



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Proyecto Fin de Carrera

**RADIO ONLINE
BASADA EN UN
MOTOR DE FILTRADO
COLABORATIVO**

Alumno: Iván Palomares Carrascosa

Tutores: Prof. Dr. Luis Martínez López
Prof. Dra. Macarena Espinilla Estévez

Departamento: Informática

Junio 2009



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Dr. D. Luis Martínez López y Dra. D^a. Macarena Espinilla Estévez, tutores del Proyecto Fin de Carrera titulado: Radio Online basada en un Motor de Filtrado Colaborativo, que presenta D. Iván Palomares Carrascosa, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2009

El alumno

Los Tutores:

D. Iván Palomares Carrascosa

Dr. D. Luis Martínez López
Dra. D^a. Macarena Espinilla Estévez

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que, de alguna forma, han contribuido a que este proyecto haya salido adelante, con especial mención a:

Mis padres, porque gracias a su esfuerzo y sacrificio, he conseguido llegar hasta aquí.

Mi hermana, por su paciencia y comprensión durante aquellos fines de semana en los que tuve que “acaparar” el ordenador los días enteros.

Mis abuelos y familia en general, por ser como son, y por la confianza que siempre han depositado en mí.

Mis tutores, Luis y Macarena, por el constante ánimo y motivación que me han brindado, por lo mucho que me han ayudado y, sobretodo, por haber sido para mí tan profesionales y al mismo tiempo tan amigos.

El Profesor Dr. D. José Moyano Fuentes, por sus sabios consejos y porque, de algún modo, es el “primer responsable” de que haya llegado a realizar este estupendo proyecto.

MUCHAS GRACIAS A TODOS.

Índice general

1.INTRODUCCIÓN	1
1.1. Introducción al Proyecto	2
1.2. Propósito.....	5
1.3. Objetivos	5
1.4. Estructura y Planificación del Proyecto	5
 2. SISTEMAS DE RECOMENDACIÓN	 9
2.1. Introducción	10
2.1.1. Esquema Básico de Funcionamiento y Elementos de un SR.....	12
2.1.2. Utilidad de los Sistemas de Recomendación.....	13
2.1.3. Datos en los SR	14
2.1.3.1. Realimentación en los SR	14
2.1.3.2. Realimentación Implícita.....	14
2.1.3.3. Realimentación Explícita	15
2.1.3.4. Datos Reales vs. Datos Sintetizados.....	15
2.1.3.5. Análisis Online vs. Análisis Offline	15
2.2. Clasificación de los Sistemas de Recomendación	16

2.2.1. Sistemas de Recomendación Basados en Contenido	16
2.2.2. Sistemas de Recomendación Colaborativos.....	19
2.2.3. Sistemas de Recomendación Basados en Conocimiento.....	20
2.2.4. Sistemas de Recomendación Híbridos.....	22
2.3. Ejemplos Reales de Sistemas de Recomendación en Internet	23
2.3.1. YouTube.....	23
2.3.2. Amazon.....	24
2.3.3. IMDb Recommendation Center.....	26
2.4. Sistemas de Recomendación Colaborativos.....	27
2.4.1. Introducción y orígenes	27
2.4.2. Fases y Ejemplo de Funcionamiento de los SR Colaborativos	28
2.4.3. Algoritmos de Filtrado Colaborativo.....	31
2.4.4. Medidas y Técnicas Empleadas en Algoritmos de Filtrado Colaborativo	33
2.4.4.1. Notación.....	34
2.4.4.2. Algoritmo K-nn.....	34
2.4.4.3. Medidas de Similitud	36
2.4.4.4. Medidas de Predicción.....	38
2.4.5. Problemas de los Sistemas de Recomendación Colaborativos.....	39
2.4.6. Sistemas de Recomendación Colaborativos en Internet	40

2.4.6.1. Zagat.....	41
2.4.6.2. MovieLens	42
3. RADIOS COLABORATIVAS EN INTERNET	45
3.1. Introducción a la difusión de contenidos Multimedia en la Online	46
3.2. Música en la Web: Streaming	46
3.3. Ejemplos de radios colaborativas en Internet	49
3.3.1. Last.fm	49
3.3.2. Pandora.....	51
3.4. Licencias de libre distribución de contenidos	52
3.4.1. Copyleft	53
3.4.2 Creative Commons.....	54
3.4.2.1. Creative Commons en España.....	55
4. DESARROLLO DEL PROYECTO	59
4.1. Descripción del Proyecto.....	60
4.2. Especificación de Requerimientos	62
4.2.1. Requerimientos funcionales.....	63
4.2.2. Requerimientos No Funcionales	66

4.2.3. Requerimientos de la Interfaz	69
4.3. Análisis del Sistema.....	71
4.3.1. Modelo de Casos de Uso	71
4.3.2. Escenarios	87
4.4. Diseño del Sistema.....	94
4.4.1. Diseño de los Datos	95
4.4.1.1. Esquema Conceptual.....	99
4.4.1.2. Esquema Conceptual Modificado	101
4.4.1.3. Tablas de la Aplicación	103
4.4.2. Diseño de la Interfaz	108
4.4.2.1. Guía de Estilo	109
4.4.2.2. Metáforas	111
4.4.2.3. Prototipos de la Aplicación	114
4.4.2.4. Caminos de navegación.....	122
4.5. Implementación	128
4.5.1. Tipo de arquitectura de la aplicación.....	128
4.5.2. Lenguajes de programación utilizados.....	129
4.5.3. Herramientas de desarrollo.....	133
4.5.4. Actualización del algoritmo de filtrado	138

5. CONCLUSIONES	139
5.1. Conclusiones	140
A. MANUAL DE INSTALACIÓN DEL SERVIDOR	143
A.1. Instalación de AppServ	144
A.2. Alojamiento del contenido de la aplicación en el Servidor	152
A.3. Importar la base de datos y activar Drupal	153
A.4. Instalar la máquina virtual Java	155
A.5. Instalar NetBeans y JDBC para la ejecución del Algoritmo de Filtrado.....	156
B. MANUAL DE USUARIO	161
C. CÓDIGO FUENTE DEL ALGORITMO DE FILTRADO COLABORATIVO	175
Bibliografía y Referencias	187

Lista de figuras

2. SISTEMAS DE RECOMENDACIÓN

2.1: Esquema básico de funcionamiento de un SR.....	12
2.2: Esquema de funcionamiento de un Sistema de Recomendación basado en contenido. ...	17
2.3: Funcionamiento de un SR basado en contenido	18
2.4: Funcionamiento de los Sistemas de Recomendación basados en conocimiento.....	20
2.5: Portal de YouTube en Internet	23
2.6: Interfaz de recomendaciones de YouTube	24
2.7: Portal de Amazon en Internet	24
2.8: Interfaz de recomendación de Amazon	25
2.9: Interfaz del centro de recomendación de IMDb	26
2.10: Interfaz del centro de recomendación de IMDb	26
2.11: Funcionamiento de los Sistemas de Recomendación Colaborativos	28
2.12: Proceso de cálculo de predicciones en SR Colaborativos	29
2.13: Ejemplo de funcionamiento de SR Colaborativos	29
2.14: Las principales partes de un SR Colaborativo basado en memoria	31
2.15: Cálculo de la similitud basada solamente en ítems co-evaluados	36
2.16: Portal de Zagat.com	41

2.17: Opiniones de usuarios en Zagat.com	42
--	----

2.18: Portal de MovieLens	42
---------------------------------	----

3. RADIOS COLABORATIVAS EN INTERNET

3.1: Ejemplo de estructura de audio-Streaming en directo.....	48
---	----

3.2: Portal de Last.fm.	49
------------------------------	----

3.3: Sección de artistas y recomendaciones en Last.fm	50
---	----

3.4: Portal de pandora.com.....	51
---------------------------------	----

3.5: Logotipo de la licencia Copyleft	53
---	----

3.6: Logotipo de la licencia Creative Commons	54
---	----

4. DESARROLLO DEL PROYECTO

4.1: Diagrama Frontera del sistema.....	74
---	----

4.2: Caso de uso 'Escuchar radio'	79
---	----

4.3: Caso de uso 'Gestión de datos de usuario'	82
--	----

4.4: Caso de uso 'Gestión de contenidos'	85
---	----

4.5: Esquema Conceptual de la aplicación	100
--	-----

4.6: Esquema Conceptual Modificado de la aplicación.....	102
--	-----

4.7: Ejemplo de tuplas en la tabla CANCIONES.....	105
---	-----

4.8: Ejemplo de tuplas en la tabla PUNTUACIONES.....	106
--	-----

4.9: Panel de puntuaciones de la aplicación	113
4.10: Disposición de la lista de álbumes	113
4.11: Pantalla principal/radio para el usuario anónimo	115
4.12: Pantalla principal/radio para el usuario registrado	116
4.13: Panel de puntuaciones	117
4.14: Pantalla Lista de Álbumes	118
4.15: Pantalla álbum	119
4.16: Pantalla Perfil de Usuario	120
4.17: Pantalla Libro de Visitas	121
4.18: Arquitectura Cliente/Servidor	129
4.19: Panel de bienvenida de NetBeans 6	133
4.20: Ejemplo de funcionamiento de NotePad++	135
4.21: Panel de administración de Drupal	136
4.22: Pantalla principal de phpMyAdmin	137

