEGRO	BOOLEAN	La película es de género negro	
TERROR	BOOLEAN	La película es de terror	
MUSICAL	BOOLEAN	La película es musical	
MISTERIO	BOOLEAN	La película es de misterio	
ROMANTICO	BOOLEAN	La película es romántica	
CIENCIA-FICCION	BOOLEAN	La película es de ciencia-ficción	
THRILLER	BOOLEAN	La película es un thriller	
GUERRA	BOOLEAN	La película es bélica	
WESTERN	BOOLEAN	La película es un western	

Tabla 3.2.2. Campos de la tabla PELICULAS

PUNTUACIONES

Tabla que contiene la información sobre las puntuaciones que hacen los distintos usuarios de la aplicación sobre las distintas películas. Esta formada por los siguientes campos:

CAMPO	TIPO	DESCRIPCIÓN	CLAVE
ID_USER	NUMBER(3)	Identificador unívoco del usuario que realiza la puntuación	*
ID_MOVIE	NUMBER(4)	Identificador unívoco de la película puntuada	*
RATING	NUMBER(1)	Valor de la puntuación. Debe ser 1, 2, 3, 4 o 5. Cualquier otro valor es erróneo	

Tabla 3.2.3. Campos de la tabla PUNTUACIONES

ALQUILADAS

Tabla que contiene la información sobre las películas alquiladas pero todavía no puntuadas por parte de cada usuario de la aplicación. Esta formada por los siguientes campos:

CAMPO	TIPO	DESCRIPCIÓN	CLAVE
ID_USER	NUMBER(3)	Identificador unívoco del usuario que alquila la película	*
ID_MOVIE	NUMBER(4)	Identificador unívoco de la película alquilada	*

Tabla 3.2.4. Campos de la tabla ALQUILADAS

4.3.2 Diseño de la interfaz

En esta fase del diseño del sistema software se define cual va a ser la **apariencia visual de la aplicación**, es decir, se define la interfaz visual entre el usuario y la aplicación. Sin duda, realizar un buen diseño de la interfaz resulta primordial ya que esta debe presentarse atractiva al usuario de la aplicación pero a la vez le debe de resultar fácil de entender y trabajar sobre ella.

Esta importancia se acrecienta aun más en nuestro caso ya que la interfaz de nuestro proyecto es una **interfaz web** y la usabilidad web es un tema candente, foco de cierta controversia. Esta controversia se debe a que para las aplicaciones con interfaces web no existe una guía de estilo estándar como la puede haber, por ejemplo, para desarrollar interfaces para aplicaciones de escritorio de **Windows XP** y que resulten, a la vez, atractivas y familiares. Cada programador, desarrollador o diseñador web debe definir su propia guía de estilo y procurar que, en base a ella, la interfaz resultante consiga unas cotas dignas de atractivo visual, familiaridad y facilidad de uso.

En este apartado vamos a definir nuestra **guía de estilo** y a describir y analizar las **metáforas** empleadas.

A) Definir guía de estilo

Antes de ponerse a diseñar una interfaz de usuario, se debe definir el estilo de la misma. Esto es de vital importancia cuando el diseño va a ser compartido entre varios diseñadores, ya que ayuda a mantener la coherencia interna de la interfaz.

Sin embargo, en contra de lo que pueda parecer en un principio, también es de mucha utilidad definir una guía de estilo cuando solo hay un diseñador encargado de la interfaz. Esto se debe a varias razones:

- A veces es posible que mantener la coherencia y consistencia de una interfaz, si esta es muy grande o muy ambiciosa, sea algo complicado incluso si sólo hay un diseñador si no tiene una base.
- El diseñador primitivo puede, por las más diversas razones, abandonar el diseño y es de utilidad para sus sustitutos contar con una guía de estilo predefinida para no tener que empezar de cero otra vez. Lo mismo puede aplicarse si no es el diseñador original el que se encarga del mantenimiento o la actualización de la interfaz.

Quedando demostrada la utilidad del uso de guías de estilo podemos pasar a definir las reglas, normas y recomendaciones que contendrá la guía de estilo de nuestra interfaz:

- **Fuentes**
 - *Tipo:* Verdana
 - *Tamaño:*
 - Cabecera: 18px
 - Párrafo: 12px
 - Formulario: 12px
 - Otras circunstancias: 9px
 - *Color:*
 - En el Cuerpo: Azul oscuro
 - En el Pie: Gris oscuro

- **Enlaces:** sin subrayado. Fondo Azul Oscuro. Los colores empleados para los enlaces son:
 - *Color:*
 - En el Cuerpo: Azul claro
 - En el Pie: Gris oscuro

- **Colores de fondo**
 - *Cabecera:* Azul claro
 - *Cuerpo:* Blanco
 - *Pie:* Azul claro

- **Logotipo:** arriba a la izquierda (en la Cabecera). Esta presente en todas las páginas del sitio web y siempre sirve como enlace de vuelta a la página de inicio.

- **Ayuda:** abajo a la izquierda (en el Pie). Esta presente en todas las páginas del sitio web.

- **Opciones:** en menús desplegados.

B) Metáforas

Al diseñar una interfaz gráfica, la utilización de metáforas resulta muy útil ya que permiten al usuario, por comparación con otro objeto o concepto, comprender de una manera más intuitiva las diversas tareas que la interfaz permite desarrollar.

Al igual que pasa en el ámbito de la literatura o la lingüística, para que una metáfora cumpla con su cometido, el desarrollador de la aplicación y el usuario final de esta deben tener una base cultural similar. Es muy posible que el uso de un icono de manera metafórica sea entendido de una manera por el usuario occidental y de otra bien distinta por un usuario de extremo oriente. Hay que intentar, por lo tanto, que las metáforas empleadas sean lo más universales posibles para que así sean comprendidas a la perfección por la mayor parte del público potencial.

Las aplicaciones de escritorio de **Windows** suelen seguir la **Guía de Estilo XP** y utilizan una serie de metáforas con las que el usuario está plenamente familiarizado (por ejemplo, una lupa con un signo '+' en su interior establece que la función del icono es, inequívocamente, la de realizar un aumento de zoom). En el mundo de las aplicaciones web también existen una cantidad de metáforas de amplia difusión como puede ser, por ejemplo, el celebre *carrito de la compra* que emplean casi todos los comercios online.

Pero las metáforas no sólo dependen del tipo de aplicación (escritorio o web) sino también del ámbito de la misma. Por ejemplo, el carrito de la compra es una metáfora conocida por todos pero si nuestra aplicación no va a vender nada al usuario no resulta conveniente utilizarla ya que puede confundir. Es por ello, que se ha realizado una revisión de los sitios webs de recomendación de características similares al que se pretende realizar en este proyecto para encontrar las metáforas más comunes entre los sistemas de recomendación y se ha llegado a la conclusión de usar la metáfora siguiente:



Información. El usuario, a la vista de este icono, comprende inmediatamente que si pincha sobre el obtendrá una información sobre la película a la que acompaña. Además, al colocarse encima del mismo, se mostrará una dirección web en la barra inferior del navegador por lo que el usuario sabrá rápidamente que se trata de un enlace externo de información.

Se pensó también en utilizar una metáfora para referirse al hecho de puntuar pero pocos son los sitios que lo utilizan al no haber ninguna lo suficientemente universal y reconocible por todos por lo que al final se desechó la idea.

4.4 Implementación

La implementación es la actividad final de la Ingeniería del Software, aquella en la que el modelo obtenido en las actividades anteriores se debe transformar en código fuente. Para ello se debe ser cuidadoso en la elección del lenguaje de programación empleado para la codificación y de la herramienta utilizada para generarla.

4.4.1. Tipo de arquitectura de la aplicación

En nuestro caso, vamos a desarrollar una sistema de recomendación con una arquitectura cliente/servidor y una interfaz web de comunicación con los usuarios. El funcionamiento de las arquitecturas de este tipo es sencilla: la aplicación se encuentra en un servidor central al que los usuarios acceden a través de un software cliente, en nuestro caso un navegador web. Una vez que ha accedido a la aplicación, el usuario realiza peticiones que el servidor tiene que atender para generar una respuesta comprensible para el cliente.

