



Universidad de Jaén

Escuela Politécnica Superior de Jaén

Departamento de Informática

Dr. D. Carlos Porcel Gallego y Dr. D. Francisco de Asís Conde Rodríguez, tutores del Proyecto Fin de Carrera titulado: Incorporación a un sistema de recomendaciones de una red de confianza, que presenta D^a. Rocío Verónica Quesada Calahorro, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Julio de 2012

El alumno

Los Tutores

D^a. Rocío Verónica Quesada Calahorro

Dr. D. Carlos Porcel Gallego
Dr. D. Francisco de Asís Conde Rodríguez

AGRADECIMIENTOS

Mis padres, a los que les debo todo.

Mis abuelos, por su eterno cariño.

José Luis, por no dejarme desistir y estar siempre a mi lado.

Miguel Ángel por su paciencia.

Mis tutores por soportarme y ayudarme en todo este proceso.

M. Carmen y Juan por ser unos magníficos amigos.

Al resto de familia y amigos porque siempre están si los necesito.

ÍNDICE

CAPÍTULO 1	INTRODUCCIÓN	13
1.1.	INTRODUCCIÓN AL PROYECTO	14
1.2.	PROPÓSITO:.....	16
1.3.	OBJETIVOS:.....	16
1.4.	ESTRUCTURA Y PLANIFICACIÓN DEL PROYECTO	16
CAPÍTULO 2	SISTEMAS DE RECOMENDACIÓN	19
2.1.	SISTEMAS DE RECOMENDACIÓN	20
2.1.3.1.	REALIMENTACIÓN EN LOS SISTEMAS DE RECOMENDACIÓN.....	23
2.1.3.2.	DATOS REALES VS. DATOS SINTETIZADOS	24
2.1.3.3.	ANÁLISIS ONLINE VS. ANÁLISIS OFFLINE	24
2.2.	EJEMPLOS DE SISTEMAS DE RECOMENDACIÓN	27
2.3.	LOS SISTEMAS DE RECOMENDACIÓN COMO NEGOCIO	32
2.4.	SISTEMAS DE RECOMENDACIÓN COLABORATIVOS BASADOS EN CONFIANZA	33
CAPÍTULO 3	DESARROLLO DEL PROYECTO	43
3.1.	DESCRIPCIÓN DEL PROYECTO.....	44
3.2.	ESPECIFICACIÓN DE REQUERIMIENTOS	44
3.3.	RECURSOS HARDWARE Y SOFTWARE REQUERIDOS	47
3.4.	PRESUPUESTO DEL PROYECTO	48
3.5.	ANÁLISIS DEL SISTEMA.....	50
3.6.	DISEÑO DEL SISTEMA.....	67
3.7.	IMPLEMENTACIÓN.....	95
CAPÍTULO 4	CONCLUSIONES	105
CAPÍTULO 5	BIBLIOGRAFÍA	109

de una red de confianza

APÉNDICE A MANUAL DE INSTALACIÓN	113
A.1. INSTALACIÓN DE APPSERV	114
A.2. INSTALACIÓN DE LA MÁQUINA VIRTUAL.....	126
APÉNDICE B MANUAL DE USUARIO	129
APÉNDICE C CODIGO FUENTE	149

LISTA DE FIGURAS

2. Sistema de Recomendación

TABLA 2 - 1: VALORACIONES	34
TABLA 2 - 2: SIMILITUD DE VECINOS	35
FIGURA 2 - 1: ESQUEMA SISTEMA RECOMENDACIÓN	22
FIGURA 2 - 2: AMAZON	27
FIGURA 2 - 3: DAILYME.....	29
FIGURA 2 - 4: FILMAFFINITY	30
FIGURA 2 - 5: MOVIELENS	31
FIGURA 2 - 6: NETFLIX	32
FIGURA 2 - 7: BRAINSINS	33
FIGURA 2 - 8: PROPAGACIÓN DE CONFIANZA.....	37
FIGURA 2 - 9: PROPAGACIÓN Y AGREGACIÓN DE CONFIANZA	40
FIGURA 2 - 10: CÁLCULO DE CONFIANZA	41

3. Desarrollo del proyecto

TABLA 3 - 1: PRESUPUESTO 1	48
TABLA 3 - 2: PRESUPUESTO 2	49
TABLA 3 - 3: METÁFORAS	70
TABLA 3 - 4: CAJA NEGRA INICIAR SESIÓN	101
TABLA 3 - 5: CAJA NEGRA REGISTRAR USUARIO	101
TABLA 3 - 6: CAJA NEGRA BUSCAR PELÍCULA	102
FIGURA 3 - 1: DIAGRAMA FRONTERA.....	51
FIGURA 3 - 2: CASO DE USO REGISTRAR USUARIO.....	52
FIGURA 3 - 3: CASO DE USO AUTENTIFICAR USUARIO	53
FIGURA 3 - 4: CASO DE USO BUSCAR PELÍCULA	55
FIGURA 3 - 5: CASO DE USO LISTAR PELÍCULAS VALORADAS.....	58
FIGURA 3 - 6: CASO DE USO LISTAR USUARIOS EN LOS QUE SE CONFÍA.....	60
FIGURA 3 - 7: CASO DE USO RECIBIR RECOMENDACIÓN	62

de una red de confianza

FIGURA 3 - 8: MODELO DEL DOMINIO	64
FIGURA 3 - 9: MODELO DE DISEÑO	66
FIGURA 3 - 10: PANTALLA ACCESO.....	70
FIGURA 3 - 11: PANTALLA REGISTRO DE USUARIOS.....	71
FIGURA 3 - 12: PANTALLA PÁGINA PRINCIPAL	72
FIGURA 3 - 13: PANTALLA BÚSQUEDA POR INICIAL	73
FIGURA 3 - 14: PANTALLA BÚSQUEDA POR AÑO	74
FIGURA 3 - 15: PANTALLA VOTACIONES.....	74
FIGURA 3 - 16: PANTALLA RECOMENDACIONES	75
FIGURA 3 - 17: PANTALLA LISTAR PELÍCULA BUSCADA	75
FIGURA 3 - 18: PANTALLA VALORES DE CONFIANZA.....	76
FIGURA 3 - 19: PANTALLA ERROR AUTENTIFICACIÓN	76
FIGURA 3 - 20: PANTALLA ERROR DE REGISTRO	77
FIGURA 3 - 21: STORYBOARD ENTRADA AL SISTEMA.....	79
FIGURA 3 - 22: STORYBOARD REGISTRO EN EL SISTEMA	80
FIGURA 3 - 23: STORYBOARD BÚSQUEDA [A-Z]	81
FIGURA 3 - 24: STORYBOARD BÚSQUEDA POR AÑO	82
FIGURA 3 - 25: STORYBOARD MIS VOTACIONES	83
FIGURA 3 - 26: STORYBOARD RECOMENDACIONES.....	84
FIGURA 3 - 27: STORYBOARD VALORES DE CONFIANZA	85
FIGURA 3 - 28: STORYBOARD CERRAR SESIÓN	86
FIGURA 3 - 29: STORYBOARD HERRAMIENTA DE BÚSQUEDA.....	87
FIGURA 3 - 30: STORYBOARD EMITIR Y MODIFICAR VALORACIÓN	88
FIGURA 3 - 31: STORYBOARD VALOR DE CONFIANZA	89
FIGURA 3 - 32: CARDINALIDAD EN DIAGRAMA ENTIDAD-RELACIÓN	90
FIGURA 3 - 33: DIAGRAMA ENTIDAD-RELACIÓN	91
FIGURA 3 - 34: DIAGRAMA ENTIDAD-RELACIÓN MODIFICADO	91
FIGURA 3 - 35: TABLA USUARIOS	93
FIGURA 3 - 36: TABLA PELÍCULAS.....	93
FIGURA 3 - 37: TABLA VALORACIONES.....	94
FIGURA 3 - 38: TABLA CONFIANZA	95
FIGURA 3 - 39: ARQUITECTURA CLIENTE/SERVIDOR.....	96
FIGURA 3 - 40: MULTIPLATAFORMA	98

A. Manual de instalación

FIGURA A.1 - 1: DESCARGA APPSERV	114
FIGURA A.1 - 2: DESCARGA DE LA APLICACIÓN.....	115
FIGURA A.1 - 3: INICIO INSTALACIÓN	116
FIGURA A.1 - 4: ACUERDO DE LICENCIA	117
FIGURA A.1 - 5: LUGAR DE INSTALACIÓN	118
FIGURA A.1 - 6: ELECCIÓN DE COMPONENTES	119
FIGURA A.1 - 7: DATOS DEL SERVIDOR.....	120
FIGURA A.1 - 8: CONFIGURACIÓN MYSQL.....	121
FIGURA A.1 - 9: FINALIZACIÓN DE LA INSTALACIÓN	122
FIGURA A.1 - 10: TERMINAL	123
FIGURA A.1 - 11: INICIO DE LA BASE DE DATOS DESDE EL NAVEGADOR	123
FIGURA A.1 - 12: PATALLA INICIO DE LA BASE DE DATOS	124
FIGURA A.1 - 13: IMPORTAR LA BASE DE DATOS	125
FIGURA A.2 - 1: DESCARGA DE LA MÁQUINA VIRTUAL JAVA	126
FIGURA A.2 - 2: INSTALACIÓN DE JAVA	127
FIGURA A.2 - 3: DESCARGA DE NETBEANS	127
FIGURA A.2 - 4: AGREGACIÓN DEL CONECTOR MYSQL	128

B. Manual de usuario

FIGURA B - 1: PORTADA	130
FIGURA B - 2: PÁGINA PRINCIPAL.....	131
FIGURA B - 3: ERROR DE AUTENTIFICACIÓN	131
FIGURA B - 4: REGISTRO DE USUARIO	132
FIGURA B - 5: ERROR DE CREACIÓN DE USUARIO	133
FIGURA B - 6: ÉXITO EN LA CREACIÓN DE USUARIO.....	134
FIGURA B - 7: CORREO DE CONFIRMACIÓN	134
FIGURA B - 8: BÚSQUEDA POR TÍTULO	135
FIGURA B - 9: RESULTADO DE LA BÚSQUEDA POR TÍTULO	136
FIGURA B - 10: ERROR EN LA BÚSQUEDA POR TÍTULO	137

de una red de confianza

FIGURA B - 11: BÚSQUEDA POR INICIAL..... 138

FIGURA B - 12: RESULTADO DE LA BÚSQUEDA POR INICIAL 139

FIGURA B - 13: PAGINACIÓN 139

FIGURA B - 14: BÚSQUEDA POR AÑO..... 140

FIGURA B - 15: RESULTADO DE LA BÚSQUEDA POR AÑO 140

FIGURA B - 16: ERROR 1 EN LA BÚSQUEDA POR AÑO..... 141

FIGURA B - 17: ERROR 2 EN LA BÚSQUEDA POR AÑO..... 142

FIGURA B - 18: MIS VOTACIONES..... 143

FIGURA B - 19: ERROR EN MIS VOTACIONES..... 144

FIGURA B - 20: VALORES DE CONFIANZA 145

FIGURA B - 21: RECOMENDACIONES..... 146

FIGURA B - 22: ERROR EN RECOMENDACIONES 147

FIGURA B - 23: CERRAR SESIÓN..... 148

CAPÍTULO 1

INTRODUCCIÓN

de una red de confianza

1.1. Introducción al proyecto

Con el imparable crecimiento de la World Wide Web y su evolución hacia la web 2.0 sustentada principalmente en la interoperabilidad, la colaboración y el diseño centrado en el usuario, en la que nos encontramos inmersos en nuestros días, estamos siendo testigos y a la vez partícipes de las infinitas posibilidades que ofrece este medio.

Es precisamente la web 2.0 y la interoperabilidad entre distintos dispositivos la que ha dado lugar a fenómenos como la web social, donde la forma de compartir información digital es más fácil que nunca, sea del tipo que sea. Contamos con multitud de dispositivos como por ejemplo smartphone y tablets, todos conectados a internet, y hacemos uso de distintos servicios, aplicaciones web interactuando con ellos.

Por otra parte, la evolución de los formatos digitales audiovisuales, junto con la facilidad para acceder, comprar o compartir la información digital de la que hablábamos, ha dado lugar, a una difusión cultural nunca vista hasta ahora. En este contexto de web social el “*boca a boca*” cobra especial importancia, ya que se destaca como la publicidad más efectiva.

Ámbitos como el del cine cuentan con infinidad de títulos y en ocasiones nos resulta difícil escoger qué película vamos a ver. También suele ocurrir con frecuencia que vamos al cine y salimos del mismo con la sensación agri dulce de haber perdido el tiempo porque la película no ha sido de nuestro agrado o no era lo que esperábamos. Bajo estas ideas nace nuestro proyecto. Contamos con la experiencia de una comunidad de usuarios que da su opinión o valoración sobre películas que ha visto y vamos a utilizar esta información para crear una aplicación web que albergará un sistema de recomendación de películas.

Existen distintas formas de definir un sistema de recomendación, una de ellas es la siguiente:

Por sistema de recomendación se entiende al sistema que utiliza las opiniones de los usuarios que forman una comunidad, para ayudar a otros usuarios (que también pertenecen a esa misma comunidad) a encontrar contenidos de su agrado entre un gran número de contenidos existentes.

Los sistemas de recomendación utilizan técnicas propias del Knowledge Discovery in Databases (KDD) [MAIMON, O.] para realizar recomendaciones personalizadas de información, productos y servicios durante una sesión online. Empresas como Lastfm dan fe de ello, ya que basan todo su

modelo de negocio además de en la publicidad, en acuerdos con tiendas online para enlazar directamente las recomendaciones con su servicio de venta [GALAN, S.M.]. De hecho, son innumerables las tiendas virtuales que nos ofrecen recomendaciones diariamente, por todo ello, es previsible que los sistemas de recomendación aumenten su presencia y utilidad en un futuro.

Hay distintos tipos de sistemas de recomendación:

- Basados en conocimiento, que parten de la información que tienen de los usuarios y productos para generar las recomendaciones que mejor se adapten a las necesidades del usuario.
- Basados en contenido, que se fundamentan en la similitud entre objetos para generar la predicción, partiendo de objetos que son del agrado del usuario.
- Colaborativos, que se basan en la similitud entre usuarios, recomendando objetos que son del agrado de usuarios con gustos similares a los del usuario en cuestión.
- Híbridos, que combinan algunos de los tipos anteriores.

Teniendo en cuenta esto y la cantidad de información que generan las aplicaciones web de hoy día, cada vez más orientadas a las redes sociales (perfiles de usuario, amigos...), se ha decidido que para este proyecto se va a desarrollar un sistema de recomendación de películas utilizando las técnicas de filtrado colaborativo, ya que son las más extendidas en el ámbito del comercio electrónico, pero insertándole el concepto de la confianza.

Como punto de partida para el desarrollo del sistema de recomendación colaborativo de películas será necesario realizar la elección de la mejor alternativa para la implementación del algoritmo de filtrado colaborativo adecuado a la colección de películas suministrada.

Una vez superadas las vicisitudes de la primera etapa del proyecto crearemos una base de datos que contenga la colección de películas y desarrollaremos un sitio web con acceso a dicha base de datos de forma paralela con la implementación del algoritmo de filtrado dando lugar al motor de recomendación de la aplicación. Una vez terminadas ambas tareas el último paso será la integración del resultado final en un servidor web que permita su uso y ofrezca la funcionalidad requerida.

Para la implementación del motor de recomendación y los algoritmos de filtrado utilizaremos el lenguaje orientado a objetos Java [LENG JAVA], dado que cumple con todos los requerimientos del proyecto. Para la generación de páginas web dinámicas haremos uso de las tecnologías JSP's y

de una red de confianza

Servlets [MURACH, J.] sobre un servidor Apache Tomcat [TOMCAT], así como de las tecnologías HTML, CSS y JavaScript, para su estructura y presentación. Por último, para gestionar las bases de datos de películas y clientes se utilizará el SGBD MySQL [MYSQL].

1.2. Propósito:

Implementación de un sistema web colaborativo basado en una red de confianza entre usuarios, para la recomendación de películas.

1.3. Objetivos:

- Búsqueda y revisión bibliográfica sobre sistemas de recomendación y algoritmos de filtrado colaborativo atendiendo especialmente a aquellos basados en la confianza entre usuarios.
- Estudio y obtención las bases de datos, para la puesta en funcionamiento de la web, bajo una licencia de libre distribución.
- Análisis y diseño de los distintos modelos de filtrado colaborativos y de confianza.
- Implementación del modelo de filtrado elegido.
- Desarrollar un sistema de recomendación colaborativo de películas, basado en la confianza aportada por los usuarios, con una interfaz web, atractiva y práctica.
- Verificación y evaluación del sistema obtenido.
- Redacción de la memoria.

1.4. Estructura y planificación del proyecto

Este apartado está dedicado a introducir brevemente los diversos capítulos que forman parte de este proyecto y el contenido de los mismos. Se ha comenzado en este capítulo con una introducción general al proyecto, lo que ha dado lugar a exponer el propósito que perseguimos con el mismo así como los objetivos asociados.

En el capítulo 2, empezamos dando una primera visión de los sistemas de recomendación en la que se incluye la definición de estos y la utilidad que tienen. Luego procederemos a mostrar la estructura que tienen y explicaremos los elementos que la componen. También analizaremos los datos que pueden ser utilizados en los sistemas de recomendación, pasando por los mecanismos de realimentación y una reflexión entre el análisis online y el offline. Una vez asentadas las bases estudiaremos los distintos sistemas de recomendación que existen juntos con los pros y contras que conllevan. Veremos distintos ejemplos de sistemas de recomendación existentes en Internet y acabaremos el capítulo profundizando en los sistemas de recomendación colaborativos y los basados en confianza.

El capítulo 3 está dedicado al proceso de desarrollo de nuestro proyecto, por lo que será el más importante que va a contener esta memoria. En él aplicaremos las diversas etapas de la Ingeniería del Software que vamos a necesitar para desarrollar nuestro sistema. Esto incluye una definición de requerimientos funcionales y no funcionales para el sistema; una fase de análisis en la que estudiar el modelo de casos de uso, el modelo del dominio y el modelo de diseño; una fase de diseño en la que veremos el diseño de la interfaz del sitio web y el diseño de los datos que va a usar el sistema; y una fase de implementación en la que mostraremos la arquitectura que usaremos para la aplicación, las herramientas usadas para su desarrollo y las pruebas de caja negra del mismo.

Terminaremos la memoria con el capítulo 4 dedicado a las conclusiones derivadas del desarrollo del proyecto y el capítulo 5 que contendrá toda la bibliografía necesitada durante el transcurso de éste.

Incluiremos también en la memoria unos apéndices en los que explicamos cómo poder instalar la aplicación y como usar la interfaz web de la misma. Y un último apéndice en el que incluimos código fuente de la aplicación que hemos desarrollado.

CAPÍTULO 2

SISTEMAS DE RECOMENDACIÓN

de una red de confianza

2.1. Sistemas de recomendación

La World Wide Web es una herramienta, muy útil para la comunicación, que ha crecido vertiginosamente desde su nacimiento. En un principio estaba destinada a ser una web estática cuya función era la de informar, pero se ha convertido en un medio abierto en el que la colaboración e interacción humana son posibles, ya sea escuchando o participando.

Debido a este dinamismo también surgen problemas, como son la enorme cantidad de información que se genera y se publica al día, lo que hace complicada su manipulación y organización. Como consecuencia de ello, es difícil, en ese entorno, encontrar la información que se necesita y que a su vez sea veraz.

Está claro, que la información no es útil si no se es capaz de encontrarla, aunque esté ahí, o si no se encuentra en el contexto idóneo. Ante esta situación, muchas empresas vieron el potencial que la red les podía ofrecer para hacer negocio si se adaptaban a los nuevos modelos e invertían capital en la evolución de la web.

Hay muchas empresas que surgieron en internet y que todavía siguen consolidadas como negocio, un ejemplo obvio son Amazon, con su tienda online, o Google que es un arquetipo de empresa tecnológica en el mundo de los Sistemas Inteligentes [SCHALKOFF, R.J.]. Pero ¿cuáles han sido los motivos que han impulsado a estas empresas hasta el lugar que hoy conservan? Si sólo se tiene en cuenta el aspecto tecnológico, ambas son empresas que han desarrollado su propia tecnología y gracias a ello han triunfado.

Como ya se ha comentado, Amazon destaca por su tienda de comercio electrónico, aunque también por vender parte de su tecnología a otros [AMAZON]. Y a pesar de que es una empresa puntera en su sector no siempre lo ha tenido fácil. Desde sus comienzos tuvo que luchar por la idea de ofrecer un catálogo de productos lo más extenso posible. Para ello tuvo que pensar en primer lugar en el comprador potencial, ya que una tienda de este estilo podría desorientarlo a la hora de realizar compras debido a su gran volumen de artículos disponibles.

La visión de Jeff Bezos, fundador y CEO de la compañía fue la siguiente: *“si tienes 3 millones de clientes en la Web, debes tener 3 millones de tiendas en la Web”*, pues para mantener una buena experiencia de usuario no sólo basta con que el sitio web sea usable, atractivo y goce de una correcta

arquitectura de la información, sino que éste debe adaptarse de la mejor manera posible al usuario final para atraer su atención.

Aquí es donde entran en juego los sistemas de recomendación que en una primera definición se podría decir que:

“Son sistemas basados en técnicas de filtrado de información con el objetivo de recomendar ítems de diversa índole, que previsiblemente serán del interés del usuario”.

Con esta idea en mente, Amazon se acabó convirtiendo en un referente dentro del comercio electrónico y de los sistemas de recomendación, gracias a su gran catálogo de productos y al desarrollo de su propia tecnología. Con ellos ofrecen a los clientes una experiencia personalizada en la que, dependiendo del comportamiento que tengan en la tienda, se les sugerirá productos que podrían ser de su interés.

Los sistemas de recomendación no están restringidos al ámbito del comercio electrónico, si no que en la situación que acontece, en la que se está dando una transición hacia la Web Semántica [SEMANTIC] y en la que se otorga un mayor significado a los contenidos a la hora de encontrar la información que esté mejor definida, es primordial adaptarlos a los usuarios para poder ofrecerles lo que buscan.

Por esta razón son tan importantes en ese aspecto, ya que su finalidad es la de proporcionar sugerencias útiles y relevantes para cada usuario de los sistemas en los que estén implementados.

Los sistemas de recomendación permitirán, previsiblemente, evitar la sobrecarga de información, no desde la perspectiva de la búsqueda de la información sino más bien desde la perspectiva del descubrimiento de la misma. Estos sistemas requieren mínima información sobre el usuario y sin embargo a partir del análisis de su comportamiento serán capaces de detectar artículos o contenidos que puedan resultarles interesantes, todo ello de manera invisible para el mismo [APRESW].

Típicamente los sistemas de recomendación comparan un perfil de usuario sobre algunas características de referencia intentando realizar una predicción acertada sobre los ítems que éste no ha considerado aún. Estas características de las que se hablan pueden tomarse de la propia información de los ítems, como ocurre en el caso del enfoque basado en contenido, o bien de una comunidad de usuarios en el caso de los sistemas basados en filtrado colaborativo.

de una red de confianza

A continuación se estudiarán, en función de estos parámetros, los distintos enfoques que se le pueden aplicar a un sistema de recomendación. También veremos la utilidad que tienen y los tipos de datos que utilizan, así como la realimentación que necesitan. Pondremos algunos ejemplos de sistemas de recomendación vigentes y mostraremos la utilidad que tienen en los negocios. Por último acabaremos el capítulo profundizando en el tipo de sistema de recomendación elegido para este proyecto.

2.1.1. Estructura de los sistemas de recomendación

En el siguiente esquema de funcionamiento de un sistema de recomendación (FIGURA 2 - 1) se pueden distinguir tres componentes clave: los usuarios, los recursos y las recomendaciones.

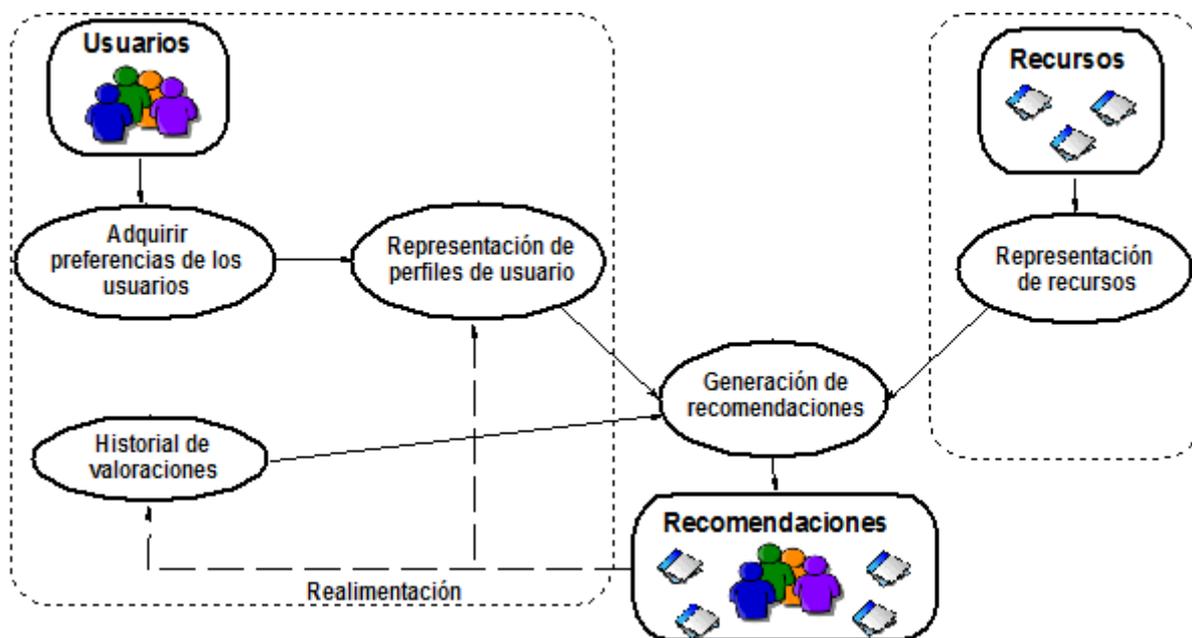


FIGURA 2 - 1: ESQUEMA SISTEMA RECOMENDACIÓN

Hay que empezar diciendo que el objetivo de cualquier sistema de recomendación es obtener una recomendación. Por ello las recomendaciones son el elemento clave del esquema. Para llevar a cabo el proceso de obtención de éstas hay que tener en cuenta:

- Los usuarios, adquiriendo sus preferencias y luego representando su perfil.
- Los recursos, cogiendo sus características más representativas.
- El historial de valoraciones.

Tanto el historial de valoraciones como los perfiles de usuarios deben ser realimentados con los nuevos datos que se vayan adquiriendo.

Con estos elementos, se procederá a aplicar el algoritmo de recomendación elegido, el cual dependerá del sistema que se quiera desarrollar. A su vez, los elementos serán considerados, en menor o mayor medida, dependiendo del sistema elegido. Por ejemplo los sistemas de recomendación basados en contenido sólo consideran los recursos valorados por los usuarios para llevar a cabo la recomendación.

2.1.2. Utilidad de los sistemas de recomendación

Los sistemas de recomendación hacen que las búsquedas sean más efectivas y que el tiempo dedicado a realizarlas sea menor, por ello se utilizan frecuentemente en aplicaciones de marketing personalizado en Internet. Todo sistema de recomendación debe realizar dos tareas para lograr alcanzar esos objetivos:

- Predecir: los sistemas de recomendación deben predecir los ítems en los que un usuario podría tener interés.
- Recomendar lo n-mejores ítems: los sistemas de recomendación calculan cuáles son los n-mejores ítems acordes a las preferencias del usuario.

2.1.3. Datos en los sistemas de recomendación

Cuando se desarrolla un sistema de recomendación hay que tomar ciertas decisiones, como el tipo de dato a utilizar, la forma en la que se van a analizar éstos o la forma en la que se va a recibir la nueva información.

2.1.3.1. Realimentación en los sistemas de recomendación

Los sistemas de recomendación deben ser capaces de mejorar la calidad de sus recomendaciones, gracias a la experiencia y a la nueva información recopilada a lo largo del tiempo de ejecución del mismo. Para conseguirlo se utilizan los mecanismos de realimentación entre usuarios y sistema. Estos pueden ser mecanismos de realimentación explícita o mecanismos de realimentación implícita.

de una red de confianza

2.1.3.1.1. Mecanismos de realimentación explícita

Son mecanismos que se fundamentan en las acciones del usuario. Para llevar a cabo la realimentación es necesario que el usuario indique los ítems del sistema que le interesan, mediante votaciones o cualquier otro tipo de medio. El inconveniente que plantean es que dependen directamente del usuario y su predisposición a la hora de facilitar la información requerida.

2.1.3.1.2. Mecanismos de realimentación implícita

Son mecanismos que proporcionan información, acerca de las preferencias de los usuarios, al sistema de recomendación, y lo hacen sin que los usuarios sean conscientes de ello. Para realizar la realimentación se usan distintas métricas como por ejemplo el número de consultas que se hacen sobre un ítem. Entre los inconvenientes destacan: que dependen mucho del contexto y que las hipótesis que se hacen sobre las preferencias de los usuarios no tienen por qué ser reales.

2.1.3.2. Datos reales vs. datos sintetizados

Es importante elegir el tipo de dato con el que va a trabajar el sistema de recomendación y por ello en este apartado se va a explicar la diferencia entre los datos reales y los sintetizados.

Los datos reales son aquellos recopilados de usuarios reales sobre objetos reales y para ello se utilizan métodos como las encuestas, por lo que su recopilación es costosa.

Los datos sintetizados son aquellos que se crean específicamente para el sistema de recomendación, no tienen por qué tener ninguna base real y se usan para probar la funcionalidad del sistema. Son más fáciles de obtener ya que evitamos tener que usar métodos de recopilación de información real. Se usan en las primeras fases de desarrollo de éste y luego son sustituidos por los reales una vez que el sistema haya recopilado la suficiente información.

2.1.3.3. Análisis online vs. análisis offline

Otra tarea fundamental es decidir si se va a trabajar sobre los datos de manera online u offline. En el análisis offline se emplea una técnica o algoritmo de filtrado para hacer predicciones sobre el conjunto de datos y los resultados de dichas predicciones son analizados mediante una o varias métricas de error. Este tipo de análisis presenta la ventaja de ser rápido y económico, pero tiene a la

vez dos inconvenientes importantes: el problema de la escasez de datos y el problema de obtener como único resultado la bondad de la predicción.

Por el contrario, el análisis online permite obtener más resultados, entre los que destacan la actuación de los usuarios participantes, la satisfacción o la participación de los mismos. El inconveniente es que es más lento y costoso que el análisis offline.

2.1.4. Clasificación de los sistemas de recomendación

Antes de estudiar más a fondo el tipo de sistema de recomendación utilizado hay que echar un vistazo general a los distintos tipos que existen. Los sistemas de recomendación pueden ser implementados utilizando diversas técnicas [CHO, Y.H.], por lo que su clasificación dependerá de su funcionamiento para calcular las predicciones. La clasificación más común es la expuesta a continuación:

2.1.4.1. Sistemas de recomendación basados en conocimiento

Son sistemas que hacen una reflexión sobre las preferencias y necesidades, para de esa forma deducir las recomendaciones [GUO, X.]. Lo que hacen es partir de la información que tienen de los usuarios y los productos y con ella generan las recomendaciones que mejor se adapten a las necesidades del usuario.

Estos sistemas se componen de tres fases: una de obtención de ejemplos que reflejen la necesidad del usuario, una segunda de búsqueda de productos que coincidan con ella y una tercera en la que el usuario añade y/o modifica la información aportada para que se filtren los resultados y se adapten mejor a sus expectativas.

Lo mejor de estos sistemas es que sólo necesitan una información global de los productos y los usuarios que estudian, y pueden llegar a ser muy ventajosos al combinarse con otros sistemas de recomendación.

El mayor problema que presentan es que la información que necesitan para llevar a cabo las recomendaciones la tienen que obtener explícitamente de los usuarios. Otro problema que presentan es que, muchas veces para refinar las recomendaciones, los usuarios deben refinar a su vez su perfil para que éstas puedan ser más fiables.

de una red de confianza

2.1.4.2. Sistemas de recomendación colaborativos

Los sistemas de recomendación colaborativos se basan en la similitud entre usuarios. Para calcular las recomendaciones de un usuario buscan productos que son del agrado de usuarios que tienen intereses similares a los de éste. Para llevar a cabo ese proceso ha de usarse un algoritmo de filtrado colaborativo que puede estar basado en memoria o en ítem.

Lo mejor de estos sistemas es que no tienen que preocuparse de la representación interna de los ítems a la hora de hacer una recomendación. Aunque, al estar tan extendidos en el mercado, también se le han descubierto una serie de problemas como son: la escalabilidad, la escasez y el problema de la agregación de ítems [CLAYPOOL, M.].

Una variante de este tipo de sistema es el que se va a utilizar en este proyecto y por esa razón se va a llevar a cabo un estudio más detallado del mismo en el apartado 2.4.

2.1.4.3. Sistemas de recomendación basados en contenido

Este tipo de sistemas está basado en un perfil que se crea gracias al análisis del contenido de los ítems que el usuario ha evaluado con anterioridad. Con este perfil se podrán llevar a cabo las recomendaciones que sólo estarán fundadas en las características de los ítems [BALABANOVIC, M.]. Para ello comparan el perfil del usuario con las características extraídas de los ítems y de esa manera predecir las preferencias que tendrán los usuarios sobre esos ítems.

Lo que se quiere conseguir es recomendar a los usuarios ítems que son similares en contenido a ítems que, previamente, ya sabemos que son del agrado de éste porque forman parte de su perfil.

Los sistemas de recomendación basados en contenido eran los más utilizados hasta que los colaborativos le arrebataron el puesto y para su funcionamiento dependen del tipo de información que se utilice en los perfiles de usuario, ya que primero analizan los datos de los distintos ítems valorados y después predicen cuál podría satisfacer al usuario.

El problema de estos sistemas es que sólo conocen parte de la información del producto, por lo que pueden no recomendar productos que hubieran sido del interés del usuario. Otro problema es la sobre-especialización, debido a que ignoran datos útiles como los gustos de un usuario.

2.1.4.4. Sistemas de recomendación híbridos

Este tipo de sistemas nacieron para intentar solucionar los problemas que los anteriores expuestos tienen. Lo que hacen es mezclar dos o más tipos diferentes de sistemas de recomendación para así reducir los inconvenientes. Para ello cogen los puntos fuertes de los distintos tipos de sistema de recomendación que se van a usar y los unen para formar el sistema de recomendación híbrido [BURKE, R.].