A. MANUAL DE INSTALACIÓN DEL SERVIDOR

A.1: Portal de AppservNetwork.com	145
A.2: Instalador de AppServ	146
A.3: Acuerdo de licencia de AppServ	146

A.4: Selección del directorio de instalación.....	147
A.5: Pantalla de selección de componentes.....	147
A.6: Configuración del Servidor Apache	148
A.7: Configuración de MySQL Server	149
A.8: Pantalla fin de instalación de AppServ	149
A.9: Obtención de la dirección IP	150
A.10: Pantalla inicial del servidor AppServ	151
A.11: Control de acceso a phpMyAdmin	151
A.12: Pantalla de inicio de phpMyAdmin en el servidor	152
A.13: Importación de la base de datos	153
A.14: Página principal de la aplicación.....	154
A.15: Descarga de Java 6.....	155
A.16: Instalador de Java 6	155
A.17: Instalación de Java 6.....	156
A.18: Descarga de NetBeans con JDK.....	157
A.19: Directorio de instalación de JDK.....	157
A.20: Directorio de colocación de JDBC	158
A.21: Abrir proyectos en NetBeans	159

B. MANUAL DE USUARIO

B.1: Página de inicio de OL-RadioUJA	163
B.2: Formulario de inicio de sesión	163
B.3: Pantalla principal para usuario autenticado	164
B.4: Error de validación en OL-RadioUJA.	165
B.5: Registro de usuario	166
B.6: Menú principal de OL-RadioUJA.....	167
B.7: Detalles de la pantalla de radio.....	167
B.8: Lista de reproducción (radio).....	168
B.9: Detalles de canción en radio del usuario.....	169
B.10: Pantalla de puntuación de canciones.....	170
B.11: Vista principal de la sección 'Álbumes'	171
B.12: Vista de los álbumes por género.....	172
B.13: Página de información del álbum.....	172
B.14: Perfil de usuario en OL-RadioUJA	173
B.15: Libro de visitas	174

CAPÍTULO 1

Introducción

1.1 Introducción al Proyecto

El uso, cada vez mayor, de los múltiples servicios que ofrece Internet por parte de los usuarios es de lo más variado, y está propiciando la existencia de sitios Web que sirven de soporte para muchos de estos servicios, no sólo en el ámbito científico, académico o empresarial, sino también para el ocio y disfrute del usuario.

La **radio por Internet** es actualmente uno de los mayores atractivos de ocio y tiempo libre para los internautas en general, debido a su facilidad de uso y su alto grado de accesibilidad. Se fundamenta en el “**Webcasting**” [16], es decir, la difusión a través de Internet de contenido multimedia, en este caso de audio. Para ello, se utiliza la tecnología conocida como “**Streaming**” [17], que consiste en brindar al usuario la posibilidad de reproducir contenidos multimedia directamente y de forma paralela en el navegador Web, sin que este tenga que descargar dicho contenido en su ordenador.

En el ámbito de la radio por Internet cabe destacar la reciente irrupción de radios personalizadas colaborativas, que ayudan al usuario a encontrar nueva música de su agrado, basándose en sus preferencias, es decir, estudiando las características de la música que ya ha escuchado.

Algunos ejemplos de radios colaborativas los podemos encontrar en :

-**Pandora** (www.pandora.com): Mediante una interfaz muy conseguida[13], ayuda al usuario a generar una lista de artistas y canciones de su agrado desde el momento en que escucha su primera canción, basándose en las similitudes entre objetos (canciones).

-**Last.fm** (www.lastfm.es): Considerada como una red social a gran escala, construye a partir de estadísticas de otros usuarios registrados perfiles sobre los gustos musicales que se adecúan al usuario. Su servicio es de código abierto y se basa en un algoritmo de filtrado colaborativo [3].

El exitoso funcionamiento de las citadas radios online radica en el uso de lo que formalmente se conoce como Sistemas de Recomendación Colaborativos.

Hay varias 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.

Existen varios tipos de Sistemas de Recomendación: Basados en Contenido, Colaborativos [1], Basados en Conocimiento, Demográficos, así como hibridaciones de los anteriores tipos. A continuación, comentaremos brevemente los dos primeros, por ser los más ampliamente utilizados:

1. Sistemas de Recomendación Basados en Contenido.

Están basados en la similitud entre objetos, por tanto, consisten en predecir para un usuario los objetos que sean de su interés, al calcular los objetos más parecidos a aquellos que ya sabemos que lo son.

2. Sistemas de Recomendación Colaborativos.

Son aquellos en los que las recomendaciones se realizan en base únicamente a los términos de similitud entre usuarios, es decir, a un usuario se le recomiendan objetos que sabemos que son del agrado de otros usuarios de gustos similares a él. Por tanto, este tipo de Sistema de Recomendación se asemeja bastante a la forma de pensar que tenemos los seres humanos.

El presente proyecto se basará en la construcción de un **Sistema de Recomendación Colaborativo**, que puede definirse como:

Los Sistemas de Recomendación basados en filtrado Colaborativo o, simplemente, Colaborativos, son aquellos en los que las recomendaciones se hacen atendiendo a las similitudes entre usuarios: se trata de recomendar objetos del interés de otros usuarios que sabemos que tienen gustos e intereses similares.

Es necesario, pues, elegir un buen algoritmo de filtrado colaborativo [5] que se ajuste a los objetivos.

En el caso del presente proyecto, se pretende, en primera instancia, escoger un algoritmo de filtrado colaborativo que se adecúe bien a un Sistema de Recomendación basado en una colección de archivos musicales de libre distribución, utilizando “Streaming” para su publicación a través de Internet, y ofreciendo así una radio colaborativa por Internet dotada de una interfaz usable.

Para ello, en primer lugar descargaremos una muestra significativa y variada de archivos musicales bajo licencia Creative Commons [18], buena parte de los cuales los hemos encontrado en la base de datos de Internet *Jamendo* (<http://www.jamendo.es>). Una vez que tenemos dicha muestra, pasamos a subirla a nuestro servidor y desarrollamos un sitio Web para el Sistema de Recomendación, con acceso a una base de datos para recuperar la información relativa a las canciones.

Posteriormente, implementamos el algoritmo de filtrado colaborativo para después integrarlo en el servidor junto con el sitio Web y la base de datos, obteniendo así una primera versión del resultado que se pretende conseguir con este proyecto. La aplicación resultante de la realización del proyecto tendrá el nombre OL-RADIOUJA.

Por último, expondremos una serie de conclusiones generales sobre la radio online colaborativa que se ha venido desarrollando para este proyecto.

1.2 Propósito

Creación de una radio colaborativa personalizada, basada en las preferencias musicales del usuario.

1.3 Objetivos

1. Búsqueda y revisión bibliográfica.
2. Estudio y obtención de una base de datos de música bajo una licencia de libre distribución.
3. Estudiar métodos de reproducción de música a través de la Web.
4. Análisis, diseño e implementación de un modelo de filtrado o motor colaborativo adecuado al tipo de información adquirido.
5. Implementación de una radio colaborativa en base al algoritmo de filtrado colaborativo estudiado.
6. Redacción de la memoria.

1.4 Estructura y Planificación del Proyecto

A continuación, haremos una breve introducción a los diferentes capítulos en los que se estructura este proyecto y los contenidos expuestos en los mismos. Como hemos visto, este primer capítulo supone una introducción general al proyecto que se ha realizado, con una justificación de su realización, la definición del propósito y los objetivos que persigue.