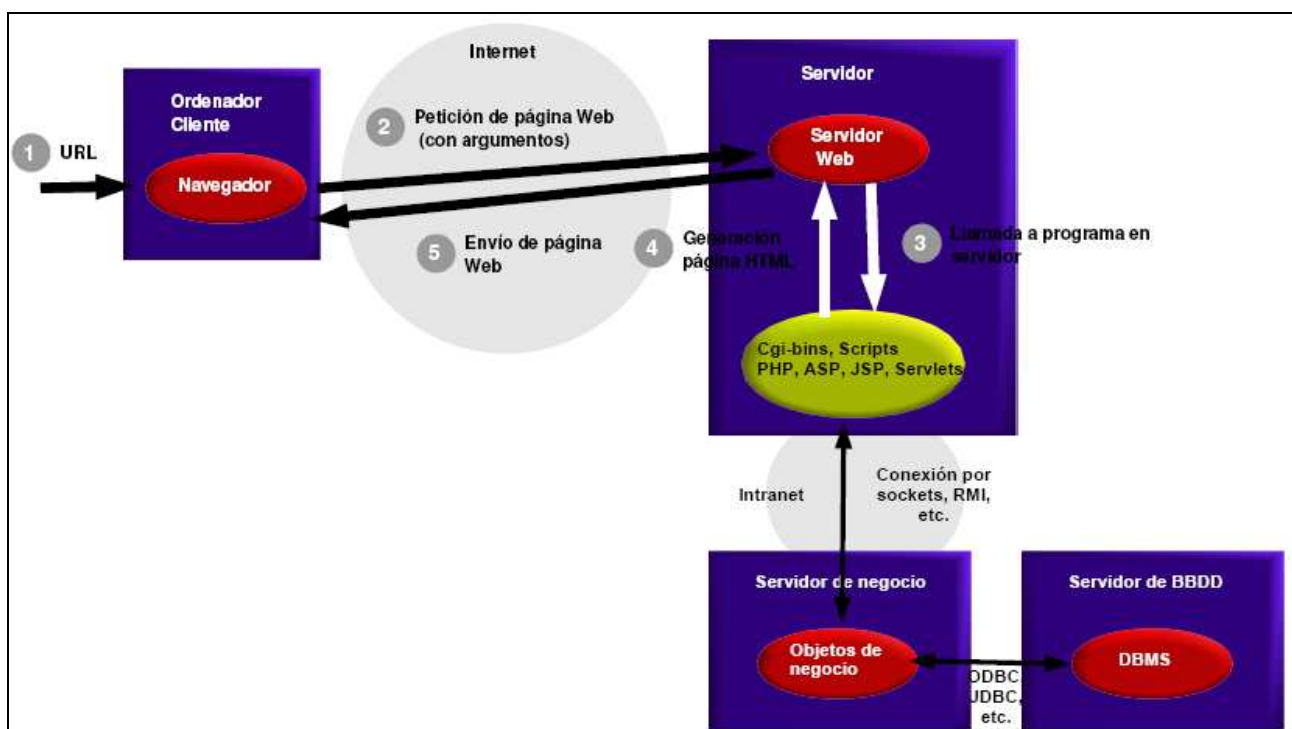


Figura 3.2.8. Arquitectura Cliente/Servidor genérica

Una arquitectura cliente/servidor libera, por lo tanto, al usuario final de la aplicación de tener que instalarla en su máquina y consigue que cada usuario sólo pueda acceder a la información que le corresponde. Además, este tipo de arquitectura, gracias a su diseño modular, es fácilmente escalable y ampliable tanto en nuevos clientes como en servidores añadidos.

4.4.2. Lenguajes de programación utilizados

Resulta obvio ante la arquitectura y el funcionamiento previsto de nuestra aplicación que el uso de HTML simple y llano no es adecuado sino que se necesita otro lenguaje capaz de generar contenido dinámico desde el servidor de manera transparente al usuario final. Existen varias alternativas para realizar esto, desde Perl a ASP (Active Server Pages) o JSP (Java Server Pages) pasando por el uso de CGIs (Common Gateway Interface). Sin embargo, finalmente, nos hemos decantado por el uso de PHP.

PHP, acrónimo de PHP Hypertext Preprocessor, es un lenguaje de programación interpretado, que se ejecuta del lado del servidor y genera contenido dinámico a petición del cliente. Es un lenguaje que tiene una importante serie de ventajas sobre otros lenguajes que realizan funciones parecidas como son las siguientes:

- Es libre, abierto y multiplataforma.
- Sintaxis similar a lenguajes estructurados como C.
- Capacidad de conexión con múltiples gestores de bases de datos.
- Cuenta con mecanismos para trabajar con ficheros, tratar textos, generar imágenes de manera dinámica y tratar documentos XML.
- Esta ampliamente documentado.
- Cuenta con un gran número de extensiones y módulos.
- No requiere declaración de variables.

Estas características le hacen ideal para nuestros propósitos:

- El cliente solicita cualquier funcionalidad.
- El servidor, mediante PHP, conecta con nuestra base de datos en Access y obtiene los datos pertinentes.
- También mediante PHP realiza los cálculos y acciones que sean necesarios sobre esos datos.

- Finalmente, genera el código XHTML adecuado y se lo presenta al cliente de manera transparente.

Con PHP es suficiente para satisfacer las funcionalidades que debe presentar la aplicación a sus usuarios. Sin embargo, para realizar una implementación de la interfaz web adecuada se hace necesario el uso de otros dos lenguajes de programación: CSS y Javascript.

CSS, acrónimo de Cascade Style Sheets, es un lenguaje formal que ayuda a separar la estructura interna de un documento de su presentación externa. Las etiquetas de estilo CSS pueden presentarse tanto dentro de un documento HTML (encerradas dentro de las etiquetas `<style type="text/css"></style>` en la cabecera) como en un documento aparte (con extensión .css) al que el documento HTML se encarga de llamar cuando es necesario. De esta última manera no solo se consigue separar la estructura de la presentación sino que se consigue la centralización del estilo ya que una sola hoja de estilos CSS puede ser invocada por distintas páginas de la aplicación web lo que ayuda de manera muy importante al mantenimiento de la coherencia y consistencia del diseño de la aplicación.

En nuestra aplicación, el uso de hojas de estilo CSS es algo ineludible ya que así se consigue que las sentencias PHP del servidor generen, simplemente, el código XHTML necesario para responder a la petición del cliente sin entrar en temas del diseño o visualización de esta respuesta, de lo que se encargará el estilo CSS predefinido.

Por su parte, Javascript, lenguaje interpretado de sintaxis similar a lenguajes como Java o C que se ejecuta del lado del cliente, ayuda a PHP de otra manera: filtrando los datos de las peticiones de los clientes, dejando realizar la petición al servidor solo cuando estos son válidos. Si los datos son erróneos informan al cliente de su error mediante mensajes de error o alerta.

Al igual que ocurre con CSS, el código Javascript puede ir incrustado dentro del documento HTML (entre las etiquetas `<script type="text/javascript"></script>` en el cuerpo o la cabecera) o estar almacenado en ficheros aparte (con extensión .js) y ser invocados por el documento. Para nuestra aplicación, tanto para los estilos CSS como para el código Javascript, nos hemos decantado por la segunda opción.

4.4.3 Herramienta de desarrollo

Para generar código XHTML, CSS, PHP y Javascript no hace falta ninguna herramienta o entorno específico de desarrollo ya que con un simple editor de textos se pueden escribir las sentencias y etiquetas y guardar el resultado con la extensión correspondiente. Sin embargo, existen multitud de herramientas de desarrollo que facilitan de manera enorme esta tarea de codificación. Las hay tanto libres y gratuitas (como pueden ser Quanta Plus o NVU) como propietarias. Para este proyecto hemos elegido una de las de este último grupo, que además es la más popular de ellas y la que mejores prestaciones ofrece: Macromedia Dreamweaver X 2004. Dreamweaver es una herramienta con capacidad WYSIWYG (What You See Is What You Get) que permite crear y trabajar sobre documentos de HTML, PHP, CSS, Javascript, ASP, Java, C#, JSP, VisualBasic y otros muchos lenguajes de manera sencilla además de proporcionar numerosas plantillas ya prediseñadas. Dispone de previsualizador en distintos navegadores, vistas de código, de diseño e híbrida, validador de código W3C y otras muchas funcionalidades que la hacen una herramienta de desarrollo muy potente para entornos web.

4.4.4 Actualización del algoritmo de filtrado

La base primordial para desarrollar un sistema de recomendación colaborativo es contar con un buen algoritmo de filtrado colaborativo. Para ello es necesario refinarlo conforme los clientes vayan mejorando sus perfiles puntuando nuevas películas. En la aplicación final, este refinamiento o actualización debería realizarse de manera inmediata cada vez que hubiera una nueva puntuación. Sin embargo, al centrarse este proyecto en realizar un modelo prototipal, no contempla esta funcionalidad sino que el refinamiento se realizará de manera periódica a petición del administrador del servicio.

Para realizar esto se ejecutará el programa de actualización en Java desde el servidor que se encargará de actualizar el algoritmo. Esta actualización requerirá de un tiempo importante para realizarse ya que se deben recalcular los grupos de vecinos, es por ello que se ha optado por la opción de ejecutarlo en segundo plano y devuelva el resultado cuando termine.

4.4.5 Instalación en el servidor y funcionamiento de MoviesRecommender II

La instalación de la aplicación MoviesRecommender II en el servidor viene documentada paso a paso en el Anexo I. Por su parte, en el Anexo II se encuentra disponible un manual de usuario donde se detalla el funcionamiento desde el punto de vista del cliente de MoviesRecommender II.

CAPÍTULO 5.

CONCLUSIONES.