Pueden estar compuestos por las diversas combinaciones formadas entre ellos, como puede ser los basados en conocimiento con los colaborativos, los basados en contenido con los colaborativos o los basados en contenido con los basados en conocimiento. Incluso se pueden combinar con connotaciones sociales, como es nuestro caso, en el que unificamos los colaborativos con la confianza que tienen los usuarios entre sí.

2.2. Ejemplos de sistemas de recomendación

Hoy día se pueden encontrar múltiples sistemas de recomendación en Internet y que abarcan distintos tipos de ámbitos. A continuación analizaremos brevemente algunos de los ejemplos que se han considerado más interesantes.

2.2.1. Amazon

The screenshot shows the Amazon.es homepage with a focus on Kindle products. The main banner features a Kindle e-reader with the text: 'Kindle El regalo perfecto para Sant Jordi. No regales 1 libro, regala 1.400. 99€'. Below this, there are several product recommendations under the heading 'Más productos a tener en cuenta'. These include: 'Kindle: dispositivo de lectura...', 'Kindle Touch: e-reader con wifi y...', 'Funda de cuero Amazon para Kindle...', 'Marware Eco-Vue - Funda para Kindle y...', and 'Marware Atlas - Funda para Kindle y...'. The page also includes promotional banners for Amazon Premium, a 10% discount on books, and a section for 'Los más vendidos' in electronics.

FIGURA 2 - 2: AMAZON

de una red de confianza

De obligada referencia, Amazon implementa a día de hoy uno de los sistemas de recomendación más populares. Utiliza un enfoque híbrido combinando las ventajas del enfoque basado en contenido y el basado en filtrado colaborativo.

Comprobamos como al realizar cualquier búsqueda el sistema recomienda ítems que fueron comprados por otros usuarios que también adquirieron el actual. Pero además, Amazon utiliza información sobre nuestro historial de búsqueda para sugerir ítems relacionados.

Al añadido éxito de su tienda virtual hay de sumarle también el emergente éxito de su filial Amazon Web Services destinada a proporcionar soluciones de cloud-computing a empresas [AMAZON].

2.2.2. DailyMe

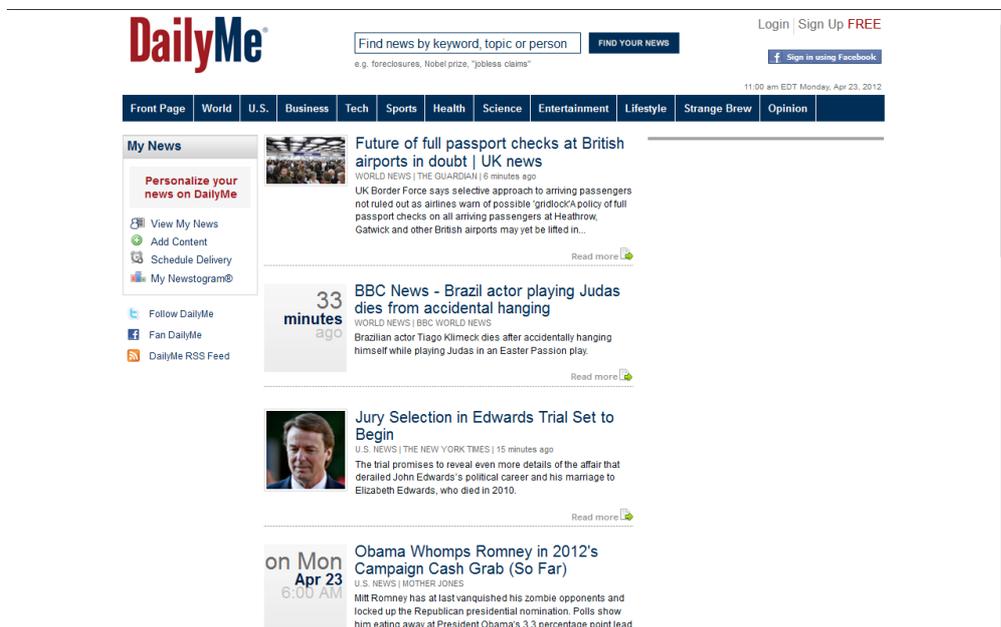


FIGURA 2 - 3: DAILYME

DailyMe es un sistema de recomendación de noticias que permite personalizar el diario con noticias que nos sean relevantes y de interés.

De forma similar a como hace Amazon este sitio web está basado en un enfoque híbrido combinando técnicas colaborativas y basadas en contenido. Para ello ha desarrollado Newstogram, una tecnología que funciona al margen de los sitios de noticias digitales, rastreando y analizando las noticias que los usuarios leen en un nivel granular para comprender los intereses específicos del visitante y sus comportamientos de consumo.

La tecnología no recolecta información personal de los usuarios. Los datos generados por este proceso permiten ofrecer recomendaciones personalizadas de información basadas en los intereses noticiosos de cada usuario y un mejor entendimiento de su audiencia sobre el contenido, comercio electrónico y publicidad.

de una red de confianza

2.2.3. Ejemplos de sistemas de recomendación de películas

Dado que nuestro proyecto consiste en desarrollar un sistema de recomendación de películas, dedicaremos este apartado a analizar algunos de ellos.

2.2.3.1. FilmAffinity

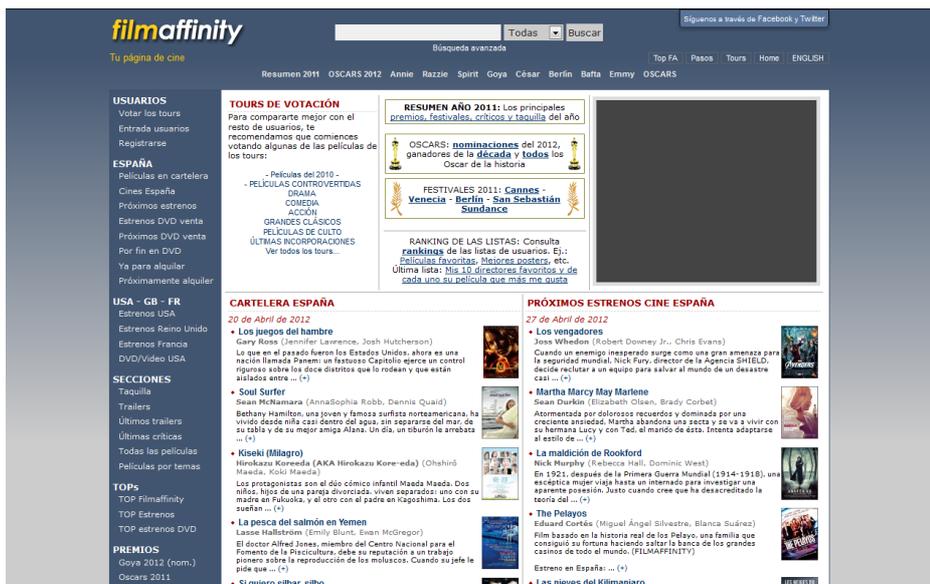


FIGURA 2 - 4: FILMAFFINITY

Se trata de un sistema de recomendación de películas muy popular al estar disponible tanto en inglés como en español.

El sitio web cuenta con una amplia base de datos muy útil para el público cinéfilo, incluyendo noticias sobre últimos estrenos así como festivales de cine celebrados. En cuanto a los usuarios registrados, éstos tienen un perfil a través del cual podrán realizar críticas y realizar votaciones sobre los títulos que hayan visto.

FilmAffinity implementa un sistema de recomendación colaborativo basado en modelo. El usuario debe votar un número suficiente de películas para que la calidad de la recomendación sea aceptable, y una vez construido su perfil, el sistema realiza la recomendación encontrando primero aquellos usuarios con gustos afines al mismo, o lo que ellos llaman “Almas Gemelas”.

Una vez encontrado el vecindario de usuarios afines el sistema cruza las valoraciones de las películas valoradas por éstos para ofrecerle la recomendación al usuario.

2.2.3.2. MovieLens



FIGURA 2 - 5: MOVIELENS

MovieLens es un sistema de recomendación de películas online, basado en un motor de filtrado colaborativo, desarrollado por el grupo investigador GroupLens Research (<http://www.grouplens.org>) de la Universidad de Minnesota.

Su principal función como sistema de recomendación colaborativo es la de agrupar a los usuarios de gustos similares, para ello recopila las puntuaciones de los usuarios sobre películas. Basándose en las puntuaciones de los usuarios del grupo al que pertenece un usuario activo, se intenta predecir para dicho usuario su valoración sobre películas aún no evaluadas por él.

La mayoría de datos de puntuaciones son privados, MovieLens mantiene como públicas dos muestras de 100.000 y un millón de puntuaciones, respectivamente.

de una red de confianza

2.2.3.3. NetFlix

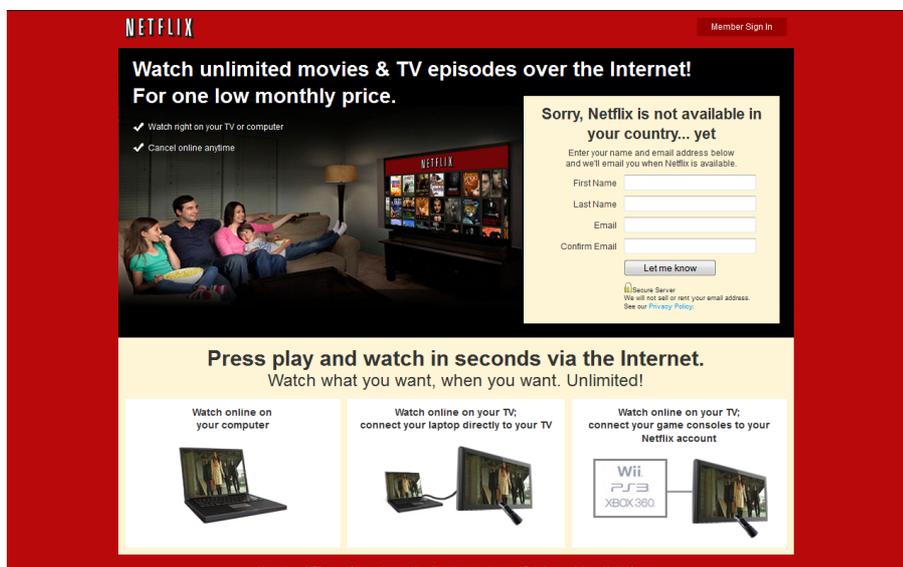


FIGURA 2 - 6: NETFLIX

Aunque no está presente en España, NetFlix es una empresa de alquiler de DVD's que además cuenta con un sistema de video bajo demanda que se ha convertido muy popular debido a la calidad de sus recomendaciones.

Pese a la quiebra de empresas similares del sector como BlockBuster, NetFlix no sólo se ha mantenido en el mercado sino que lo ha hecho de manera exitosa debido a su modelo de negocio y probablemente a su sistema de recomendación. En 2009 entregaron un premio de 1 millón de dólares tras promover un concurso para quien consiguiese mejorar un 10% la calidad de las recomendaciones de su sistema, lo cual se convirtió en una muy buena maniobra de mercado.

2.3. Los sistemas de recomendación como negocio

Dado el auge de este tipo de sistemas han surgido algunas empresas dedicadas a ofrecer servicios de recomendación a terceros bajo las premisas de que un sistema de recomendación bien implementado conlleva:

- Un aumento de ventas (productos). Un aumento entre un 20-30%.
- Un aumento de visualizaciones (contenidos). Un aumento similar al anterior.
- Un aumento de fidelidad (existe un trato personalizado). Un aumento del doble o triple.

- Una diferenciación con la competencia.



FIGURA 2 - 7: BRAINSINS

De este modo empresas como BrainSins (FIGURA 2-7) o ScarabResearch ya están integrando sus soluciones de recomendación en todo tipo de sitios web.

2.4. Sistemas de recomendación colaborativos basados en confianza

En este apartado se analizará en profundidad qué es un sistema de recomendación colaborativo y una variante de éste que es el sistema de recomendación colaborativo basado en confianza, ya que es el tipo de sistema que vamos a implantar en la aplicación [JUNG, K.Y.].

A continuación se estudiarán en primer lugar los sistemas de recomendación colaborativos, para dar una idea general de su funcionamiento, para después abordar los basados en confianza y así profundizar en las ideas concretas que aportan.

2.4.1. Sistemas de recomendación colaborativos

Antes los sistemas de recomendación colaborativos consistían en usar técnicas que se basaban en minería de datos [DAWSON, C.W.], aunque hoy en día se usa un paradigma de aprendizaje relajado

de una red de confianza

debido a que existe una gran interacción de usuarios con el sistema y éste debe actualizarse con frecuencia.

Los sistemas de recomendación colaborativos se basan en usar información conocida, acerca de la preferencias de usuarios, para realizar recomendaciones a otro usuario que la esté solicitando. Lo que hacen es identificar usuarios cuyas preferencias son similares a las de otros y de esa forma les recomiendan los elementos que hayan satisfecho a los segundos. Debido a esto es muy importante definir medidas de similitud entre preferencias. Para representar esta situación se puede recurrir a usar una matriz de usuarios e ítems, donde cada celda representa la valoración de un usuario con respecto a un ítem concreto. De esta manera lo que habría que hacer sería predecir valores para las celdas que estén vacías. En la TABLA 2 - 1 se puede ver un ejemplo de ello en el que el usuario1 y el usuario2 han valorado el ítem1 y el ítem2, y el usuario3 sólo ha valorado el ítem2 con lo que podría predecirse la valoración que tendría para el ítem1.

	Item1	Item2
Usuario1	Valoración1_1	Valoración1_2
Usuario2	Valoración2_1	Valoración2_2
Usuario3		Valoración3_2

TABLA 2 - 1: VALORACIONES

En estos sistemas los usuarios deben evaluar previamente algunos elementos para así poder ir creándoles un perfil de usuario. Además para cada usuario se creará un conjunto de usuarios, cuyas valoraciones tienen una gran semejanza con este, llamado vecinos cercanos (TABLA 2 - 2). De esa forma los ítems que no estén valorados se podrán predecir en base a este conjunto de vecinos cercanos.

Estos sistemas se caracterizan porque no se analizan los ítems valorados, sino que las recomendaciones se basan solamente en la similitud entre usuarios, con lo que pueden surgir distintos problemas. Uno de ellos es que el sistema debe tener un perfil lo suficientemente completo de un usuario antes de poder encontrarle un conjunto de vecinos cercanos. También surge el inconveniente de que si los gustos de un usuario son poco comunes, encontrarle un conjunto de vecinos cercanos será una tarea complicada. De esto podemos deducir que las recomendaciones dependen directamente de la variedad y el número de usuarios que haya en el sistema.

	Vecino1	Similitud1	Vecino2	Similitud2
Usuario1	IdVecino1	ValorSimilitud	IdVecino2	ValorSimilitud
Usuario2	IdVecino1	ValorSimilitud	IdVecino2	ValorSimilitud

TABLA 2 - 2: SIMILITUD DE VECINOS

Las técnicas más prometedoras en este aspecto son las de filtrado colaborativo, que utilizan una base de datos en función de las preferencias de los usuarios sobre los ítems. En estos casos, las opiniones de los usuarios sobre los productos suelen venir dadas a partir de un rating o puntuación sobre la satisfacción.

Para desarrollar un sistema de recomendación colaborativo contamos con dos enfoques [PAPAGELISA, M.]:

- Algoritmos de filtrado colaborativo basados en memoria o usuario: utilizan toda la base de datos de elementos y usuarios para generar predicciones. Mediante técnicas estadísticas se hallan los vecinos que corresponden a aquellos usuarios con gustos similares a los del usuario actual y una vez obtenido el vecindario se usan distintos algoritmos para combinar las preferencias entre estos y servir el top-N de recomendaciones. Sitios Web como www.filmaffinity.com nos indica aquellos usuarios con gustos similares a los nuestros (almas gemelas).
- Algoritmos de filtrado colaborativo basados en modelo o ítem: el principio es similar sólo que en este caso buscamos las similitudes entre ítems en lugar de buscar los usuarios más afines, es decir, los ítems son comparados por la apreciación compartida de los usuarios. Este tipo de sistemas construyen un modelo intermedio con la matriz de valoraciones y obtienen la predicción a partir de él. El proceso de construcción del modelo se realiza a partir de técnicas de machine learning como Redes Bayesianas, Clustering y enfoques basados en reglas obteniendo buenos resultados [UNGAR, L.H.]. Un ejemplo de este tipo de sistemas lo encontramos en www.amazon.com que ofrece una recomendación de ítems similares al visitar un producto.

El filtrado colaborativo presenta, no obstante, dos importantes desafíos: mejorar la escalabilidad de los algoritmos y mejorar la calidad de las recomendaciones de los usuarios. El cuello de botella de los algoritmos de filtrado colaborativo convencionales se encuentra en la búsqueda de vecinos potenciales en una población inmensa.

de una red de confianza

Los algoritmos basados en modelo (ítem-based) evitan este cuello de botella explorando las relaciones entre ítems primero. Las recomendaciones son computadas encontrando ítems que son similares a otros que han gustado al usuario. Esto es debido a que las relaciones entre ítems son relativamente estáticas, a diferencia de las relaciones entre usuarios, más variables. De esta manera se demuestra que los algoritmos basados en modelo son capaces de proporcionar idéntica calidad en la recomendación que los algoritmos basados en usuario pero con un coste computacional menor [GALAN, S.M.].

2.4.2. Sistemas de recomendación basados en confianza

Hoy en día es difícil llevar un control centralizado de entornos donde todo el mundo puede añadir contenidos, con lo que es complicado medir la calidad de los mismos. Por ello la Web 2.0 está teniendo tanta aceptación, ya que se permite la interacción, colaboración e intercambio de información entre usuarios.

Aplicando esta idea a los sistemas de recomendación obtenemos sistemas de recomendación basados en confianza o también conocidos como sociales, en los que las recomendaciones se calculan gracias a los perfiles de usuarios y las relaciones entre estos. Estos sistemas cada vez tienen más aceptación debido a que las redes sociales se propagan rápidamente y a que un usuario tiende a creer antes en recomendaciones que le llegan de personas en las que confía [SINHA, R.R.], que las que se le pueden calcular con un sistema online.

En el siguiente apartado hablaremos del concepto de confianza y se analizarán los distintos modelos de confianza.

2.4.2.1. Confianza

El concepto de confianza se está extendiendo cada vez más en la web, lo que resulta beneficioso ya que se nos permite acceder a esa información de una manera más sencilla. Esto contribuye a que puedan llevarse a cabo proyectos como este en el que se intenta mejorar los sistemas de recomendación colaborativos para así obtener sistemas fiables en entornos descentralizados [CHANG, H.]. La labor de calificar a los usuarios recae en ellos mismos al igual que se hacía con los ítems. Con lo que se obtiene una web de confianza, una vez que se hayan valorado entre sí los usuarios, que puede agregarse en una red social o red global de confianza.

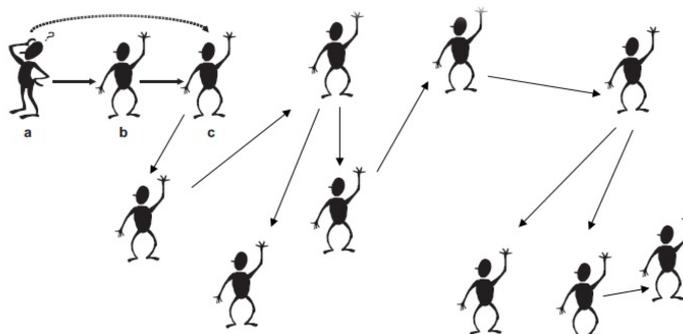


FIGURA 2 - 8: PROPAGACIÓN DE CONFIANZA

Como se puede ver en la FIGURA 2 - 8 los usuarios crean su red de confianza y a partir de ella se puede crear otra más completa estimando las conexiones que faltan. Una vez que un usuario ha declarado su confianza hacia otros se puede calcular la que podría tener hacia otros usuarios que no ha valorado mediante la propagación de la confianza. Un ejemplo de esto se puede observar de nuevo en la FIGURA 2 - 8. En la que el usuario A confía en B y a su vez el usuario B confía en otro C, con lo que se podría estimar que el usuario A puede confiar en el C. Esto también desencadena algunos problemas, debido a que la confianza es muy relativa, y es que aunque el usuario B esté confiando en el C no implica que el A deba confiar en él. Además la confianza tampoco implica simetría porque el usuario A puede confiar en el B y sin embargo el B no confiar el A. Debido a esto las métricas de confianza [FALCONE, R.] [GUHA, R.] tienen como objetivo el poder predecir dado unos usuarios, la confianza en usuarios desconocidos. En este proyecto se ha optado por seguir la implementada en [MASSA, P.].

2.4.2.2. Algoritmos basados en confianza

Desde hace bastante tiempo se está usando la confianza para ayudar en el ámbito de las ciencias de la computación. Podemos encontrar diversos modelos para ello en [ARTZ, D.]. Para este proyecto nos hemos quedado con los modelos de confianza basados en reputación debido a que son los que mejor se adaptan a los sistemas de recomendación.

Dentro del modelo elegido podemos encontrar dos tipos de sistemas [VICTOR, P.]:

- Sistemas probabilísticos: se basan en la probabilidad de poder confiar en un usuario. Si un usuario tiene un alto grado de confianza en otro es que tiene una alta probabilidad de confiar en él, aunque eso no significa que se pueda confiar en él.

de una red de confianza

- **Sistemas graduales:** tienen en cuenta la estimación de confianza que tiene un usuario en otro, por lo que si un usuario tiene un alto grado de confianza en otro significa que confía bastante en él.

En nuestro proyecto usaremos los sistemas graduales ya que son los que representan la confianza de manera más real. Hay distintos estudios que se centran en ello, nosotros seguiremos el que propone [MASSA, P]. A continuación explicaremos los diversos algoritmos que se pueden usar para añadir confianza a los sistemas de recomendación:

- **Algoritmo Always5.** Este es un algoritmo muy sencillo en el que la predicción que tendría un usuario respecto a un ítem siempre sería 5

$$prediction_{always5}(u, i) = 5$$

Apenas tiene relevancia para los sistemas de recomendación debido a que no permite hacer diferenciación entre los elementos, pero es un punto de partida para introducirnos en la confianza.

- **Algoritmo UserMean.** Lo que hace es coger la media de las valoraciones de un usuario

$$prediction_{userMean}(u, i) = \frac{\sum_{j=1}^n(r_{u,j})}{n}$$

donde n es el número de ítems i valorados por el usuario u. Para que este algoritmo obtenga resultados medianamente aceptables habría que filtrar las valoraciones antes del cálculo de la predicción.

- **Algoritmo TrustAll.** En este algoritmo la predicción que tendría un usuario para un ítem se calcula como una media no ponderada de todas las calificaciones dadas por todos los usuarios en los que confía el usuario en cuestión.

$$prediction_{trustAll}(a, i) = \bar{r}_a + \frac{\sum_{u=1}^n(r_{u,i} - \bar{r}_u)}{n}$$

donde a es el usuario activo, i el ítem a predecir y n el total de usuarios en los que ha confiado a .

TrustAll es un algoritmo que usa una técnica no personalizada, que es como asignar una similitud o confianza absoluta a todos los usuarios en los que confía el usuario activo.

- **Algoritmo MT1.** En este algoritmo la confianza ya no es absoluta, sino que oscila en un ranking, que suele estar entre $[0,1]$

$$prediction_{MT1}(a, i) = \bar{r}_a + \frac{\sum_{u=1}^n trust_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u=1}^n trust_{a,u}}$$

donde a es el usuario activo, i el ítem a predecir, n el total de usuarios en los que ha confiado a y $trust_{a,u}$ es el valor de confianza que ha depositado el usuario a en el u .

Para nuestro proyecto nos hemos descargado los datasets usados en Epinions, ya que están disponibles en la web para todo aquel que quiera hacer uso de ellos. En estos los datos de confianza usados son absolutos, es decir de confianza total o de desconfianza, y por esa razón hemos decidido implementar el algoritmo TrustAll que es el que mejor se adapta a ellos.

2.4.2.3. Cálculo de confianza

Los sistemas de recomendación basados en confianza preguntan a sus usuarios por la confianza que depositan en ellos, gracias a esto son capaces de hacer una red de relaciones que les ayuda a la hora de hacer las predicciones.

Surge el inconveniente de que un usuario no haya estimado la confianza que tiene respecto a usuarios útiles para poder llevar a cabo la predicción, ya que es posible que los usuarios en los que confía no hayan valorado los ítems necesarios. Cuando esto sucede puede ocurrir que existan diferentes opiniones acerca del usuario en cuestión o que no exista un usuario directamente conectado al usuario activo que conozca a ese usuario concreto. Para estimar la confianza que tendría el usuario activo en otro con el que no tiene una conexión directa, es decir que sólo lo conocen los conocidos del usuario activo, se usa la propagación y la agregación.

de una red de confianza

La propagación nos permite ir desde un usuario origen a uno destino por un solo camino y la agregación combina los diversos caminos que llegan del usuario origen al usuario destino. Puede verse claramente en el ejemplo de la FIGURA 2 - 9, en el que un usuario a quiere saber si se puede confiar en el usuario c pero como no lo conoce directamente puede tomar dos caminos: el que pasa por b o el que pasa por d. Para poder obtener el valor que le dará cada uno acerca del usuario c, se necesita que este valor se propague entre los usuarios hasta llegar al usuario a. Los valores de propagación serían $P_1(ab, bc)$ y $P_2(ad, de, ef, fc)$. Después de obtener estos valores el usuario a deberá combinarlos haciendo uso de la agregación.

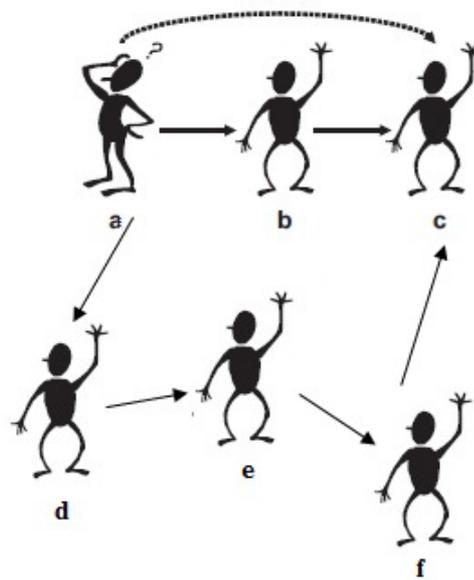


FIGURA 2 - 9: PROPAGACIÓN Y AGREGACIÓN DE CONFIANZA

Existen diferentes formas de calcular la confianza pero en este proyecto se ha optado por seguir la fórmula propuesta por [GOLBECK, J.A.]

$$trust_{is} = \frac{\sum_{j \in adj(j) | trust_{ij} \geq max} trust_{ij} trust_{js}}{\sum_{j \in adj(j) | trust_{ij} \geq max} trust_{ij}}$$

donde $trust_{is}$ es la estimación de la confianza que tendría el usuario i respecto al usuario s, j son todos los predecesores que tiene el usuario s hasta llegar al usuario i y max es un valor fijo que para nuestro

proyecto hemos optado por que sea de 0.6. Para clarificar un poco esto vamos a ver un ejemplo FIGURA 2 - 10.

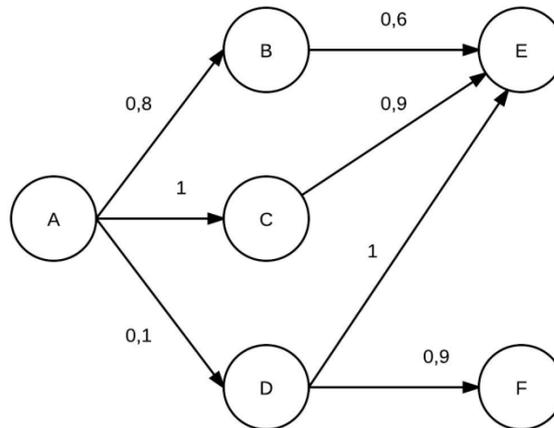


FIGURA 2 - 10: CÁLCULO DE CONFIANZA

Como se puede observar no hay un camino directo entre al usuario A y el usuario E, por lo que el cálculo de su confianza se haría de la siguiente manera:

$$trust_{AE} = \frac{0.8 * 0.6 + 1 * 0.9}{0.8 + 1}$$

Si observamos detenidamente se puede comprobar que el camino del usuario C no se ha incluido en el sumatorio. Esto es debido a que, como hemos comentado antes, el valor max es igual a 0.6 y como el valor de $trust_{AD}$ es menor, ya que es de 0.1, entonces no debe incluirse.

CAPÍTULO 3

DESARROLLO DEL PROYECTO

de una red de confianza

3.1. Descripción del proyecto

En este apartado vamos a proceder a especificar y explicar las distintas partes que componen el presente proyecto, y como tal, debe incluir el modelo de casos de uso y el modelo del dominio. Las partes que vamos a incluir en ellos son:

- La especificación de requerimientos, en la que narraremos los requerimientos funcionales y no funcionales que debe cumplir nuestra aplicación.
- Los diagramas de casos de uso, en los que veremos el diagrama frontera, que nos da una visión global de la funcionalidad del sistema y los distintos casos de uso junto con sus narrativas correspondientes
- El diagrama del dominio, en el que vamos a poder ver de una forma clara y concisa las clases conceptuales del dominio del problema.

3.2. Especificación de requerimientos

3.2.1. Requerimientos funcionales

Los requerimientos funcionales son aquellos que se encargan de describir el funcionamiento del sistema, son las funcionalidades que el usuario espera del sistema.

Este tipo de requerimientos son obtenidos mediante entrevistas y encuestas que se realizan sobre el cliente. Este proyecto que se va a desarrollar, no necesita de estas entrevistas, puesto que los requerimientos han sido identificados previamente, y se suponen que serían los que un usuario habitual podría esperar de un sistema de recomendación.

Las funcionalidades del sistema, sobre un usuario registrado, serán las siguientes:

- Permitir a un usuario no registrado darse de alta en el sistema.
- Autenticarse en el sistema.
- Buscar películas en el sistema.
- Poder recibir recomendaciones de objetos que aun no ha valorado y que podrían ser de su agrado.

- Puntuar aquellos objetos que haya probado.
- Modificar puntuaciones otorgadas previamente a objetos.
- Visualizar partes estructuradas del conjunto total de objetos.
- Mostrar objetos que el usuario ha valorado.

Después de haber definido cuáles son las funcionalidades que nuestro sistema de recomendación va a tener, procedemos a concretar de manera formal las funcionalidades anteriores, basándonos en que nuestro sistema de recomendación se realizará sobre la recomendación de películas a los usuarios del sistema:

- Permitir a un usuario no registrado registrarse en el sistema.
- Autenticarse en el sistema.
- Recibir recomendaciones de películas aún no puntuadas.
- Buscar películas en el sistema.
- Buscar alfabéticamente.
- Buscar por año.
- Buscar por título.
- Puntuar películas ya vistas.
- Modificar la puntuación de las películas.
- Modificar la puntuación de los valores de confianza.
- Listar el subconjunto de películas valoradas, del total de películas.
- Listar el conjunto de usuarios en los que se ha depositado confianza.

Este sistema es un prototipo de un sistema real, por lo tanto un sistema real debería de proporcionar una serie ampliada de funcionalidades:

- Visualizar datos personales.
- Modificar datos personales.
- Consulta de una ayuda sobre el funcionamiento del sistema.
- Recalcular las recomendaciones de manera instantánea, después de una actualización de la base de datos.

de una red de confianza

- Registro de nuevas películas.
- Posibilidad de realizar comentarios sobre películas.

3.2.2. Requerimientos no funcionales

Los requerimientos no funcionales son aquellos que restringen los requerimientos funcionales. Son tan importantes como los requerimientos funcionales y pueden incluso llegar a ser críticos para la aceptación del sistema. Estos requerimientos suelen especificar las propiedades de la interfaz, las propiedades del sistema software, como el rendimiento o la velocidad, y también las restricciones que imponga la organización tales como los plazos de entrega o la política empresarial.

- Atributos de calidad del sistema:
 - Disponibilidad:
 - El sistema debe estar disponible al 100% o cercano a esta disponibilidad durante todo el día.
 - Facilidad de uso e ingreso de información:
 - El sistema debe ser fácil de usar por parte de los usuarios.
 - El sistema no debe permitir el cierre de una operación hasta que todos sus procesos, subprocesos y tareas relacionados, hayan sido terminados y cerrados satisfactoriamente.
 - El ingreso de información al sistema de realizarse de manera que el tamaño de las páginas de registro o formularios de información sean adecuados.
 - El sistema debe presentar mensajes de error que permitan al usuario identificar el tipo de error.
 - Mantenibilidad:
 - Todo el sistema será completamente documentado, tanto en el código fuente como en los manuales de administración y de usuario.
 - El sistema debe ser capacitado para su fácil mantenimiento en el futuro, debido a los posibles errores que puedan presentarse en el futuro.
 - Seguridad:
 - El acceso al sistema estará restringido por el uso de contraseñas definidas por los usuarios. Sólo podrán acceder al sistema aquellos usuarios que estén registrados.

- Otros requerimientos no funcionales:
 - Arquitectura:
 - El sistema deberá operar de manera independiente del navegador que se esté utilizando.

3.3. Recursos hardware y software requeridos

3.3.1 Requerimientos hardware:

- Ordenador con los siguientes requisitos mínimos:
 - Cualquier microprocesador actual, ya que van a ser lo bastante rápidos como para ejecutar la aplicación.
 - Memoria RAM suficiente para la base de datos y las aplicaciones.
 - Disco duro de capacidad suficiente para almacenar las películas y las distintas aplicaciones necesarias.
 - Tarjeta gráfica básica.
 - Conexión a internet.

3.3.2 Requerimientos software:

- Cualquier sistema operativo.
- IDE cualquiera para desarrollar código en Java (en nuestro caso NetBeans).
- Visual Paradigm para realizar los diagramas.
- Sistema de Gestión de Base de Datos para trabajar con las películas (MySQL).
- Servidor Apache Tomcat y tecnología JSP's.

de una red de confianza

3.4. Presupuesto del proyecto

Para llevar a cabo este proyecto es necesario cubrir las necesidades de recursos expuestos en el apartado anterior. Partimos del supuesto de que se comienza sin nada y hay que adquirirlo todo.