En el capítulo 2, abordamos una visión general de los Sistemas de Recomendación. En primer lugar, daremos la definición de Sistema de Recomendación, así como la justificación de su utilización. A continuación estudiaremos la estructura y elementos básicos de los Sistemas de Recomendación, y veremos una clasificación de Sistemas de Recomendación, atendiendo a su funcionamiento, ventajas e inconvenientes. Esta visión de los Sistemas de Recomendación irá acompañada de algunos ejemplos reales de sistemas de colaboración existentes en Internet.

El capítulo 3 está dedicado al concepto de radio colaborativa en Internet. Haremos un breve repaso a las tecnologías de difusión de contenidos multimedia en Internet de *Streaming*, nos centraremos en la difusión de contenido musical y en las radios colaborativas, conociendo además algunos ejemplos reales de ellas. También revisaremos en este capítulo las licencias de distribución de contenidos, ya que han sido un importante factor a tener en cuenta para la recopilación de los datos para este proyecto.

El capítulo 4 supone el eje principal de la presente memoria, al ser el de mayor extensión y estar dedicado al proceso completo de desarrollo de este proyecto. Como proyecto software que es, haremos en primer lugar un breve repaso a las diferentes etapas de la Ingeniería de Software, para acto seguido aplicarlas en el desarrollo de nuestro sistema. Así, definiremos los requerimientos funcionales y no funcionales para el sistema, abordaremos la etapa de análisis, incluyendo el modelo de casos de uso y el análisis de la base de datos. Acto seguido, pasaremos a la etapa de diseño, haciendo especial hincapié en el diseño de la interfaz del sitio Web y las pruebas de usabilidad llevadas a cabo. Por último, repasaremos el proceso seguido para la etapa de implementación de la radio y el motor de filtrado colaborativo.

Una vez expuesto el desarrollo del proyecto, llegamos al quinto y último capítulo, dedicado a las conclusiones generales derivadas del desarrollo del mismo.

La sección final de esta memoria contiene los anexos dedicados a la instalación y configuración del servidor, el manual de usuario de la aplicación OL-RADIOUJA, y el código fuente del algoritmo de filtrado.

CAPÍTULO 2

Sistemas de Recomendación

2.1. Introducción

El actual auge de las Tecnologías de la Información, germen de lo que conocemos como Sociedad de la Información y su evolución hacia la Sociedad del Conocimiento, han propiciado que las personas dispongamos cada vez de más información para realizar nuestros acometidos. Esto es en cierto modo una ventaja, pero también nos encontramos con frecuencia el problema de sobrecarga de información, lo cual puede llegar a dificultar seriamente la tarea de escoger la información más adecuada a nuestras necesidades.

Es en Internet donde esta situación se hace presente a menudo, debido al cada vez mayor número de sitios dedicados a múltiples propósitos que ofrecen una considerable colección de información. El usuario necesita algún tipo de “ayuda” para elegir aquel contenido que sea de su interés. Así, en los últimos años los servicios de las citadas aplicaciones Web se han ido centrando en personalizar sus productos y/o servicios, de manera que consigan satisfacer las necesidades de cada usuario concreto.

Pensemos por ejemplo en una tienda online. El repertorio de productos que ofrecen dichas tiendas suele ser muy amplio y variado, lo cual provoca un gran número de elecciones posibles para los clientes y, por consiguiente, un aumento de la información que el cliente debe procesar para tomar una decisión. Esto puede traducirse, en muchos casos, en una sobrecarga de información, lo que hace que el usuario tenga una sensación de saturación que podría, en el peor caso, provocar que el usuario renuncie a realizar la compra al desistir de realizar una elección costosa en tiempo. En cambio, si el sistema de la tienda online fuese capaz de recomendar, entre el amplio conjunto de productos, aquellos que se estimen de mayor interés para cada usuario, la elección será mucho más sencilla para este.

Es aquí donde interviene una herramienta ampliamente utilizada, de forma satisfactoria, para resolver este inconveniente: se trata de los Sistemas de Recomendación.

Una primera y sencilla definición de Sistema de Recomendación es la siguiente:

Cualquier sistema que provee recomendaciones, predicciones u opiniones, normalmente en forma de conjunto de ítems, con el fin de ayudar al usuario a descubrir y evaluar ítems.

En adelante, utilizaremos el término *ítem* para referirnos a cualquier objeto, servicio o producto existente en un Sistema de Recomendación. Otra definición, empleada por algunos autores para los Sistemas de Recomendación, es la siguiente:

Aquellos sistemas que utilizan las opiniones de los usuarios de una comunidad para ayudar a los usuarios de la misma a encontrar contenidos de su gusto entre un conjunto sobrecargado de elecciones posibles.

A partir de estas definiciones, no nos resulta muy difícil concluir que los Sistemas de Recomendación actúan de un modo muy similar a como lo hace la inteligencia humana, al basarse en actos tan frecuentes en la vida cotidiana como: pedir asesoramiento (recomendaciones) a expertos en un dominio particular, guiarse por individuos con gustos y preferencias similares a las del usuario (lo que se conoce como Sistema de Recomendación Colaborativo), o elegir objetos con cualidades similares a las de otros objetos que ya han sido positivamente valorados por el usuario con anterioridad (Sistema de Recomendación basado en contenido).

Actualmente existen varios tipos de Sistemas de Recomendación [1], entre los que destacan los Sistemas de Recomendación Colaborativos, basados en contenido, basados en conocimiento e híbridos. Veremos con mayor profundidad las características de cada uno de ellos en un epígrafe posterior.

Tanto los Sistemas de Recomendación Colaborativos como los basados en contenido presentan el inconveniente de requerir una gran cantidad de información, tanto de usuarios del mismo como de los ítems que lo componen, para realizar recomendaciones de calidad y, por tanto, funcionar correctamente. Para resolver este inconveniente aparecieron otros tipos

de Sistemas de Recomendación capaces de realizar recomendaciones de calidad sin necesidad de una gran cantidad de información: se trata de los Sistemas de Recomendación basados en conocimiento. Por último, de cara a mejorar aún más los resultados de los Sistemas de Recomendación, han aparecido en los últimos años los Sistemas de Recomendación híbridos, en los que se sintetizan las ventajas de dos o más de los anteriores tipos de Sistemas de Recomendación, para así lograr un resultado mejor al que cada uno de ellos lograría por separado.

A continuación, estudiaremos brevemente la estructura básica de funcionamiento de los Sistemas de Recomendación (que en adelante también nombraremos de forma abreviada como SR), los principales elementos que los componen, y la utilidad de los mismos.

2.1.1. Esquema Básico de Funcionamiento y Elementos de un SR

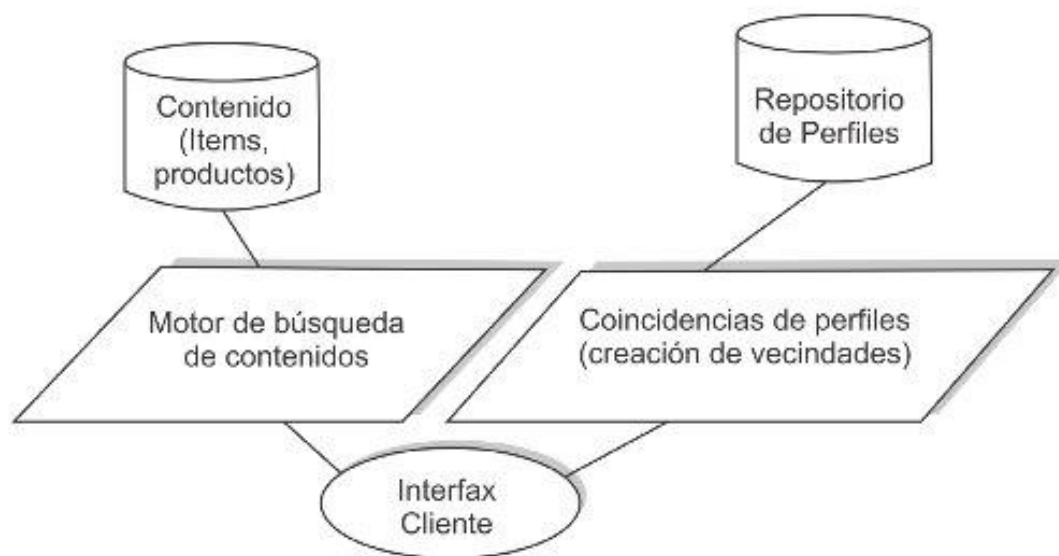


Figura 2.1: Esquema básico de funcionamiento de un SR

En la figura 2.1 se puede observar un esquema básico de un SR en el que se puede distinguir los siguientes elementos:

- **Base de datos:** La calidad de los datos almacenados en nuestra base de datos jugará un papel crucial a la hora de realizar recomendaciones con mayor o menor calidad.