El objetivo de este proyecto es obtener un Sistema de Recomendación basado en Filtrado Colaborativo. Dado que no existe un algoritmo de filtrado colaborativo mejor que otro, hemos realizado un estudio comparativo entre los distintos algoritmos básicos de filtrado colaborativo y algoritmos básicos mejorados de filtrado colaborativo (aplicando el voto por defecto, la amplificación de casos y la frecuencia inversa/directa de usuario), para obtener el algoritmo que mejor convenga para nuestra base de datos.

Este estudio ha usado la base de datos MovieLens. Esta base de datos proporciona 943 usuarios, 1.682 películas y 100.000 puntuaciones realizadas por los usuarios. Para trabajar con ella, se ha preferido implementar un algoritmo de filtrado colaborativo basado en ítem con el fin de evitar trabajar con todos los datos almacenados como hacen los algoritmos basados en usuario.

Sabemos que existen tantos algoritmos de filtrado colaborativo basados en ítem como combinaciones posibles de los parámetros que intervengan en ellos. Por tanto, se ha realizado un estudio comparativo entre los resultados obtenidos por los algoritmos básicos de filtrado colaborativo y los algoritmos básicos mejorados para estudiar el que mejor comportamiento presente en el caso particular de la base de datos tratada.

Una vez realizado este estudio comparativo hemos concluido que el algoritmo básico mejorado de filtrado colaborativo con frecuencia directa de usuario es el que mejores resultados ha conseguido respecto a la base de datos. Por tanto, este es el algoritmo que se ha empleado para implementar el sistema de recomendación basado en algoritmos de filtrado colaborativo.

El sistema de recomendación implementado es una versión prototipal y se basa en una arquitectura cliente/servidor con una interfaz web que permite a los clientes conectarse de manera remota desde su navegador web a la aplicación que se encuentra alojada en un servidor central.

Esta versión de prueba toma en consideración los gustos de un usuario dado sobre las películas que haya alquilado o visto anteriormente por medio de sus puntuaciones. En función de estas puntuaciones el sistema crea un perfil de cada cliente y lo compara con los perfiles del resto de clientes. A continuación recomienda a un usuario determinado aquellas películas que puedan ser más atractivas para él de entre las favoritas del grupo de clientes más afines a

él.

Finalmente, comentar que para el autor, el haber realizado este proyecto ha sido todo un desafío ya que en un principio la temática en la que se encuadra dentro de la informática era totalmente desconocida. Como aspecto positivo destacar el hecho de haber podido aplicar muchas de las metodologías y habilidades adquiridas durante los años de estudio, así como la posibilidad de aprender otras muchas durante el desarrollo del propio proyecto.

ANEXO I.

MANUAL DE INSTALACIÓN

DEL SERVIDOR.

Existen dos alternativas para poner en funcionamiento el software **MoviesRecommender II**. La primera consiste en alojar la aplicación en un servidor de hosting que de soporte al lenguaje **PHP** y al gestor de bases de datos **MS Access**. La otra alternativa, la cual ha resultado la elegida, consiste en montar nuestro propio servidor. En este anexo se especificará paso a paso como realizar esta operación en el entorno de un sistema operativo **Windows XP Professional Service Pack 2**.

Material necesario

Todo el material necesario para instalar y dejar operativo el servidor se encuentra disponible en el CD que acompaña a esta memoria. Vaya a **D:\moviesrecommenderII** (suponiendo que D: es su unidad de cd o dvd) y compruebe que en el directorio se encuentran los siguientes archivos:

- apache_2.2.3-win32-x86-no_ssl.ins
- bbdd.zip
- httpd.conf
- jdk-6-windows-i586.exe
- mod_jk.os
- moviesrecommenderII_files.zip
- php5_2.zip

Si es así podemos proceder al montaje de nuestro servidor inmediatamente. Si falta algún archivo o alguno de ellos se encuentra dañado pongase en contacto con el responsable de la aplicación para subsanar el percance.

Paso 1: Instalar Apache

Apache es un servidor HTTP de código abierto y multiplataforma desarrollado por la Apache Software Foundation, en cuya web (<http://www.apache.org>) se pueden conseguir la última versión del servidor, sus múltiples módulos de desarrollo y ampliación y toda la documentación necesaria para su correcto funcionamiento. Se trata, con diferencia, del más popular de los servidores HTTP de la actualidad.

Si ya dispone de de la versión 2.2.3 de Apache instalada en su computador dirígase al

Paso 2 de este manual. Si dispone de una versión anterior sería conveniente que la eliminara y siguiera leyendo este paso para instalar esta versión, que es la que nos asegura que el servidor va a ser montado correctamente. Si no dispone de ninguna versión de Apache siga leyendo.

Vaya a **D:\moviesrecomender** y ejecute el archivo instalador **apache_2.2.3-win32-x86-no_ssl.ins**. Le aparecerá una pantalla de bienvenida como la siguiente:



Figura 1. Bienvenida del instalador de Apache

Pulse el botón **Next**. Aparecerá la pantalla de aceptación de la licencia del producto (Figura 2). En ella hay dos botones de radio, seleccione el de la confirmación de aceptación y pulse el botón **Next**.



Figura 2. Aceptación de la licencia de Apache

La siguiente pantalla que aparecerá ante usted es una pantalla de información sobre el servidor HTTP Apache (Figura 3). Una vez leído, pulse el botón **Next**.



Figura 3. Lectura de información sobre Apache

Una vez realizado este paso nos encontramos ante una de las pantallas más importantes de la instalación: la de la información del servidor (Figura 4). Se deben introducir el nombre del dominio, el nombre del servidor y el e-mail del administrador. Usted puede rellenar estos campos con los datos por defecto que se muestran en la figura o introducir los suyos propios. En cualquiera de los casos es recomendable que apunte la información introducida ya que será necesaria en un futuro.



Figura 4. Introducir información del servidor

En la misma pantalla hay dos botones de radio: el de arriba permite la utilización como servicio del servidor Apache a todos los usuarios del computador; el de abajo, por su parte, permite solo su utilización al usuario actual. Lo recomendado es seleccionar el la primera opción. Una vez hecho esto pulse el botón **Next**.

La siguiente pantalla (Figura 5) nos ofrece dos tipos de instalación: típica y personalizada por el usuario (custom). Se recomienda elegir la primera opción. Una vez hecho esto pulse el botón **Next**.



Figura 5. Elegir tipo de instalación

Una vez elegido el tipo de instalación llega el momento de elegir el directorio donde se va a instalar Apache (Figura 6). Es recomendable realizar la instalación en el directorio por defecto que muestra el instalador (normalmente dentro del directorio **Archivos de Programa** de su disco duro principal) pero si prefiere otra localización pulse sobre el botón **Change** y elija la que mejor le parezca. Una vez elegido el directorio de instalación pulse el botón **Next**.



Figura 6. Seleccionar directorio de destino

La siguiente pantalla que aparecerá será una de confirmación (Figura 7). Si esta seguro de realizar la instalación pulse sobre el botón **Install**, si quiere cancelar la instalación pulse **Cancel** y si no esta seguro de alguno de los pasos anteriores pulse **Back** para realizar los cambios oportunos.

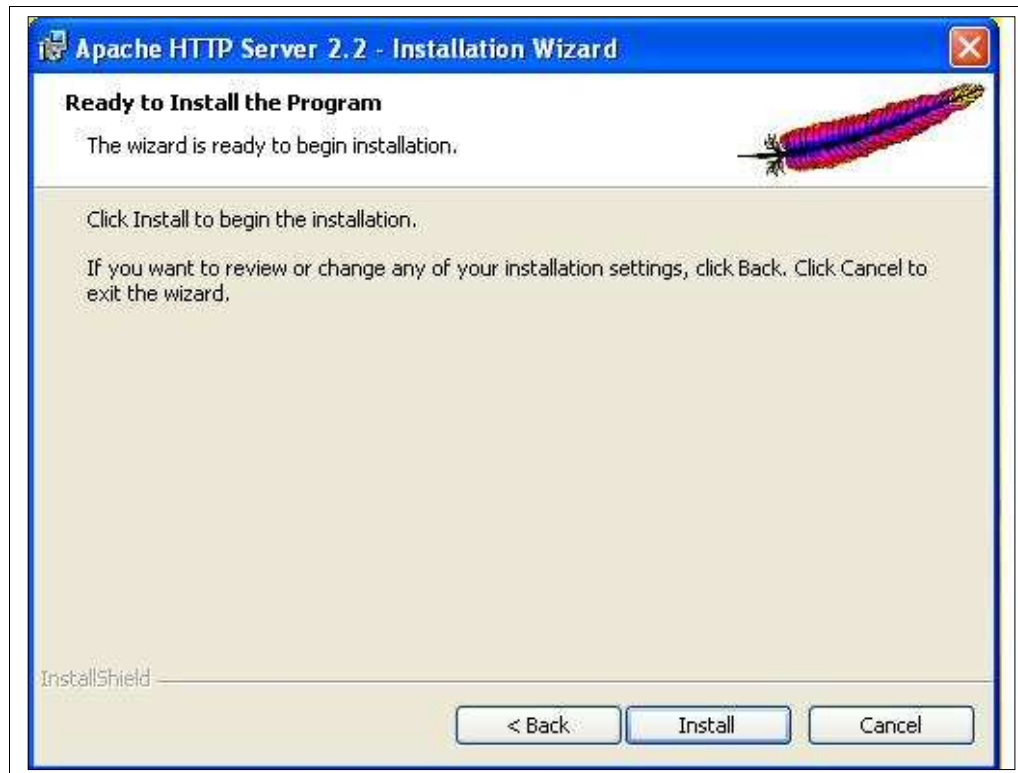


Figura 7. Confirmar la instalación

Si ha pulsado **Install** le aparecerá una pantalla donde se mostrará el progreso de la instalación (Figura 8).