Hemos confeccionado dos presupuestos básicos en los que se incluye todo lo necesario. La única diferencia que hay entre ellos es la de optar por software de libre distribución o comprarlo. Son los a continuación expuestos:

PRESUPUESTO 1

PRODUCTO	DESCRIPCIÓN	PRECIO
Servidor	Alojamiento de la página web y la base de datos Registro del dominio	10€/mes
PC	- Disco duro Seagate Barracuda 7200.12 500GB SATA3 -Tarjeta gráfica Asus GeForce EN210 Silent 1GB GDDR3 Low Profile -Procesador Intel Core i-5 2500k 3.3Ghz Box Socket 1155 -Memoria RAM Kingston 4GB DDR3 1333 Para Lenovo Memoria DDR3	599€
Total Costes Hardware		609€
Aplicaciones Software	Licencias y actualizaciones:	
	• Windows 7 Professional	131€
	• Visual Paradigm	79€
Total Costes Software		220€
Desarrollo Instalación Mantenimiento	Mano de obra	1500€
Total Costes Personal		1500€
TOTAL (IVA 18% incluido)		2748.22€

TABLA 3 - 1: PRESUPUESTO 1

PRESUPUESTO 2

PRODUCTO	DESCRIPCIÓN	PRECIO
Servidor	Alojamiento de la página web y la base de datos Registro del dominio	10€/mes
PC	- Disco duro Seagate Barracuda 7200.12 500GB SATA3 -Tarjeta gráfica Asus GeForce EN210 Silent 1GB GDDR3 Low Profile -Procesador Intel Core i-5 2500k 3.3Ghz Box Socket 1155 -Memoria RAM Kingston 4GB DDR3 1333 Para Lenovo Memoria DDR3	599€
Total Costes Hardware		609€
Aplicaciones Software de libre distribución	Licencias y actualizaciones	0€
Total Costes Software		0€
Desarrollo, instalación y mantenimiento	Mano de obra	1500€
TOTAL (IVA 18% incluido)		2488,62€

TABLA 3 - 2: PRESUPUESTO 2

de una red de confianza

3.5. Análisis del sistema

3.4.1 Modelo de casos de uso

Un caso de uso representa una situación o tarea de interacción de uno o más usuarios con la aplicación. Los casos de uso son tareas con significado, coherentes y relativamente independientes, que los actores realizan en su trabajo cotidiano.

Los casos de uso constan de los siguientes elementos:

- Nombre único y unívoco.
- Diagrama de caso de uso
- Actores participantes.
- Condiciones de entrada.
- Flujo de eventos.
- Condiciones de salida.
- Excepciones.

En nuestro sistema contamos con dos actores:

- El usuario registrado, es un usuario que ya está registrado en nuestro sistema y que puede hacer uso de todas las funcionalidades de éste como son pedir recomendaciones, buscar y valorar películas, ver las personas en las que confía, etc.
- El usuario no registrado, es un usuario aún sin registrar en el sistema y que sólo puede pedir ser registrado en éste.

Después de definir quienes van a ser los actores de sistema hay que crear los distintos casos de uso, y para ello se debe tener en cuenta que los requisitos funcionales que hemos especificado antes deben aparecer al menos en uno de los casos de uso.

A continuación vamos a proceder a mostrar el diagrama frontera, el cual describe la completa funcionalidad del sistema (FIGURA 3 - 1).

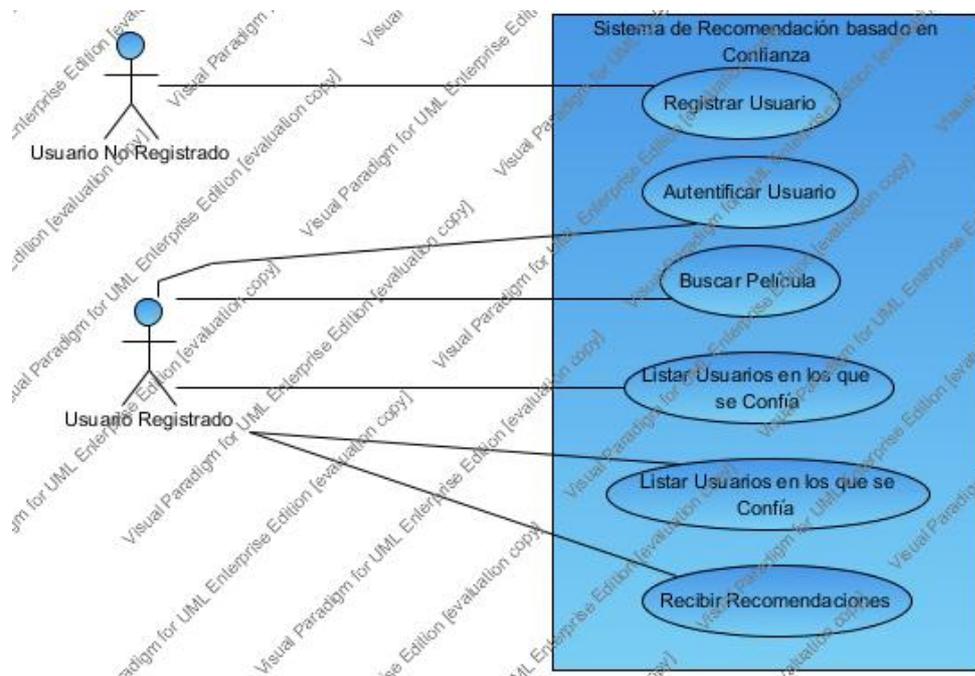


FIGURA 3 - 1: DIAGRAMA FRONTERA

Una vez definidos los casos de usos en función del estudio del sistema pasaremos a desglosar cada uno de ellos con sus respectivos diagramas y narrativas.

Tenemos que tener en cuenta que 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. Por ello a la hora de detallar un caso de uso se pueden emplear dos tipos de relaciones:

<<extend>>: es una relación que representa comportamientos excepcionales del caso de uso.

<<include>>: es una relación que representa la necesidad de otro caso de uso.

de una red de confianza

CASO DE USO: REGISTRAR USUARIO

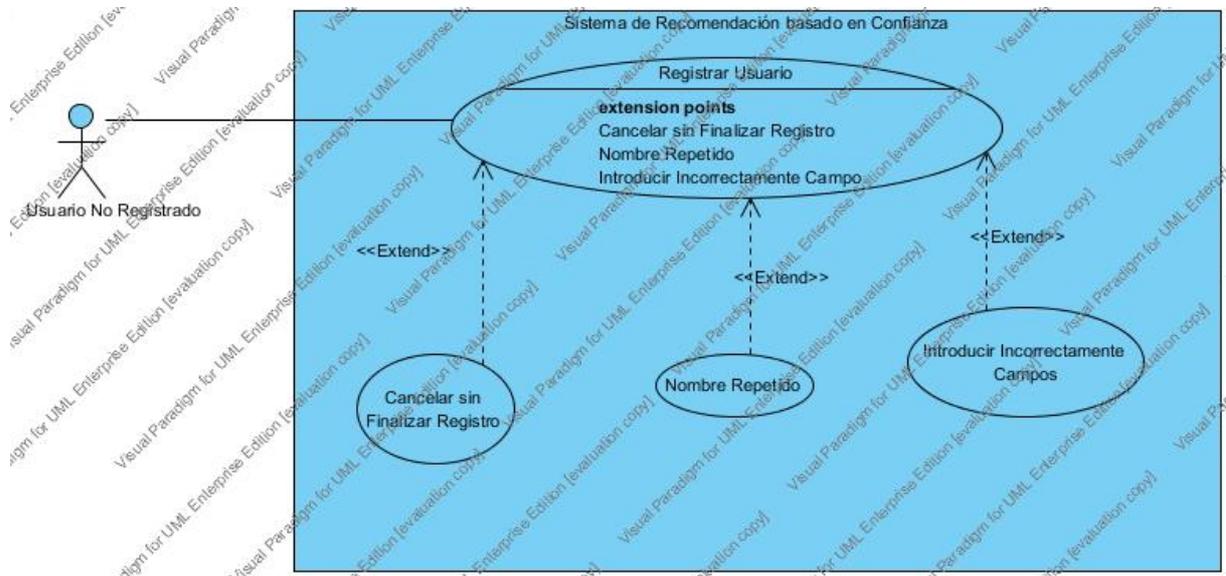


FIGURA 3 - 2: CASO DE USO REGISTRAR USUARIO

NARRATIVA

Nombre: Registrar usuario

Actor participante: Usuario no registrado

Condiciones de entrada (precondiciones):

- El usuario no debe estar registrado en el sistema.

Flujo de eventos (escenario principal):

1. El usuario no registrado pulsa la opción de registro.
2. Rellena los campos usuario y contraseña.
3. Presiona aceptar.
4. Recibe confirmación de registro.

Excepciones

- 3.a. El usuario presiona cancelar sin finalizar el registro.

- Terminar caso de uso.
- 4.a. El usuario no ha introducido correctamente los campos.
 - El sistema informa de que ocurrió un error en el registro.
 - Volver al paso 2.
- 4.b. Existe en el sistema otro usuario con el mismo nombre.
 - El sistema informa de que ocurrió un error en el registro.
 - Volver al paso 2.

Condiciones de salida (postcondiciones)

- El usuario es registrado en el sistema y recibe confirmación.

CASO DE USO: AUTENTIFICAR USUARIO

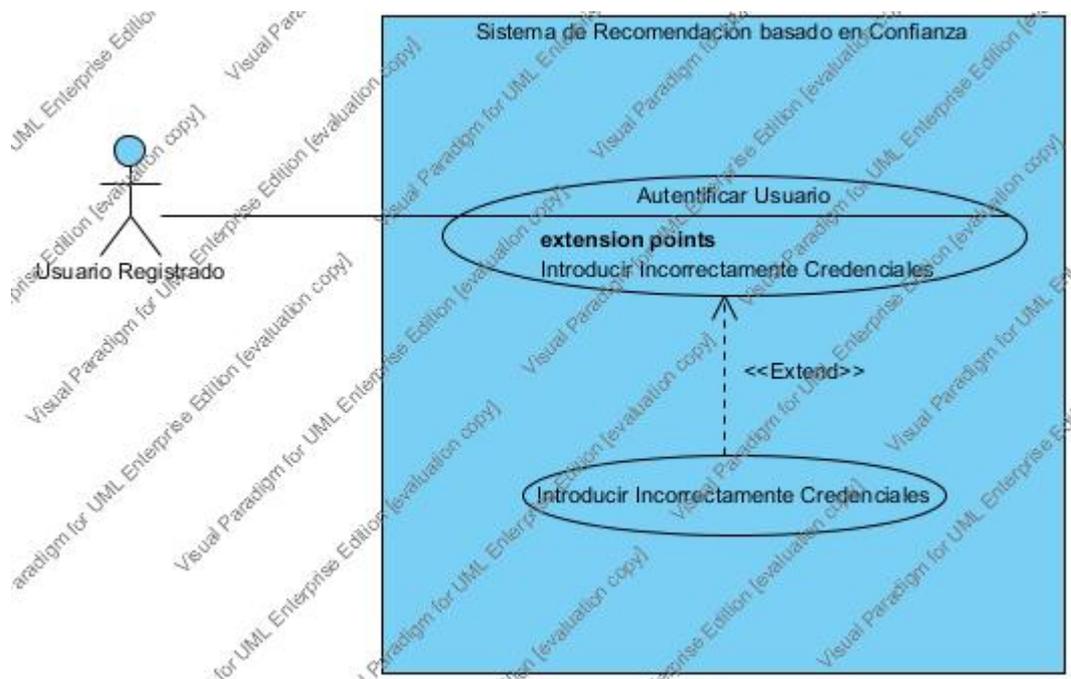


FIGURA 3 - 3: CASO DE USO AUTENTIFICAR USUARIO

de una red de confianza

NARRATIVA

Nombre: Autenticar usuario

Actor participante: Usuario registrado

Condiciones de entrada (precondiciones):

- El usuario debe estar registrado en el sistema.

Flujo de eventos (escenario principal):

1. Usuario introduce sus credenciales: usuario y contraseña.
2. Sistema autentifica correctamente al usuario.

Excepciones:

1.a. Usuario no introduce correctamente sus credenciales:

- Sistema no puede autentificar al usuario y muestra un error.
- Volver al paso 1.

Condiciones de salida (postcondiciones):

- Usuario accede al sistema.

CASO DE USO: BUSCAR PELÍCULA

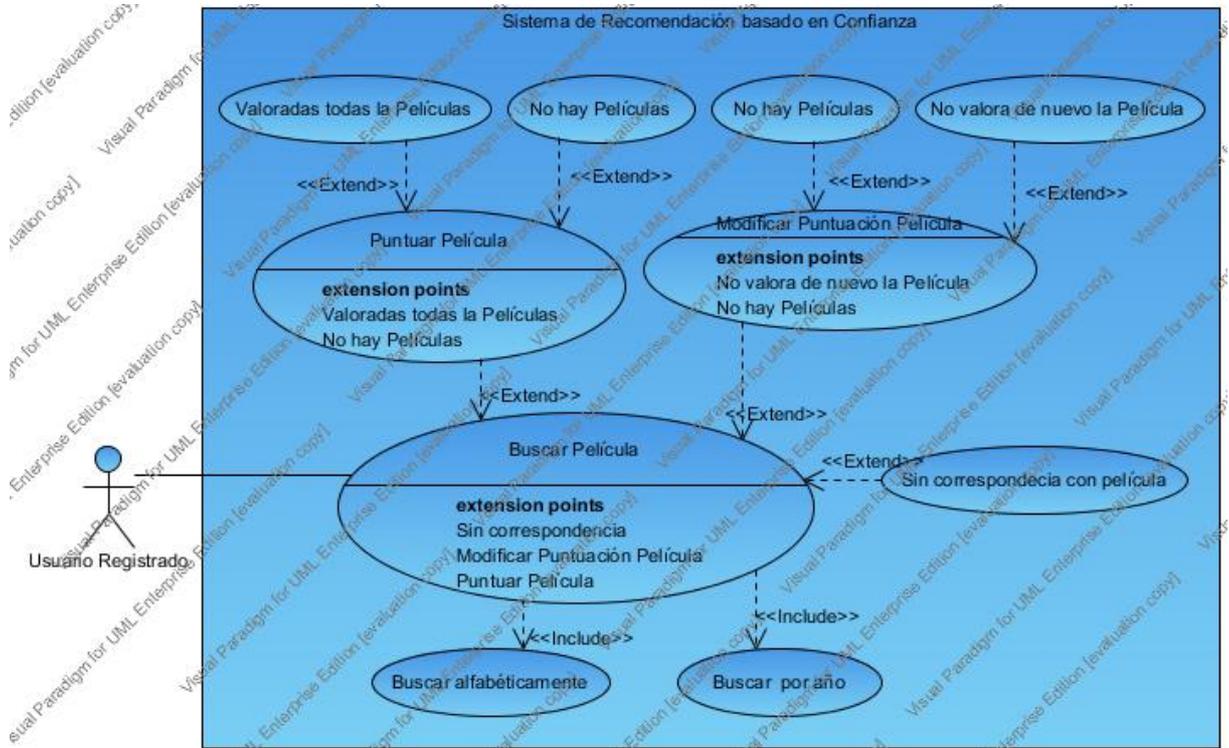


FIGURA 3 - 4: CASO DE USO BUSCAR PELÍCULA

NARRATIVA

Nombre: Buscar película

Actor participante: Usuario registrado

Condiciones de entrada (precondiciones):

- El usuario debe estar autenticado en el sistema.

Flujo de eventos (escenario principal):

1. El usuario introduce el nombre de la película que quiere buscar en el cuadro de texto y acciona Buscar.
2. El sistema muestra la película o películas que corresponden con la cadena que el usuario ha introducido.

de una red de confianza

Flujo Alternativo:

- 1.a. El usuario acciona Buscar Alfabéticamente y elige una inicial concreta.
- 1.b. El usuario acciona Buscar por Año y elige un año concreto.

Excepciones:

- 2.a. La cadena introducida por el usuario no se corresponde con ninguna película que haya en el sistema.
 - El sistema informa con un mensaje de error diciendo que no hay ninguna película que corresponda con la búsqueda realizada por el usuario.
- 2.b. El usuario pulsa buscar sin haber introducido ningún año.
 - El sistema informa con un mensaje de error diciendo que debe introducir un año.
- 2.c. El año introducido por el usuario no se corresponde con ninguna película.
 - El sistema informa con un mensaje de error diciendo que no se dispone de películas para ese año.

Condiciones de salida (postcondiciones):

- El sistema mostrará la película o películas buscadas.

NARRATIVA

Nombre: Puntuar película

Actor participante: Usuario registrado

Condiciones de entrada (precondiciones):

- El usuario debe estar autenticado en el sistema.

Flujo de eventos (escenario principal):

1. El usuario acciona la opción Buscar.
2. Sistema muestra la lista de películas que se corresponde con la búsqueda realizada.

3. El usuario valora la película o películas, de la lista que ha mostrado el sistema, que estime oportuno.
4. El sistema le informa que la valoración se ha realizado correctamente.

Excepciones:

- 2.a. El usuario ha valorado todas las películas.
 - El sistema mostrará un mensaje advirtiendo que todas las películas han sido valoradas. Termina el caso de uso.
- 2.b. No hay películas en el sistema.
 - El sistema muestra un mensaje de error advirtiendo que no hay películas actualmente. Termina el caso de uso.

Condiciones de salida (postcondiciones):

- La película o películas elegidas por el usuario quedan valoradas.

NARRATIVA

Nombre: Modificar puntuación película

Actor participante: Usuario registrado.

Condiciones de entrada (precondiciones):

- El usuario debe estar autenticado en el sistema.

Flujo de eventos (escenario principal):

1. El usuario acciona la opción Buscar.
2. El sistema le muestra una lista de películas que se corresponden con la búsqueda.
3. El usuario elige la película de la lista que desea modificar y vuelve a valorarla.
4. El sistema le informa que la valoración se ha realizado correctamente.

Excepciones:

de una red de confianza

2.a. No existen películas en el sistema para mostrar.

- El sistema muestra un mensaje de error advirtiendo que no hay películas actualmente. Termina el caso de uso.

4.a. El usuario no valora de nuevo la película.

- No se realiza ninguna modificación en la puntuación de la película. Termina el caso de uso.

Condiciones de salida (postcondiciones):

- La película elegida por el usuario va a sufrir una modificación en su valoración.

CASO DE USO: LISTAR PELÍCULAS VALORADAS

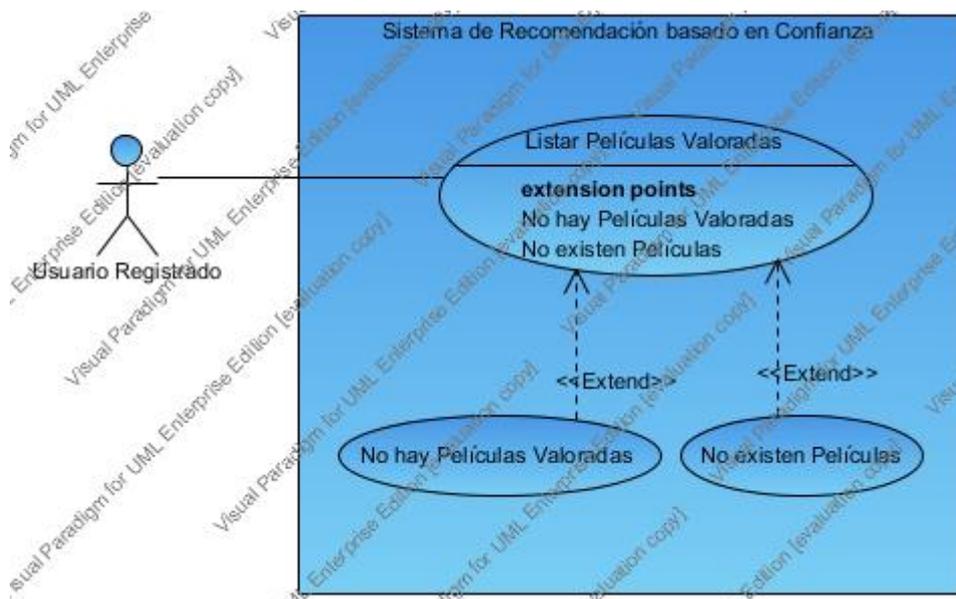


FIGURA 3 - 5: CASO DE USO LISTAR PELÍCULAS VALORADAS

NARRATIVA

Nombre: Listar películas valoradas

Actor participante: Usuario registrado

Condiciones de entrada (precondiciones):

- El usuario debe estar autenticado en el sistema.

Flujo de eventos (escenario principal):

1. El usuario acciona la opción Listar Películas Valoradas.
2. El sistema selecciona las películas que el usuario ha valorado.
3. Mostrar las películas que el sistema ha seleccionado.

Excepciones:

- 2.a. El usuario no ha votado ninguna película.
 - El sistema advierte con un mensaje que no se ha votado ninguna película. Termina el caso de uso
- 2.b. No existen películas en el sistema para mostrar.
 - El sistema muestra un mensaje de error advirtiendo que no hay películas actualmente. Termina el caso de uso.

Condiciones de salida (postcondiciones):

- El usuario obtendrá una lista, con todas las películas valoradas por él, que hay en el sistema.

de una red de confianza

CASO DE USO: LISTAR USUARIOS EN LOS QUE SE CONFÍA

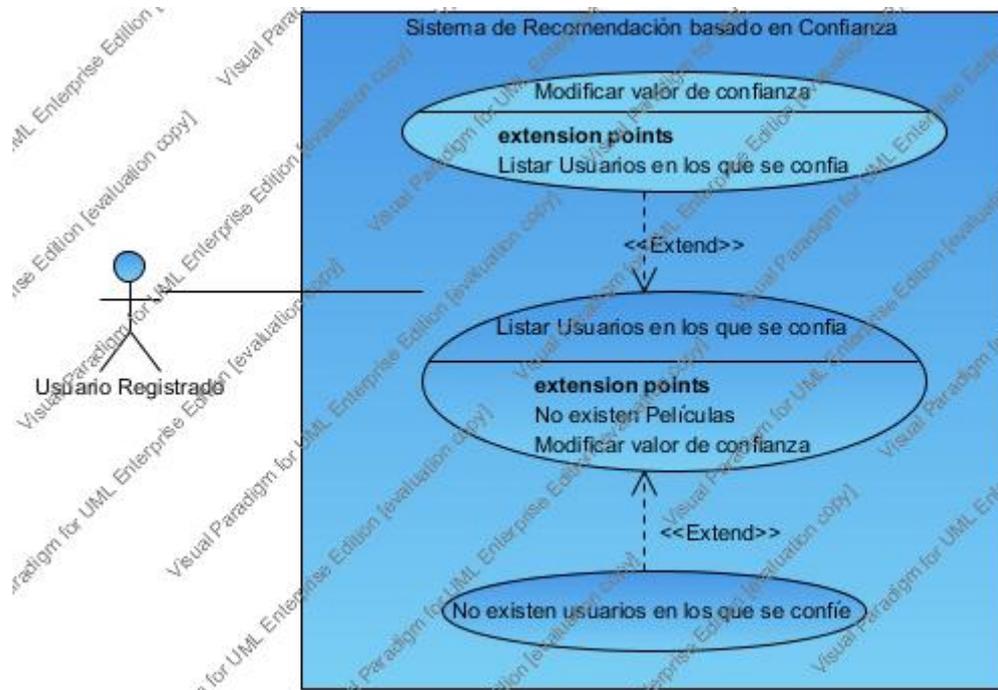


FIGURA 3 - 6: CASO DE USO LISTAR USUARIOS EN LOS QUE SE CONFÍA

NARRATIVA

Nombre: Listar usuarios en los que se confía

Actor participante: Usuario registrado

Condiciones de entrada (precondiciones):

- El usuario debe estar autenticado en el sistema.

Flujo de eventos (escenario principal):

1. El usuario acciona la opción Listar Valores de Confianza.
2. El sistema selecciona los usuarios que el usuario registrado ha valorado.
3. Mostrar los usuarios que el sistema ha seleccionado.

Excepciones:

- 2.a. El usuario no ha votado ningún usuario.

- El sistema advierte con un mensaje que no se ha votado ningún usuario. Termina el caso de uso

Condiciones de salida (postcondiciones):

- El usuario obtendrá una lista, con todos los usuarios valorados por él, que hay en el sistema.

NARRATIVA

Nombre: Modificar puntuación del valor de la confianza

Actor participante: Usuario registrado.

Condiciones de entrada (precondiciones):

- El usuario debe estar autenticado en el sistema.

Flujo de eventos (escenario principal):

1. El usuario acciona la opción Valores de Confianza.
2. El sistema le muestra una lista de los usuarios en los que confía el usuario.
3. El usuario elige el usuario de la lista que desea modificar y vuelve a valorarlo.
4. El sistema le informa de que la valoración se ha realizado correctamente.

Excepciones:

- 4.a. El usuario no valora de nuevo al usuario.
 - No se realiza ninguna modificación en la puntuación del usuario. Termina el caso de uso.

Condiciones de salida (postcondiciones):

- El usuario elegido por el usuario en cuestión va a sufrir una modificación en su valoración.

de una red de confianza

CASO DE USO: RECIBIR RECOMENDACIÓN

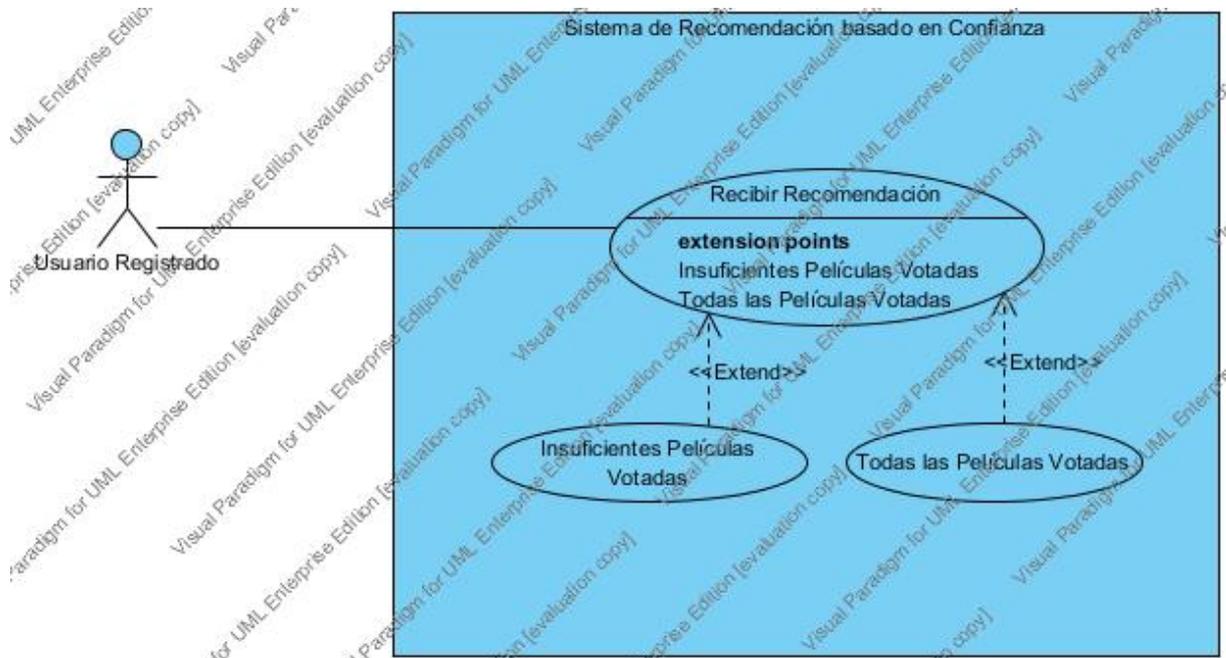


FIGURA 3 - 7: CASO DE USO RECIBIR RECOMENDACIÓN

NARRATIVA

Nombre: Recibir recomendación

Actor participante: Usuario registrado

Condiciones de entrada (precondiciones):

- El usuario debe estar autenticado en el sistema.
- El usuario debe haber votado n películas (siendo n un número que definiremos más adelante).

Flujo de eventos (esc. principal):

1. El usuario acciona la opción Recomendaciones.
2. Sistema muestra una lista de películas recomendadas para el usuario.

Excepciones:

- 2.a. El usuario no ha votado películas suficientes en el sistema.
 - Sistema informa que no hay suficientes películas recomendadas.
- 2.b. El usuario ha votado todas las películas
 - Sistema informa que no hay recomendaciones.

Condiciones de salida (postcondiciones):

- Una lista de películas recomendadas en función de las votadas por el usuario.

3.4.2 Modelo del dominio

Definición

El Modelo de Dominio (o Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software.

Una clase conceptual puede ser una idea o un objeto físico (símbolo, definición y extensión).

El modelo de dominio se representa en UML (Unified Modeling Language, lenguaje unificado de modelado) con un Diagrama de Clases en los que se muestra la siguiente información:

- Conceptos u objetos del dominio del problema: clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

El Modelo de Dominio no muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos.

Cualquiera que sea la solución de casos de uso que se haya elegido, los conceptos e ideas propias del dominio del problema son las mismas; un mismo modelo de dominio contempla cualquiera de las

de una red de confianza

soluciones analizadas. El modelo de dominio es global, es decir, se realiza para todos los casos de uso y no para uno en particular.

Características:

- No hay cronología.
- No se diferencia entre dentro y fuera del sistema.
- Es global, no por caso de uso.
- No es completo: esquemático, las asociaciones están resumidas.

DIAGRAMA DE MODELO DE DOMINIO

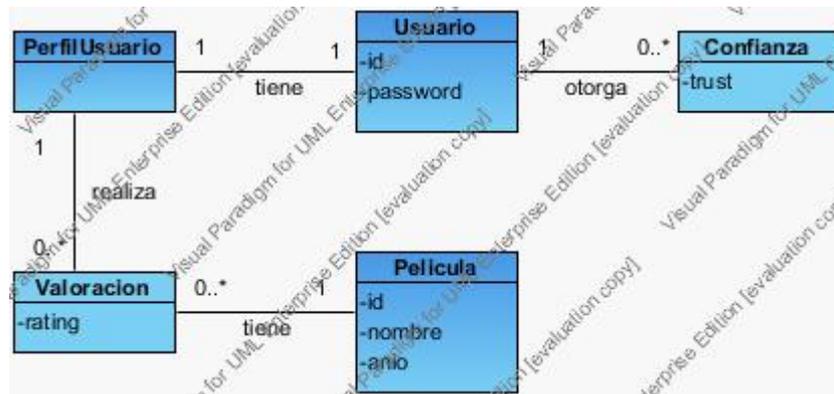


FIGURA 3 - 8: MODELO DEL DOMINIO

Hemos contemplado cinco entidades: Usuario, PerfilUsuario, Confianza, Valoración y Película. Un usuario va a tener un perfil de usuario que se compondrá de muchas valoraciones. A su vez un usuario podrá otorgar confianza a diversos usuarios y una película podrá tener varias valoraciones. En el caso inicial el usuario no tiene porqué haber otorgado confianza a nadie, una película no tiene porqué haber sido valorada o un perfil de usuario no tiene porqué tener valoraciones.

3.4.3 Modelo de diseño

En este apartado vamos a efectuar el diagrama de clases de diseño, por lo que empezaremos con la definición del mismo.

Un diagrama de clases de diseño representa las especificaciones de las clases e interfaces software en una aplicación. Entre la información general encontramos:

- Clases, asociaciones y atributos.
- Métodos.
- Información acerca del tipo de los atributos.
- Navegabilidad.
- Dependencias.

A diferencia de las clases conceptuales del Modelo de Dominio, las clases de diseño de los diagramas de clases muestran las definiciones de las clases software en lugar de los conceptos del mundo real.

En una aplicación útil los objetos colaboran e intercambian información, estableciendo distintos tipos de relaciones entre ellos. A nivel de diseño, podemos distinguir entre cinco tipos de relaciones básicas entre clases de objetos: dependencia, asociación, agregación, composición y herencia.

Las únicas que se utilizarán en este proyecto será la asociación. La asociación es la relación más importante y común. Refleja una relación entre dos clases independientes que se mantiene durante la vida de los objetos de dichas clases, o al menos, durante un tiempo prolongado. En notación UML se representa de la siguiente manera:

- El *nombre* de la relación.
- Las *cardinalidades* (o multiplicidades). Un asterisco significa varios.
- Dirección de la relación.

Si la cardinalidad es mayor que 1, será necesario utilizar un contenedor de datos, lo más común es utilizar vectores o listas. Los objetos normalmente son creados y destruidos de manera independiente el uno del otro.

3.6. Diseño del sistema

3.5.1 Diseño de la Interfaz

Una vez definida la fase de análisis, la siguiente fase del proceso de ingeniería del software es el diseño. Con el diseño se persigue desarrollar el diagrama de clases del sistema que se ha obtenido en el apartado anterior, cuya traducción a un lenguaje de programación permitirá la implantación del prototipo web usable.

El diseño de la interfaz va a consistir en la definición de:

- La guía de estilo, donde como su nombre dice se define el estilo por defecto que tendrá la interfaz.
- Las metáforas, que mediante la comparación con otro objeto o concepto, permiten al usuario comprender de una forma más intuitiva diversas tareas de la interfaz.
- El diseño de las pantallas, que nos da una idea de cómo será la interfaz.
- Los caminos de navegación o storyboards, que ayudan a comprender el desplazamiento entre pantallas.

3.5.1.1. Guía de Estilo

En este apartado definiremos el estilo de la interfaz web. Esto es útil ya que puede que exista más de un diseñador encargado de su realización y con esta guía se lograría mantener una coherencia interna. También es útil aunque sólo haya un diseñador ya que muchas veces, si el proyecto es muy grande, podemos perdernos en un determinado momento y la guía ayudaría a evitarlo.

A continuación pasamos a exponer las distintas reglas y normas que contendrá nuestra guía de estilo. Esta versión no es ni mucho menos la definitiva pero sí nos sirve como punto de partida para dar estilo a la estructura del sitio que se ha reflejado en los prototipos obtenidos.