- **Perfiles de Usuario:** Un usuario va “dándole forma” a su perfil personal a medida que utiliza el sistema. El perfil refleja los gustos/preferencias del usuario, fundamentales a la hora de discriminar objetos durante la recomendación.

- **Predicciones:** La predicción es el elemento clave en el esquema básico de todo SR. La predicción se basa en el perfil del usuario y en la información disponible en la base de datos con la que contamos.

2.1.2. Utilidad de los Sistemas de Recomendación

Los Sistemas de Recomendación han sido utilizados con gran éxito en aplicaciones de marketing personalizado en Internet, dado que reducen el tiempo invertido en realizar búsquedas de productos y hacen las búsquedas más efectivas, lo cual conlleva una mayor satisfacción a los clientes. Además, en los últimos tiempos están cobrando importancia los Sistemas de Recomendación en portales de difusión de contenidos multimedia, tales como música o vídeos. En ellos, el usuario recibe recomendaciones de canciones o vídeos que se prevé que serán de su agrado.

Para conseguir estos objetivos, todos los SR llevan a cabo dos tareas básicas:

- **Predecir:** Los Sistemas de Recomendación realizan predicciones de un conjunto de objetos, productos o servicios en los cuales podría estar interesado un usuario o cliente concreto.

- **Recomendar los N-mejores objetos:** Se trata de determinar los N objetos en los que un usuario o cliente estará más interesado.

En el siguiente epígrafe trataremos algunas cuestiones más específicas de los Sistemas de Recomendación, tales como la política de tratamiento de los datos que utilizan, además de los mecanismos de realimentación empleados.

2.1.3. Datos en los SR

Es importante tomar algunas decisiones a la hora de desarrollar un Sistema de Recomendación, tales como la forma en que recibirá información nueva, el tipo de datos a utilizar o la forma en que dichos datos se analizarán.

2.1.3.1. Realimentación en los SR

Un Sistema de Recomendación no debe ser una entidad estática, sino que la calidad de sus recomendaciones ha de evolucionar con el tiempo, en base a la experiencia y nueva información adquirida. Esto se consigue mediante los mecanismos de realimentación entre el sistema y los usuarios. Existen dos mecanismos de realimentación: la realimentación implícita y la realimentación explícita.

2.1.3.2. Realimentación Implícita

Un mecanismo de realimentación implícita es aquel que proporciona al SR información sobre las preferencias de los usuarios sin que estos sean conscientes de ello. Estas realimentaciones no se hacen de forma directa, sino mediante algunas medidas como pueden ser: el tiempo de visualización del objeto, el número de consultas del mismo, etc.

Presenta el problema de que depende demasiado del contexto y es bastante hipotética, ya que se hacen suposiciones (a partir de las mencionadas medidas) sobre los gustos del usuario que no necesariamente tienen por qué ser ciertas.

2.1.3.3. Realimentación Explícita

La realimentación explícita se basa en la acción directa y deliberada del usuario para indicar aquellos objetos del sistema que le interesan. Esta acción se puede conseguir mediante votaciones numéricas o, simplemente, indicando si el objeto es o no del agrado del usuario. Este tipo de realimentación también presenta problemas, como son la voluntariedad del cliente o el tiempo invertido en ello.

2.1.3.4. Datos Reales vs. Datos Sintetizados

Otra cuestión interesante es la de elegir un conjunto de datos reales (recopilados de usuarios reales sobre objetos reales) o un conjunto de datos sintetizados (sin ninguna base real, específicamente creados para el Sistema de Recomendación). Estos últimos son más fáciles de obtener, ya que evitamos tener que realizar encuestas u otros métodos de recopilación de información real, aunque sólo se utilizan en las primeras fases del desarrollo del sistema, para posteriormente ser sustituidos por los datos reales una vez que se haya recopilado la información suficiente.

2.1.3.5. Análisis Online vs. Análisis Offline

Es importante decidir si vamos 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, evaluando los resultados de dichas predicciones 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 de los mismos, etc. Sin embargo, resulta ser más lento y costoso que el análisis offline.

2.2. Clasificación de los Sistemas de Recomendación

Una vez sentadas las bases de los Sistemas de Recomendación, es el momento de revisar los diferentes tipos existentes de los mismos. Los Sistemas de Recomendación pueden ser implementados utilizando diversas técnicas [1], por lo que su clasificación dependerá de su funcionamiento para calcular recomendaciones. La clasificación más habitual es la siguiente:

2.2.1. Sistemas de Recomendación Basados en Contenido

Un Sistema de Recomendación basado en contenido es aquel que está basado en las características de los objetos, es decir, las recomendaciones se llevan a cabo basándose únicamente en un perfil creado a partir del análisis del contenido de los objetos que el usuario ha evaluado con anterioridad.

En otras palabras, estos sistemas extraen características de los objetos y las comparan con el perfil del usuario para predecir las preferencias de los usuarios sobre tales objetos. La idea es recomendar objetos similares en su contenido a objetos que ya sabemos que son del agrado del usuario en cuestión, es decir, los que forman parte de su perfil.

El filtrado basado en contenido era el tipo de Sistema de Recomendación más extendido antes de que se produjese la explosión del filtrado colaborativo, ya que los sistemas del primer tipo tienen un claro problema de sobre-especialización.

El funcionamiento de los sistemas de colaboración basados en contenido (Figura 2.2) comprende dos grandes etapas: analizar las descripciones de los productos valorados por los usuarios y predecir qué productos le pueden gustar.

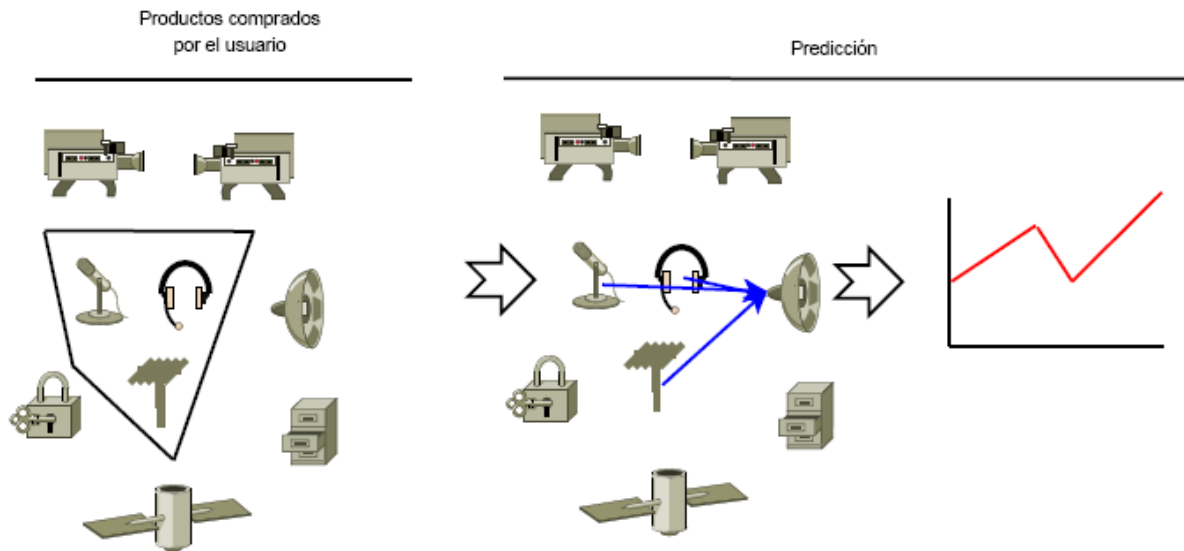


Figura 2.2: Esquema de funcionamiento de un Sistema de Recomendación basado en contenido.

Este funcionamiento dependerá en gran medida del tipo de información descriptiva que se utilice en el perfil del usuario. Esta información se divide en:

- *Conjunto de características:* Asociado a cada producto tenemos un conjunto de características que lo describen [2]. Por ejemplo, kilometraje o consumo medio si estamos hablando de vehículos.
- *Información textual sobre el producto:* Se trata de un documento que describe al producto. Se diferencia del conjunto de características principalmente en que la información no está estructurada.