Figura 8. Progreso de la instalación

Una vez terminada la instalación, se le mostrará una pantalla (Figura 9) confirmando que la instalación se ha realizado de manera correcta y exitosa.



Figura 9. Instalación finalizada correctamente

Una vez pulsado el botón **Finish** podremos comprobar que la instalación ha sido un éxito con la aparición del siguiente icono en la barra de inicio:



Figura 10. Icono de Apache 2.2

Paso 2: Instalar PHP

La instalación del lenguaje PHP es mucho menos laboriosa y costosa en tiempo que la del servidor Apache. Si usted ya dispone de la versión 5.2 de PHP instalada en su computador vaya a **Paso 3**. Si, por el contrario, dispone de una versión anterior o no dispone de ninguna siga leyendo este paso:

Vaya a **D:\moviesrecommenderII**, descomprima el archivo **php5_2.zip** y extraiga los archivos que se encuentran en su interior. Es recomendable que la extracción se produzca en **C:\php5_2** pero usted puede elegir la carpeta o directorio de destino que desee siempre y cuando tenga bien presente la ruta elegida ya que será de utilidad para el siguiente paso a realizar.

Paso 3: Configurar Apache y PHP

Una vez llegados a este punto, tanto Apache como PHP se encuentran instalados en la máquina pero la aplicación no puede ser puesta en funcionamiento debido a que no están conectados entre sí. Para conseguir esta conexión entre el servidor y el lenguaje se hace necesario modificar el archivo de configuración de Apache para insertar a PHP como un módulo del servidor.

Si usted ha instalado PHP en una ruta diferente a la especificada por defecto en **Paso 2** deberá realizar un paso previo antes de la modificación del archivo de configuración de Apache:

- Vaya a la carpeta donde tenga instalado PHP, abra con cualquier editor de textos el archivo **php.ini**, y encuentre las siguientes líneas:

```
    ; Directory in which the loadable extensions (modules) reside.  
extension_dir = "c:\php5_2\ext"
```

- Sustituya la ruta señalada en negrita por la que corresponda a la carpeta **ext** dentro del directorio donde tenga instalado PHP y guarde los cambios realizados.

Una vez hecho esto puede pasar ya a modificar el archivo de configuración de Apache aunque primero debemos asegurarnos de que Apache esta detenido y, si no lo esta, detenerlo (Figura 11).



Figura 11. Parar la ejecución de Apache

Ahora sustituya el archivo de configuración **httpd.conf** que se encuentra en **C:\Archivos de programa\Apache Software Foundation\Apache2.2\conf** (o, si no ha instalado Apache en el directorio por defecto, en la carpeta **conf** de la ruta donde lo haya instalado) por el que se encuentra en **D:\moviesrecommenderII**. Abra este nuevo archivo de configuración con un editor de texto cualquiera y busque las siguientes líneas:

```
# ServerRoot: The top of the directory tree under which the server's  
# configuration, error, and log files are kept.  
#  
# Do not add a slash at the end of the directory path. If you point  
# ServerRoot at a non-local disk, be sure to point the LockFile directive  
# at a local disk. If you wish to share the same ServerRoot for multiple  
# httpd daemons, you will need to change at least LockFile and PidFile.  
#  
ServerRoot "C:/Archivos de programa/Apache Software Foundation/Apache2.2"
```

Sustituya, si es necesario, la ruta donde se encuentra Apache teniendo en cuenta de aquí en adelante que dentro del archivo **httpd.conf** las barras van invertidas con respecto a las

barras normales de las rutas de Windows (es decir, si hay una “\” deberá transformarla en una “/” al copiarla en **httpd.conf**).

Una vez realizado este cambio busque las siguientes líneas:

```
#LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule php5_module "c:/php5_2/php5apache2_2.dll"
#LoadModule proxy_module modules/mod_proxy.so
```

Sustituya, en caso de ser necesario, la ruta donde se encuentra el archivo **php5apache2_2.dll** teniendo en cuenta la misma consideración que en el párrafo anterior. Una vez realizado el cambio busque las siguientes líneas:

```
ServerAdmin yo@localhost
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost:80
```

Si durante la instalación de Apache en el **Paso 1** usó otros datos para el nombre del servidor y la dirección de e-mail sustituya los valores de **ServerName** y **ServerAdmin** por esos datos. Una vez hechos estos cambios busque las siguientes líneas:

```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/Archivos de programa/Apache Software
Foundation/Apache2.2/htdocs"
```

Sustituya la ruta de **DocumentRoot** por la que usted considere oportuna (por ejemplo, **C:\moviesrecommenderII\WebDocs**) teniendo en cuenta que será en esa ruta donde deberán ir los archivos que quiera ejecutar con el servidor Apache. Una vez hecho este cambio guarde los cambios efectuados en **httpd.conf** y ponga en marcha de nuevo Apache.

Paso 4: Descomprimir archivos

Vaya a **D:\moviesrecommenderII** y descomprima el archivo **moviesrecommenderII_files.zip** extrayendo su contenido en el directorio que eligiera en el **Paso 3** como **DocumentRoot**. Luego vaya a su navegador Firefox (**MoviesRecommender II** es una aplicación web optimizada para este navegador) o en su defecto cualquier otro y copie en la barra de direcciones lo siguiente: <http://localhost/index.php> (sustituyendo “localhost” por el nombre de dominio que eligiera al instalar Apache en el **Paso 1**). Si obtiene la siguiente pantalla en su navegador (Figura 12) la instalación habrá sido un éxito:

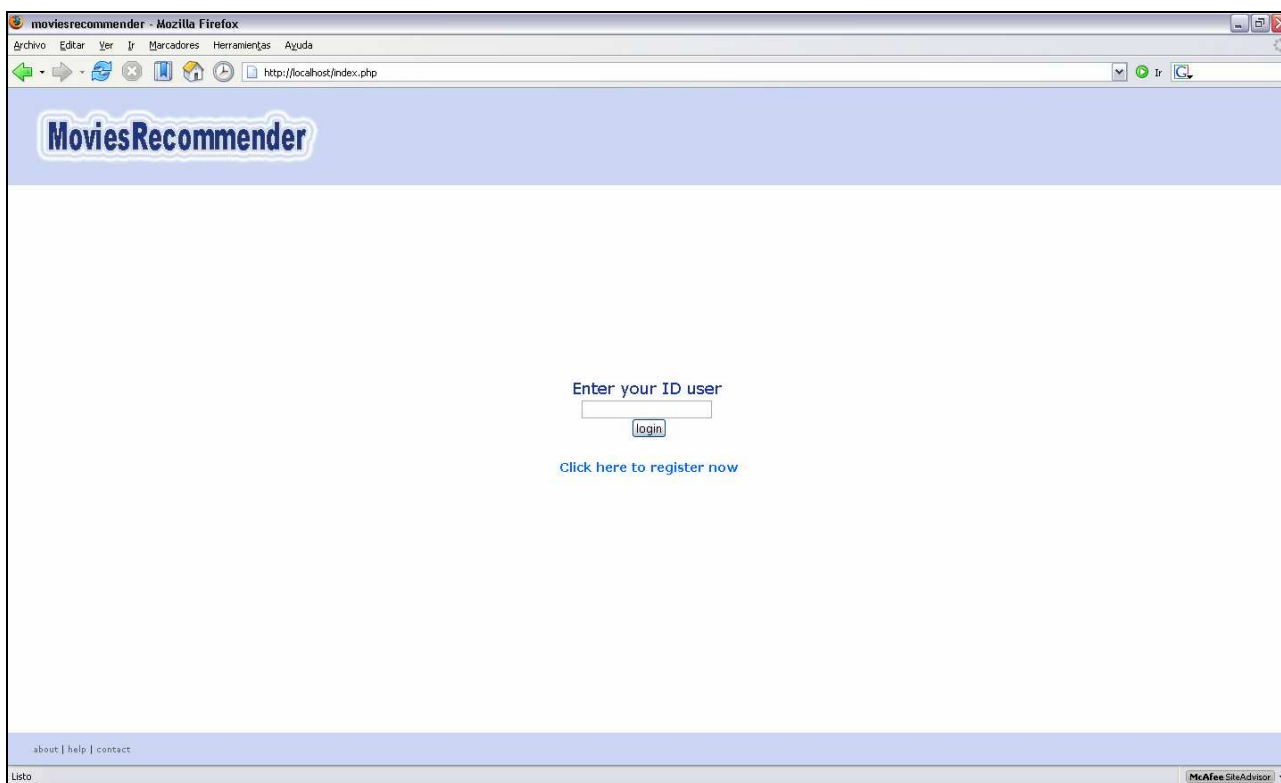


Figura 12. Página de inicio de MoviesRecommender II

Paso 5: Conectar base de datos

La instalación del servidor web ya ha terminado pero todavía queda otro paso para que la aplicación sea operativa: conectarla con la bases de datos utilizada. Este tipo de conexión se llama **ODBC** y el primer paso es ir a **D:\moviesrecommenderII** y descomprimir el archivo **bbdd.zip** que en el se encuentra extrayendo el fichero *movieranks.mdb* en el lugar de su disco duro que prefiera (recomendado: **C:\moviesrecommenderII\BasesdeDatos**).