Estilo por defecto

- Texto

Tamaño de fuente: 13px

Fuente: arial, helvetica, sans-serif

de una red de confianza

Color: negro

- Enlaces

Tamaño y fuente heredados de Texto

Color: amarillo

Efectos: subrayado, color púrpura en visitado

- Excepciones y mensajes de error

Tamaño y fuente heredados de Texto

Color: rojo

Combinación de colores

- Fondo de la página en color negro
- Color de fondo de capa por defecto blanco

Cabecera

- Título: 36px
- Información usuario y herramienta de búsqueda: heredados del estilo por defecto

Menú lateral

- Elemento activo:
 - Tamaño de la fuente: 26px
 - Color de texto: negro
 - Efecto de fuente: negrita
 - Color de fondo: blanco
- Elemento inactivo:
 - Tamaño de la fuente: 26px
 - Color de texto: blanco

Efecto de fuente: negrita

Color de fondo: negro

Footer o pie de página

- Tamaño de la fuente: 8px.
- Color: negro

Como hemos comentado, estas son las reglas de estilo básicas que vamos a utilizar para desarrollar la interfaz web de la aplicación. Una vez que dotemos de estructura al contenido identificando los distintos elementos con HTML procederemos a dotar de un estilo más rico al mismo gracias a las hojas de estilo en cascada (CSS), en un intento por dotar a la aplicación además de la funcionalidad requerida de una usabilidad acorde con los estándares de la web 2.0.[SEMANTIC]. Precisamente por este motivo utilizaremos técnicas basadas en el lenguaje interpretado Javascript, que nos permitirá implementar mejoras en la interfaz de usuario.

3.5.1.2. Metáforas

El uso de metáforas adecuadas en el diseño de una interfaz, facilita y acelera el aprendizaje del funcionamiento de una aplicación.

La clave se encuentra en las similitudes con otros mecanismos y procesos conocidos por el usuario, que aplica lo que ya conoce a los elementos de un dominio no tan familiar como por ejemplo una interfaz web, ayudando al mismo a entender rápidamente cómo interactuar con el producto.

Para que las metáforas sean utilizadas de forma eficiente deben ser consistentes y auto explicativas haciendo que el esfuerzo cognitivo a realizar por el usuario sea ínfimo.

Con el propósito de garantizar la usabilidad del sistema y ofrecer una correcta experiencia de usuario las metáforas que se van a utilizar en la interfaz del sistema de recomendación serán las convencionalmente utilizadas valoraciones mediante estrellas, estableciéndose la siguiente correspondencia:

de una red de confianza

Estrellas	Puntuación	Valor numérico
★★★★★	Película no valorada	0
☆★★★★	Película muy mala	1
☆☆★★★★	Película mala	2
☆☆☆★★★	Película regular	3
☆☆☆☆★★	Película buena	4
☆☆☆☆☆★	Película muy buena	5

TABLA 3 - 3: METÁFORAS

3.5.1.3. Prototipos

Para realizar el diseño preliminar de la interfaz de usuario es posible recurrir a distintas herramientas, siendo una de ellas la creación de prototipos de las pantallas, ya sea con dibujos a mano alzada, cortando y pegando trozos de papel con los que se representan componentes o bien recurriendo a medios digitales. En nuestro caso, hemos realizado el diseño con dibujos a mano alzada.

Además decir que la presentación de las pantallas que mostramos a continuación son la forma en la que los usuarios van a percibir el contenido.

PANTALLA: ACCESO

El usuario deberá introducir sus credenciales para acceder a las principales funcionalidades de la aplicación Web. (FIGURA 3 - 10).

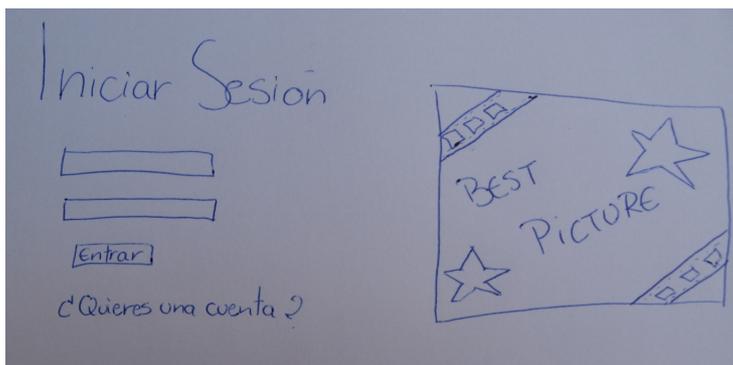


FIGURA 3 - 10: PANTALLA ACCESO

PANTALLA: REGISTRO DE USUARIOS

El usuario que quiera registrarse deberá introducir su usuario y su correo electrónico (FIGURA 3 - 11).

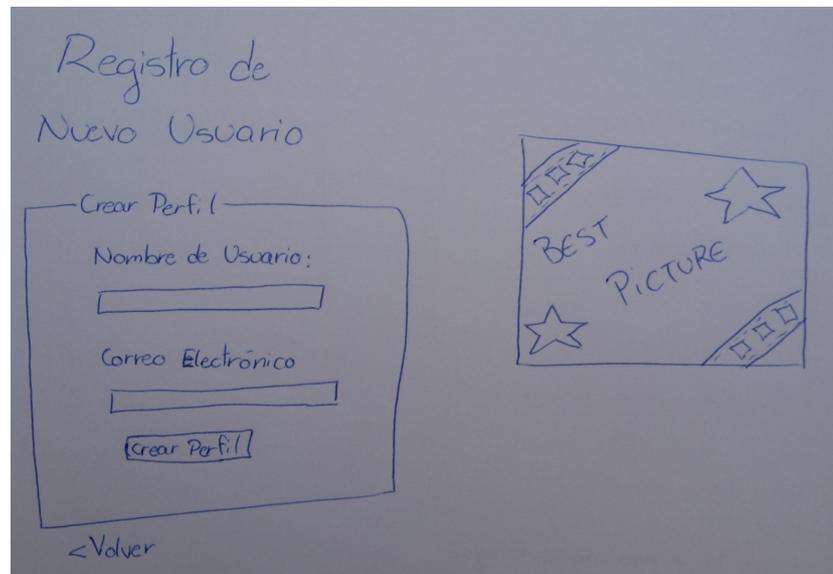


FIGURA 3 - 11: PANTALLA REGISTRO DE USUARIOS

de una red de confianza

PANTALLA: PÁGINA PRINCIPAL DE USUARIO

La apariencia de la aplicación Web es la misma en la mayoría de los casos, como vemos en la FIGURA 3 - 12. Aparecerá el nombre de la aplicación arriba a la izquierda (BEST PICTURE). Arriba a la derecha aparece el usuario que está usando la aplicación y la opción de cerrar sesión. Justo debajo de esto el usuario podrá realizar una búsqueda introduciendo palabras o el título completo de la película. Y a la izquierda aparecen todas las demás posibles opciones que el usuario podrá usar de la aplicación.

El usuario a partir de aquí podrá realizar todas las funciones que tiene la aplicación Web. Podemos observar que el usuario podrá Cerrar Sesión, obtener un listado de películas de la A a la Z, obtener listados de películas de un año en concreto, ver sus votaciones realizadas, obtener recomendación y ver sus valores de confianza.

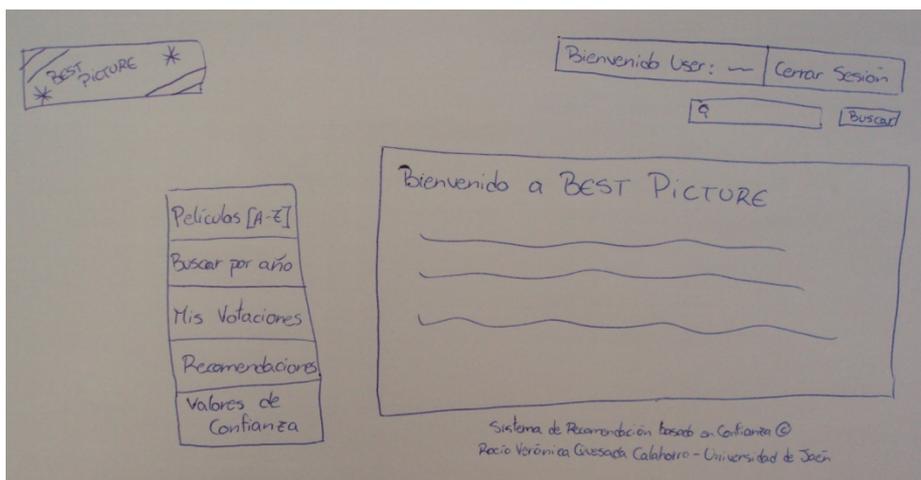


FIGURA 3 - 12: PANTALLA PÁGINA PRINCIPAL

PANTALLA: BÚSQUEDA POR INICIAL

En esta pantalla aparecerá una lista de películas que coincidan con la letra inicial del título. Dicha letra es elegida por el usuario.

Además aparecerá la opción de poder puntuar las películas y también modificar la puntuación si ha sido votada con anterioridad. Podemos ver esta pantalla en la **¡Error! No se encuentra el origen de la referencia.3 - 13**.

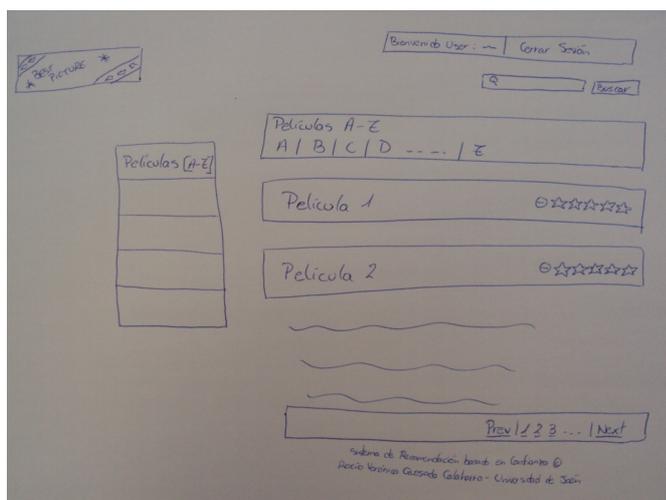


FIGURA 3 - 13: PANTALLA BÚSQUEDA POR INICIAL

PANTALLA: BÚSQUEDA POR AÑO

En esta pantalla (**¡Error! No se encuentra el origen de la referencia.14**), aparecerá una lista de películas que coincidan con el año del estreno de la película. Dicho año es elegido por el usuario.

de una red de confianza

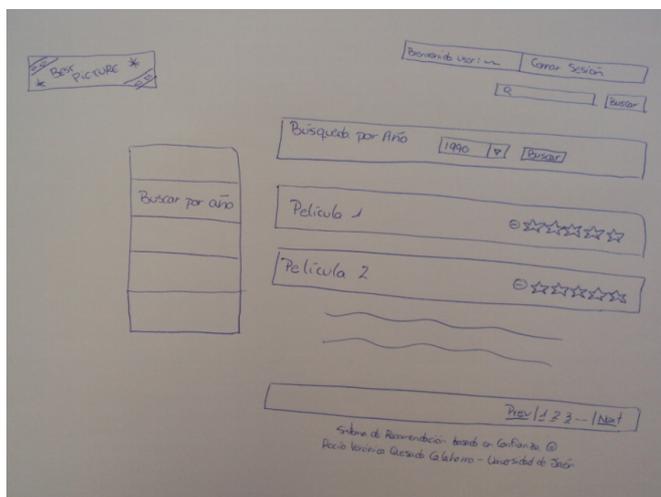


FIGURA 3 - 14: PANTALLA BÚSQUEDA POR AÑO

PANTALLA: VOTACIONES DE LOS USUARIOS

En esta pantalla (**¡Error! No se encuentra el origen de la referencia.15**), se muestran las películas que hemos votado y además la puntuación que le hemos dado a cada una, con la opción de modificar el valor de dicha votación.

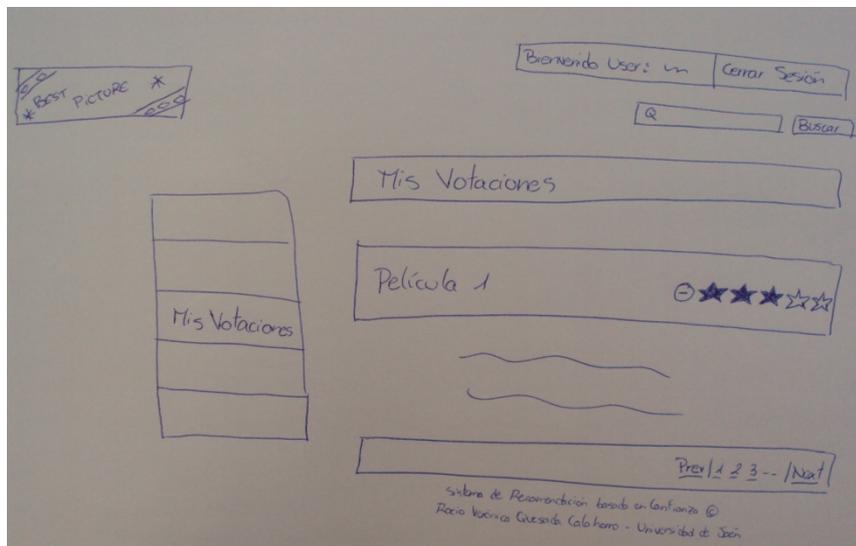


FIGURA 3 - 15: PANTALLA VOTACIONES

PANTALLA: RECOMENDACIONES

Esta pantalla (**¡Error! No se encuentra el origen de la referencia.16**), muestra un top10 de recomendaciones al usuario. Para llevarlo a cabo el sistema predice las valoraciones de todas aquellas películas que el usuario aún no ha visto y luego se queda con las 10 mejores.

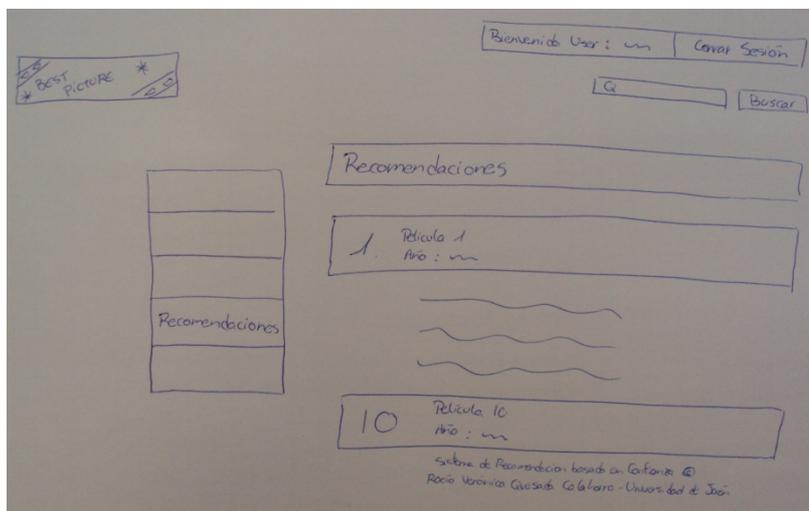


FIGURA 3 - 16: PANTALLA RECOMENDACIONES

PANTALLA: LISTAR PELÍCULA BUSCADA

Si el usuario decide realizar la búsqueda mediante la caja de texto en la que se introduce una cadena y se aplica la opción buscar, el sistema devolverá una lista de películas en las que el título cumplan las restricciones de búsqueda que ha realizado el usuario (FIGURA 3 – 17). También puede no devolver ninguna película si no encuentra ninguna que cumpla con dicha búsqueda.

de una red de confianza

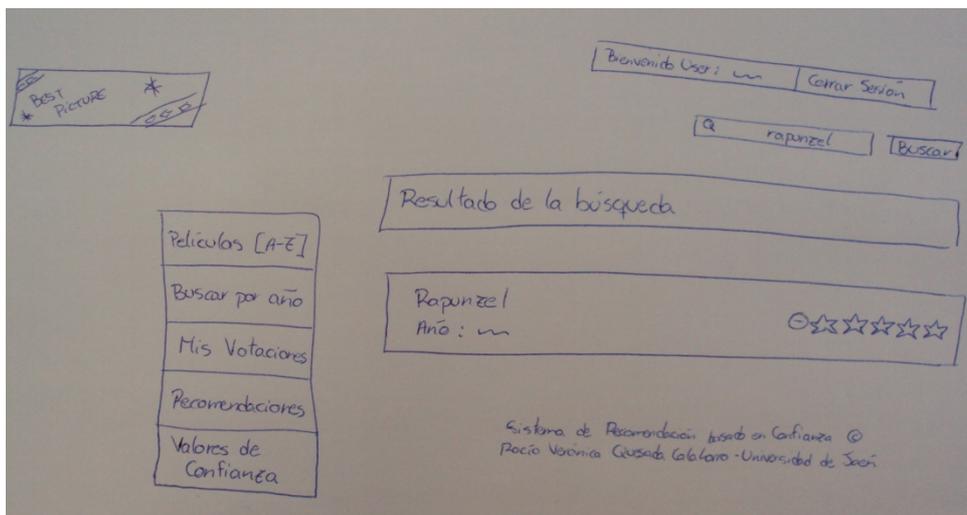


FIGURA 3 - 17: PANTALLA LISTAR PELÍCULA BUSCADA

PANTALLA: VALORES DE CONFIANZA

En esta pantalla (FIGURA 3 – 18) se muestran los usuarios y los valores de confianza que se le han otorgado.

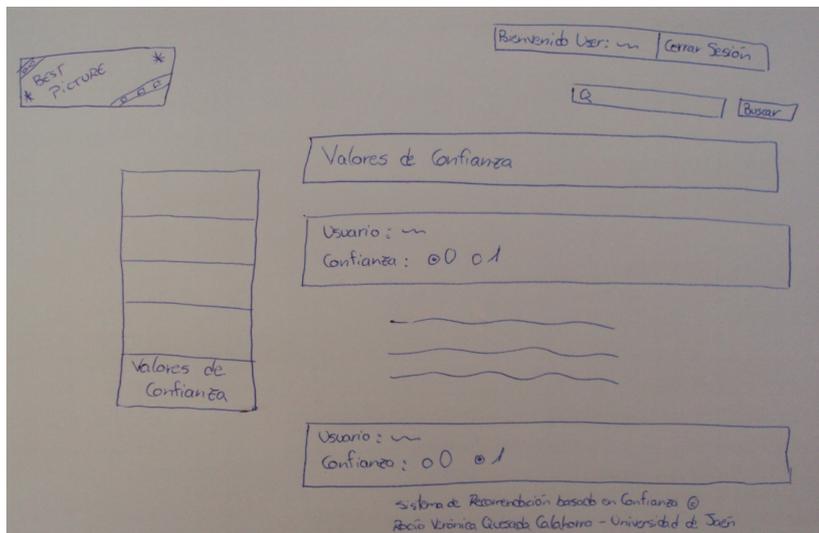


FIGURA 3 - 18: PANTALLA VALORES DE CONFIANZA

Además hemos tenido en cuenta que en el sistema se pueden producir diferentes errores, para ello hemos contemplado posibles errores y por tanto mostramos a continuación las siguientes pantallas:

PANTALLA: ERROR DE AUTENTIFICACIÓN

Podemos encontrar un error que hemos contemplado a la hora de iniciar sesión. Si un usuario introduce erróneamente el usuario y/o la contraseña, el sistema muestra un error de autenticación. (**¡Error! No se encuentra el origen de la referencia.9**).



FIGURA 3 - 19: PANTALLA ERROR AUTENTIFICACIÓN

de una red de confianza

PANTALLA: ERROR DE REGISTRO

Por otra parte también podemos encontrarnos un error de registro, que el usuario ya esté registrado en el sistema o que no coincidan las contraseñas introducidas. Por estos motivos aparecería dicha pantalla. (**¡Error! No se encuentra el origen de la referencia.20**).

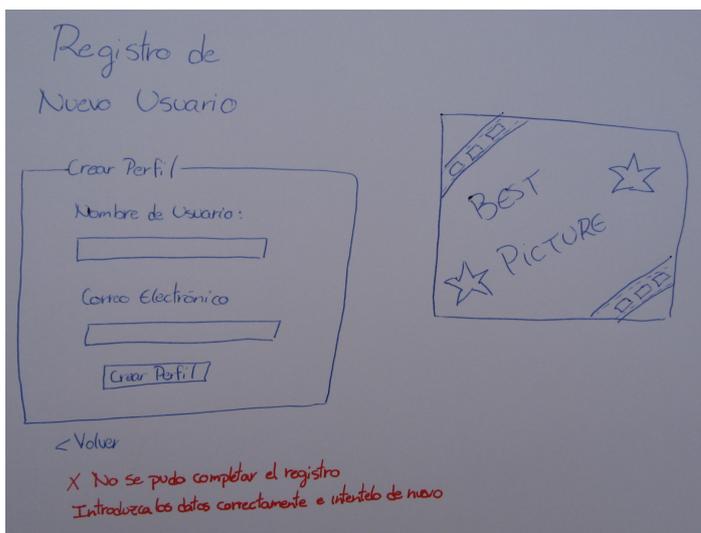


FIGURA 3 - 20: PANTALLA ERROR DE REGISTRO

3.5.1.4. Caminos de navegación

Una vez visto como es el diseño visual de cada una de las pantallas de la interfaz de una forma estática, es el momento de ver si la interacción entre ellas es la apropiada. Para ello usaremos una herramienta llamada stoyboard, que nos permitirá visualizar la secuencia de acciones que realiza un usuario con la aplicación, y a su vez ver el transcurso de pantallas que éstas conllevan.

Los storyboards en términos de diseño Web hacen referencia a una serie de diagramas que resumen la secuencia y disposición del contenido, información y funcionalidad, alrededor de un sitio Web y en cada una de sus páginas.

Lo que se pretende es simular las transiciones y el comportamiento de la aplicación Web y para ello es necesario realizar prototipos o dibujos que representen cómo estarán organizados los elementos en las páginas.

El procedimiento es el siguiente: se sitúan capturas de las pantallas de la interfaz unidas mediante flechas para indicar el camino que sigue la interacción. La posición de origen de las flechas debe ayudar a entender cuál es el elemento que ha desencadenado el paso de una pantalla a otra.

En nuestro caso hemos decidido desarrollar prototipos de alta fidelidad (wireframes o mockups en diseño Web) que representan aspectos más precisos de forma interactiva. El objetivo de estos prototipos es mostrar el contenido de las páginas con alto nivel de detalle y aunque no significan en ningún caso un compromiso final con el diseño gráfico, sirven como herramienta de comunicación entre arquitectos de información, programadores, diseñadores y clientes. Utilizando esta técnica conseguimos centrarnos en el diseño de contenidos en lugar del diseño visual, evitando cambios posteriores más costosos.

Los elementos a tener en cuenta son los siguientes:

- Inventario del contenido de cada página.
- Elementos de cada página: cabeceras, enlaces, botones, formularios, etc.
- Etiquetado de vínculos, títulos, etc.
- Layout o estructura: ubicación y agrupamiento de los elementos en la página.

El comportamiento se muestra mediante notas asociadas a los elementos para indicar cómo se deben mostrar.

de una red de confianza

A continuación exponemos los Storyboards desarrollados.

STOYBOARD: ENTRADA AL SISTEMA

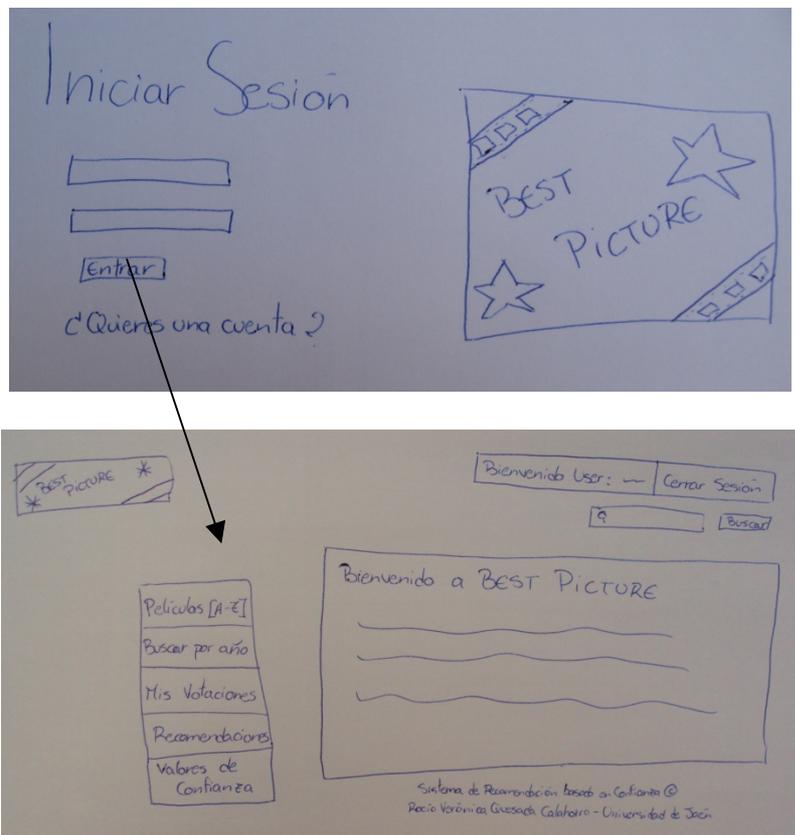


FIGURA 3 - 21: STORYBOARD ENTRADA AL SISTEMA

Transiciones:

- El usuario introduce sus credenciales, si el login es correcto es conducido a la página principal de la aplicación.
- Evento alternativo - En caso de que la autenticación no tenga éxito, se permanece en la pantalla de login y se mostrará un mensaje de error.

STOYBOARD: REGISTRO EN EL SISTEMA

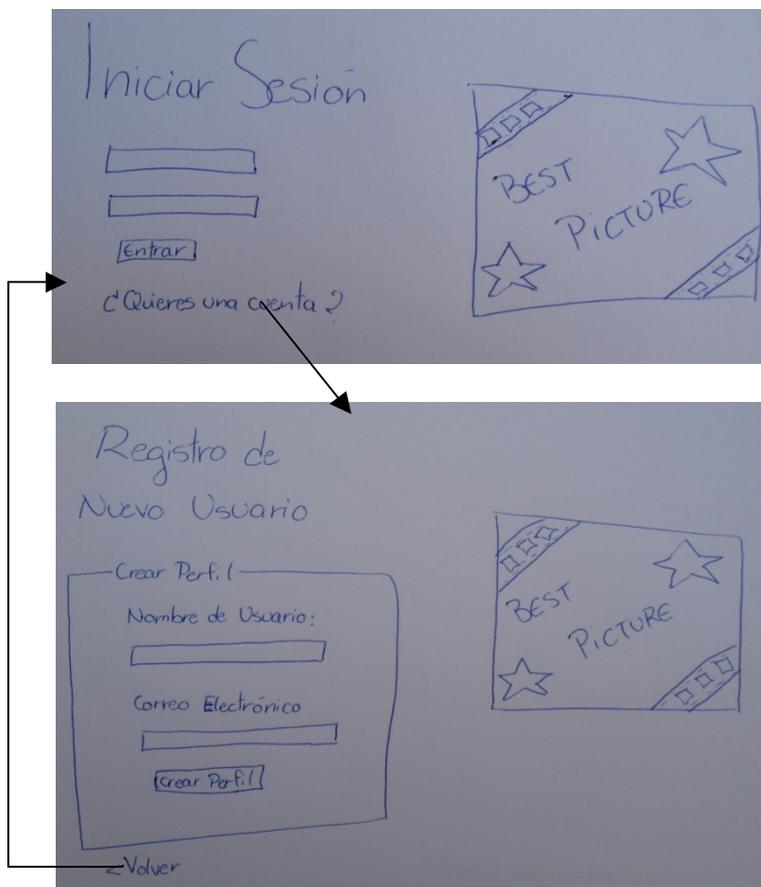


FIGURA 3 - 22: STORYBOARD REGISTRO EN EL SISTEMA

Transiciones:

- El usuario pulsa la opción de “¿Quieres una cuenta?” y es dirigido a la página de registro.
 - El usuario introduce sus credenciales y pincha en “Crear Perfil”. Si el registro se ha realizado correctamente es redirigido a la página de login mostrándose un mensaje de éxito.
- Evento alternativo - En caso de no realizarse el registro correctamente se mostrará un mensaje de error. En este punto el usuario puede seguir intentando el registro o hacer click en volver.
- Cuando el usuario pulse en el enlace volver será dirigido hacia la página de login.

de una red de confianza

STOYBOARD: PELÍCULAS [A-Z]

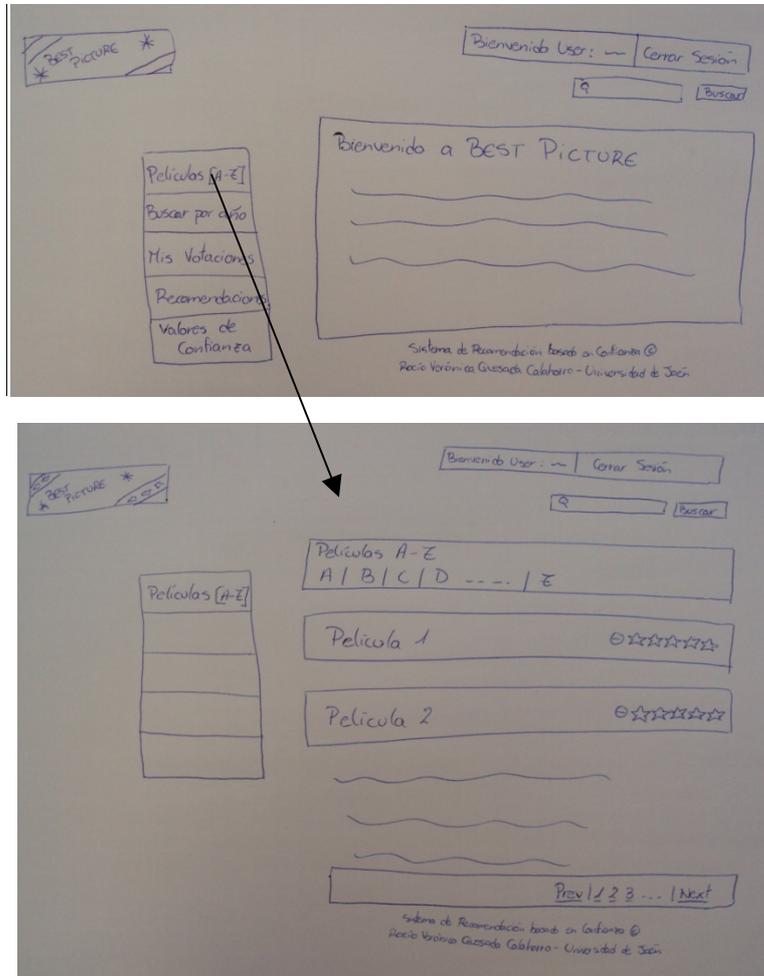


FIGURA 3 - 23: STORYBOARD BÚSQUEDA [A-Z]

Transiciones

- El usuario pincha en “Películas [A-Z]” desde cualquiera de las páginas del sitio web, siendo dirigido a la sección de búsqueda de películas alfabéticamente. Aquí se presentará un método para navegar alfabéticamente por la base de datos de películas.

STOYBOARD: BUSCAR POR AÑO

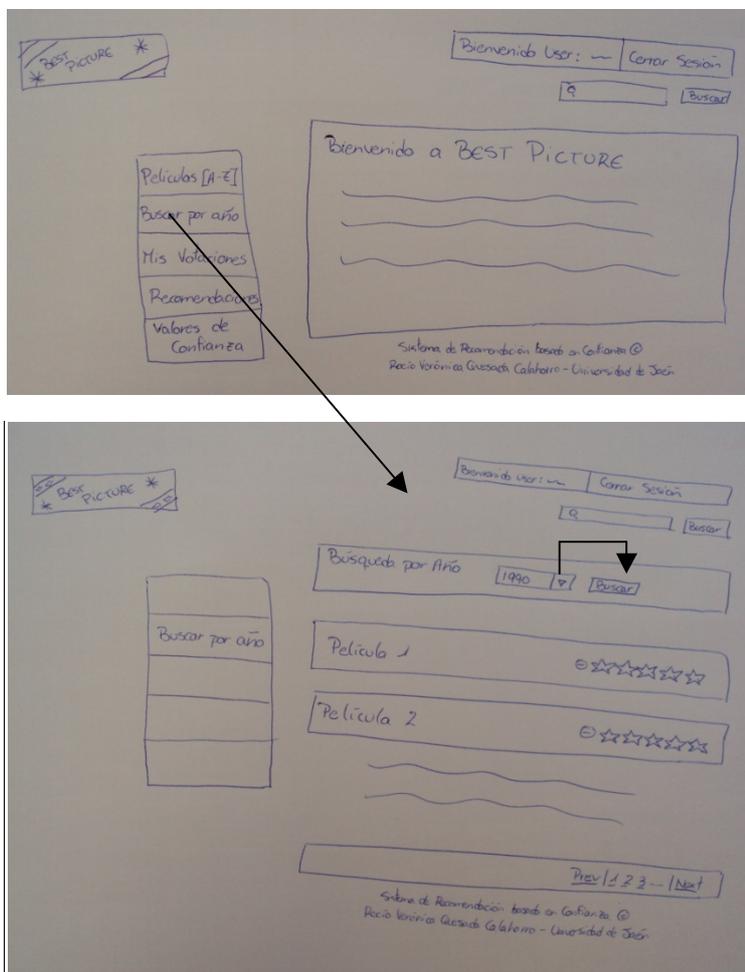


FIGURA 3 - 24: STORYBOARD BÚSQUEDA POR AÑO

Transiciones

- En caso de seleccionar la opción “Buscar por Año” el usuario será dirigido hacia el método de busca de películas en función del año en que fueron publicadas.
- Mediante el menú desplegable se podrá buscar el año y clickando en buscar se mostrarán los resultados deseados.
- Evento alternativo – En caso de que no existan películas disponibles para un año determinado se mostrará en la página el mensaje de error correspondiente.

de una red de confianza

STOYBOARD: MIS VOTACIONES

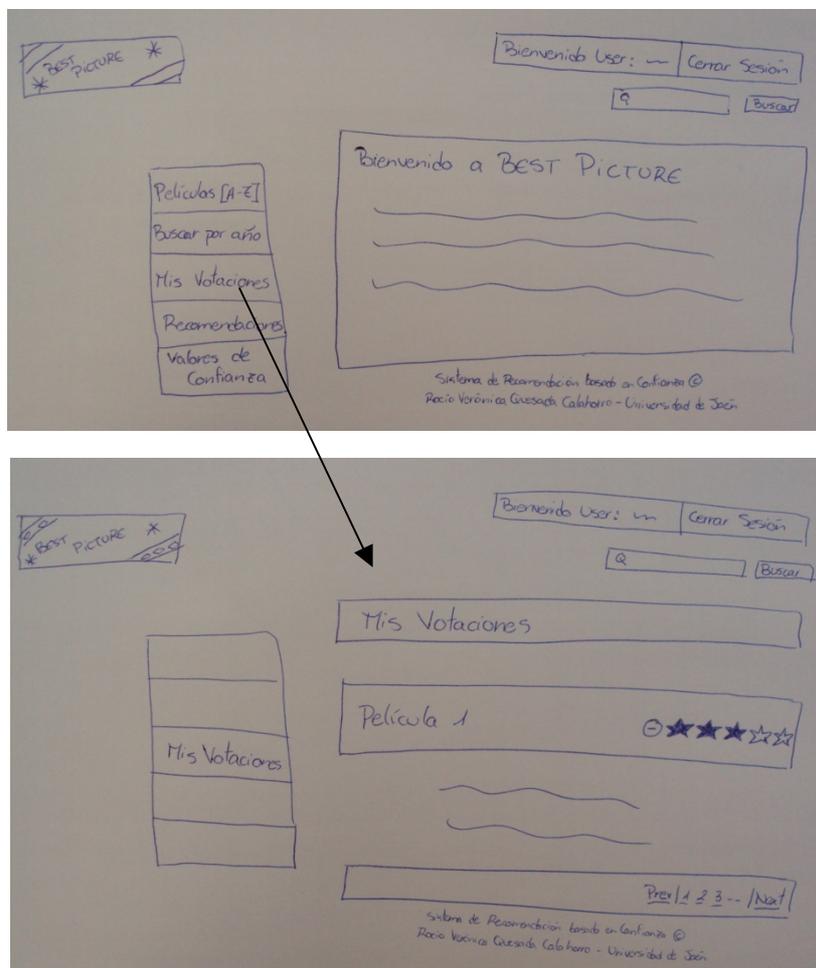


FIGURA 3 - 25: STORYBOARD MIS VOTACIONES

Transiciones

- Pinchando en la sección “Mis Votaciones” el usuario será dirigido hacia una página donde serán mostradas aquellas películas que el usuario ha votado con anterioridad.
- Evento alternativo – Si el usuario no ha realizado ninguna valoración se le mostrará el mensaje de error correspondiente.