Muchos Sistemas de Recomendación basados en contenido trabajan únicamente en base a información del conjunto de características de los objetos. En tal caso, el funcionamiento es similar al mostrado en la figura 2.3.

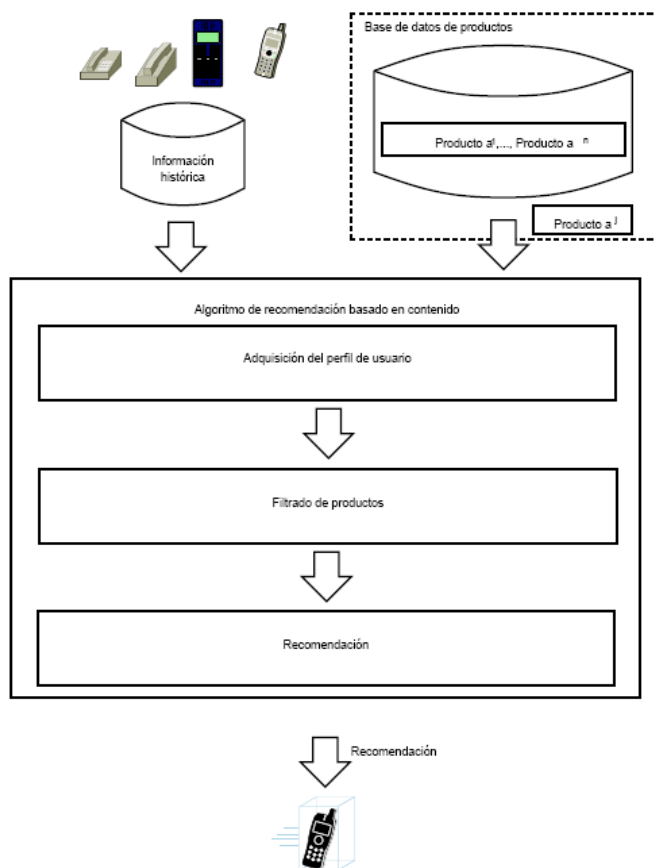


Figura 2.3: Funcionamiento de un SR basado en contenido.

Los Sistemas de Recomendación basados en contenido resultan de gran utilidad para la recomendación de documentos y páginas Web, donde el cálculo de similitud entre objetos da unos resultados bastante satisfactorios.

Este tipo de Sistema de Recomendación presenta dos grandes inconvenientes:

- **Sobre-especialización:** Se da como consecuencia de reducir las recomendaciones a unos contenidos muy similares, sin dar importancia a la posible arbitrariedad de los gustos e intereses del usuario.

• Sólo se puede conocer información parcial de un objeto, normalmente información textual, dejando así de lado información contextual, visual o semántica, la cual puede resultar muy útil para realizar las recomendaciones, pero al mismo tiempo es la más difícil de obtener. Como consecuencia de ello, pueden perderse conexiones no inmediatas entre objetos similares.

Ante estos problemas, se han propuesto múltiples soluciones, entre las que cabe destacar las siguientes:

- Incorporación de cierta aleatoriedad en las búsquedas.
- Indexación semántica latente (LSI) de la información textual.
- Medidas de similitud basadas en ontologías.

No obstante, la mejor solución a dichos problemas ha resultado ser la existencia de una buena realimentación entre el sistema y sus usuarios.

2.2.2. Sistemas de Recomendación Colaborativos.

Se trata de los Sistemas de Recomendación más extendidos y consolidados en el mercado actual. Los Sistemas de Recomendación basados en un filtrado colaborativo son aquellos que realizan recomendaciones basándose en los términos de similitud entre usuarios, esto es, recomiendan objetos que son del agrado de otros usuarios de intereses similares.

Para la realización de un buen Sistema de Recomendación Colaborativo, que ofrezca recomendaciones de calidad, es fundamental emplear un buen algoritmo de filtrado colaborativo. Estos algoritmos se pueden clasificar en dos categorías: los algoritmos basados en memoria o usuario y los algoritmos basados en ítem. La mayor ventaja de las técnicas colaborativas es que son totalmente independientes de la representación interna de los objetos que se pueden recomendar.

Sin embargo, a medida que se ha extendido su utilización, se han detectado problemas como son: la escasez, la escalabilidad y el problema del ítem nuevo. Por ello, han llevado a cabo multitud de estudios y experimentos orientados a minimizar el efecto de estos problemas.

Este modelo de Sistemas de Recomendación constituye el principal eje en torno al cual versa el presente proyecto, por lo que veremos un estudio más detallado de los mismos en un epígrafe posterior.

2.2.3. Sistemas de Recomendación Basados en Conocimiento

Los Sistemas de Recomendación basados en conocimiento realizan un razonamiento sobre las necesidades de un usuario y sus preferencias, haciendo inferencias para sugerir recomendaciones [23]. Este tipo de Sistemas de Recomendación emplea el conocimiento que tiene sobre usuarios y productos para generar recomendaciones sobre qué productos satisfarán las necesidades o gustos del usuario.

En la siguiente figura podemos ver un ejemplo del funcionamiento de los Sistemas de Recomendación basados en conocimiento, utilizando un razonamiento basado en casos para generar las recomendaciones:

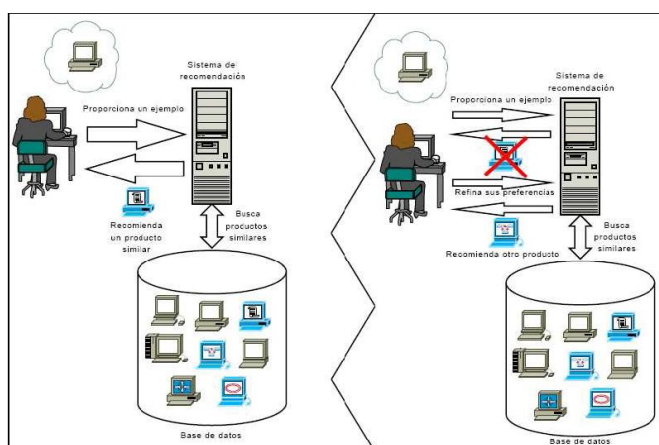


Figura 2.4: Funcionamiento de los Sistemas de Recomendación basados en conocimiento.

Veamos las fases que sigue el esquema de funcionamiento:

1. *Fase de obtención de un ejemplo de las necesidades del usuario:* El usuario proporciona al sistema un ejemplo que represente sus necesidades.

2. *Fase de búsqueda de productos que cumplen las necesidades del usuario:* A partir de la descripción del ejemplo proporcionado por el usuario, el SR buscará otros productos similares a él y los devolverá como recomendaciones (Figura 2.4: Parte izquierda).

3. *Fase de refinamiento:* En algunas circunstancias puede ocurrir que los productos recomendados a partir del ejemplo no cubran con exactitud las necesidades del usuario. En estos casos, el usuario debe refinar las características del producto seleccionado como ejemplo, añadiendo y/o modificando nuevas características (Figura 2.4: Parte derecha).

La principal ventaja de este tipo de sistemas es que, a diferencia de otros Sistemas de Recomendación, los basados en conocimiento no necesitan grandes cantidades de información sobre los objetos evaluados (contenido) o sobre los perfiles de los usuarios (colaborativos). Sólo necesitan un conocimiento general sobre el conjunto de objetos y un conocimiento informal de las necesidades del usuario. Además, presentan muchos beneficios y unos resultados exitosos al ser hibridados con otro tipo de Sistemas de Recomendación.

Los SR basados en conocimiento presentan 2 grandes inconvenientes:

- La mayor parte del conocimiento que utilizan para realizar recomendaciones debe ser explícitamente proporcionado por el usuario, por lo que muchas veces requieren un esfuerzo humano considerable para que las heurísticas de inferencia empleadas realicen recomendaciones de calidad.

- Problemas en la fase de refinamiento: Es muy difícil que un usuario encuentre un ejemplo exacto de lo que necesita. Por ello, es bastante probable que éste deba refinar su perfil de usuario modificando o añadiendo algunas de las características del ejemplo dado.

2.2.4 Sistemas de Recomendación Híbridos

Como hemos mencionado, cada uno de los modelos de recomendación anteriormente vistos presenta, además de sus ventajas, algún tipo de problema. Por ello, surgió la idea de solventar esta situación aunando sus puntos fuertes e intentando minimizar sus inconvenientes, mediante la hibridación de al menos dos de los diferentes tipos de Sistemas de Recomendación. De esta forma nacieron los Sistemas de Recomendación híbridos.