Luego debe ir a su **Panel de Control -> Herramientas administrativas -> Orígenes de datos ODBC -> DNS del Sistema** (Figura 13):

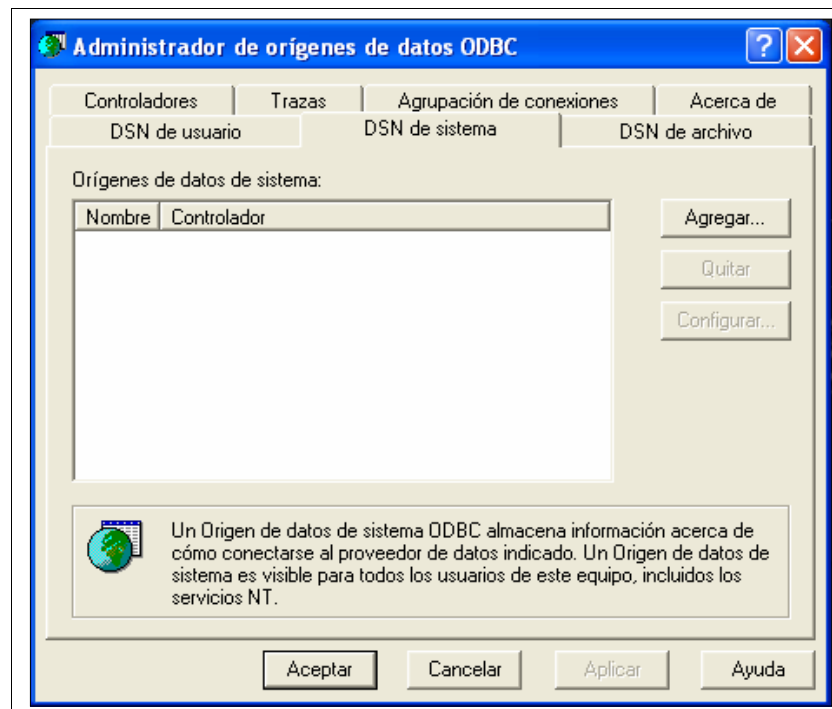


Figura 13. Administrador de orígenes ODBC

Pulse sobre el botón **Agregar** para añadir el origen de la primera de las bases de datos indicando que se trata de una base de datos MS Access (Figura 14):



Figura 14. Crear nuevo origen de datos

Pulse el botón **Finalizar** y en la siguiente pantalla (Figura 15) escriba **movierank** como **Nombre del origen de datos** y pulse el botón **Seleccionar**.



Figura 15. Configurar ODBC para base de datos movierank

Al pulsar **Seleccionar** se entra en una pantalla de selección (Figura 16) donde tiene que buscar la base de datos principal (movieranks.mdb). Una vez localizada y seleccionada pulse **Aceptar** para volver a la pantalla anterior donde también debe pulsar **Aceptar**.



Figura 16. Seleccionar base de datos

Ahora la aplicación está conectada a la base de datos con la que debe trabajar y el servidor HTTP Apache está correctamente configurado con el lenguaje PHP como uno de sus módulos. La instalación ha terminado satisfactoriamente.

ANEXO II.

MANUAL DE USUARIO.

Este manual de usuario esta organizado como una visita guiada por la aplicación pero antes de embarcarse en ella es conveniente que el usuario tenga claros algunos aspectos:

- **MoviesRecommender II** es una aplicación web optimizada para su visualización en un navegador **Firefox** (a ser posible su versión más reciente, la cual se puede descargar en <http://www.mozilla-europe.org/es/>) y con una resolución no inferior a **1024x768** pixels. Si se utiliza otro navegador o una resolución inferior a la recomendada se pueden producir fallos de visualización aunque la funcionalidad de la aplicación esta completamente asegurada.
- En este punto de su desarrollo, **MoviesRecommender II** es sólo un prototipo que no permite la inserción de nuevas películas en la base de datos. Cuando la aplicación esté completamente desarrollada, estas funcionalidades y otras muchas estarán disponibles para los usuarios.
- Debido a que los datos de la base de datos son sobre usuarios norteamericanos y se encuentran en inglés se ha decido implementar **MoviesRecommender II** en ese idioma. De cualquier forma se trata de un ingles muy básico y estandarizado por lo que no hace falta que el usuario sea un experto para disfrutar de las funcionalidades que ofrece la aplicación.

Una vez aclarados los puntos anteriores puede empezar la visita guiada por la aplicación.

Lo primero que se encontrará el usuario al iniciar **MoviesRecommender II** (<http://localhost/index.php>) será una **página de inicio** como la que se puede observar en la figura:



Figura 1. Página de inicio de MoviesRecommender II

Usuario no registrado

Si el usuario no dispone de ID puede registrarse y obtenerlo automáticamente pulsando [If you are not a register user, click here to register now](#) y rellenar los datos obligatorios del siguiente formulario.

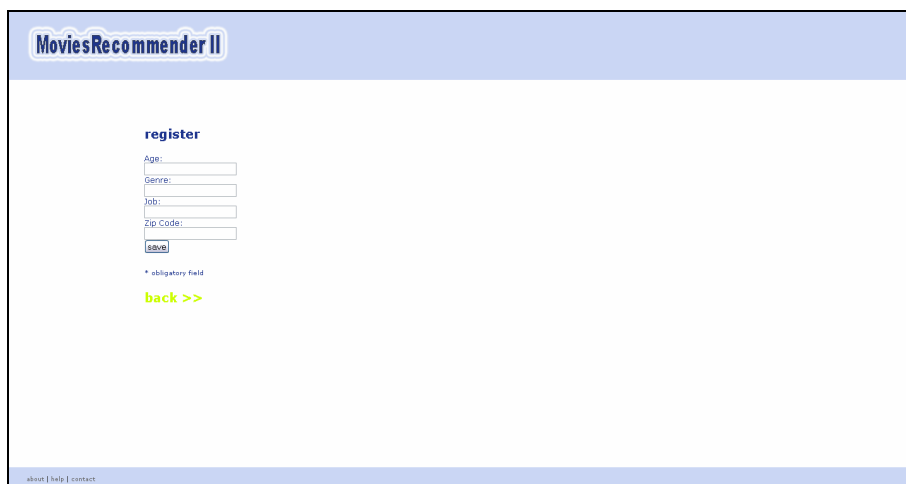


Figura 2. Página de registro de usuario

Si envía el formulario (con el botón **save**) sin haberlo rellenado, la base de datos no se actualiza pero si introduce datos en los campos estos deben ser correctos. Por ejemplo, si introduce una edad incorrecta recibirá el aviso siguiente:



Figura 3. Aviso de edad incorrecta

Y si introduce algo que no sea "M" o "F" en el campo de género el aviso será como el siguiente:



Figura 4. Aviso de género incorrecto

Si los datos introducidos son correctos y la operación con la base de datos se desarrolla de manera exitosa será conducido a la siguiente pantalla:

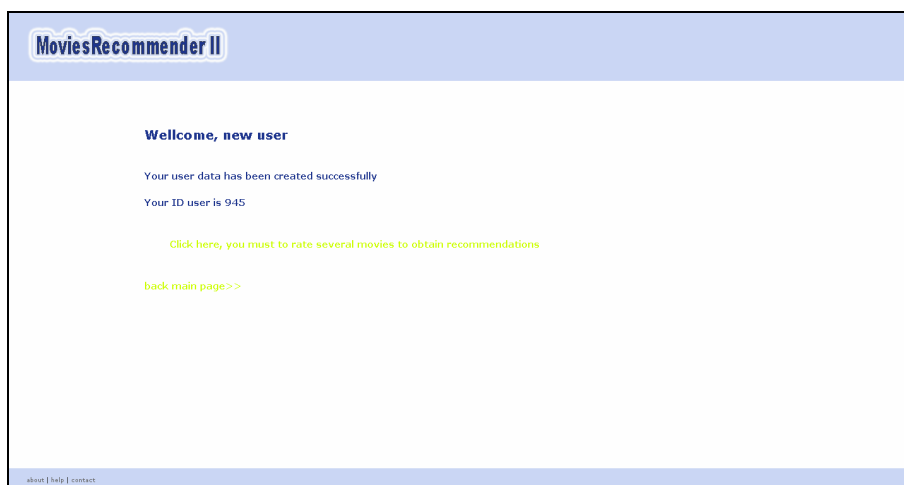


Figura 5. Éxito en el registro

A continuación, el nuevo usuario registrado tendrá que puntuar 20 películas para poder obtener recomendaciones, por este motivo tiene que pinchar sobre el enlace [Clic here, you must to rate several movies to obtain recommendations.](#)

La página que se nos mostrará tendrá el siguiente aspecto:



Figura 6. Página de nuevo usuario

En ella se indica al usuario nuevo que debe valorar 20 películas para obtener recomendaciones de películas interesantes. Por eso, seleccionando la opción [View all movies](#), se podrán ver todas las películas posibles para poder puntuarlas.