STOYBOARD: RECOMENDACIONES

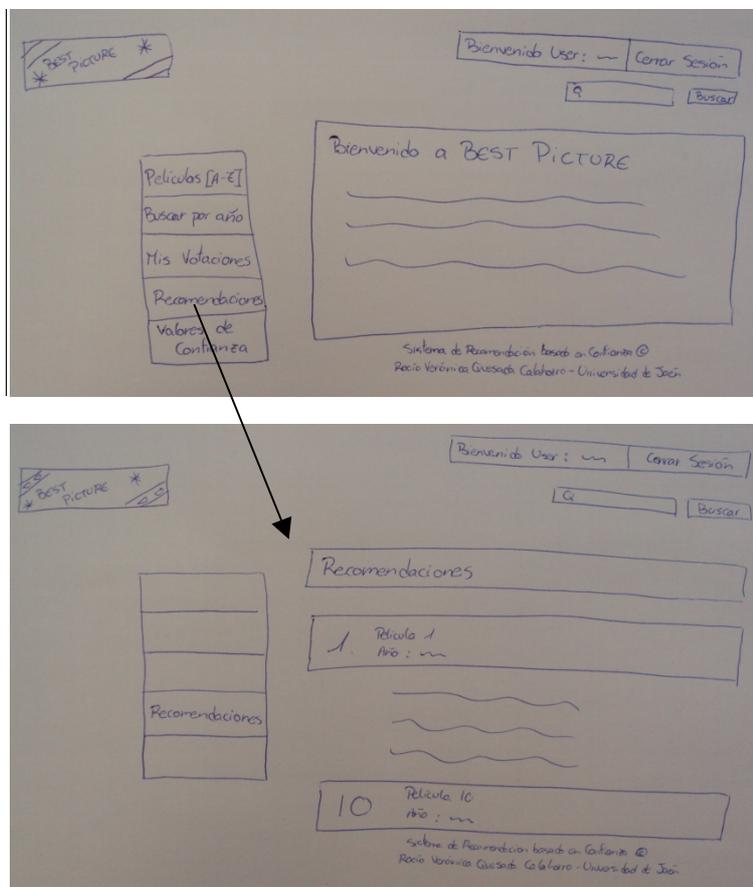


FIGURA 3 - 26: STORYBOARD RECOMENDACIONES

Transiciones:

- El usuario pincha en “Recomendaciones” desde cualquier página de la aplicación web. En condiciones normales se le mostrarán al usuario el Top-N de recomendaciones ofrecidas por el sistema. No obstante, para que el sistema pueda realizar el cómputo de las valoraciones el usuario debe haber realizado previamente un total de M valoraciones.
- Evento alternativo – En caso de no haber realizado suficientes valoraciones, el sistema mostrará un mensaje de error.

de una red de confianza

STOYBOARD: VALORES DE CONFIANZA

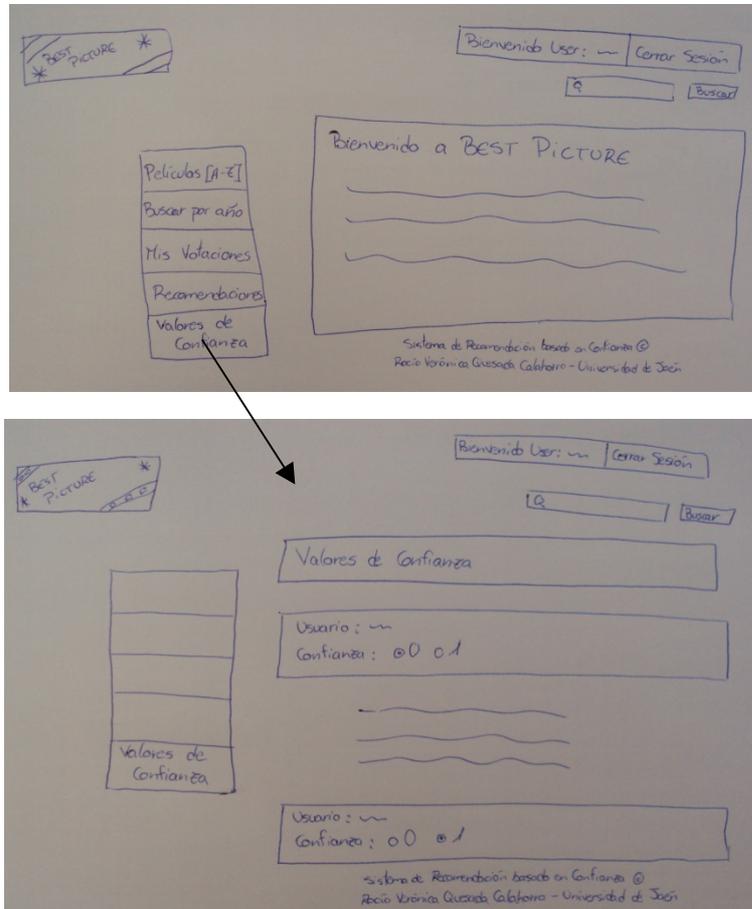


FIGURA 3 - 27: STORYBOARD VALORES DE CONFIANZA

Transiciones

- Pinchando en la sección “Valores de Confianza” el usuario será dirigido hacia una página donde se mostrarán los usuarios y la confianza que el usuario en cuestión les ha otorgado.

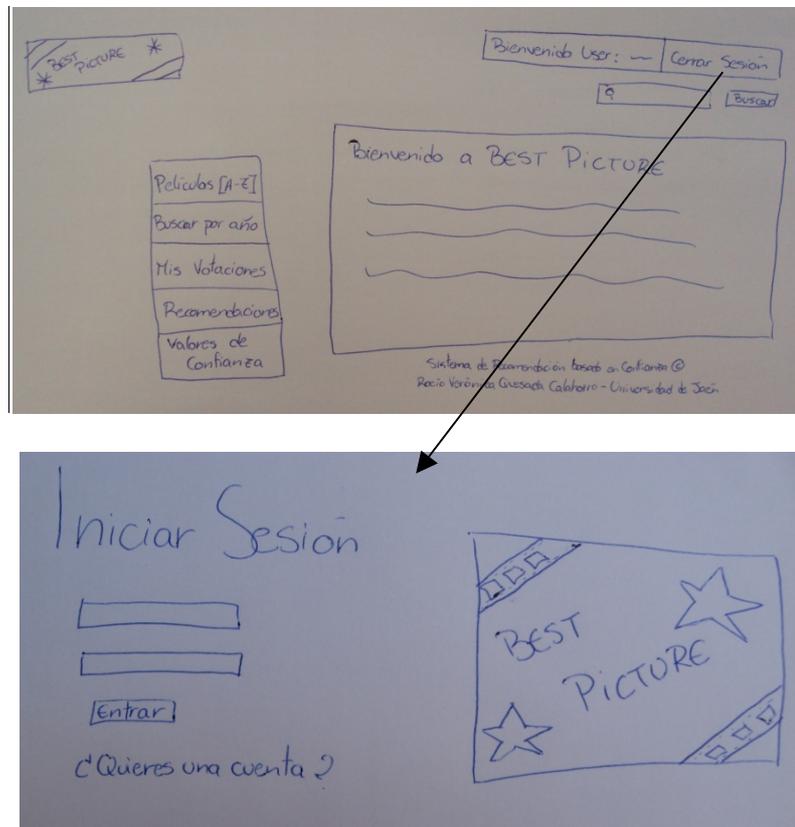
STOYBOARD: CERRAR SESIÓN

FIGURA 3 - 28: STORYBOARD CERRAR SESIÓN

Transiciones

- Desde cualquiera de las páginas del sistema, a excepción de la página de login y registro, el usuario pincha en “Cerrar Sesión”, cerrando la sesión de usuario y siendo dirigido a la página de login.

de una red de confianza

STORYBOARD: HERRAMIENTA DE BÚSQUEDA

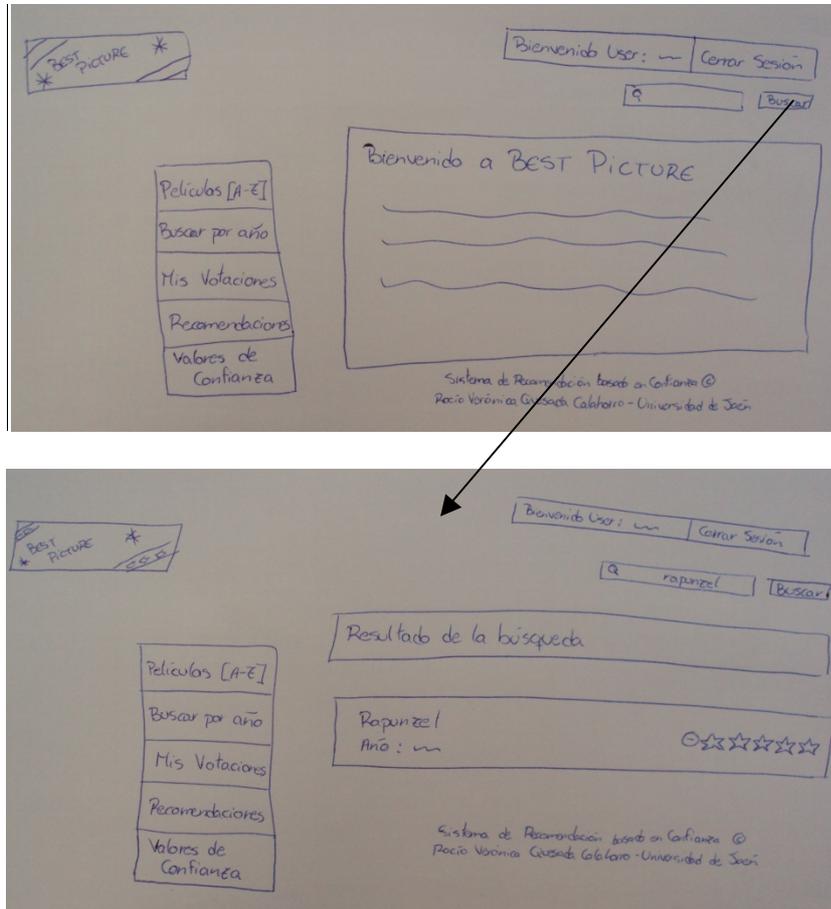


FIGURA 3 - 29: STORYBOARD HERRAMIENTA DE BÚSQUEDA

Transiciones

- El usuario introduce una consulta sobre una película en la herramienta de búsqueda. Si se encuentra la película, ésta es mostrada en el área principal de la página.
- Evento alternativo - En caso de no encontrarse la película solicitada en la base de datos, se muestra un mensaje de error.

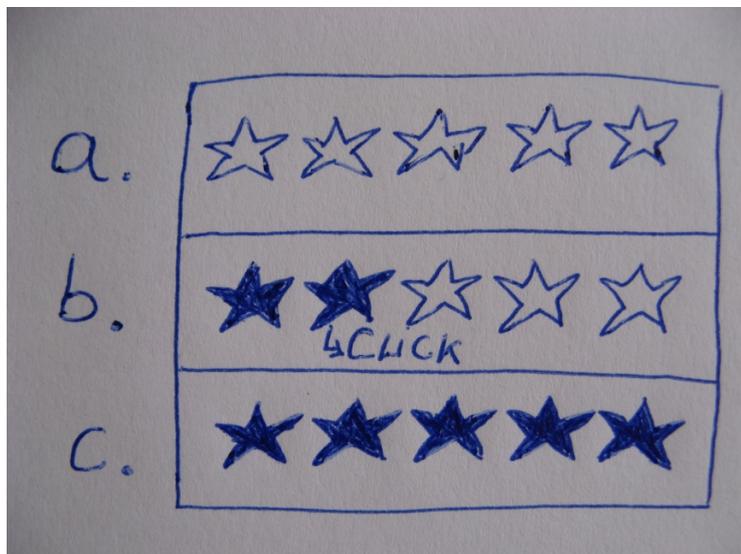
STORYBOARD: EMITIR Y MODIFICAR VALORACIÓN

FIGURA 3 - 30: STORYBOARD EMITIR Y MODIFICAR VALORACIÓN

Transiciones:

Desde cualquier página donde mostremos resultados de películas, como en las de búsqueda y en “Mis Votaciones” observamos las valoraciones actuales de los resultados mostrados:

- a - La película no ha sido valorada hasta el momento
- b - El usuario podrá emitir la valoración o modificar una ya realizada mediante el sistema de valoración con estrellas basado en “radio buttons”. El usuario podrá advertir cómo al pasar el ratón por encima de las estrellas éstas se encienden. La valoración se producirá cuando el usuario haga click sobre la estrella correspondiente a la valoración que desea realizar. En el caso del ejemplo, el usuario vota la película con un 2.
- c - La película tiene la máxima puntuación.

de una red de confianza

STORYBOARD: VALOR DE CONFIANZA

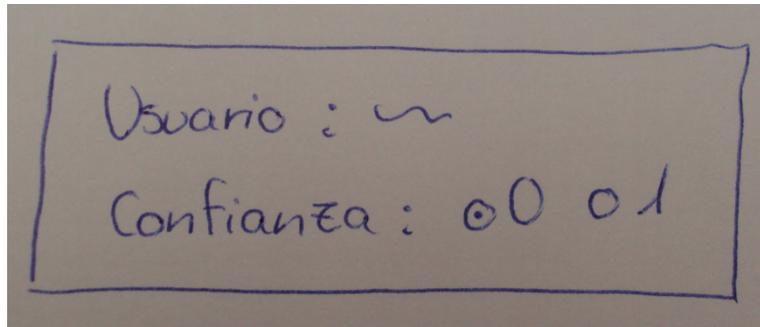


FIGURA 3 - 31: STORYBOARD VALOR DE CONFIANZA

Transiciones:

Desde la página de Valores de Confianza observamos los valores de confianza que se le han otorgado a los distintos usuarios. Este valor puede ser 0 si no confía en el usuario y 1 si confía en él.

3.5.2 Diseño de datos

3.5.2.1. Esquema E - R

El significado de cada uno de los componentes del diagrama de entidad-relación es:

- Entidad fuerte: está compuesto por el nombre de la entidad y por sus atributos; el atributo que está subrayado se trata de la clave principal de la entidad, que identifica cada elemento.
- Entidad débil: Es igual que una entidad fuerte, pero en este caso la clave principal no es suficiente para identificar de forma única cada objeto. Se considera débil respecto de una entidad fuerte.
- Relación:
 - Uno a uno: Esta relación la utilizaremos cuando solamente haya una única correspondencia entre dos entidades, y en ambas direcciones. Por ejemplo: un grupo solamente posee un profesor tutor, y a la vez, un profesor solo es tutor de un grupo.
 - Uno a muchos: Con esta relación se representa el caso en el que una entidad tenga varias ocurrencias con respecto a otra. Por ejemplo: un alumno tiene muchas notas, pero las notas solo son de un alumno.
 - Muchos a muchos: Con esta relación se representa el caso en que una entidad primera tenga varias ocurrencias con respecto de otra segunda, y a la vez esta entidad segunda tendrá varias ocurrencias en la primera entidad. Por ejemplo: una asignatura podrá impartirse en varios grupos, y además en un grupo se imparten varias asignaturas.

En los siguientes diagramas de entidad – relación la cardinalidad la especificamos de la siguiente forma:

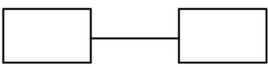
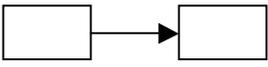
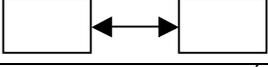
Relación es de uno a uno	
Relación es de uno a muchos	
Relación es de muchos a muchos	

FIGURA 3 - 32: CARDINALIDAD EN DIAGRAMA ENTIDAD-RELACIÓN

de una red de confianza

DIAGRAMA ENTIDAD-RELACIÓN

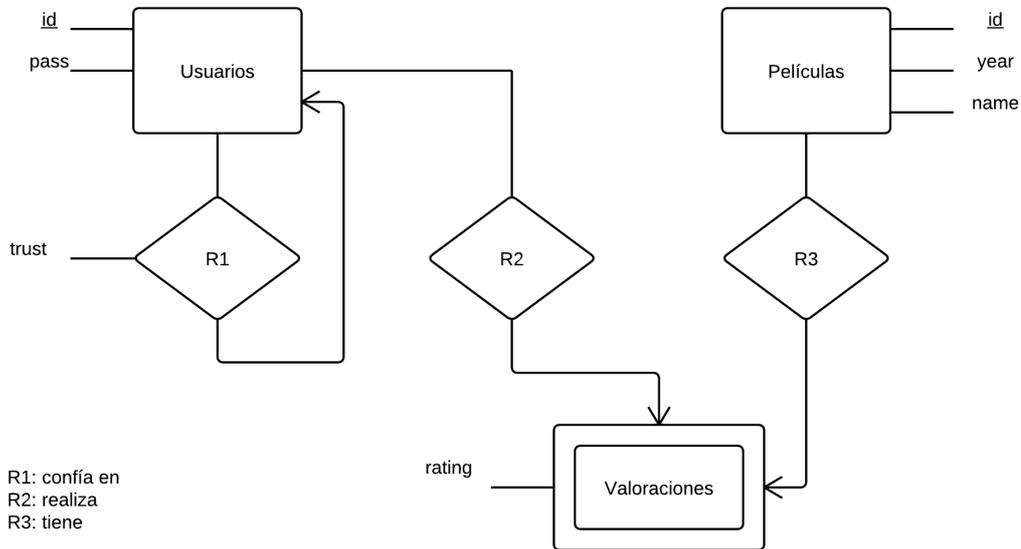


FIGURA 3 - 33: DIAGRAMA ENTIDAD-RELACIÓN

DIAGRAMA ENTIDAD-RELACIÓN MODIFICADO

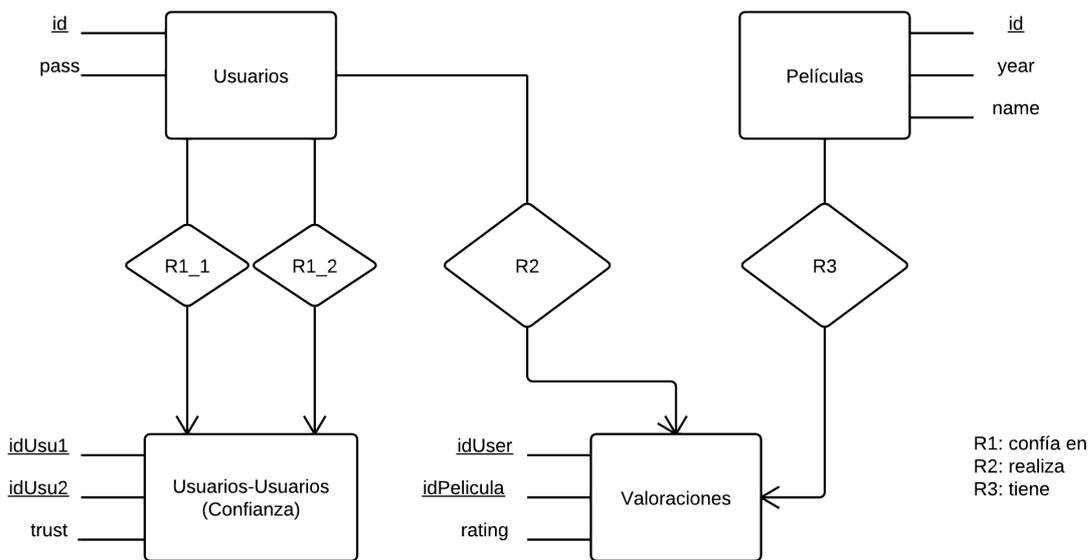


FIGURA 3 - 34: DIAGRAMA ENTIDAD-RELACIÓN MODIFICADO

3.5.2.2. Tablas

Una tabla es un conjunto de filas y columnas. Las filas se llaman registros y a veces también tuplas. Todas las filas tienen la misma estructura pero son diferentes en contenido. Cada fila tiene el mismo número de columnas, con el mismo nombre y tipo de información, pero el contenido de cada una de estas es diferente. Las columnas se llaman campos o atributos.

La información de una base de datos, por tanto, se almacena siempre en una o más tablas. Esta es la perspectiva lógica, independientemente por completo respecto al formato de almacenamiento físico de la información. El software de gestión de los datos puede estar guardándolos en archivos individuales o bien en un volumen único ajeno incluso al sistema de archivos del sistema operativo.

El conjunto de tablas han de tener una coherencia, esto es, que a pesar de que contengan datos diferentes deben de tener una relación entre sí.

Si hay un conjunto de tablas que es totalmente diferente al resto sería conveniente que éste conjunto de tablas formara parte de otra base de datos independiente. Pero este no es nuestro caso.

La relación entre dos tablas de una base de datos se encuentra en la coincidencia entre los valores existentes en unas columnas concretas.

Usuarios

Tabla para almacenar la información de los usuarios que contiene los siguientes atributos (FIGURA 3 - 35):

- id: Entero. Llave primaria. Llave externa. Identificador numérico y único para cada usuario.
- pass: cadena de 25 caracteres. Contraseña del usuario.

de una red de confianza

←T→			id	pass
<input type="checkbox"/>			1	1
<input type="checkbox"/>			4	4
<input type="checkbox"/>			9	9
<input type="checkbox"/>			13	13
<input type="checkbox"/>			14	14
<input type="checkbox"/>			16	16
<input type="checkbox"/>			17	17
<input type="checkbox"/>			19	19
<input type="checkbox"/>			20	20

FIGURA 3 - 35: TABLA USUARIOS

Película

Es una tabla que almacena todas las películas del sistema, y tiene los siguientes atributos (FIGURA 3-36):

- id: Entero. Llave primaria. Llave externa. Identificador numérico y único para cada película.
- year: entero, año en el que la película fue estrenada.
- name: cadena de 100 caracteres. Es el nombre de la película.

←T→			id	year	name
<input type="checkbox"/>			1	2003	Dinosaur Planet
<input type="checkbox"/>			2	2004	Isle of Man TT 2004 Review
<input type="checkbox"/>			3	1997	Character
<input type="checkbox"/>			4	1994	Paula Abdul's Get Up & Dance
<input type="checkbox"/>			5	2004	The Rise and Fall of ECW
<input type="checkbox"/>			6	1997	Sick
<input type="checkbox"/>			7	1992	8 Man
<input type="checkbox"/>			8	2004	What the #\$*! Do We Know!?
<input type="checkbox"/>			9	1991	Class of Nuke 'Em High 2
<input type="checkbox"/>			10	2001	Fighter
<input type="checkbox"/>			11	1999	Full Frame: Documentary Shorts
<input type="checkbox"/>			12	1947	My Favorite Brunette
<input type="checkbox"/>			13	2003	Lord of the Rings: The Return of the King: Extende...
<input type="checkbox"/>			14	1982	Nature: Antarctica

FIGURA 3 - 36: TABLA PELÍCULAS

Valoraciones

Es una tabla que almacena las valoraciones que los usuarios hacen de las películas y que contiene los siguientes atributos (FIGURA 3 - 37):

- IdUser: Entero. Llave primaria. Identificador del usuario.
- IdPelícula: Entero. Llave primaria. Identificador numérico y único para cada película.
- Rating: Entero. Puntuación (1, 2, 3, 4 ó 5) del usuario sobre la película.

←T→			idUser	idPelícula	rating
<input type="checkbox"/>			1	100	4
<input type="checkbox"/>			1	101	5
<input type="checkbox"/>			1	102	3
<input type="checkbox"/>			1	10	3
<input type="checkbox"/>			1	103	5
<input type="checkbox"/>			1	104	2
<input type="checkbox"/>			1	105	5
<input type="checkbox"/>			1	106	5
<input type="checkbox"/>			1	107	5
<input type="checkbox"/>			1	108	5
<input type="checkbox"/>			1	109	3

FIGURA 3 - 37: TABLA VALORACIONES

Confianza

En esta tabla se almacena la confianza que tiene un usuario respecto a otro (FIGURA 3-38):

- idUsu1: Entero. Llave primaria. Identificador del usuario que otorga el valor de confianza.
- idUsu2: Entero. Llave primaria. Identificador del usuario al que se le está dando el valor de confianza.
- Trust: Entero. Puntuación (0,1) de la confianza que tiene el usuario 1 respecto al usuario 2.

de una red de confianza

← T →			idUsu1	idUsu2	trust
<input type="checkbox"/>			1	2	1
<input type="checkbox"/>			1	3	1
<input type="checkbox"/>			1	4	1
<input type="checkbox"/>			1	5	1
<input type="checkbox"/>			1	6	1
<input type="checkbox"/>			1	7	1
<input type="checkbox"/>			1	8	1
<input type="checkbox"/>			1	9	1
<input type="checkbox"/>			1	10	1
<input type="checkbox"/>			1	11	1
<input type="checkbox"/>			1	12	1
<input type="checkbox"/>			1	13	1

FIGURA 3 - 38: TABLA CONFIANZA

3.7. Implementación

El último paso que hay que dar en la Ingeniería del Software es la implementación. Ahora es cuando toca implementar todo el proceso que hemos llevado a cabo anteriormente, traduciendo diagramas de clases y obtener de esa manera el código fuente de nuestra aplicación, el cual podrá ser ejecutado en un ordenador.

Nuestro proyecto, como vimos en el diagrama de clases (FIGURA 3 – 9), está compuesto por 4 clases DAO (una por cada tabla de datos de la base de datos) en las que hemos implementado todos los métodos necesarios para la consulta y modificación de los datos almacenados en las tablas del apartado anterior. También tiene una clase Valoración, una clase Usuario, una clase Película, una clase Confianza y una clase PerfilUsuario en las que se definen sus atributos junto con los métodos necesarios para interactuar con ellas. Tiene otra clase llamada SistemaRecomendación en la que se implementan todos los métodos necesarios para llevar a cabo el filtrado y el cálculo de la confianza (se muestran algunos de ellos en el APÉNDICE C). Y por último tiene una clase App que es la encargada de llamar a todo el mundo y de dar conexión a quien se lo pida.

En los siguientes apartados veremos las herramientas y tecnologías utilizadas, las pruebas de caja negra, la información relativa a la instalación de la aplicación y el manual de usuario.

3.6.1 Tipo de arquitectura de la aplicación

Nosotros hemos llevado a cabo un sistema de recomendación con una arquitectura cliente/servidor y una interfaz Web de comunicación con el usuario.

Gracias a la arquitectura cliente/servidor **¡Error! No se encuentra el origen de la referencia.**3 - 39 podemos conseguir que el usuario no acceda a información que no debe, y además le evitamos tener que instalar una gran cantidad de software para ejecutar la aplicación.

El funcionamiento que tiene esta arquitectura es el siguiente: la aplicación se encuentra en un servidor central, al que los usuarios acceden por medio un software cliente, en nuestro caso un navegador Web. Una vez que ha accedido a la aplicación, el usuario realiza peticiones que son atendidas por el servidor, generando así una respuesta adecuada y comprensible para el cliente.

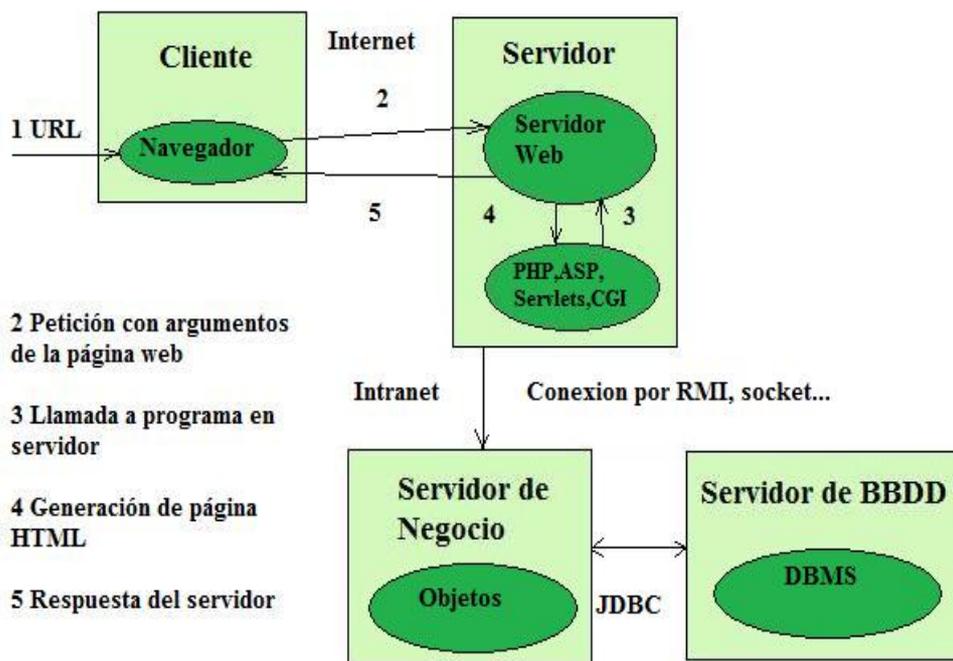


FIGURA 3 - 39: ARQUITECTURA CLIENTE/SERVIDOR

de una red de confianza

3.6.2 Tecnologías y herramientas de desarrollo utilizadas

Necesitamos un lenguaje que sea capaz de generar contenido dinámico desde el navegador, y que sea transparente al usuario. Existen distintas opciones que pueden resolver nuestro problema, pero nosotros nos vamos a decantar por JSP (Java Server Pages). Además la implementación de esta parte se ha realizado usando Java y sentencias en lenguaje SQL (MySQL) mediante el empleo de un driver para conectar la base de datos con la aplicación. A continuación vamos a describir el lenguaje Java, caracterizado principalmente por su seguridad, sencillez e independencia de la plataforma, siendo el lenguaje usado en la programación del prototipo web.

Principales características de Java

Java es un lenguaje de programación orientado a objetos, con una sintaxis similar a C o C++, pero ofreciendo una mayor simplicidad y robustez en el ciclo de desarrollo: las construcciones y características más complicadas de C y C++ han sido eliminadas, y el lenguaje contiene mecanismos implícitos para garantizar la seguridad de las aplicaciones construidas con él. También incorpora dos mecanismos a la hora de escribir programas simples, potentes y robustos: un tratamiento interno de multitarea y un sistema de excepciones que normaliza el procesado de errores por parte del programador. La principal característica de Java es que es independiente de la plataforma, pudiendo ejecutarlo sobre distintas arquitecturas y sistemas operativos sin que sea necesario modificar el código del programa. Esta independencia se logra debido a que el lenguaje está soportado por dos elementos fundamentales: el compilador y la máquina virtual. El compilador traduce los programas a un formato especial llamado bytecode, que es el formato que se le pasa a la máquina virtual. Sun describe el lenguaje Java como simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico.

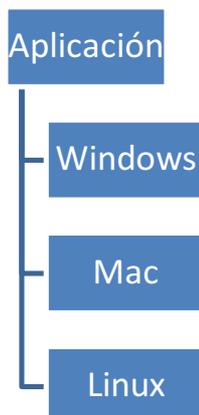


FIGURA 3 - 40: MULTIPLATAFORMA

Aplicaciones web en Java

Cuándo se realiza una aplicación web en Java lo usual es programar los objetos de negocio con clases Java, y para presentar la información se suele usar JSP y/o Servlet. Pero, en realidad, existen varios enfoques a la hora de construir una aplicación web en Java mediante JSP/Servlets.

El primero se basa en el uso exclusivo de servlets para presentar la información y tiene varios inconvenientes:

- Mezcla aspectos de la interfaz y de procesamiento.
- Dificulta el trabajo de un diseñador web.
- El mantenimiento se hace casi imposible.
- La generación de largos fragmentos de código HTML desde el Servlet resulta compleja.

El segundo enfoque se basa en el uso exclusivo de JSP, el cual resulta igualmente inadecuado:

- Mezcla aspectos de la interfaz y procesamiento.
- Dificulta el trabajo del diseñador web.
- Genera páginas con grandes fragmentos de Java, difíciles de mantener.

El tercer enfoque, conocido como JSP Model 1 se basa en el uso de JSP y beans:

- Los JSP procesan las peticiones y respuestas del cliente e implementan los aspectos de la interfaz.

de una red de confianza

- Los beans implementan el modelo de la aplicación.
- Adecuado para aplicaciones sencillas con poco procesamiento en los JSP.

El cuarto y último enfoque, conocido como JSP Model 2 combina JSP, Servlet y beans:

- Los JSP implementan la interfaz, con un procesamiento sencillo que permite acceder a los objetos de negocio y obtener de ellos el contenido dinámico a mostrar en la página.
- Los servlets actúan como controladores, instanciando objetos del modelo y transfiriendo el control a los JSP correspondientes.
- Los beans implementan el modelo de la aplicación.
- Adecuado para aplicaciones de complejidad media o alta.