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

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

2.3. Ejemplos Reales de Sistemas de Recomendación en Internet

A continuación, citaremos algunos Sistemas de Recomendación que pueden ser encontrados en conocidos portales de Internet.

2.3.1. YouTube

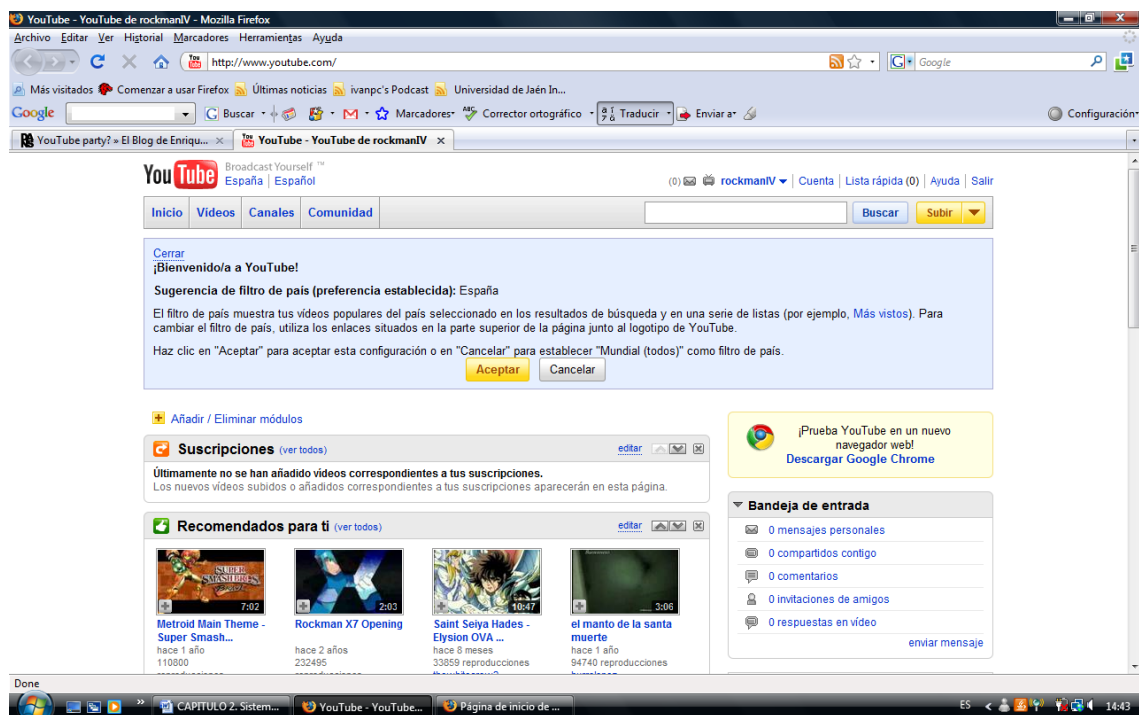


Figura 2.5: Portal de YouTube en Internet

El ya famosísimo portal de difusión de vídeos a través de Internet, Youtube, ha incorporado recientemente un Sistema de Recomendación basado en el contenido, en el que los usuarios registrados obtienen un conjunto de vídeos recomendados, similares a aquellos que suelen ver con mayor frecuencia o durante un porcentaje prolongado de la duración de estos. Dicho modelo de recomendación se plasma en una interfaz fácilmente editable y flexible.



Figura 2.6: Interfaz de recomendaciones de YouTube

2.3.2. Amazon

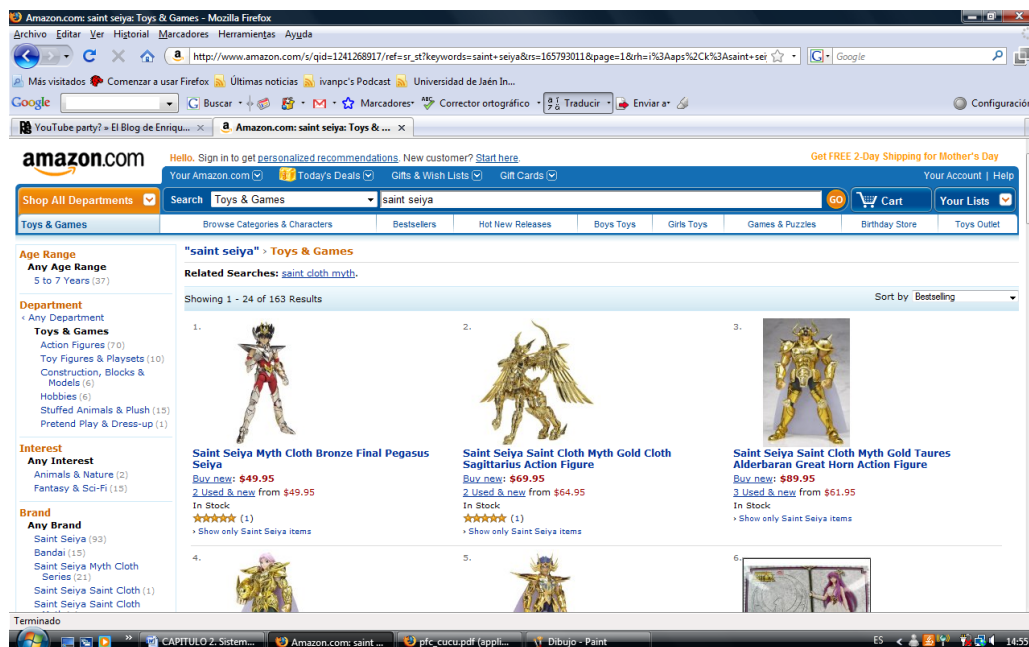
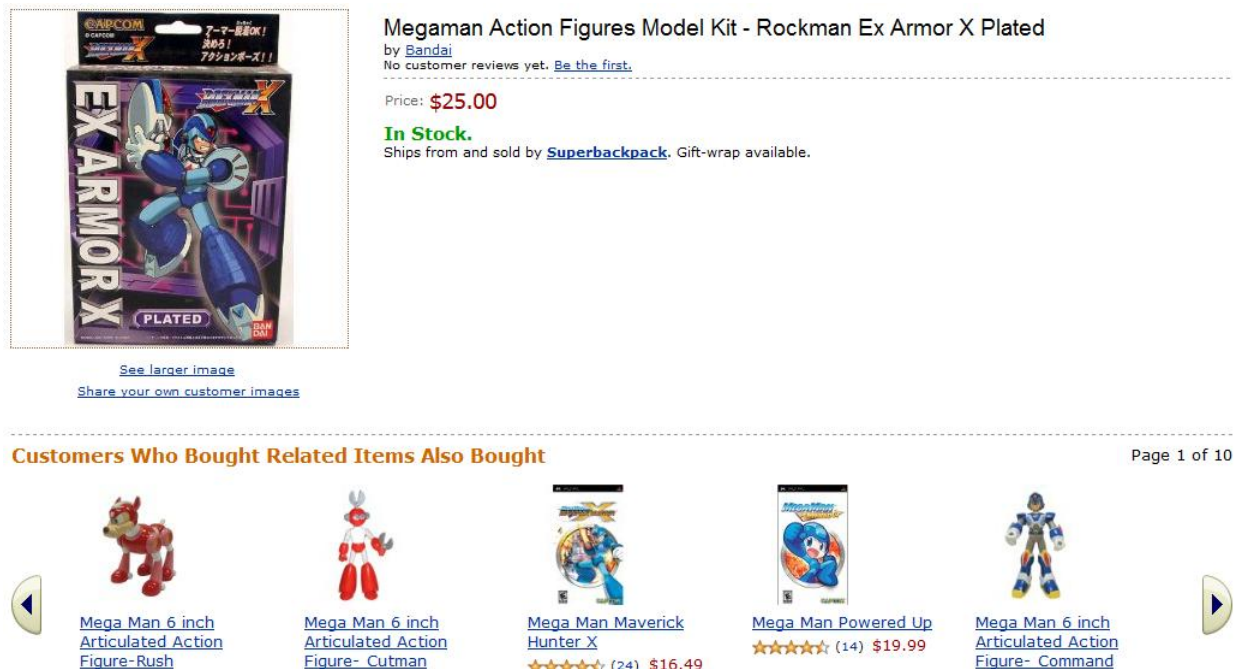


Figura 2.7: Portal de Amazon en Internet

La popular empresa norteamericana de comercio electrónico es ya todo un referente en el mundo de los Sistemas de Recomendación, con un paradigma que combina las ventajas de los enfoques colaborativo y basado en contenido. El sistema guarda las preferencias del usuario para posteriormente combinarlas con objetos relevantes para realizar las recomendaciones. Dichas recomendaciones se muestran al usuario en el tradicional formato: “People who bought this ítem ... also bought...”).