Figura 7. Página de listado de películas para nuevo usuario

Seleccionando alguna de ellas nos llevará a otra página en la cual se mostrará la información de la película y se nos da la opción de puntuarla:

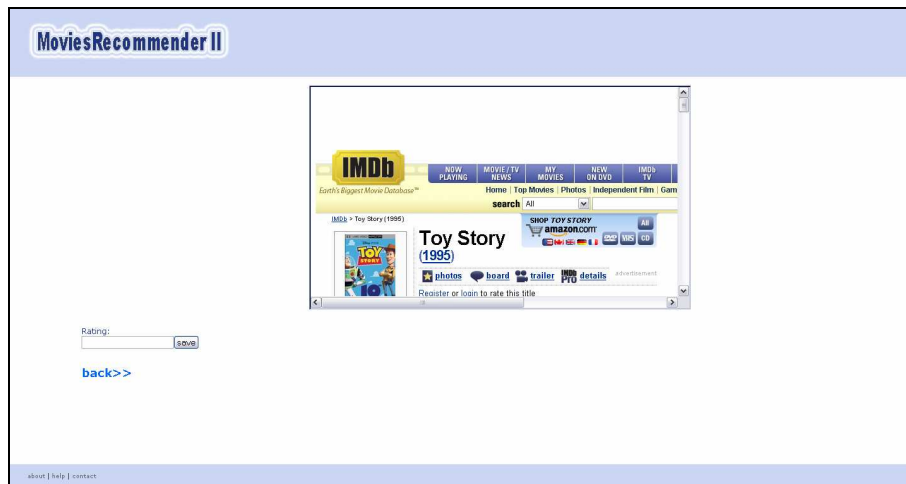


Figura 8. Página de puntuación de películas para nuevo usuario

Cuando el usuario introduce la puntuación aparecerá una ventana de éxito indicando que la puntuación se ha realizado correctamente:

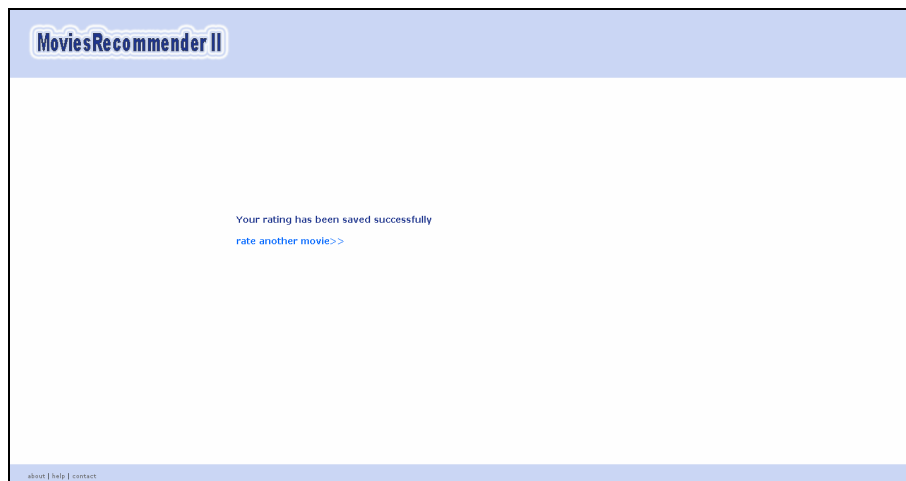


Figura 9. Página de éxito en la puntuación de películas para nuevo usuario

A continuación, para volver a la página para realizar otra puntuación, basta con pinchar sobre el enlace [rate another movie>>](#). De esa manera veremos que ya nos queda una película menos por puntuar para obtener las recomendaciones

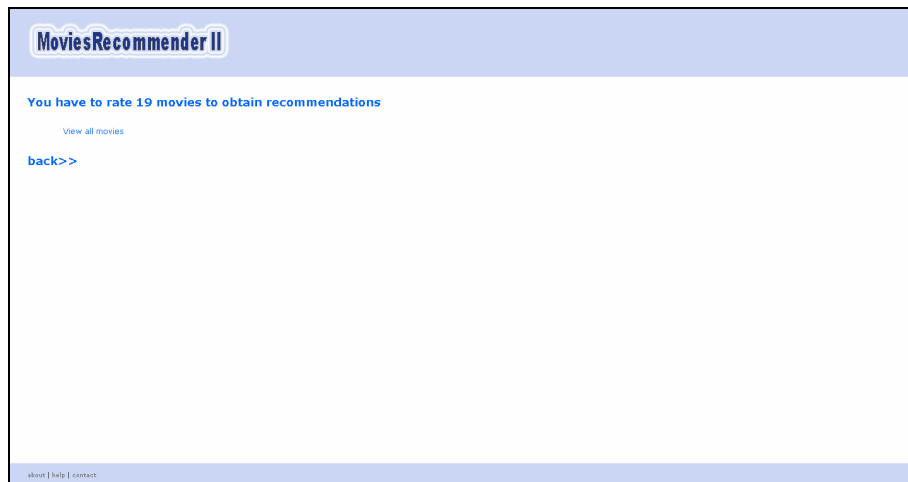


Figura 10. Página de nuevo usuario indicando las películas que faltan por puntuar

Usuario registrado

Para entrar en el sistema el usuario deberá introducir su identificador de usuario en la página de inicio (ver Figura 1) y pulsar el botón **login**. Si pulsa el botón sin haber introducido nada se mostrará el siguiente aviso:



Figura 11. Aviso para insertar una ID de usuario

Si por contra pulsa el botón habiendo introducido un identificador incorrecto recibirá el siguiente aviso:



Figura 12. Aviso de ID de usuario incorrecta

Finalmente, si el usuario introduce una ID correcta accederá a su **página principal de usuario**:



Figura 13. Página principal de usuario

En esta página principal el sistema dará la bienvenida al usuario y le ofrecerá siete posibles opciones:

1. View my user data

Si el usuario pincha sobre el enlace se desplegará un menú en el que podrá comprobar su edad, género (Male or Female), profesión, código postal y el número de películas que ha visto. Pinchando de nuevo sobre el enlace este menú se plegará.



Figura 14. Datos del usuario

2. Modify my user data

Si el usuario pincha sobre el enlace se desplegará un menú con un formulario que le permitirá cambiar sus datos de usuario.



The screenshot shows the 'MoviesRecommender II' web application interface. At the top, there is a header with the application name. Below the header, a welcome message reads 'Hi, user 24. Wellcome to moviesrecommender...'. Underneath, a section titled 'What do you wanna do?' contains several links: 'View my user data', 'Modify my user data', 'View my rated movies', 'View my rent but not rated movies', 'View all movies', 'Make me a recomendabon, please', and 'Log Out'. The 'Modify my user data' link is selected, displaying a form with input fields for 'Age:', 'Genre:', 'Job:', and 'Zip Code:'. A 'save' button is located at the bottom of the form. At the very bottom of the page, there are small links for 'about | help | contact'.

Figura 15. Modificar datos del usuario

Si envía el formulario (con el botón **save**) sin haberlo rellenado, la base de datos no se actualiza pero si introduce datos en los campos estos deben ser correctos. Por ejemplo, si introduce una edad incorrecta recibirá el aviso siguiente:



Figura 16. Aviso de edad incorrecta

Y si introduce algo que no sea “M” o “F” en el campo de género el aviso será como el siguiente:



Figura 17. Aviso de género incorrecto

Si los datos introducidos son correctos y la operación con la base de datos se desarrolla de manera exitosa será conducido a la siguiente pantalla:

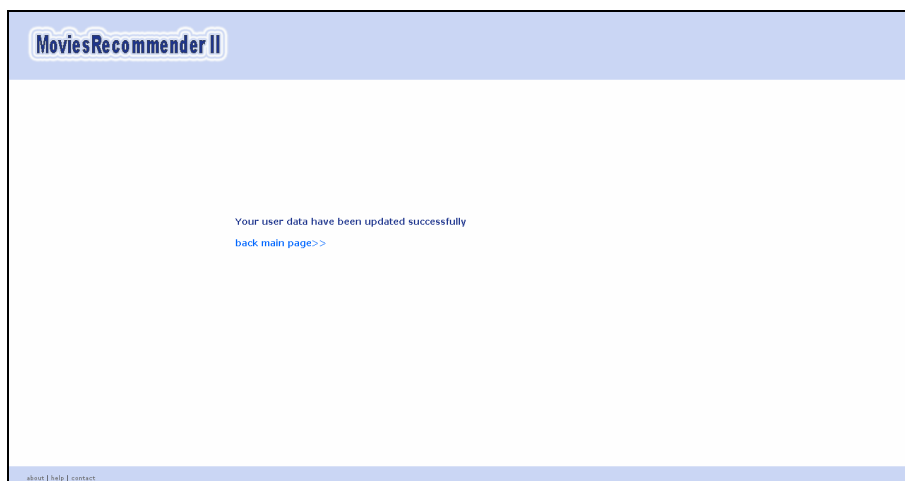



Figura 18. Éxito al modificar datos

Para volver a su página principal de usuario pulse el enlace.