NetBeans

En el mercado existen multitud de productos que facilitan el desarrollo de aplicaciones Java, herramientas compuestas de editores de código, diseñadores de interfaces, administradores de paquetes y configuraciones, etc. Una gran parte de ellos son de código libre y multiplataforma, por lo que la elección de la herramienta de desarrollo dependerá principalmente de las necesidades de cada proyecto en particular y, en gran medida, de las preferencias de cada desarrollador.

Nosotros optamos por usar la última versión de NetBeans para implementar el proyecto por dos razones: ser una herramienta libre del mismo fabricante que la plataforma, y estar disponible para múltiples sistemas operativos.

Esta herramienta se ha empleado para editar el código, compilar y depurar. Asimismo, NetBeans ha facilitado la administración de los módulos de código, distribuidos en los paquetes.

Una vez establecido el modelo de la aplicación, elegida la plataforma sobre la que se desarrollará, y seleccionadas las herramientas de trabajo, se ha procedido a la implementación de la aplicación y su puesta en funcionamiento.

3.6.3 Pruebas de caja negra

Este tipo de pruebas se llevan a cabo sobre la interfaz del software y pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma adecuada.
- La salida que se produce es correcta.
- La integridad de la información externa se mantiene.

Los errores que intentan encontrar estas pruebas son:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores de estructuras de datos.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Tipos de pruebas de caja negra:

- Método de la participación equivalente (Trata de obtener un pequeño subconjunto de todas las posibles entradas con la mayor probabilidad de encontrar errores).
- Análisis de valores límite (Situaciones que se dan cuando se introducen valores que están justo en el límite de las clases de equivalencia de entrada y de salida).

Así pues, como las pruebas de caja negra se basan en la interfaz software, por cada componente del paquete Vista realizaremos éstas pruebas. Los componentes de la vista y sus respectivas pruebas de caja negra quedan expuestos de la siguiente forma:

de una red de confianza

TIPO DE PRUEBA: PARTICIÓN EQUIVALENTE (A la hora de Iniciar Sesión).

Condiciones de Entrada	Clases válidas	Clases inválidas
\$usuario	1. Cadena no vacía	2. Cadena vacía
\$contraseña	3. Cadena no vacía	4. Cadena vacía
\$clave = \$contraseña	5. TRUE	6. False
\$usuario identificado	7. SI	8. NO

TABLA 3 - 4: CAJA NEGRA INICIAR SESIÓN

Clases válidas (Se espera que se acceda al perfil del usuario)

\$usuario="1" y \$contraseña = "1" (1,3).

\$clave = "1" y \$contraseña = "1" (3,5).

Usuario identificado (7)

Clases inválidas (Se espera un mensaje de error)

\$usuario = "" y \$ contraseña = "" (2,4).

\$clave = "1" y \$contraseña = "" (4,6).

Usuario identificado (8).

TIPO DE PRUEBA: PARTICIÓN EQUIVALENTE (A la hora de registrarse el usuario)

Condiciones de entrada	Clases válidas	Clases inválidas
\$usuario	1. Cadena no vacía	2. Cadena vacía
\$email	3. Cadena no vacía	4. Cadena vacía
\$email enviado	5. SI	6. NO

TABLA 3 - 5: CAJA NEGRA REGISTRAR USUARIO

Clases válidas (Se espera que se registre el usuario)

\$usuario = "pepe" y \$email = "pepe@ujaen.es" (1,3).

Email enviado (5).

Clases inválidas (Se espera un mensaje de error)

\$usuario = "" y \$email = "" (2,4).

\$email enviado (6).

TIPO DE PRUEBA: PARTICIÓN EQUIVALENTE (A la hora de buscar una película)

Condiciones de entrada	Clases válidas	Clases inválidas
\$película	1. Cadena no vacía	2. Cadena vacía
\$película = \$nom_película	3. TRUE	4. FALSE
\$película encontrada	5. SI	6. NO

TABLA 3 - 6: CAJA NEGRA BUSCAR PELÍCULA

Clases válidas (Se espera una película o una lista de películas)

\$película = "Moon" (1).

\$película = \$nom_película (3).

\$Película/as encontrada/as (5).

Clases inválidas (Se espera un mensaje informando que no hay películas que cumplan los requisitos de búsqueda)

\$película = "" (2).

\$película != \$nom_película (4).

\$Película/as encontrada/as (6).

de una red de confianza

TIPO DE PRUEBA: ANÁLISIS DE VALORES LÍMITE (A la hora de recomendar películas)

El usuario cuando realiza una petición para recibir una recomendación tiene que haber votado antes como mínimo 20 películas. Esto es debido porque para que el sistema sea fiable y de recomendaciones acertadas debe tener información del usuario.

Este planteamiento nuestro sistema lo solventa, ya que si el usuario intenta pedir una recomendación y no ha votado como mínimo 20 películas, el usuario no podrá ver las recomendaciones y aparecerá un mensaje de error informando que debe de votar más películas.

CAPÍTULO 4

CONCLUSIONES

Los sistemas de recomendación nacieron para dar una solución a la gran cantidad de información que podemos encontrar en la red. Debido a esto un usuario que navegue por ella puede sentirse sobrecogido ya que no encuentra la información que desea, y malgasta mucho tiempo en ello. Por esta razón los sistemas de recomendación filtran esta información, adecuándola al perfil de cada usuario, en función de distintos criterios. Y debido a esto existen distintos enfoques para la implementación de los sistemas de recomendación.

Este proyecto surgió con la idea de estudiar los sistemas de recomendación en los que poder agregar el matiz de la confianza, debido a que actualmente hay muy pocos trabajos realizados respecto a ello y que además en la vida real normalmente damos más importancia a las valoraciones de una persona en la que confiamos que a las de una persona en la que no. Para poder desarrollar esta idea tuvimos que partir de cero e indagar en todo lo relacionado con los sistemas de recomendación. Estudiamos sus distintas clasificaciones y sopesamos los pro y contra de cada uno de ellos. Gracias a ellos pudimos concienciarnos de las posibilidades que nos ofrecían los sistemas de recomendación y pudimos cerciorarnos de que la opción escogida era la correcta.

Durante el proceso aprendimos mucho de los distintos sistemas de recomendación, como que los basados en contenido se caracterizan por las propiedades de los elementos o que los colaborativos se basan en la similitud de las valoraciones que distintos usuarios hacen sobre un mismo ítem.

Una vez aprendida la base y el abanico de posibilidades de los que disponíamos, centramos nuestra atención en los sistemas de recomendación colaborativos, por ser los más extendidos y utilizados en sistemas web reales, y contemplamos las opciones que había dentro de ellos: los sistemas de recomendación basados en user – user, que se basan en la similitud que existe entre los distintos usuarios, y los basados en ítem – ítem, que se basan en la similitud entre ítems. Por último todo esto nos llevó a un híbrido en el que se le añadía el matiz de la confianza.

Para poder desarrollar el sistema tuvimos que elegir los algoritmos que estimamos más adecuados para nuestro proyecto y a su vez seguir todas las etapas de la Ingeniería de Software, pasando por el análisis, el diseño y por último la implementación del prototipo.

Todo este sistema en su conjunto lo hemos realizado con la arquitectura cliente/servidor. La información la obtuvimos de Epinions, que suministran una base de datos de libre disposición que se utiliza habitualmente en procesos de evaluación de sistemas de recomendaciones, dada la

de una red de confianza

imposibilidad de tiempo que nos suponía el recopilar los suficientes datos reales para que funcionase y la almacenamos en una base de datos.

Terminar diciendo que los resultados obtenidos con el sistema no han sido los esperados, debido a que los datos de los que disponíamos no eran los más adecuados y no estaban realmente adaptados a la aplicación concreta. En el ámbito de los sistemas de recomendación es de sobra conocido que no hay unos enfoques mejores que otros, sino que unos se comportan mejor que otros dependiendo del ámbito donde se apliquen.

Con esa finalidad en mente se optó por hacer un sistema Web, para dejar que diferentes usuarios entrasen y lo visitasen, y así poder realizar valoraciones de las películas que hubiesen visto. De esa forma el sistema podría realimentarse y ofrecer a éstos unas recomendaciones mucho más fiables. Si que debemos incidir en que aunque los resultados que hemos obtenido no han sido los esperados de cara al rendimiento de las recomendaciones generada, si que serían útiles para los usuarios ya que dichas recomendaciones están basadas en la confianza que sienten hacia otros usuarios.

Por todo ello nuestra propuesta como futura mejora es que se recopilen datos de manera online, una vez puesto en marcha el sistema, y así poder incrementar la viabilidad del proyecto.

CAPÍTULO 5

BIBLIOGRAFÍA

[**AMAZON**] Amazon Web Services - <http://aws.amazon.com/es/>

[**APRESW**] 1st International Workshop on Adaptation, Personalization and REcommendation in the Social-semantic Web (APRESW 2010)

[**ARTZ, D.**] Artz, D., and Gil, Y. A survey of trust in computer science and the semantic web. *Web Semant.* 5 (2007), 58-71.

[**BALABANOVIC, M.**] Balabanovic, M. and Shoham, Y. (1997), "Content-based, collaborative recommendation". *Communications of the ACM*, Vol.40.

[**BURKE, R.**] Burke, R. "Hybrid recommender systems: Survey and experiments". *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[**CHANG, H.**] Chang, H. and Wong, K. "Adoption of e-procurement and participation of e-marketplace on firm performance: Trust as a moderator". *Inf. Manage.* 47 (2010), 262-270.

[**CHO, Y.H.**] Cho, Y.H. and Kyeong Kim, J. "A personalized recommender system based on Web usage mining and decision tree induction". *Expert Systems with Applications*, Volume 23, Issue 3, 1 October 2002, Pages 329-342.

[**CLAYPOOL, M.**] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. and 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.

[**DAWSON, C.W.**] Dawson, C. W. y Martin, G. (2002), "El proyecto fin de carrera en Ingeniería Informática: Una guía para el estudiante" Ed. Prentice Hall

[**GALAN, S.M.**] Galán Nieto, Sergio Manuel. *Filtrado Colaborativo y Sistemas de Recomendación*

[**GOLBECK, J.A.**] Golbeck, J. A. *Computing and applying trust in web-based social networks*. PhD thesis, 2005

[**GUHA, R.**] Guha, R., Kumar, R., Raghavan, P. and Tomkins, A. "Propagation of trust and distrust". In *Proceedings of the 13th international conference on World Wide Web (2004)*, WWW '04, pp. 403-412.

de una red de confianza

[**GUO, X.**] Guo, X. (2006), Personalized Government Online Services with Recommendation Techniques, tesis Phd, University Graduate School, University of Technology Sydney.

[**FALCONE, R.**] Falcone, R., Pezzulo, G. and Castelfranchi, C. “A fuzzy approach to a belief-based trust computation”. In Proceedings of the 2002 international conference on Trust, reputation, and security: theories and practice (2003), AAMAS'02, pp. 73-86.

[**JUNG, K.Y.**] Jung, K.Y., Choi, J.H., Rim, K.W. and Lee, J.H., “Development of Design Recommender System Using Collaborative Filtering”, Digital libraries: technology and management of indigenous knowledge for global access, Vol. 2911, 2003

[**LENG JAVA**] Lenguaje Java [Http://www.oracle.com/technetwork/java/javase/overview/index-jsp-136246.html](http://www.oracle.com/technetwork/java/javase/overview/index-jsp-136246.html)

[**MAIMON, O.**] Maimon, O. and Rokach, L. (Eds.): The Data Mining and Knowledge Discovery Handbook. Springer 2005, ISBN 0-387-24435-2.

[**MASSA, P.**] Massa, P. and Avesani, P. “Trust-aware recommender systems”. In Proceedings of the 2007 ACM conference on Recommender systems (2007), RecSys '07, pp. 17-24.

[**MURACH, J.**] Murach, J. and Steelman, A. “Murach's Java Servlets and JSP”, 2nd Edition

[**MYSQL**] MySQL - <http://www.mysql.com/> (Último acceso: 09/11/2010)

[**PAPAGELISA, M.**] Papagelisa, M. and Plexousakisa, D. “Qualitative analysis of user-based and item - based prediction”, Engineering Applications of Artificial Intelligence 18 (2005) 781–789

[**SCHALKOFF, R.J.**] Schalkoff, Robert J. “Intelligent Systems: Principles, Paradigms and Pragmatics”. 2009 J & B Publishers.

[**SEMANT**] Semantic Web - <http://www.w3.org/standards/semanticweb/>

[**SINHA, R.R.**] Sinha, R. R. and Swearingen, K. “Comparing Recommendations Made by Online Systems and Friends”. In DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries (2001).

[**TOMCAT**] Apache Tomcat - <http://tomcat.apache.org/> (Último acceso: 09/11/2010)

[UNGAR, L.H.] Ungar, L. H. and Foster, D. P. (1998) “Clustering Methods for Collaborative Filtering”. In Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence.

[VICTOR, P.] Victor, P. Trust Networks for Recommender System. PhD thesis, 2010.

APÉNDICE A

MANUAL DE INSTALACIÓN

de una red de confianza

En este manual vamos a explicar cómo instalar todo lo necesario, para poder montar nuestra aplicación. Vamos a mostrar paso a paso cómo se ha llevado a cabo la instalación de los distintos elementos necesarios. Éstos son servidor Apache, PHP, MySQL, phpMyAdmin (que es una interfaz que nos permite acceder a la base de datos), la máquina virtual java y por último NeatBeans.

Para instalarlos de manera cómoda y rápida hemos optado por usar un paquete de instalación software (de libre distribución), que es AppServ, en el que se incluyen Apache, PHP, MySQL, phpMyAdmin. Este es muy fácil de instalar y ya nos deja todo configurado para poder empezar a utilizarlo. La máquina virtual java y Netbeans lo instalaremos después, ya que no van incorporados en el paquete anterior.

Por último, antes de empezar, debemos comentar que este manual está orientado a ordenadores con sistemas operativos Windows.

A.1. INSTALACIÓN DE APPSERV

Este es un paquete de libre distribución, como ya hemos comentado, que se puede descargar de la página web <http://www.appservnetwork.com/index.php?newlang=spanish>

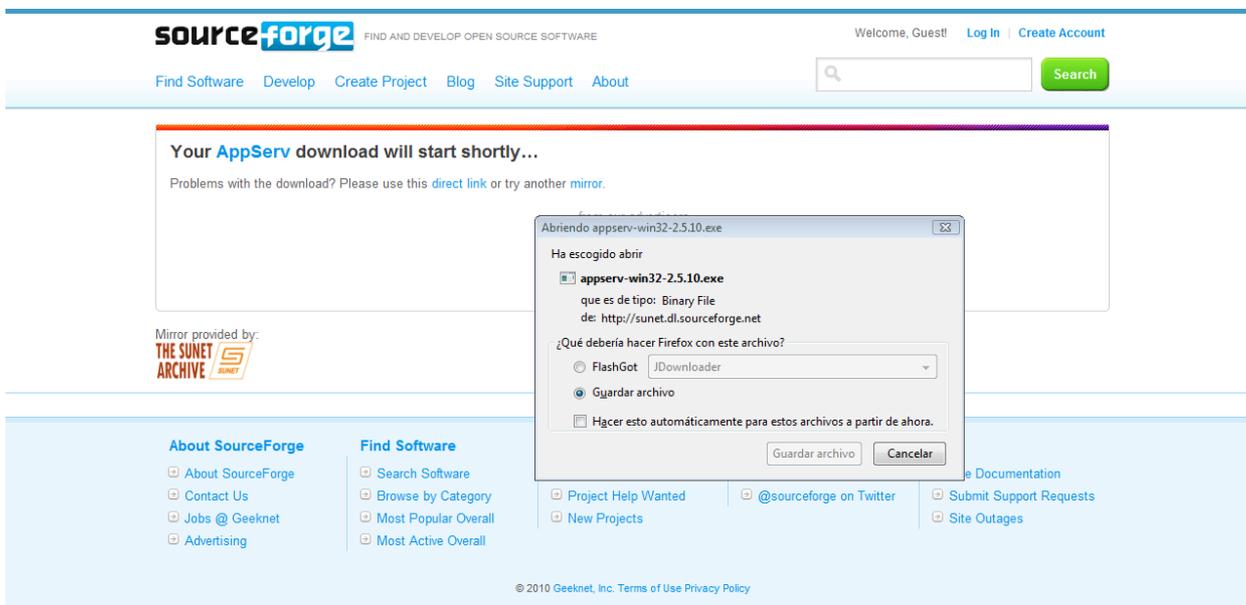
The screenshot shows the AppServ Open Project website. The main menu includes links for Home, Downloads, AddOns, Community, FAQ, Stats Activity, Shopping, and AJAX Script. The main content area features a 'Main Menu' with links like 'What is AppServ?', 'How to install AppServ', and 'Download Now!'. The 'Lastest Release' section highlights 'AppServ 2.6.0' and 'AppServ 2.5.10' with their respective download links and MD5SUMs. The 'Web Link' section promotes 'AppServ 2.5.9 & 2.4.9 with Zend Optimizer AddOns Released!'. The right sidebar contains a 'Languages' dropdown set to Spanish, a 'Donate to AppServ' section with PayPal buttons for \$5, \$10, \$15, and \$20, and a 'Google ads' section.

de una red de confianza

Nosotros nos hemos descargado la versión AppServ 2.5.10, que es el que más se adapta a nuestras necesidades, y que contiene:

- Apache 2.2.8
- PHP 5.2.6
- MySQL 5.0.51b
- phpMyAdmin-2.10.3

Para llevar a cabo la descarga sólo hay que pulsar en el enlace de Download y nos aparecerá la siguiente página:



Luego procedemos a guardar el ejecutable (donde queramos), dándole al botón Guardar archivo y ya lo tendremos listo para empezar la instalación.

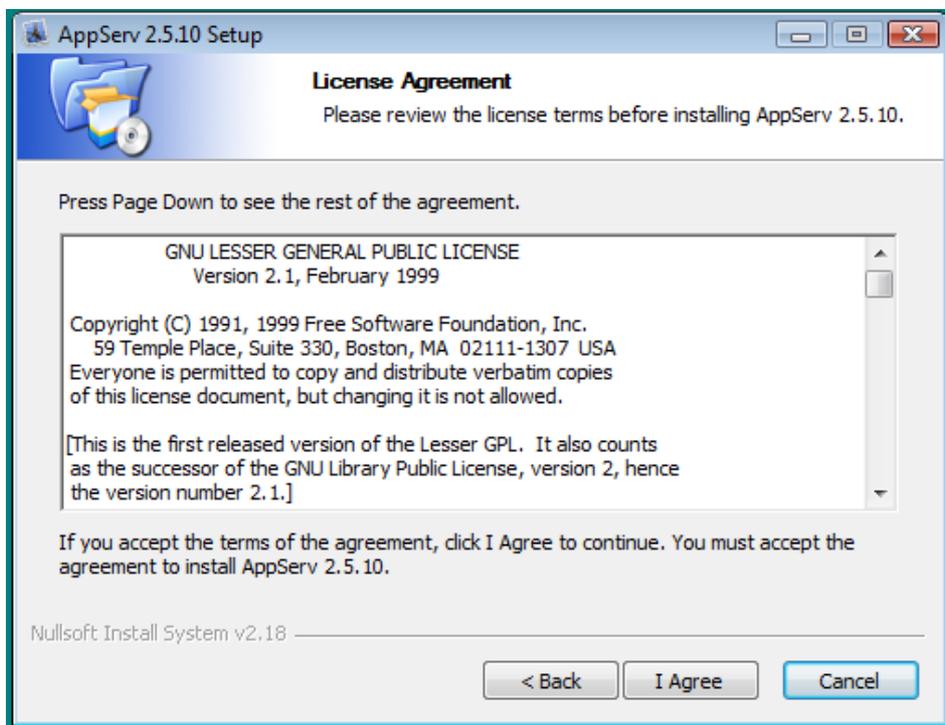
Nos vamos a la ruta de nuestro ordenador donde lo hemos guardado y hacemos doble click en el ejecutable, entonces nos aparecerá una pantalla como la siguiente (**¡Error! No se encuentra el origen de la referencia.**):



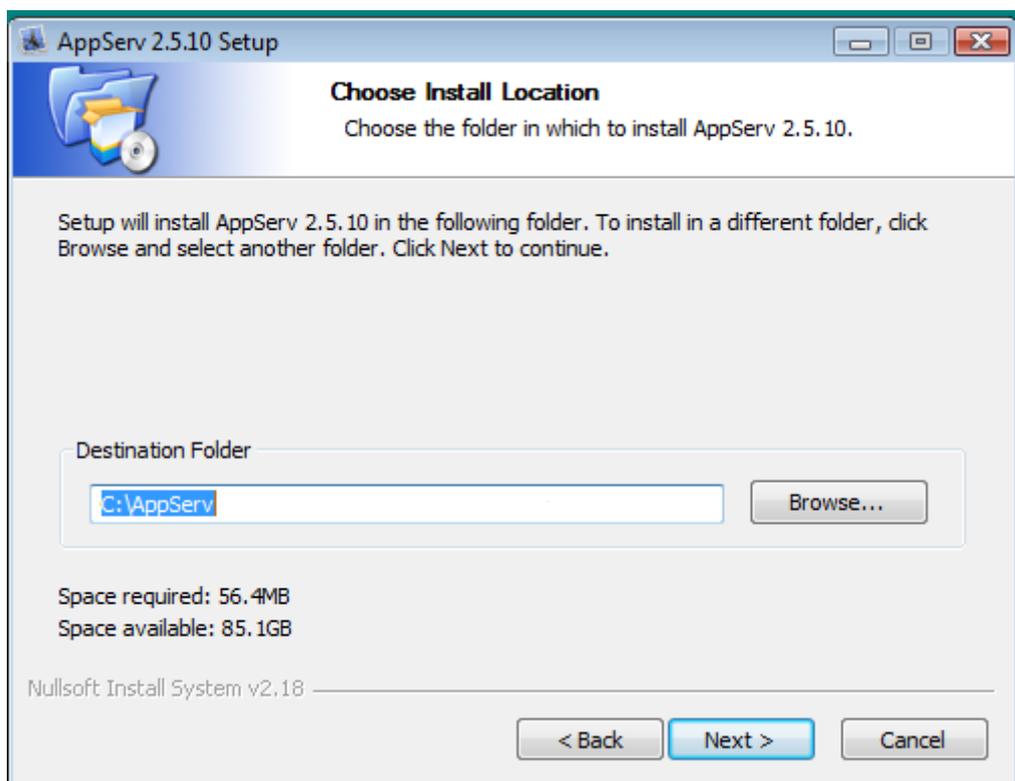
FIGURA A.1 - 3: INICIO INSTALACIÓN

Le damos a Next para comenzar y nos aparecerá el contrato de licencia (**¡Error! No se encuentra el origen de la referencia.**), tras leerlo detenidamente, pulsamos I Agree, si estamos de acuerdo con él, sino pulsamos Cancel para salir de la instalación.

de una red de confianza

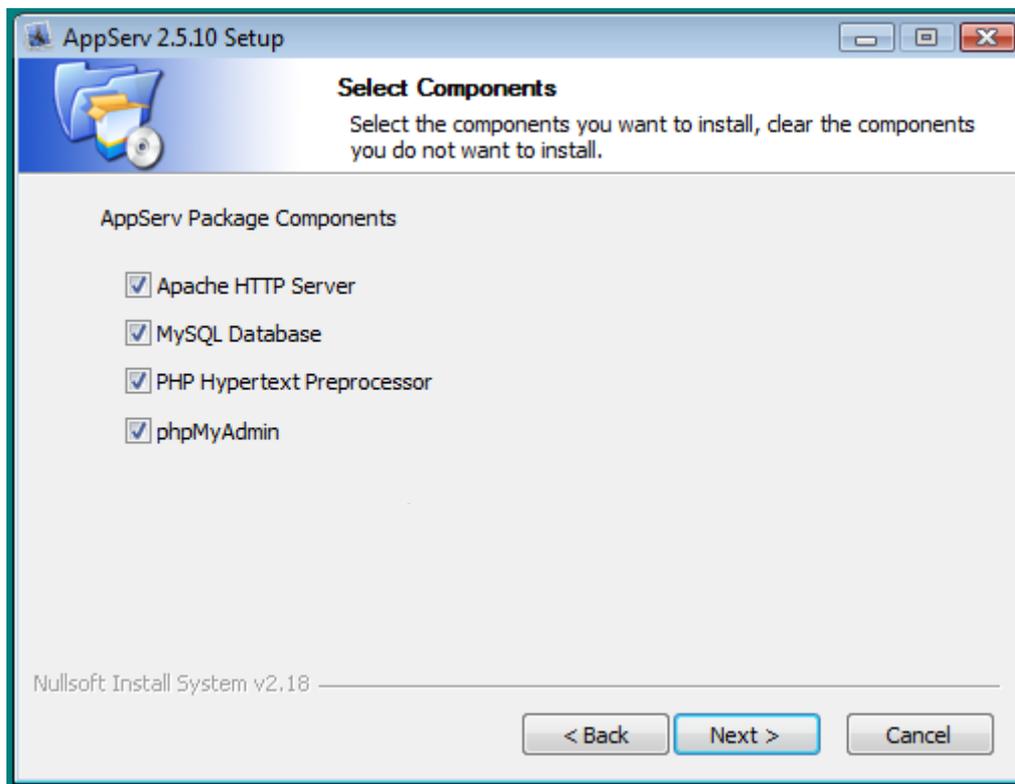


Ahora nos aparecerá el lugar donde el usuario quiere que se lleve a cabo la instalación (**¡Error! No se encuentra el origen de la referencia.**). Por defecto el programa nos da una, si no se está de acuerdo con ella se debe pulsar la tecla Browse... y elegir la que se estime oportuna, nosotros optaremos por dejar la ruta por defecto. Una vez elegida se le debe dar a la tecla Next para continuar.

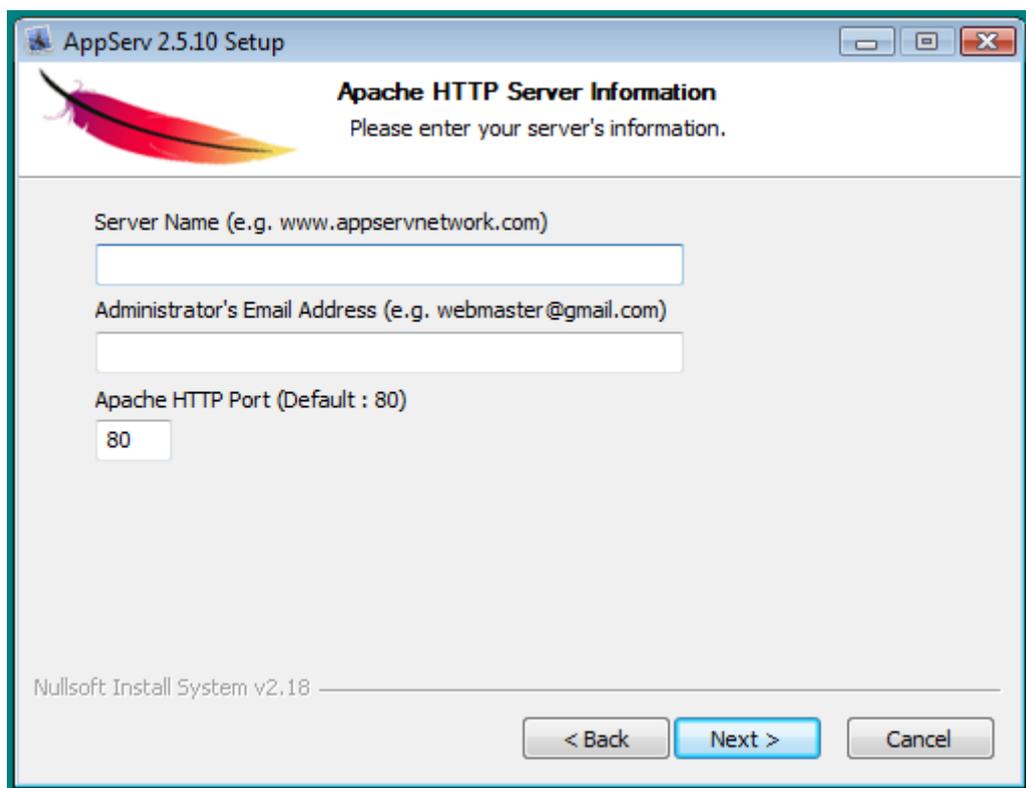


A continuación toca elegir los distintos componentes que se quieren instalar, en nuestro caso son todos, así que habrá que marcar todas las casillas como muestra la **¡Error! No se encuentra el origen de la referencia.:**

de una red de confianza

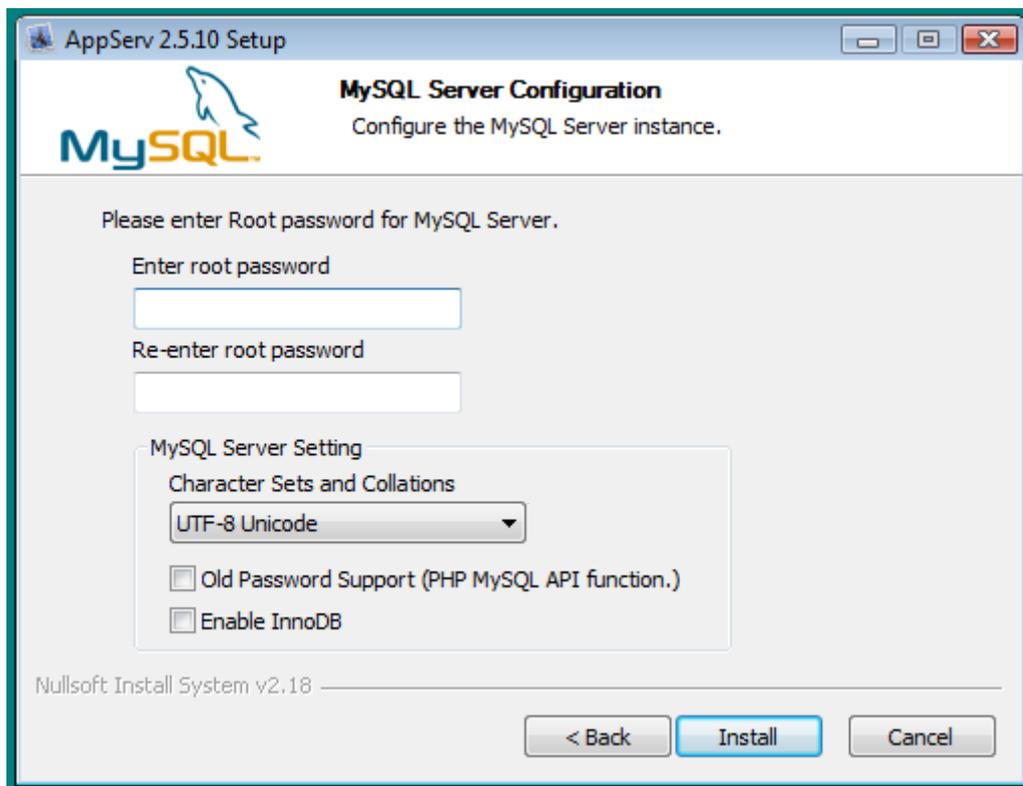


Se pulsa Next y nos aparecerá la pantalla de configuración del servidor Apache (**¡Error! No se encuentra el origen de la referencia.**). Como puede apreciarse nos pide muy pocos campos, ya que AppServ se encarga el sólo de configurar la mayoría de los parámetros. Nosotros debemos de preocuparnos de meter el Server Name, que en nuestro caso será localhost, una dirección de correo válida, y el puerto, que dejaremos el que viene por defecto.



Le damos a Next para continuar y nos aparece la pantalla para configurar MySQL Server (**¡Error! No se encuentra el origen de la referencia.**). En ella debemos introducir una clave (dos veces por seguridad), que debemos recordar, ya que es con la que podremos acceder a la base de datos. El resto de parámetros los dejamos como vienen por defecto.

de una red de confianza



Por último tras pulsar Install se llevará a cabo la instalación de los componentes que antes hemos elegido, y nos aparecerá la ventana de finalización (**¡Error! No se encuentra el origen de la referencia.**).

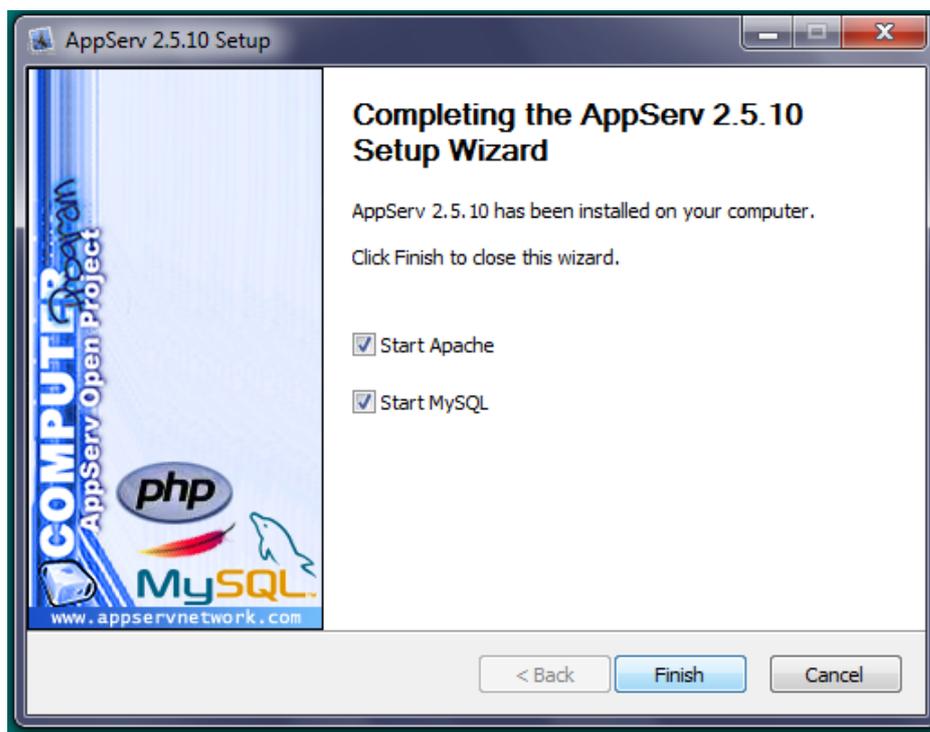


FIGURA A.1 - 9: FINALIZACIÓN DE LA INSTALACIÓN

Se nos pregunta si deseamos iniciar la ejecución de Apache y MySQL, dejamos marcadas las opciones por defecto y pulsamos la opción Finish.