Megaman Action Figures Model Kit - Rockman Ex Armor X Plated
by [Bandai](#)
No customer reviews yet. [Be the first.](#)

Price: **\$25.00**

In Stock.
Ships from and sold by [Superbackpack](#). Gift-wrap available.

[See larger image](#)
[Share your own customer images](#)

Customers Who Bought Related Items Also Bought Page 1 of 10



[Mega Man 6 inch Articulated Action Figure-Rush](#)



[Mega Man 6 inch Articulated Action Figure- Cutman](#)



[Mega Man Maverick Hunter X](#)
★★★★☆ (24) \$16.49



[Mega Man Powered Up](#)
★★★★☆ (14) \$19.99



[Mega Man 6 inch Articulated Action Figure- Command](#)

Figura 2.8: Interfaz de recomendación de Amazon

2.3.3. IMDb Recommendation Center

La Internet Movie DataBase (IMDb) es la mayor base de datos del mundo con información en línea sobre películas, directores, actores, series de televisión, etc. La IMDb cuenta con un motor de recomendaciones basado en contenido, llamado Recommendation Center.



Figura 2.9: Interfaz del centro de recomendación de IMDb



Figura 2.10: Interfaz del centro de recomendación de IMDb

El usuario introduce el programa de televisión o película que le guste y el sistema le ofrece una lista con al menos diez recomendaciones (Figura 2.10). Además, incorpora un mecanismo de realimentación en el que el usuario indica los elementos que no sean de su agrado entre aquellos que le han sido recomendados, para así ir recibiendo recomendaciones nuevas, más depuradas. Pese a ser muy famoso, no es un motor de recomendación demasiado bueno, por lo que no es el punto fuerte de esta compañía.

2.4. Sistemas de Recomendación Colaborativos

Debido a que este proyecto se sustenta sobre un algoritmo de filtrado colaborativo, a continuación vamos a hacer una revisión un poco más en detalle de este tipo de Sistemas de Recomendación [6].

2.4.1. Introducción y Orígenes

Los Sistemas de Recomendación Colaborativos son aquellos que realizan recomendaciones basándose únicamente en términos de similitud entre los usuarios. Es decir, combinan las valoraciones de los objetos, identifican los gustos comunes entre usuarios en base a dichas valoraciones y recomiendan así objetos que son del gusto de otros usuarios de gustos similares al usuario actual.

Las técnicas para desarrollar los primeros Sistemas de Recomendación de filtrado colaborativo estaban basadas en métodos provenientes de la minería de datos [2]. Para ello, se distinguía entre una fase de aprendizaje (offline) en la que se aprende el modelo, al igual que ocurre en la minería de datos, y una fase de recomendación (online) en la que se aplica el modelo obtenido de la fase anterior a un problema de la vida real, produciéndose así las recomendaciones para los usuarios del sistema. No obstante, actualmente este tipo de técnica no se suele utilizar, ya que debido a la interacción de los usuarios con el sistema es más

conveniente emplear un paradigma de aprendizaje relajado (el modelo se construye y actualiza durante el funcionamiento del sistema).

La base teórica de los Sistemas de Recomendación es simple: se forman grupos de usuarios más cercanos, que serán los que mantengan unos perfiles similares, y a un usuario de un grupo se le recomiendan objetos que aún no ha experimentado, pero sí han experimentado y valorado positivamente otros usuarios de su “grupo”.

2.4.2. Fases y Ejemplo de Funcionamiento de los SR Colaborativos

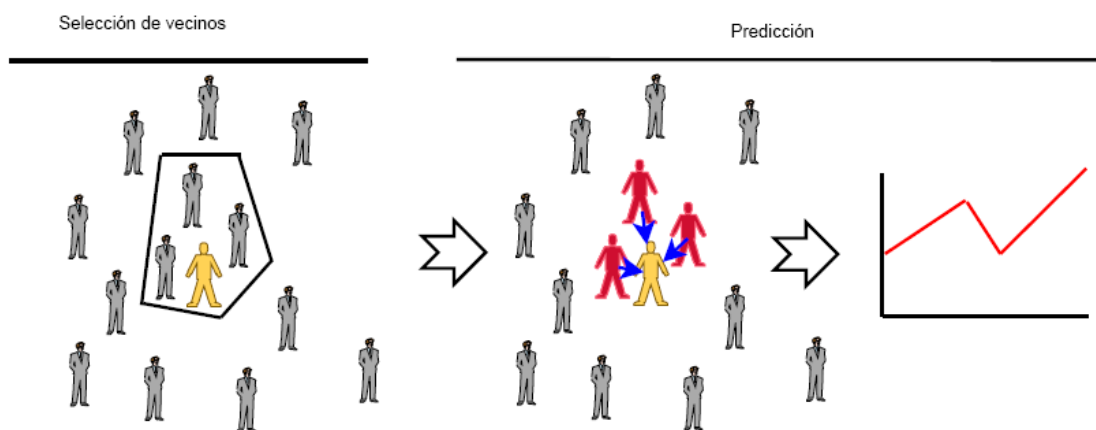


Figura 2.11: Funcionamiento de los Sistemas de Recomendación Colaborativos.

Shardanand y Maes [10] distinguieron tres etapas fundamentales en el funcionamiento de todo Sistema de Recomendación Colaborativo, que puede verse gráficamente en la Figura 2.11.

1. El sistema guarda un perfil de cada usuario, que consta de las evaluaciones de objetos conocidos por él y que pertenecen a la base de datos sobre la que se trabajará.
2. En base a estos perfiles, se mide el grado de similitud entre los usuarios del sistema y se crean grupos de usuarios con características similares.

- El sistema utiliza la información obtenida en los dos pasos anteriores para calcular las predicciones (Figura 2.12). A cada usuario se le recomendarán objetos que no haya evaluado previamente y que hayan obtenido los mayores valores para dicha predicción.

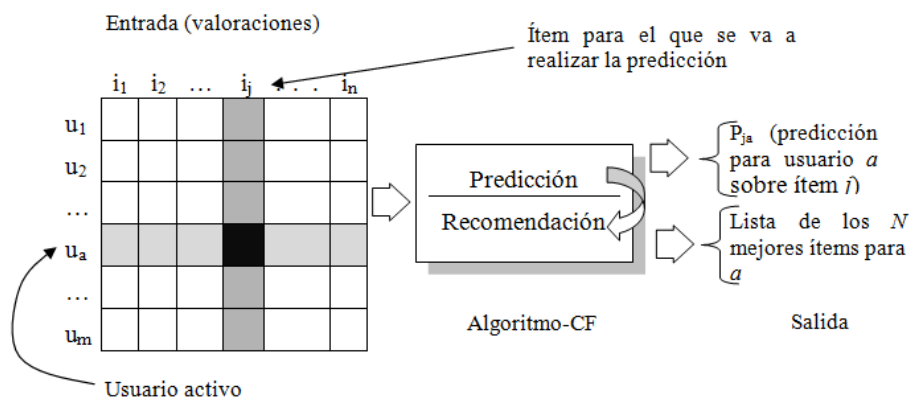


Figura 2.12: Proceso de cálculo de predicciones en SR Colaborativos.

Por tanto, estos Sistemas de Recomendación no toman en consideración el contenido y las características de los productos que recomiendan, sino que sean del gusto de usuarios con un perfil semejante al del usuario que demanda el servicio.

Para entender mejor el funcionamiento de estos sistemas, utilizaremos el siguiente ejemplo que muestra una relación de usuarios y salones de celebraciones de la provincia:

	Juan	Luis	Manuel	Pedro	Antonio
Sierra Mágina	+	-	+	+	-
Puerta Cerro	+	+	+	+	+
Mistral	-	+	-	+	+
California	+	-	+	-	-
Ruta del Sol	-	+	-	+	?

Figura 2.13: Ejemplo de funcionamiento de SR Colaborativos.