3. View my rated movies

En esta opción se despliegan todas las películas que el usuario ha puntuado con su título, su año de estreno, la puntuación recibida y un icono  a la izquierda que nos dirige a la entrada en **IMDB** (<http://us.imdb.com>) de la película en cuestión.

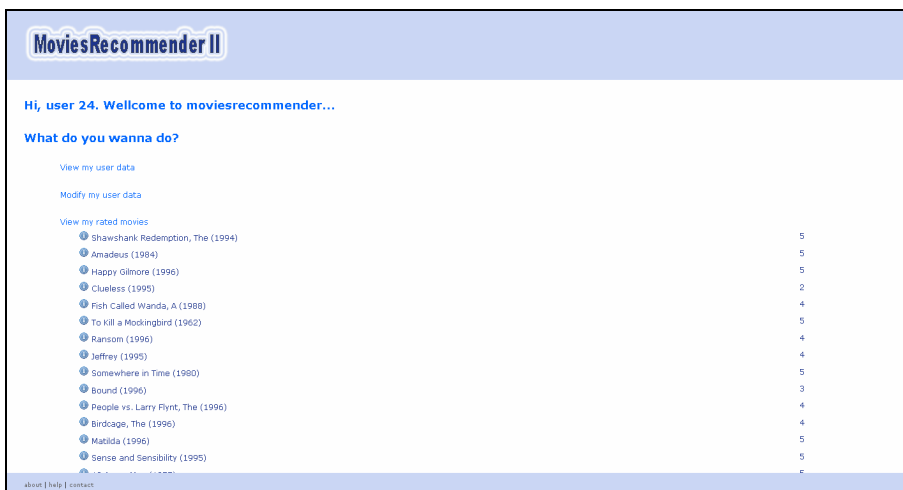


Figura 19. Películas puntuadas por el usuario

4. View my rent but not rated movies

En esta opción el usuario puede visualizar aquellas películas que ha alquilado pero que todavía no ha puntuado además de poder puntuarlas:

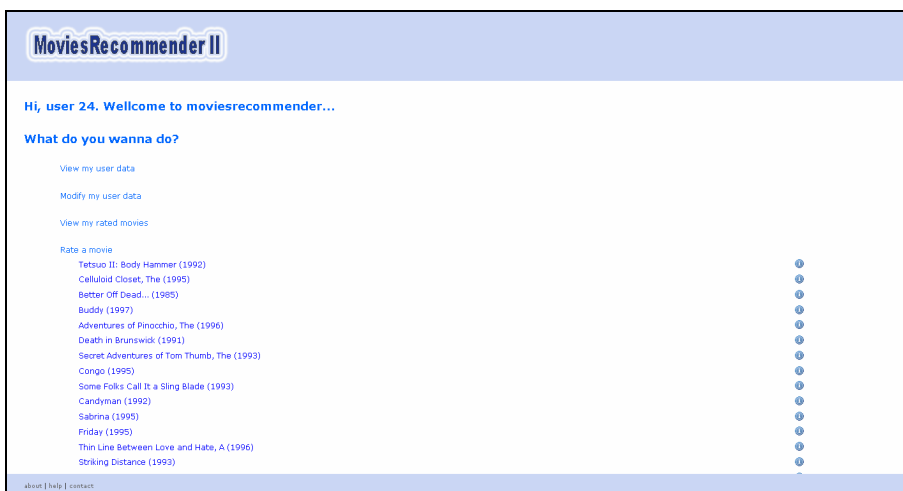


Figura 20. Películas alquiladas pero no puntuadas

Es conveniente que el usuario sea conocedor de la importancia de realizar puntuaciones ya que esto hace crecer y mejorar el perfil de usuario lo cual permite mejorar las predicciones del sistema de recomendación.

Para poder realizar una puntuación el usuario deberá pulsar sobre el título de la película y accederá a una nueva página que le permitirá realizar la acción:

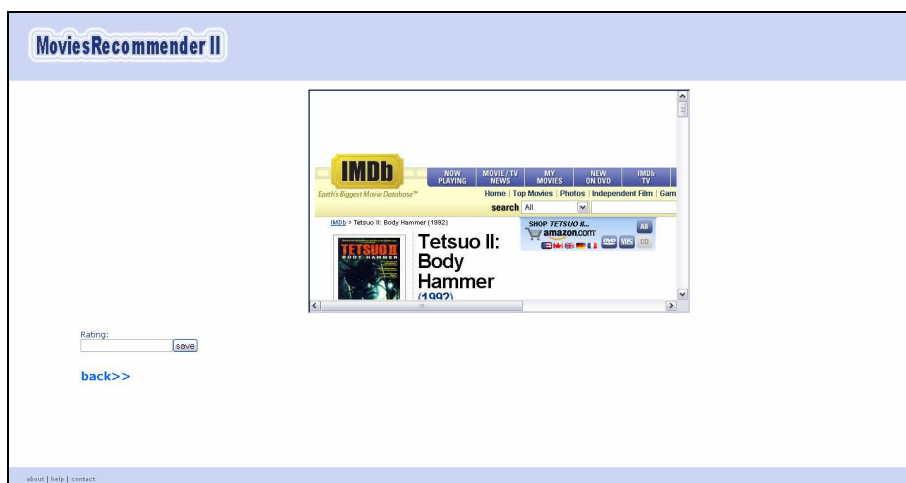


Figura 21. Página para puntuar película

En esta página el usuario encontrará una ventana o *frame* con la entrada en **IMDB** de la película (sobre la cual si pincha irá a la entrada misma) y un pequeño formulario con un campo para introducir la puntuación numérica entre 1 y 5, un botón (**save**) para aceptar esta puntuación y añadirla a la base de datos y otro botón (**back**) para volver a la página principal del usuario.

Si la operación de puntuación se realiza correctamente, el usuario será conducido a la pantalla siguiente (donde tendrá que pulsar sobre el enlace si desea volver a su página principal):

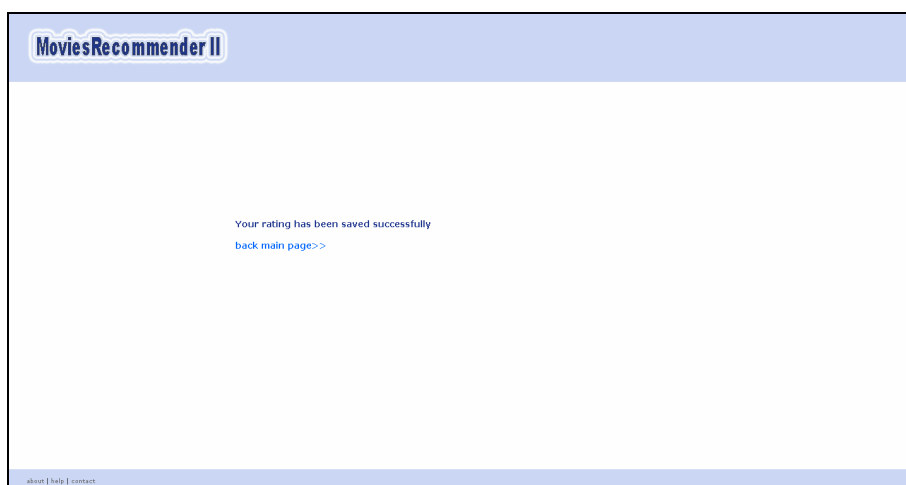


Figura 22. Éxito al realizar una puntuación

5. View all movies

Si el usuario pincha sobre este enlace se desplegará un menú con todas las películas disponibles en la base de datos:



Figura 23. Todas las películas de la base de datos

Esta opción se ofrece porque es muy posible que el usuario haya visto más películas de la lista además de las que ha ido alquilando (ya sea en el cine, en la televisión o alquilada previamente a la introducción del sistema de recomendación) y es muy posible que resulte interesante conocer sus puntuaciones sobre esas películas.

Además con esta opción se da la oportunidad al usuario de en un momento dado cambiar la puntuación otorgada a una película ya sea este cambio de postura debido a cualquier circunstancia. Las películas ya puntuadas se diferencian del resto al estar tachado su título por una línea roja horizontal:

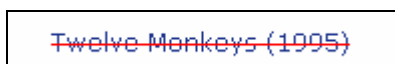


Figura 24. Película ya puntuada

La forma de puntuar es idéntica a la del apartado anterior: se pulsa sobre el título de la película y se accede a la página específica para puntuaciones operando en ella de la manera ya vista.