Ya hemos terminado la instalación de las aplicaciones y se ha empezado su ejecución, por lo que ya podemos acceder a ellas. Nosotros lo vamos a hacer desde el navegador (**¡Error! No se encuentra el origen de la referencia.**), para ello pondremos en la barra de dirección de éste, la dirección local de nuestra máquina, que es localhost en la mayoría de los casos (si no nos funciona se puede probar, en lugar de con el nombre, con la dirección, que es 127.1.1.0 o con la dirección de nuestra IP (**¡Error! No se encuentra el origen de la referencia.**), que puede mirarse en el terminal poniendo ipconfig).

de una red de confianza

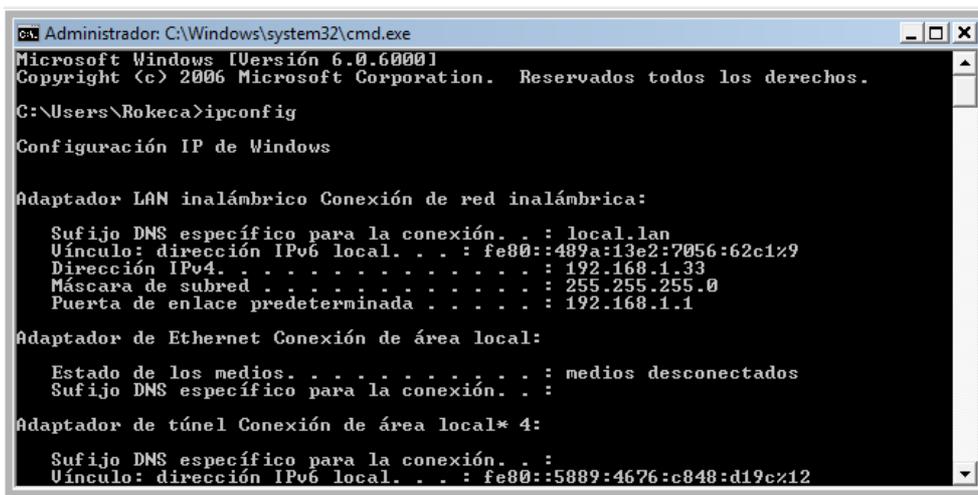


FIGURA A.1 - 10: TERMINAL

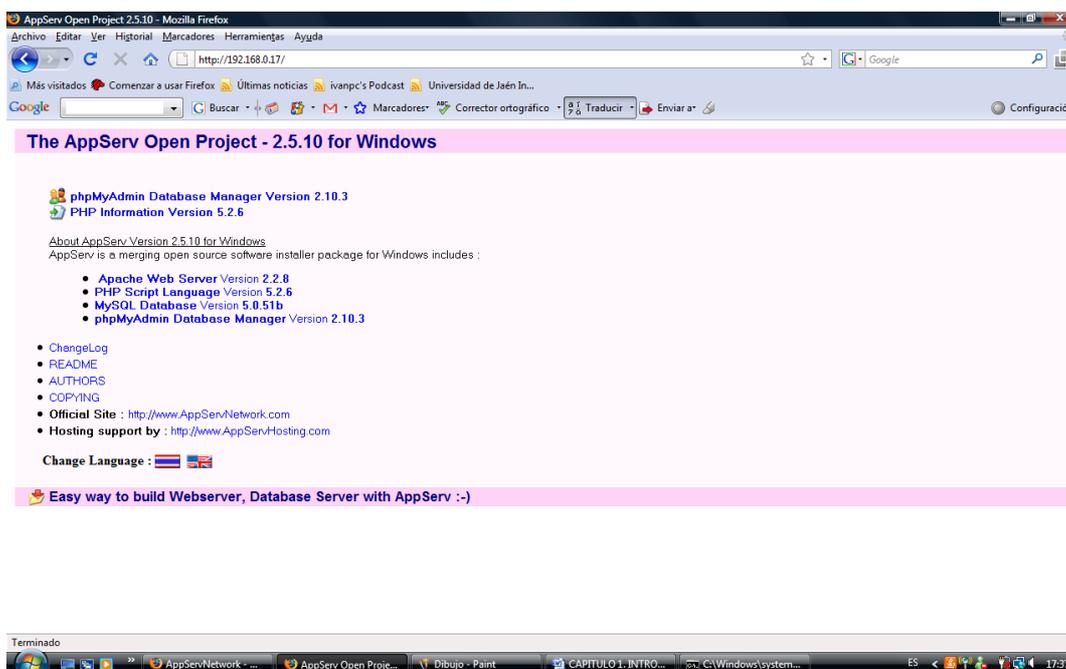


FIGURA A.1 - 11: INICIO DE LA BASE DE DATOS DESDE EL NAVEGADOR

Pinchando en phpMyAdmin Database Mannager, nos aparece una ventana pidiendo el nombre y la contraseña, introducimos en el nombre root y en la contraseña la que pusimos durante la instalación, y al darle a Aceptar entraremos en la base de datos (**¡Error! No se encuentra el origen de la referencia.**).

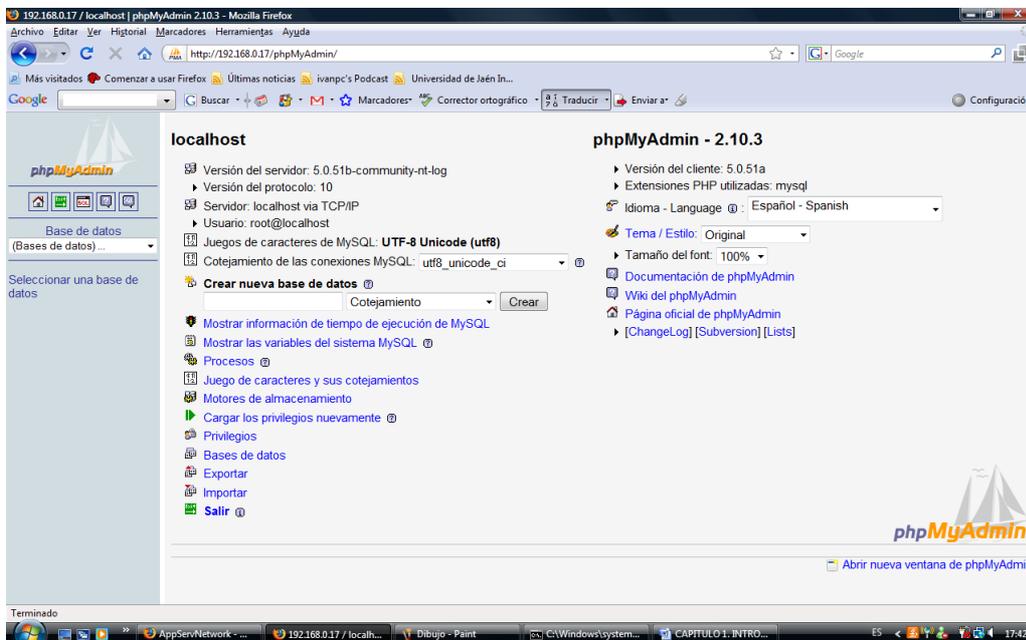


FIGURA A.1 - 12: PATALLA INICIO DE LA BASE DE DATOS

Lo que debemos hacer ahora es crear una nueva base de datos, en nuestro caso la hemos llamado “sr”. Nosotros hemos hecho una copia de seguridad de los datos con los hemos trabajado en los ficheros ratings2.csv y películas.csv, así que esos son los ficheros que importamos como se observa en la **¡Error! No se encuentra el origen de la referencia.**

de una red de confianza

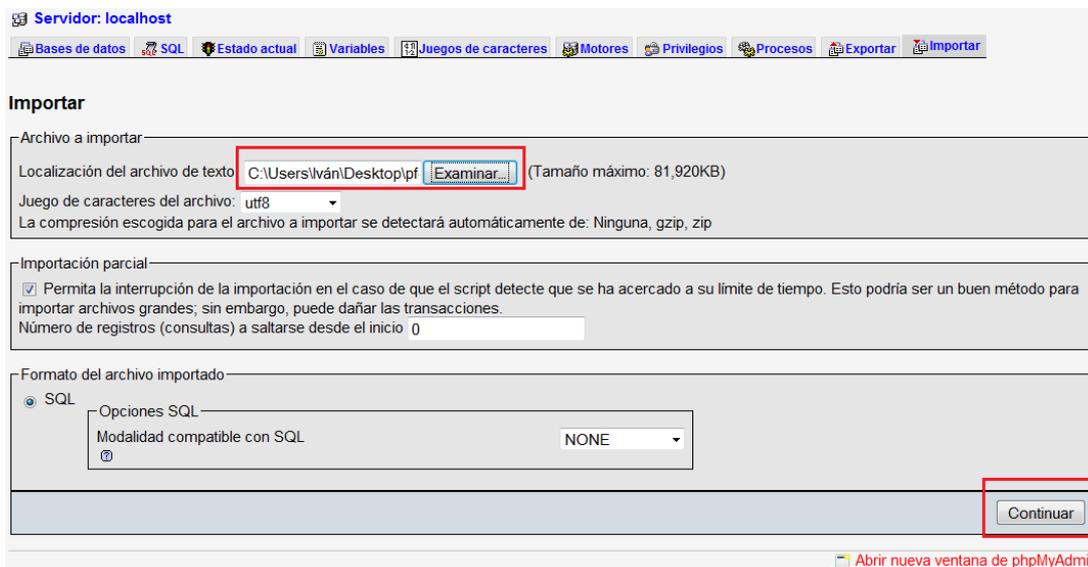


FIGURA A.1 - 13: IMPORTAR LA BASE DE DATOS

A.2. INSTALACIÓN DE LA MÁQUINA VIRTUAL

La instalación de la máquina virtual java es muy sencilla, y nos va a permitir ejecutar diversidad de aplicaciones hechas en java. Para llevarla a cabo nos debemos dirigir al lugar web, donde nos descargaremos gratuitamente el ejecutable, ya que es código abierto, que es <http://www.java.com/es/download/> (**¡Error! No se encuentra el origen de la referencia.**)

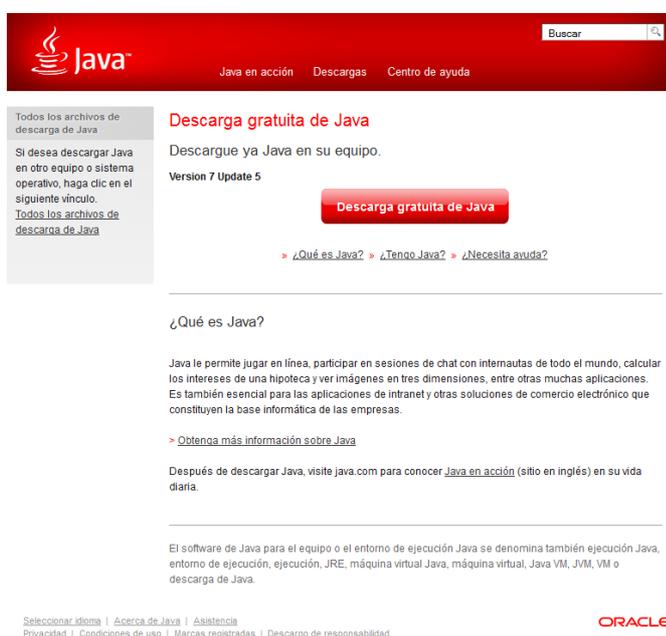


FIGURA A.2 - 1: DESCARGA DE LA MÁQUINA VIRTUAL JAVA

Tras descargárnoslo lo ejecutaremos, para comenzar con la instalación y luego hay que aceptar los términos de licencia y darle a continuar en las ventanas que sea necesario (**¡Error! No se encuentra el origen de la referencia.**)

de una red de confianza



FIGURA A.2 - 2: INSTALACIÓN DE JAVA

Ahora pasaremos a descargarnos NetBeans con JDK, que lo podemos encontrar en la siguiente dirección <http://www.oracle.com/technetwork/java/javase/downloads/index.html> (**Error! No se encuentra el origen de la referencia.**) e instalarlo al igual que hemos hecho con la maquina virtual, siguiendo unas sencillas ventanas.

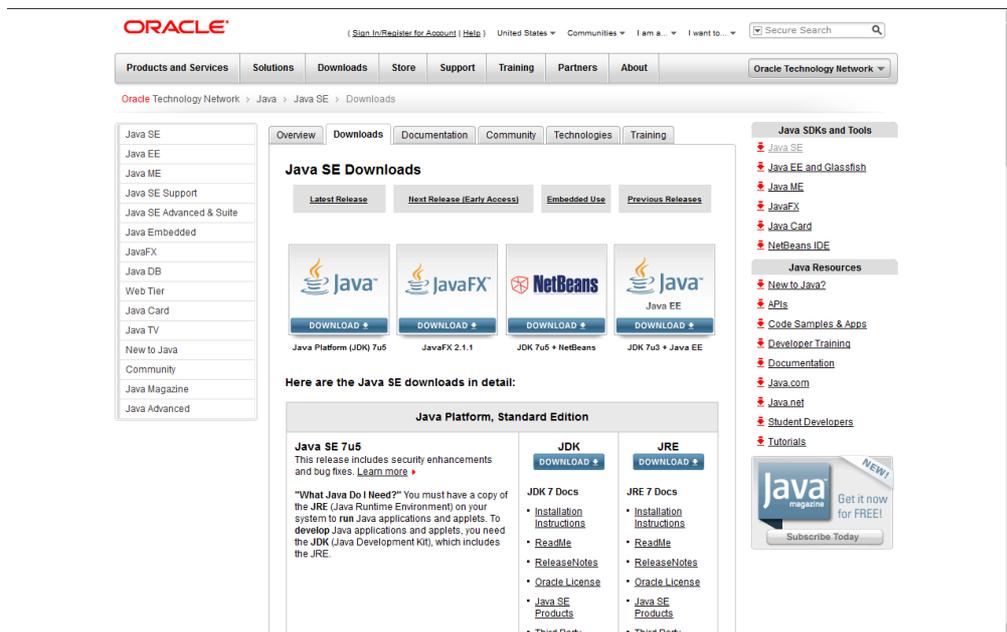


FIGURA A.2 - 3: DESCARGA DE NETBEANS

Ya tenemos todo lo que necesitamos para poder ejecutar nuestra aplicación, sólo nos queda incluir la librería JDBC de forma que sea accesible por el entorno de desarrollo. Para llevarlo a cabo tenemos que descargarnos un archivo que podemos encontrar en la página de MySQL <http://dev.mysql.com/downloads/connector/j/>, que es mysql-connector-java, y adjuntarlo en nuestro proyecto NetBeans como una biblioteca. Otra opción sino queremos adjuntarlo es copiar el archivo en la ruta donde está el JDK que antes hemos instalado, en concreto es la siguiente C:\Program Files\Java\jdk1.6.0_21\jre\lib\ext (**Error! No se encuentra el origen de la referencia.**).

de una red de confianza

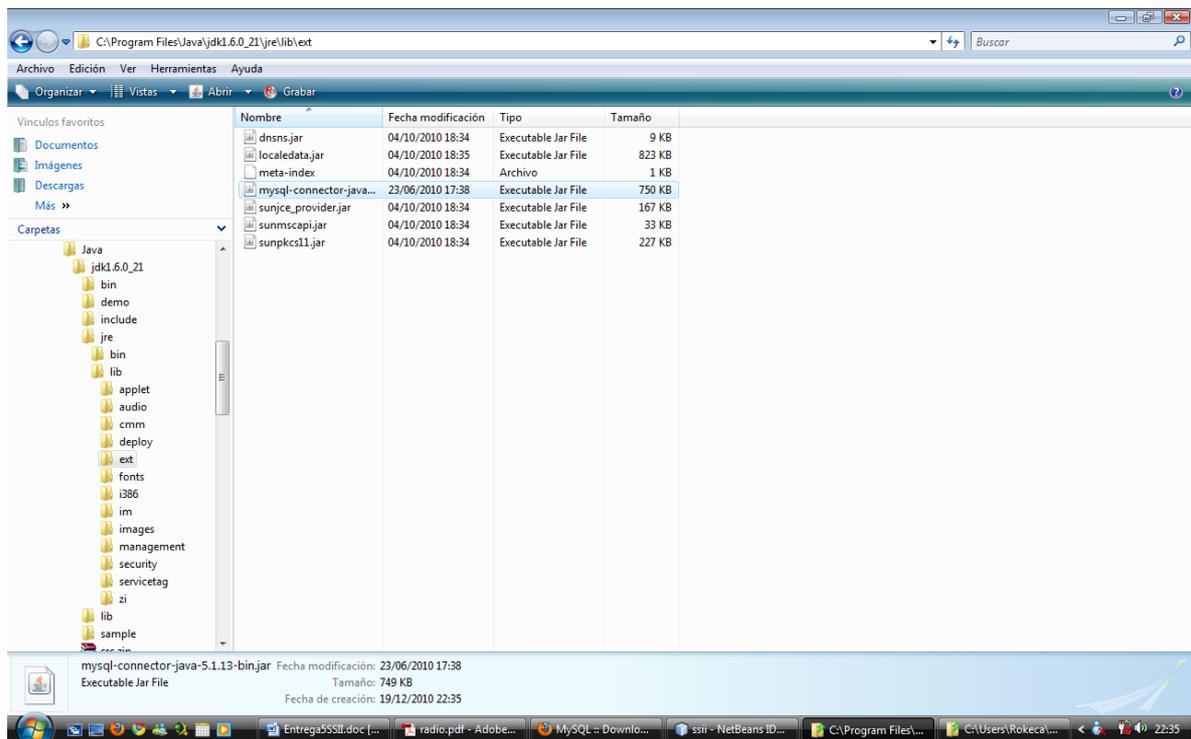


FIGURA A.2 - 4: AGREGACIÓN DEL CONECTOR MYSQL

APÉNDICE B

MANUAL DE USUARIO

de una red de confianza

BEST PICTURE es una aplicación que se encarga de realizar recomendaciones de películas basándose en la confianza entre usuarios. Se trata de una aplicación web a la que tienen acceso usuarios sin registrar (ya que el sistema permite registrarse) o registrados con anterioridad. BEST PICTURE permite:

- Iniciar Sesión a usuarios.
- Registrar Usuarios.
- Proporcionar recomendaciones y poder puntuarlas.
- Ver las valoraciones que un usuario ha realizado hasta el momento.
- Ver los valores de confianza que tiene con los usuarios.

Vamos a exponer las acciones que definen nuestro sistema y como el usuario interactúa con las mismas:

PORTADA

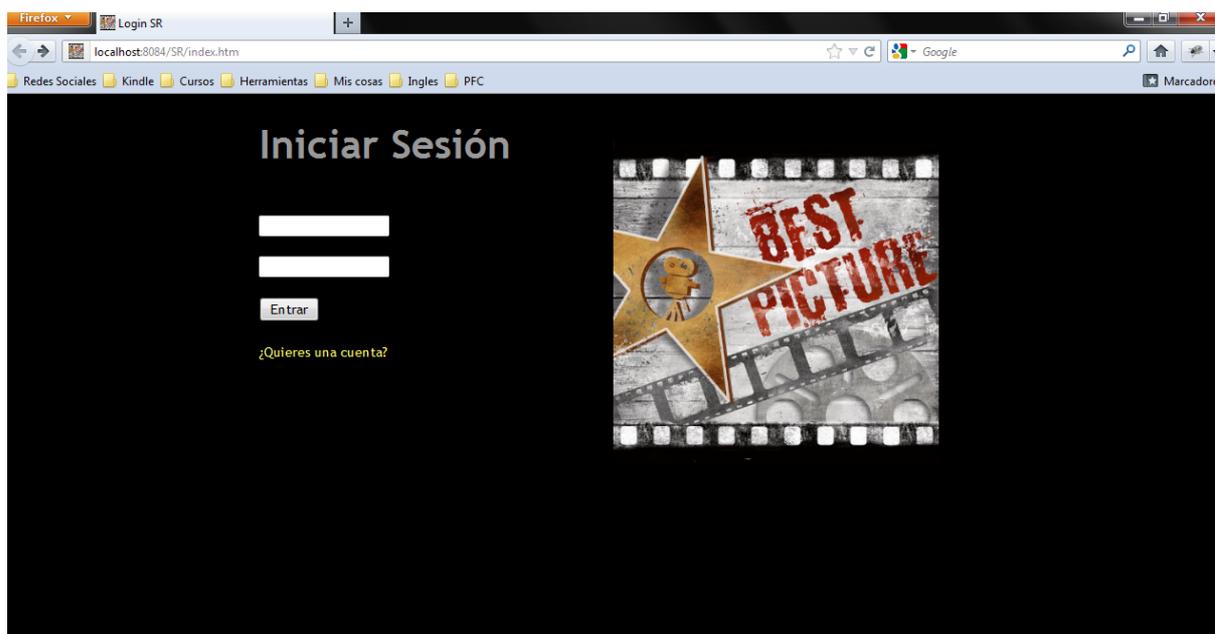


FIGURA B - 1: PORTADA

El usuario accede a la portada (**¡Error! No se encuentra el origen de la referencia.**). En ella puede hacer dos cosas: Registrar Usuarios (Se explicará posteriormente) o Iniciar Sesión a usuarios. Para iniciar sesión el usuario teclea su usuario y contraseña, y pulsa el botón “Entrar”. Si el sistema lo

valida correctamente, se irá a la página principal (**¡Error! No se encuentra el origen de la referencia.**), sino se mostrará un mensaje de error indicando que el usuario o clave son incorrectos (**¡Error! No se encuentra el origen de la referencia.**).

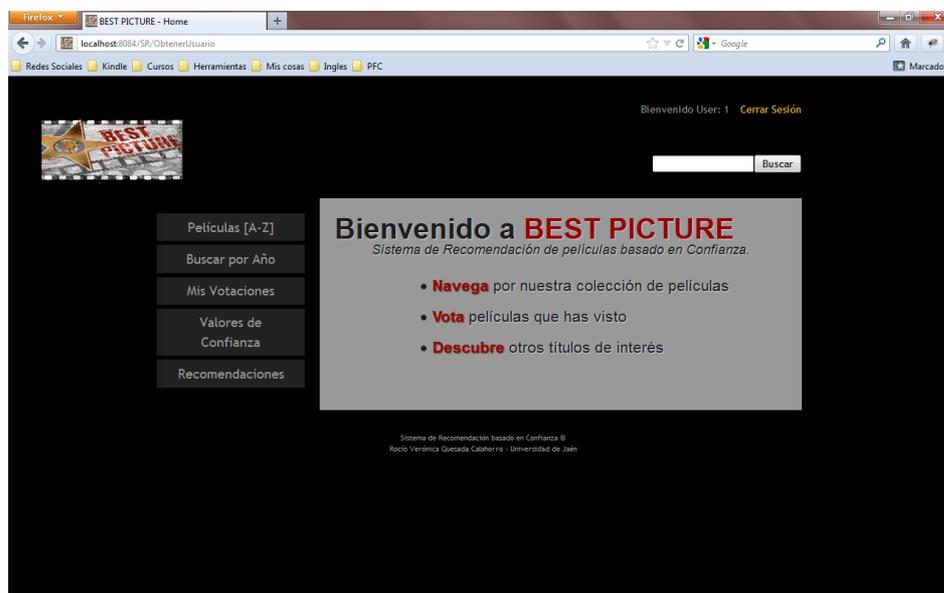


FIGURA B - 2: PÁGINA PRINCIPAL

Aquí en la página principal (**¡Error! No se encuentra el origen de la referencia.**) podemos ver las distintas secciones de las que dispone la aplicación, el mensaje de bienvenida, el botón de cerrar sesión y la cabecera con el nombre de la aplicación. Cabe destacar que si pinchamos en la cabecera, estemos donde estemos, se nos redirige a la página principal.

de una red de confianza

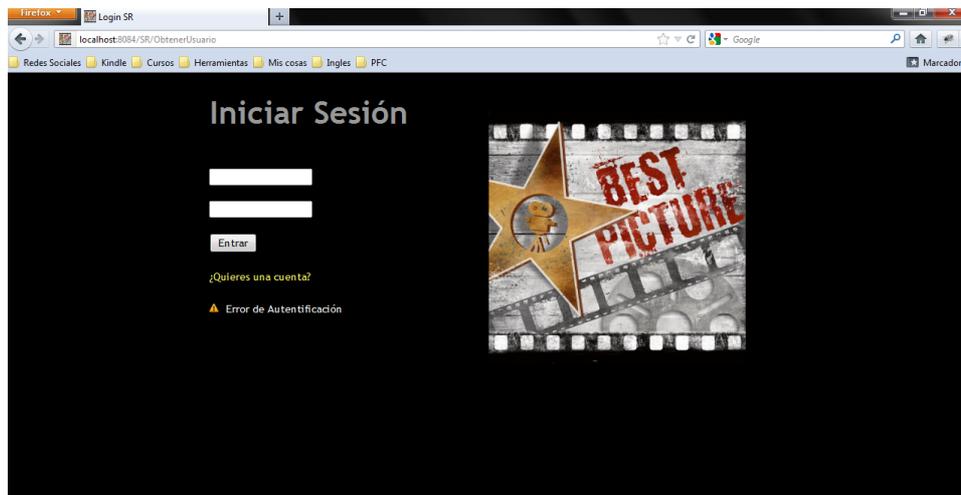


FIGURA B - 3: ERROR DE AUTENTICACIÓN

REGISTRAR USUARIOS

En la **¡Error! No se encuentra el origen de la referencia.** hemos podido observar que existe un botón llamado “¿Quieres una cuenta?”. El usuario lo pulsará e irá a la pantalla de Registro de usuario **¡Error! No se encuentra el origen de la referencia.**, donde el usuario enviará un nombre de usuario y un email. El sistema le notificará a dicho email un nuevo usuario y una contraseña para acceder como usuario registrado.

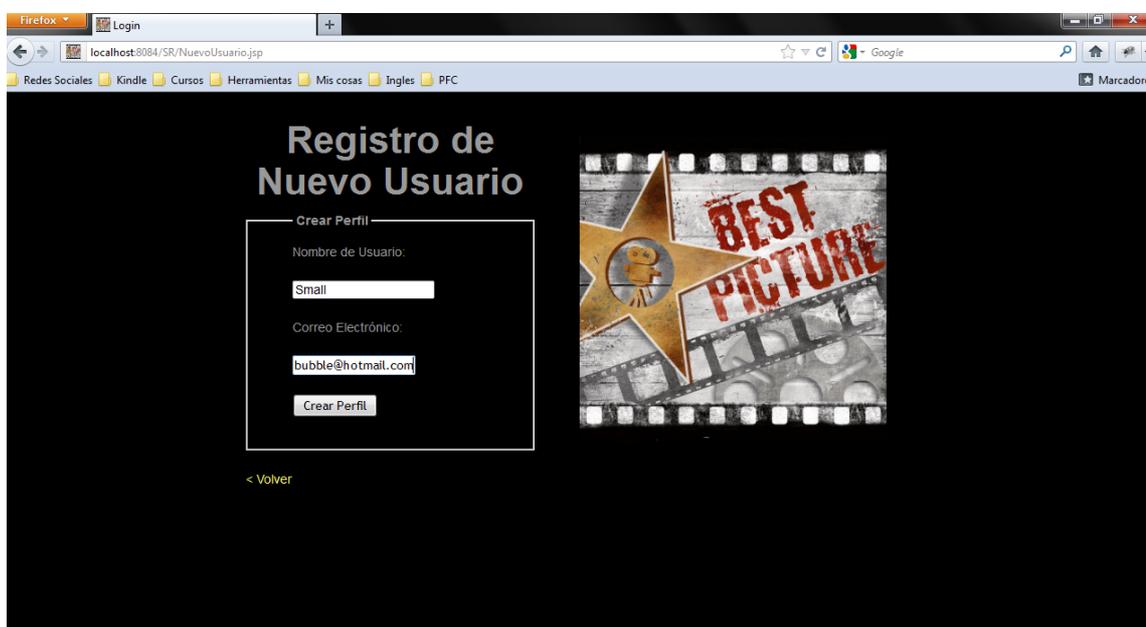


FIGURA B - 4: REGISTRO DE USUARIO

de una red de confianza

Si el registro no se ha llevado a cabo correctamente, porque la cuenta de correo sea incorrecta u omite algún dato, nos saltará un aviso indicando que el registro no se completó (**¡Error! No se encuentra el origen de la referencia.**).



FIGURA B - 5: ERROR DE CREACIÓN DE USUARIO

Si el registro se lleva a cabo de forma correcta vuelve al inicio (**¡Error! No se encuentra el origen de la referencia.**) donde se muestra un aviso que dice que la operación fue un éxito.

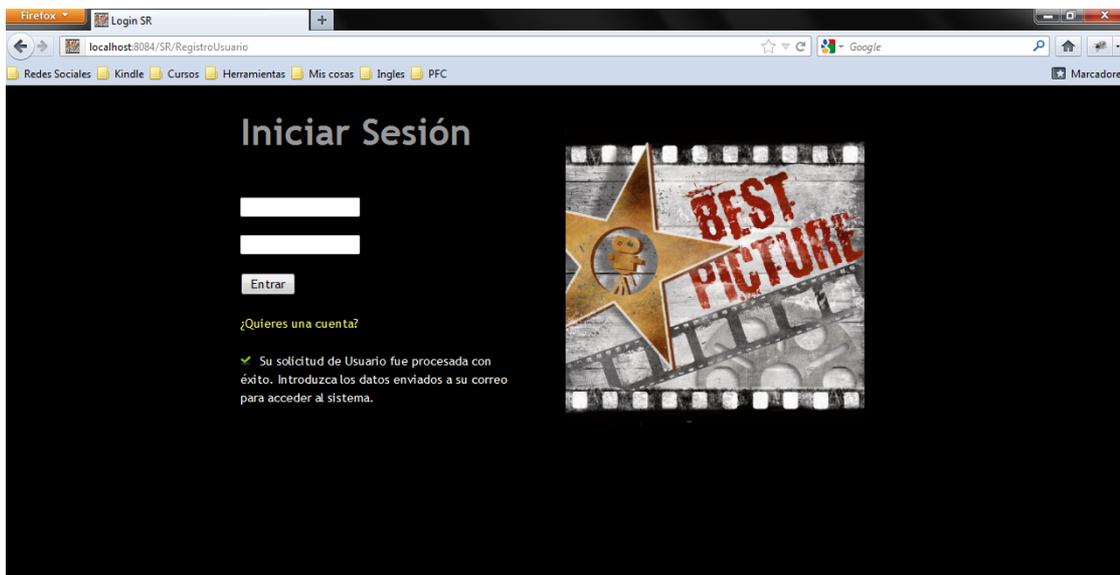


FIGURA B - 6: ÉXITO EN LA CREACIÓN DE USUARIO

Ahora podremos introducir el usuario y contraseña que hemos obtenido en el email (**¡Error! No se encuentra el origen de la referencia.**) que hemos recibido. Si hemos metido bien los datos se nos remitirá a la **¡Error! No se encuentra el origen de la referencia.** como hemos visto anteriormente.

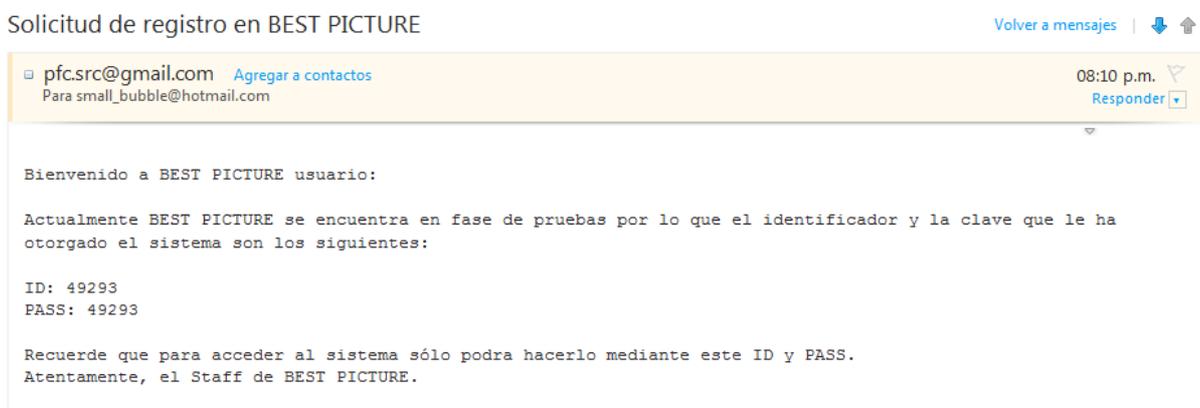


FIGURA B - 7: CORREO DE CONFIRMACIÓN

de una red de confianza

BUSCAR UNA PELÍCULA

Tenemos 3 formas diferentes para realizar una **búsqueda de películas**.

La primera es introducir el título, o parte del título, de la película en la parte de arriba a la derecha (**¡Error! No se encuentra el origen de la referencia.**). El sistema nos devolverá un listado de películas que cumplen esta característica.