6. Make me a recommendation, please

Esta opción es la más importante dentro de la aplicación: cuando el usuario pinche sobre el enlace se le desplegará ante él un menú con las diez películas todavía no vistas ni alquiladas por el usuario que el sistema considera que van a ser más de su agrado ordenadas de mayor y menor y con un enlace hacia la entrada en el **IMDB** de cada de ellas para que el usuario pueda recabar toda la información que necesite:

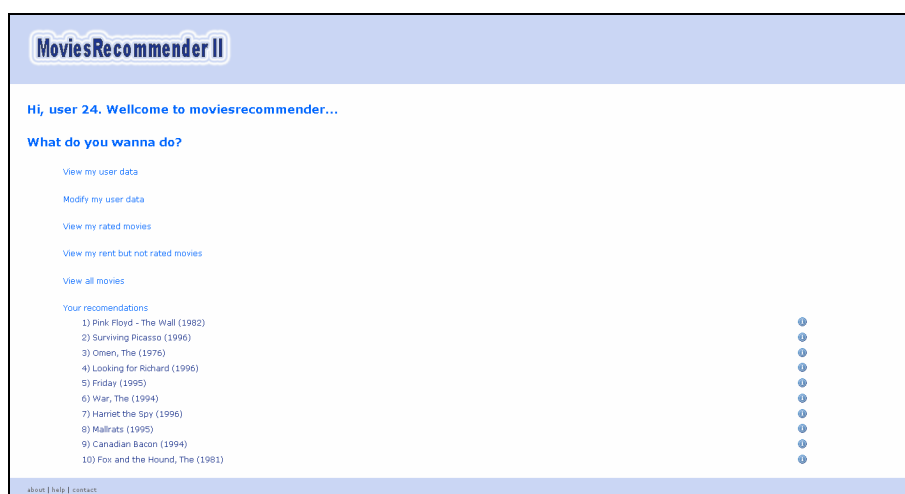


Figura 25. Películas recomendadas para el usuario dado

Para que el usuario obtenga una recomendación, es necesario que al menos haya puntuado 20 películas.

7. Logout

Cuando el usuario pulse sobre este enlace, terminará su sesión y volverá a la **página de inicio**.

Con lo visto hasta el momento en esta visita guiada el usuario ya debe haberse familiarizado con la funcionalidad principal de la aplicación pero todas las páginas de la aplicación integran una cabecera y un pie de página comunes que dotan de otras características a la aplicación completa y que sería conveniente que el usuario conociera.

La **cabecera** esta formada por un *banner* que a su vez es un enlace que lleva desde

cualquier parte de la aplicación hasta la **página de inicio** terminando la sesión en curso si la hubiera.



Figura 26. Banner de la cabecera

Por su parte, el **pie de página** nos muestra tres enlaces: uno a la página de contacto con el administrador de la aplicación (**contact**); otro a una página acerca de **MoviesRecommender II** (**about**) y el último a una página de ayuda (**help**):

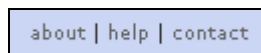


Figura 27. Pie de página

La **página de contacto** presenta un formulario para que el usuario (logueado o no) pueda ponerse en contacto con la administración de la aplicación:

A screenshot of a web page titled "MoviesRecommender II" in the header. The main content area is titled "contact" and contains a message: "If you have any doubts, suggestions or complaints, you can fill this contact form up. Webmasters will answer you as soon as possible, we promise it." Below the message are four input fields: "name *", "mail *", "subject", and "message *". The "message" field is a larger text area. A "send" button is located below the "message" field. At the bottom of the form, there is a note "* obligatory field" and a "back >>" link. The footer of the page contains the text "about | help | contact".

Figura 28. Formulario de contacto

Por su parte, la **página acerca de MoviesRecommender II** es una breve semblanza de la aplicación y de los autores de la misma.



Figura 29. Página acerca de MoviesRecommender II

Esta página se puede leer también en español.

Finalmente, la **página de ayuda** se encarga de dar un enlace en el que el usuario pueda descargar esta misma **guía de usuario** con la que aclarar cualquier duda que le surja:



Figura 30. Página de ayuda de moviesrecomender

Estas tres páginas accesibles desde el pie de página tienen en común un enlace que, si no hay ninguna sesión iniciada (es decir, ningún usuario logeado), nos devuelve a la **página de inicio** o, si el usuario está logeado, a la **página principal del usuario**.

Una vez llegado a este punto el usuario debe ser capaz de manejarse con soltura por **MoviesRecommender II** y disfrutar de sus características y funcionalidades.

BIBLIOGRAFÍA.

1. Bibliografía específica sobre Sistemas de Recomendación Colaborativos

- [1] Balabanovic, M. y Shoham, Y. (1997), "Content-based, collaborative recommendation" en *Communications of ACM* 40.
- [2] Breese, J. S., Heckerman, D. y Kadie, C. (1998), "Empirical analysis of predictive algorithms for collaborative filtering" en *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin, USA.
- [3] Cho, Y. H., Kim, J. K. Y Kim, S. H. (2002), "A personalized recommender system based on web usage mining and decision tree induction" en *Expert Systems with Applications*, vol. 23, pp. 233-342.
- [4] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. y Sartin, M. (1999), "Combining content-based and collaborative filters in an online newspaper" en *Proceedings of ACM SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, CA.
- [5] Dai, H. y Mobasher, B., "Integrated semantic knowledge with web usage mining and personalization" en *Web Mining: Applications and Techniques*, Ed. IRM Press
- [6] Guo, X. (2006), *Personalized Government Online Services with Recommendation Techniques*, tesis Phd, University Graduate School, University of Technology Sydney.
- [7] Herlocker, J., Konstan, J., Terveen, L. y Riedl, J. (2004), "Evaluating Collaborative Filtering Recommender Systems" en *ACM Vol. Transactions on Information Systems*, vol. 22, pp. 5-53.
- [8] Karypis, G. (2001), "Evaluation of item-based top-N recommendation algorithms" en *Proceedings of ACM 10th International Conference on Information and Knowledge Management*, Atlanta, Georgia.
- [9] Papagelis, M., Plexousakis, D., Rousadis, I. y Theorapoulos, E. (2004), "Qualitative Analysis of User-based and Item-based Prediction Algorithms for Recommendation Systems", *Proceedings of the 3rd Hellenic Data Management Symposium*.

- [10] Sarwar, B., Konstan, J., Riedl, J., Borchers, A., Herlocker, J. y Miller, B. (1998), "Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system" en *Proceedings of the 1998 ACM Conference on Computer Support Cooperative Work*, Seattle, Washington, USA.
- [11] Sarwar, B., Konstan, J., Terveen, L. y Riedl, J. (2000), "Analysis of recommendation algorithms for e-commerce" en *Proceedings of the 2nd ACM Conference on Electronic Commerce*, Minneapolis, Minnesota, USA.
- [12] Sarwar, B., Konstan, J., Terveen, L. y Riedl, J. (2001), "Item-Based Collaborative Filtering Recommendation" en *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, China.
- [13] Shardanan, U. y Maes, P. (1995), "Social information filtering: algorithms for automating 'word of mouth'", en *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*.
- [14] RICH, E. (1979): Building And Exploiting User Models
- [15] B. Krulwich. 1997. Lifestyle Finder: Intelligent User Profiling Using Large-Scale DemographicData. *AI Magazine*, 18(2): 37–45.
- [16] Pazzani, M. J. (1999). A Framework for Collaborative, content-based and Demographic Filtering. *Artificial Intelligence Review*, 13(5-6), 393-408.
- [17] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User Adapted Interaction*, 12, 331-370.
- [18] Konstan, J. A., Riedl, J., Borchers, A. y Herlocker, J. L. (1998). Recommender Systems: A GroupLens perspective. En *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-00-04)* (p'ags. 60-64). Menlo Park, CA, Estados Unidos.

2. Bibliografía general para la realización del proyecto

- [19] Dawson, C. W. y Martin, G. (2002), *El Proyecto Fin de Carrera en Ingeniería Informática: Una guía para el Estudiante*, Prentice Hall, Madrid.
- [20] Gutiérrez, A. y Bravo, G. (2005), *PHP5 a través de ejemplos*, Editorial Ra-Ma
- [21] Krug, S. (2000), *No me hagas pensar. Una aproximación a la usabilidad en la Web*, Prentice Hall, Madrid.
- [22] Manchón, E. (2002), *¿Qué es usabilidad?*, disponible en Internet (http://www.ainda.info/que_es_usabilidad.htm).
- [23] García Gómez, J. C. y Saorín Pérez, T. (2006), *Usabilidad para principiantes*, disponible en Internet (<http://usalo.es/?p=117>).
- [24] *scriptaculous wiki*, disponible en Internet (<http://wiki.script.aculo.us/scriptaculous>).
- [25] GroupLens Research: sitio web en el que está disponible la base de datos movieranks utilizada (<http://www.grouplens.org>).