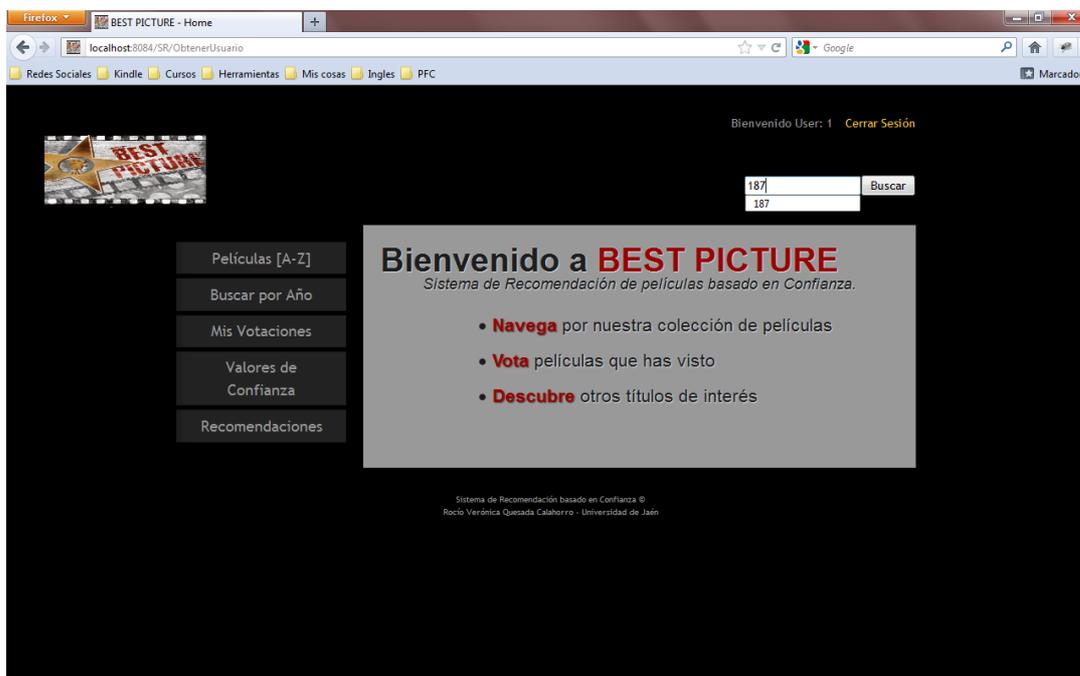


FIGURA B - 8: BÚSQUEDA POR TÍTULO

Hemos introducido la palabra “187” en el buscador. El resultado es el siguiente (**¡Error! No se encuentra el origen de la referencia.**):

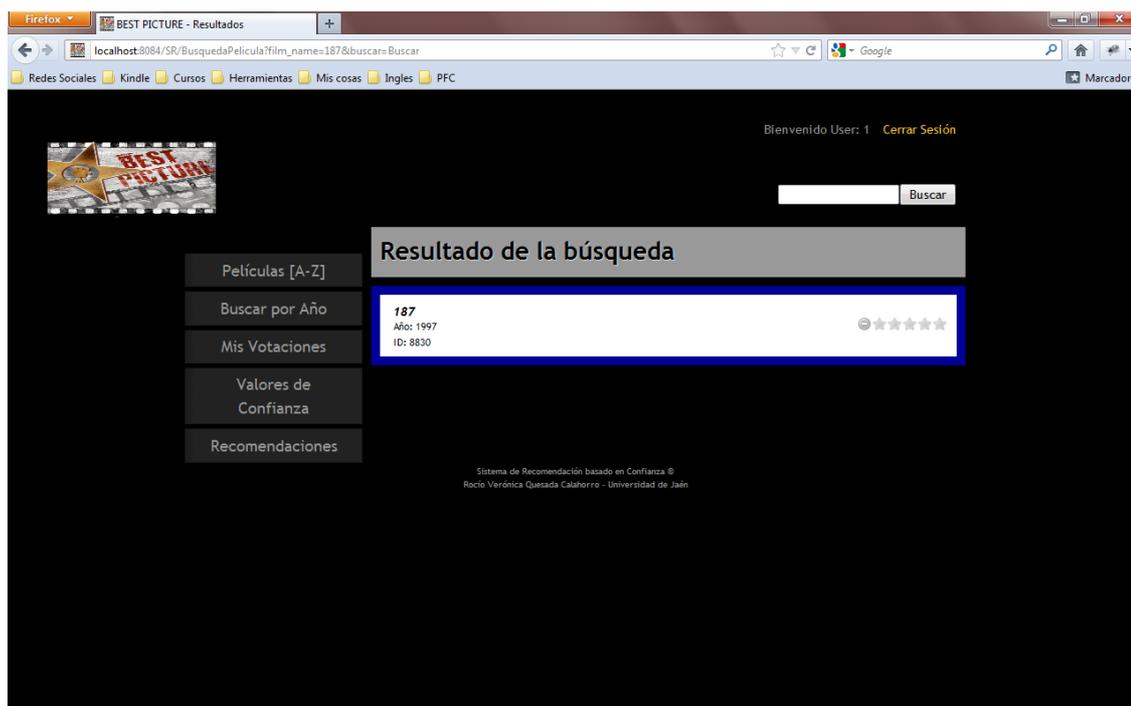


FIGURA B - 9: RESULTADO DE LA BÚSQUEDA POR TÍTULO

A partir de aquí el usuario puede votar una película o si la tiene votada cambiar su votación, utilizando las estrellas que están situadas en la parte derecha de cada película.

de una red de confianza

Si no encontrase ningún resultado nos mostrará un aviso (**Error! No se encuentra el origen de la referencia.**)



FIGURA B - 10: ERROR EN LA BÚSQUEDA POR TÍTULO

Otra forma es buscar una película es, según la letra inicial por la que empiece dicha película. El sistema devuelve una lista de películas que coinciden con la letra seleccionada (**¡Error! No se encuentra el origen de la referencia.**). Igualmente se puede realizar la votación de las películas o la modificación de dicha votación.



FIGURA B - 11: BÚSQUEDA POR INICIAL

de una red de confianza

Como comprobamos en la **¡Error! No se encuentra el origen de la referencia.** al pinchar en una letra nos muestra un listado de películas cuyo título empieza por ella.

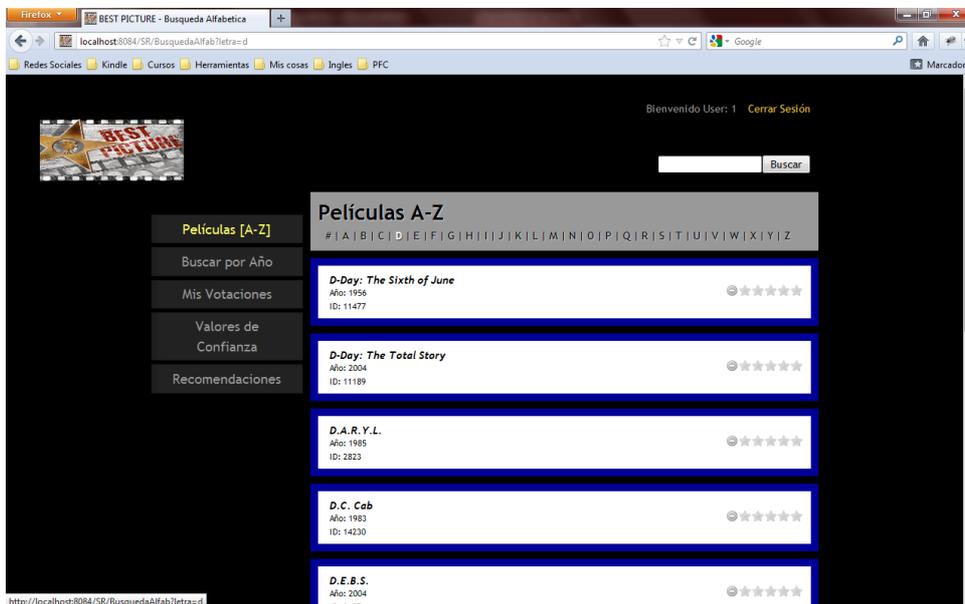


FIGURA B - 12: RESULTADO DE LA BÚSQUEDA POR INICIAL

Además si el listado de películas es muy largo se nos mostrarán varias páginas de películas, y nos podremos desplazar a cada una de ellas gracias a la paginación (**¡Error! No se encuentra el origen de la referencia.**).

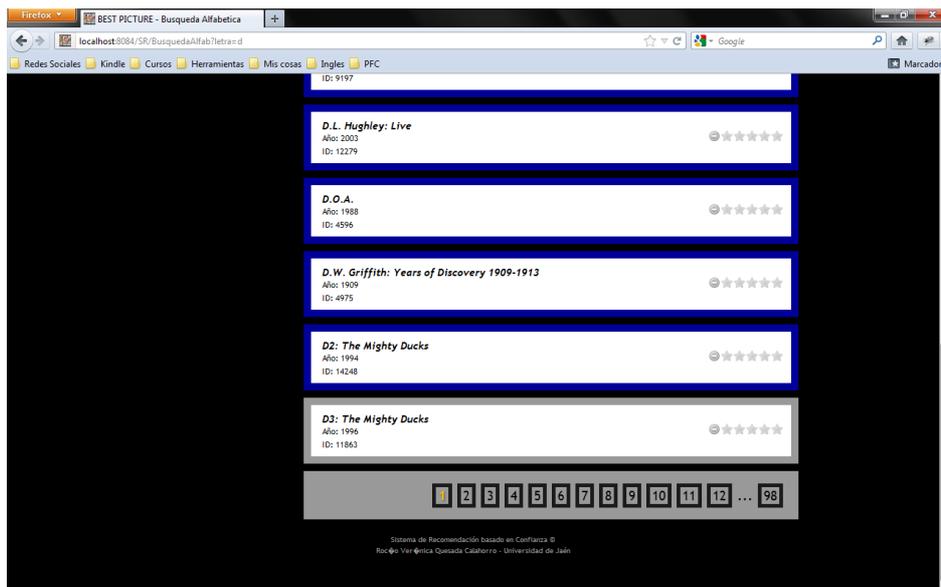


FIGURA B - 13: PAGINACIÓN

La última forma de **buscar una película es por año** (**¡Error! No se encuentra el origen de la referencia.**). El sistema devolverá una lista de películas que coincidan con el año de búsqueda (**¡Error! No se encuentra el origen de la referencia.**).

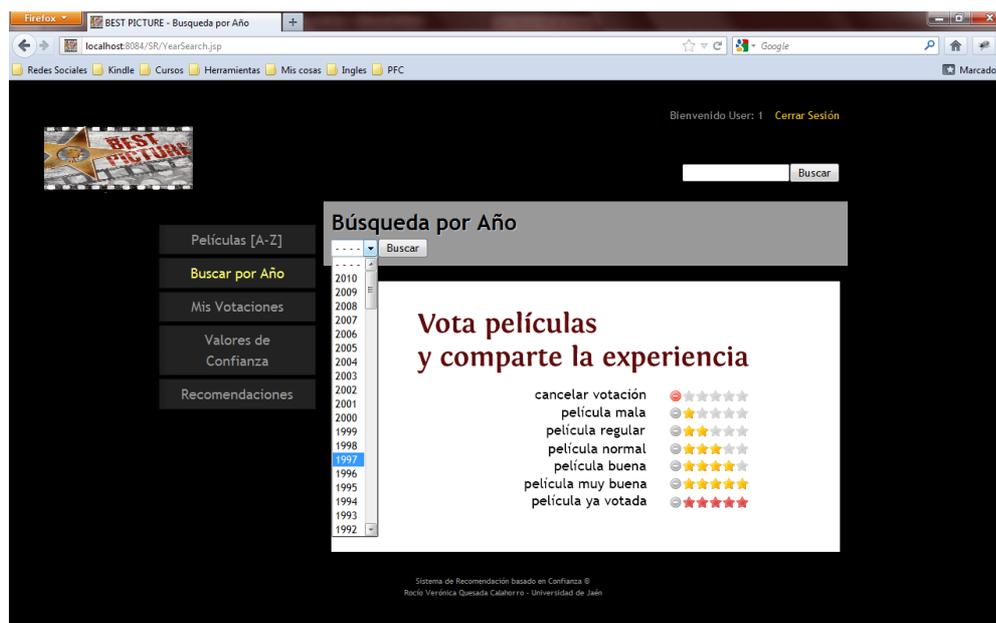


FIGURA B - 14: BÚSQUEDA POR AÑO

de una red de confianza

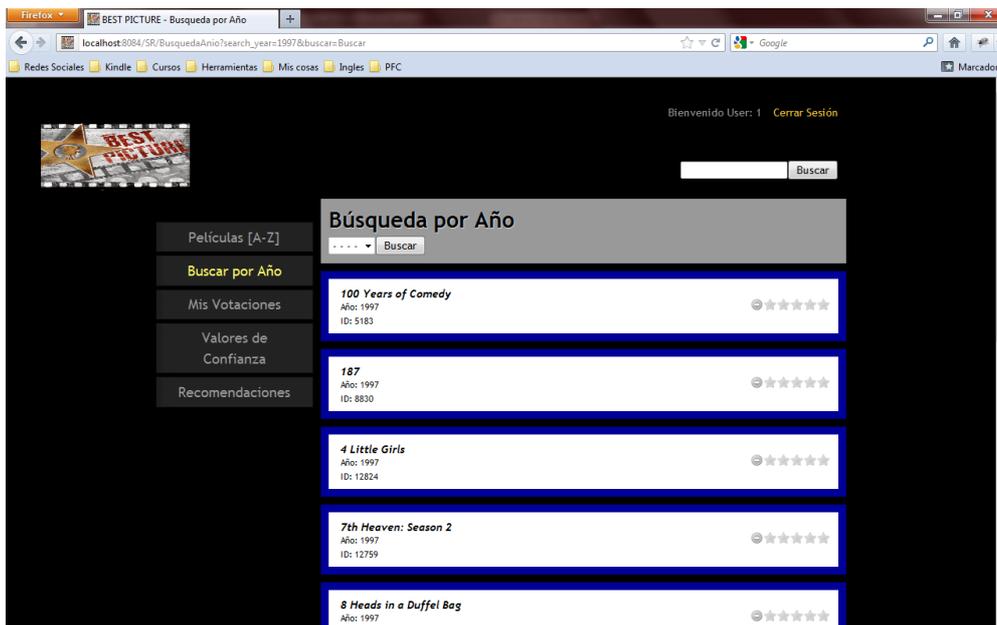


FIGURA B - 15: RESULTADO DE LA BÚSQUEDA POR AÑO

De nuevo el usuario puede modificar o puntuar por primera vez una película.

En el caso de introducir un año en el que no existen película se nos mostrará un aviso (**¡Error! No se encuentra el origen de la referencia.**).

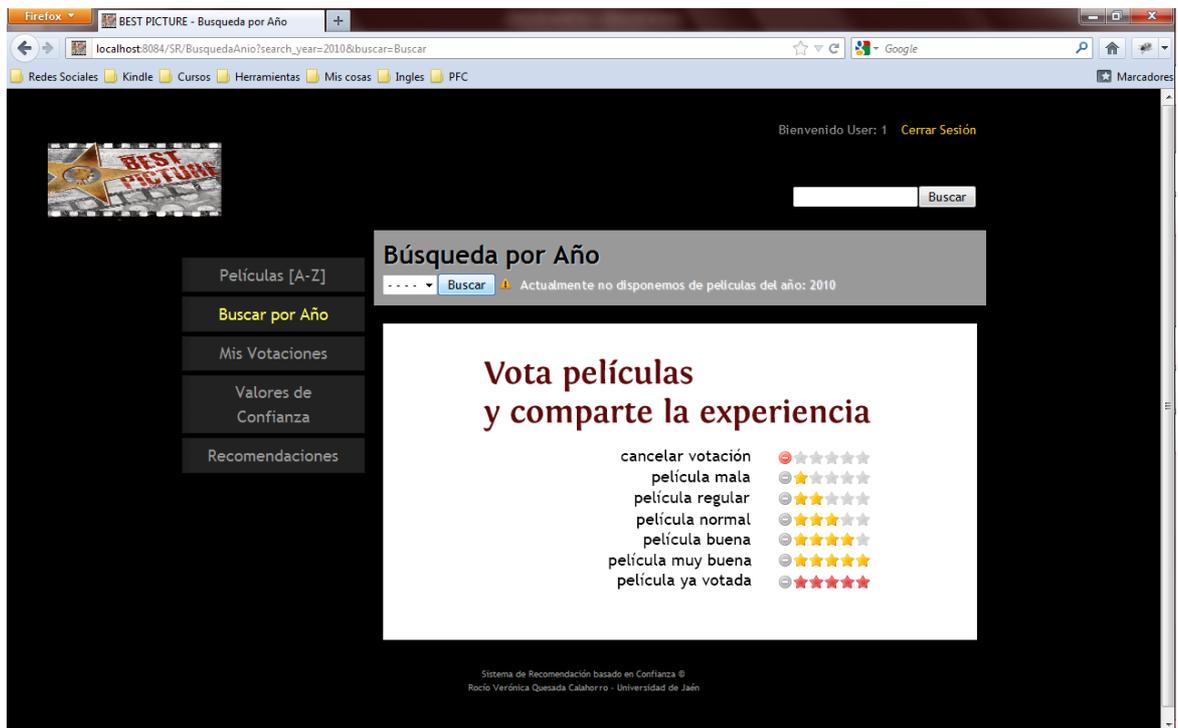


FIGURA B - 16: ERROR 1 EN LA BÚSQUEDA POR AÑO

de una red de confianza

Si no se introduce ningún año y se le da a buscar también se nos mostrará un aviso (**¡Error! No se encuentra el origen de la referencia.**).

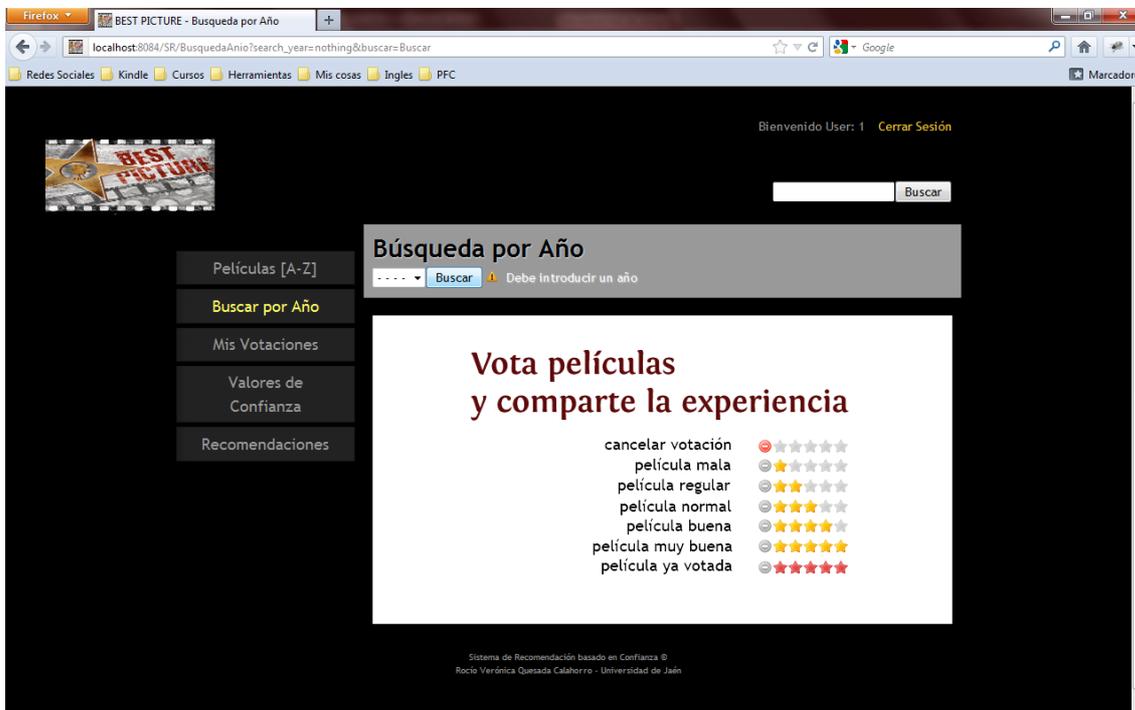


FIGURA B - 17: ERROR 2 EN LA BÚSQUEDA POR AÑO

Otra opción es **listar las películas que hemos votado**. En la sección “Mis votaciones” (**¡Error! No se encuentra el origen de la referencia.**).

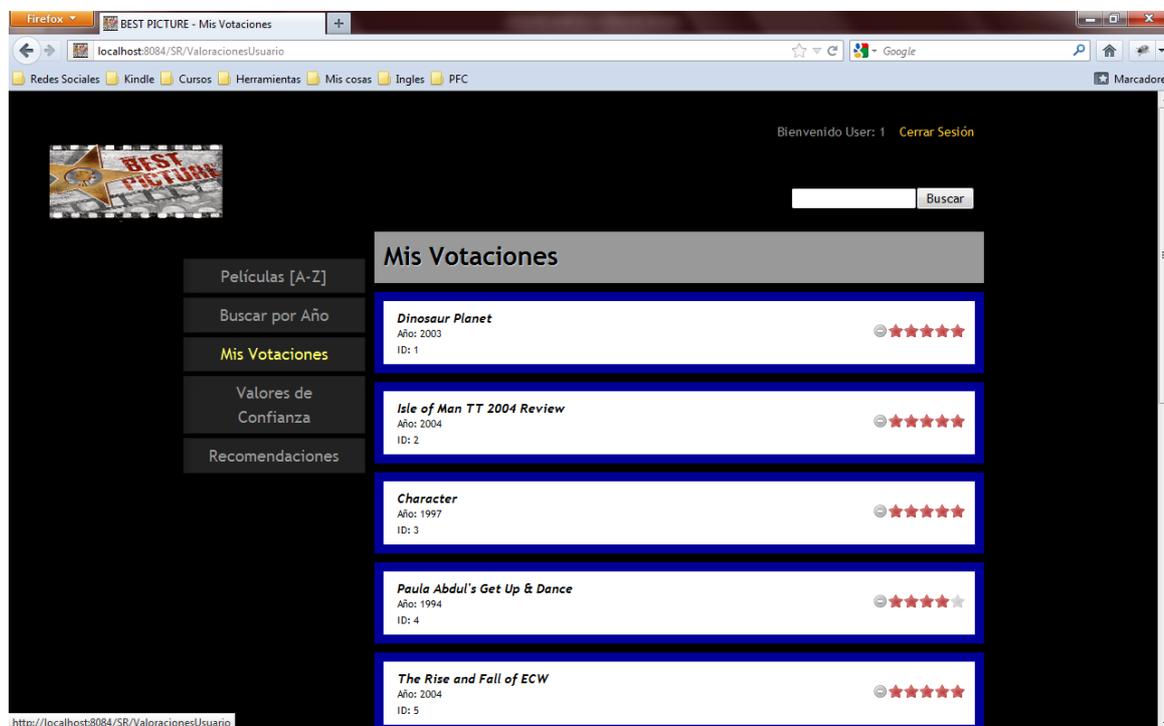


FIGURA B - 18: MIS VOTACIONES

En este caso igualmente se pueden modificar las puntuaciones que anteriormente había hecho.

de una red de confianza

Si el usuario no ha llevado a cabo ninguna votación se nos mostrará un aviso (**¡Error! No se encuentra el origen de la referencia.**).



FIGURA B - 19: ERROR EN MIS VOTACIONES

También podemos **ver los valores de confianza** que un usuario ha dado a otros usuarios del sistema en la sección “Valores de Confianza” (**¡Error! No se encuentra el origen de la referencia.**).



FIGURA B - 20: VALORES DE CONFIANZA

de una red de confianza

Por último podemos ver **las recomendaciones** que el sistema nos hace según las votaciones que nosotros hemos hecho anteriormente, en la sección “Recomendaciones” (**¡Error! No se encuentra el origen de la referencia.1**).

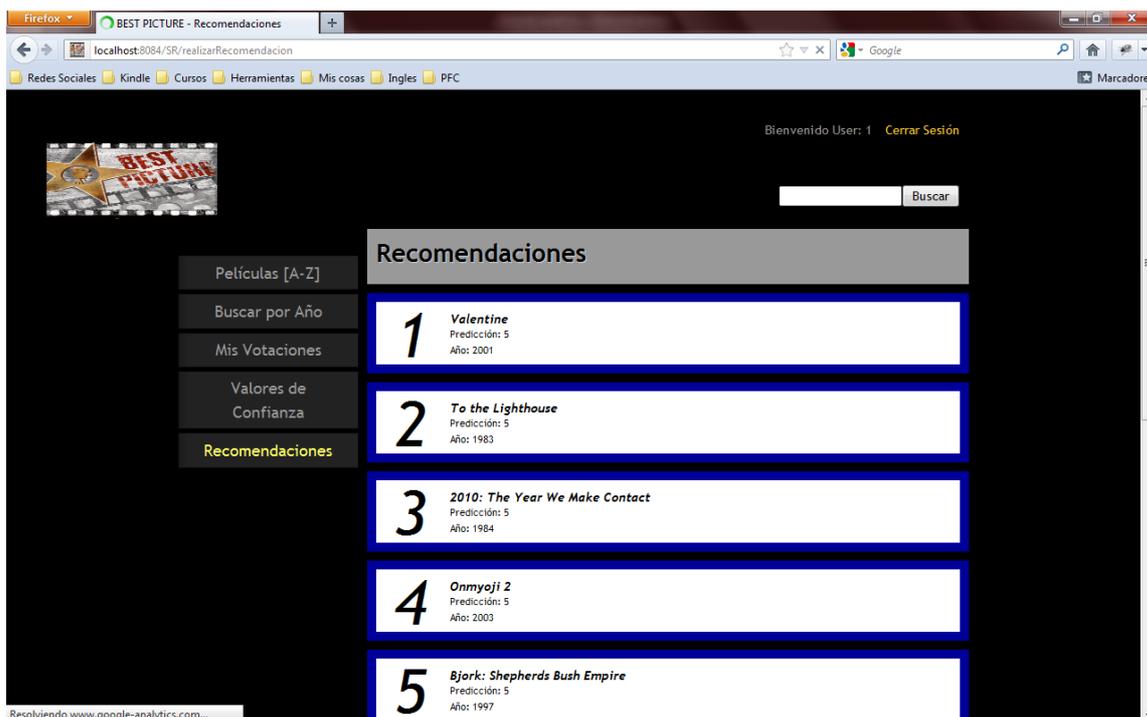


FIGURA B - 21: RECOMENDACIONES

Como podemos ver, es una lista ordenada de recomendaciones del uno al diez.

Si el usuario no dispone de suficientes películas votadas se nos mostrará un aviso, ya que la aplicación no nos puede dar una recomendación (**¡Error! No se encuentra el origen de la referencia.2**).

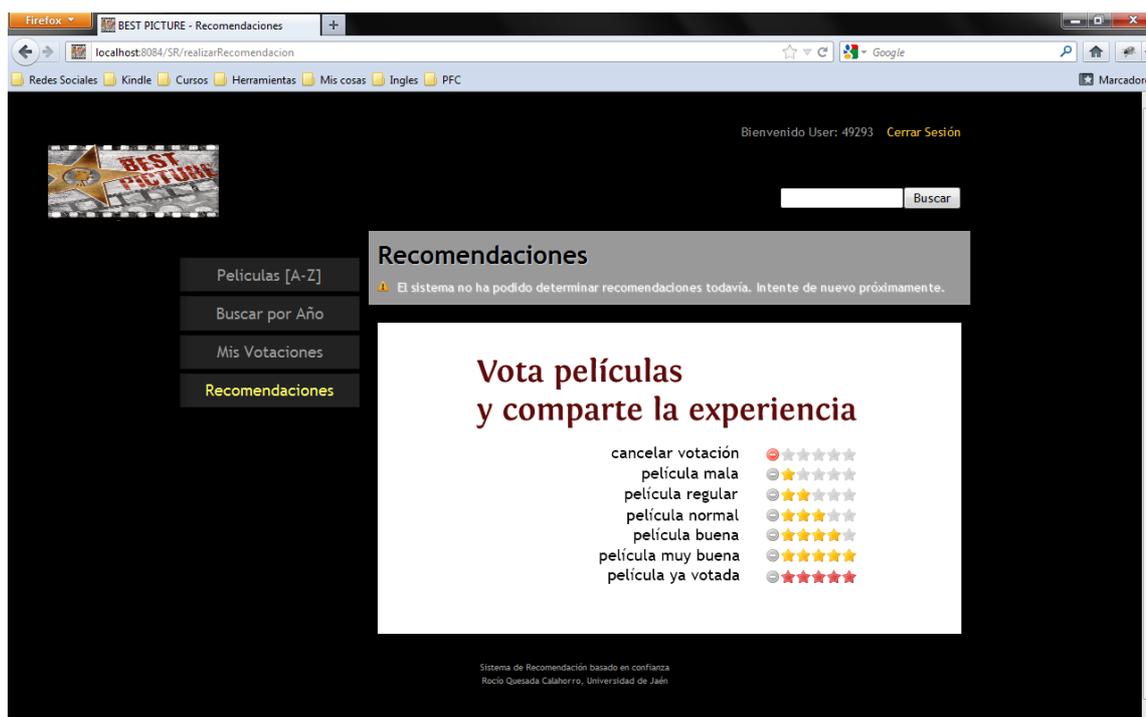


FIGURA B - 22: ERROR EN RECOMENDACIONES

de una red de confianza

Para cerrar sesión el usuario deberá pulsar la opción de **Cerrar sesión** situada en todas las páginas en la parte superior derecha (**¡Error! No se encuentra el origen de la referencia.3**).

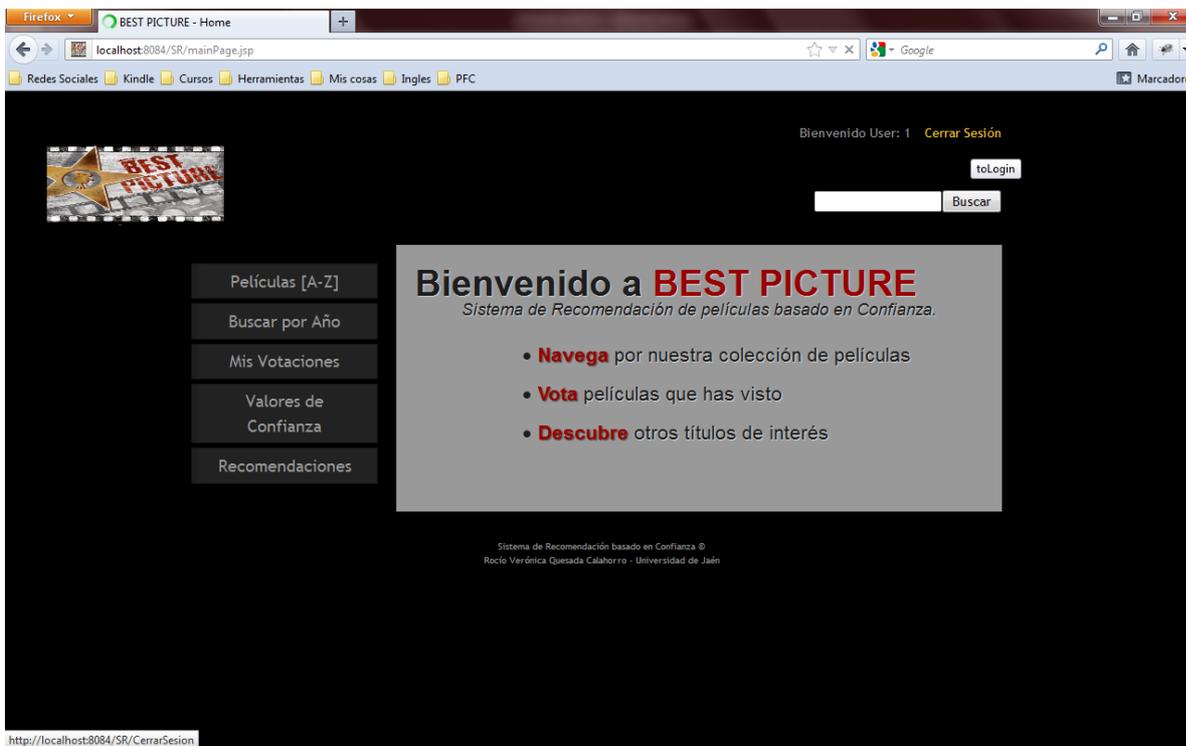


FIGURA B - 23: CERRAR SESIÓN

APÉNDICE C

CODIGO FUENTE

de una red de confianza

Este último apéndice está dedicado a mostrar las partes del código fuente de nuestro proyecto que más relevancia tienen en él. A continuación mostramos la implementación en Java de los métodos `trustall` y `calculaConfianza`:

```
/** Realiza el cálculo la predicción
 *
 * @param con conexión que nos para la App
 *
 * @param pelicula es el id de la película que se va a predecir
 *
 * @param usuario es el perfil del usuario al que se le está haciendo la predicción
 *
 * @param mapaPerfilesUsu es un mapa con todos los perfiles de usuario en los que confía el usuario
 *
 * @return valor de la predicción
 *
 * @throws GetValoracionException
 *
 * @throws GetConfianzaException
 *
 * @throws SQLException
 *
 * @throws TrustException
 */
public static double trustall(Connection conn, int pelicula, PerfilUsuario usuario, Map mapaPerfilesUsu)
        throws GetConfianzaException, SQLException, TrustException,
        GetValoracionException {
    double ru = 0;
    double ra = 0;
    if(conn!=null){
        try {
            ra = usuario.getAverage(); //media de las valoraciones del usuario

```

```
double numerador = 0;

double denominador = 0;

Iterator itPerfiles = mapaPerfilesUsu.entrySet().iterator();

while (itPerfiles.hasNext()){ //recorre los perfiles de usuario

    Map.Entry entrada = (Map.Entry)itPerfiles.next();

    PerfilUsuario perfil = (PerfilUsuario)entrada.getValue(); //coge un perfil de usuario

    ru = perfil.buscaValoracion(pelicula); //busca la valoración de ese usuario para la película en cuestión

    double average_ru = perfil.getAverage(); //coge la media de valoraciones de ese usuario

    if (ru > 0){ //si la valoración que ha hecho ese usuario es mayor que cero

        numerador += (ru-average_ru); //suma para cada usuario en los que confía el usuario

        denominador++;

    }

}

if (denominador != 0.0)

    return ra + (numerador / denominador); //devuelve el valor de la predicción

else return -1; //devuelve -1 si el denominador vale 0

} catch(Exception e){

    throw new TrustException();

}

}

return 0; //devuelve 0 si no se ha establecido la conexión

}
```

de una red de confianza

```
/** Calcula la confianza que un usuario tendría en otro
 *
 * @param conn
 * @param usu1
 * @param usu2
 * @return valor de la confianza que tendría usu1 de usu2
 * @throws GetConfianzaException
 * @throws SQLException
 * @throws TrustException
 */
public double calculaConfianza(Connection conn, int usu1, int usu2) throws GetConfianzaException,
                                SQLException,
                                TrustException{
    DAOConfianza daoConfianza = new DAOConfianza(conn);
    if(conn!=null){
        ArrayList confianzas = daoConfianza.getUsu1(usu1);//array de confianzas del usu1
        double numerador = 0;
        double denominador = 0;
        for (int i=0; i<confianzas.size(); i++){ //recorre el array de confianza del usu1
            Confianza conf = (Confianza) confianzas.get(i); //coge la confianza i del array de confianzas
            if(conf.getTrust(>0.6){ // si esa confianza es mayor que 0.6
```

```
double trust = daoConfianza.confía(conf.getIdUsu2(), usu2); //comprueba si el usu i confía en usu2

if (trust > 0.6){

//si esa confianza es a su vez mayor que 0.6 multiplica la confianza que tiene el usu i en el usu2 por
la confianza del usu1 en el usu i

    numerador += trust*conf.getTrust();

    denominador += conf.getTrust(); //voy sumando la confianza de usu1 con el usu i

}

}

}

if (denominador == 0) throw new TrustException();

return (numerador/denominador); //devuelve el valor de la confianza

}

return -1; // si devuelve -1 es que no se ha llevado a cabo el cálculo

}

}
```