El símbolo “+” significa que al usuario le ha gustado la descripción del lugar, el símbolo “-” indica que al usuario no le ha gustado, y el símbolo “?” representa la recomendación que queremos calcular. Queremos saber, pues, la valoración que podría tener Antonio sobre los salones Ruta del Sol. Para hacer esta predicción, buscaremos usuarios que tengan un patrón de valoraciones (perfil) similares a Antonio. Podemos ver que Luis tiene unas valoraciones muy parecidas a las de Antonio, por lo que podríamos suponer que, dado que a Luis le gustan los salones Ruta del Sol, a Antonio también le gustarán. Esta solución no sería del todo correcta, ya que al basarse solo en un usuario con gustos parecidos puede ofrecer recomendaciones equivocadas.

Lo ideal es buscar el grado de correlación entre Antonio y cada uno de los usuarios del sistema, y usar dicho grado para realizar una media ponderada de sus valoraciones. Una forma de obtener el grado de correlación entre un usuario “x” y otro usuario “y” es mediante el coeficiente de correlación de Pearson, donde $r(x,y)$ representa la valoración del usuario “x” sobre el producto “y”. Así, la correlación entre los usuarios “x” e “y” se puede obtener de la siguiente forma:

$$r(x,y) = \frac{\sum_{d \in \text{Productos}} (R_{x,d} - \bar{R}_x) (R_{y,d} - \bar{R}_y)}{\sqrt{\sum_{d \in \text{Productos}} (R_{x,d} - \bar{R}_x)^2 (R_{y,d} - \bar{R}_y)^2}}$$

Donde $\bar{R}_x$ es el valor medio de las valoraciones hechas por el usuario “x”.

Siguiendo con el ejemplo, sabemos que la correlación entre Antonio y Luis es 1, entre Antonio y Pedro es 0.577, entre Antonio y Juan es 0.577, y entre Antonio y Manuel es -0.577. Si calculamos media ponderada de la valoración de cada usuario sobre los salones Ruta del Sol por su correlación con Antonio, obtendremos un valor de 0.682.

Un algoritmo de filtrado colaborativo predeciría, pues, que a Antonio le gustarían los salones Ruta del Sol. En el siguiente apartado veremos una clasificación de los algoritmos de filtrado colaborativo.

2.4.3. Algoritmos de Filtrado Colaborativo

Para desarrollar un buen Sistema de Recomendación Colaborativo, es de vital importancia elegir un buen algoritmo de filtrado colaborativo [5]. Existen distintas posibilidades a la hora implementar dicho algoritmo. Veamos con un poco de detalle los diferentes tipos de algoritmos de filtrado [8].

- Algoritmos basados en memoria o basados en usuario.

Estos algoritmos realizan las recomendaciones basándose en la base de datos completa, teniendo en cuenta todos los ítems previamente evaluados por el usuario. El funcionamiento de los algoritmos es el siguiente: se utilizan técnicas estadísticas para determinar un conjunto de vecinos al usuario objetivo (Figura 2.14) y, posteriormente, se aplican algoritmos que combinan preferencias de esta vecindad para realizar las predicciones y recomendaciones.

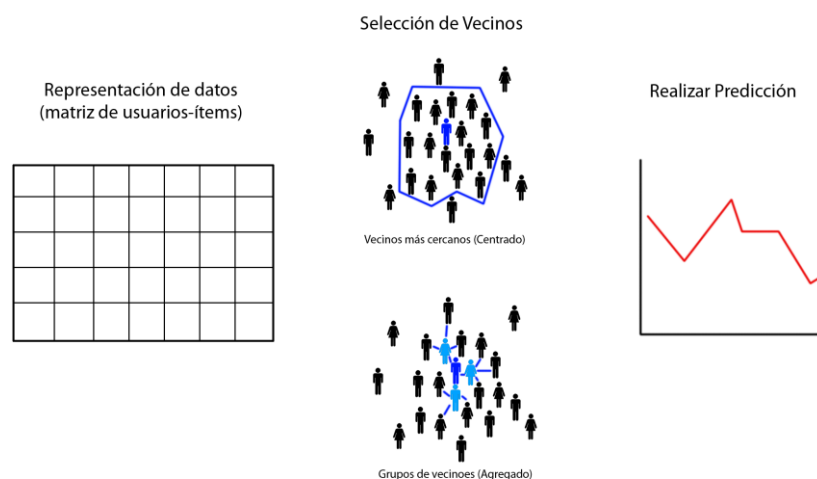


Figura 2.14: Las principales partes de un SR Colaborativo basado en memoria

Pese a ser bastante populares y exitosos en la práctica, suelen sufrir especialmente los problemas de escalabilidad y escasez. Se hizo necesario, pues, el desarrollo de otro tipo de algoritmos de filtrado colaborativo.

- Algoritmos basados en modelos o basados en ítem

Estos algoritmos se centran en desarrollar un modelo de las puntuaciones de los usuarios sobre los ítems, para proporcionar las recomendaciones. Se utiliza para ello una aproximación probabilística que calcula el valor esperado de una predicción del usuario a partir de sus puntuaciones sobre otros ítems.

Dado que este es el tipo de algoritmo sobre el que se centrará el presente proyecto, pasamos a continuación a describir distintas técnicas para obtener el modelo de recomendación.

- *Redes bayesianas*: Esta técnica consiste en obtener el modelo a partir de un conjunto de entrenamiento con un árbol de decisión, donde los nodos y ramas representan información de los usuarios. Suele resultar útil cuando el conocimiento sobre el perfil de usuario cambia de forma lenta.
- *Modelos basados en reglas*: El procedimiento es similar al de la técnica anterior, con la diferencia de que el modelo obtenido tiene la forma de un conjunto de reglas del tipo “Antecedente $\rightarrow$ Consecuente”.
- *Técnicas de agrupamiento (clustering)*: Hacen grupos de usuarios, denominados *clusters*, con gustos similares. Las predicciones para un individuo se realizan mediante la agregación de opiniones de otros usuarios del mismo grupo. La tarea de creación de los *clusters* conlleva un gran esfuerzo, pero una vez creados se obtiene un rendimiento bueno, ya que el grupo a considerar para realizar las recomendaciones queda considerablemente reducido.

- *Modelos basados en vecino más cercano*: Estos algoritmos proporcionan recomendaciones desarrollando en primer lugar un modelo, utilizando cualquiera de las tres técnicas anteriormente descritas, de las puntuaciones de los usuarios sobre los ítems.

Estos algoritmos miran en el conjunto de ítems evaluados por el usuario activo para calcular cómo de parecidas son estas puntuaciones al ítem activo, con el fin de realizar una predicción para el mismo. Para realizar las recomendaciones se realizan las siguientes tareas:

1. Exploración del conjunto de ítems que el usuario ha valorado.
2. Cálculo de la similitud de los ítems anteriores con respecto al ítem o producto del cual queremos predecir la puntuación que el usuario le daría.
3. Selección de los k ítems más cercanos (Knn).
4. Cálculo de la predicción del usuario objetivo sobre el ítem dado como la media ponderada de las valoraciones del usuario hacia los k ítems más similares.

Dentro de los Sistemas de Recomendación Colaborativos basados en ítem, emplearemos este último modelo para el algoritmo de filtrado de este proyecto.

2.4.4. Medidas y Técnicas Empleadas en Algoritmos de Filtrado Colaborativo

Una vez estudiada la clasificación de algoritmos de filtrado colaborativo, pasamos a definir algunas medidas empleadas para realizar las recomendaciones en este tipo de sistemas.

El primer paso para la realización de dichas recomendaciones consiste en formar grupos con los usuarios o ítems más similares entre sí. Para ello se emplearán medidas de similitud y un algoritmo de clasificación K-nn [7]. Acto seguido, se empleará una técnica de predicción para estimar la valoración del usuario sobre ciertos ítems.

2.4.4.1. Notación

Antes de mostrar formalmente la formulación del problema, es conveniente dejar clara la notación que se va a emplear para no producir confusión al lector:

- Un usuario es aquel elemento representado por $u_i \in U = \{u_1, u_2, \dots, u_n\}$
- Un ítem será aquel elemento representado por $i_j \in I = \{i_1, i_2, \dots, i_m\}$
- La similitud entre dos ítems i_j e i_k vendrá dada por $s(i_j, i_k)$
- Una evaluación de un usuario u_i sobre un ítem i_j vendrá dada por r_{u_i, i_j}
- Una predicción de un usuario u_i sobre el ítem i_j se representará como p_{u_i, i_j}

2.4.4.2. Algoritmo K-nn

Un paso crucial en la realización de un Sistema de Recomendación Colaborativo de calidad es la formación de grupos de usuarios (si es un SR Colaborativo basado en memoria) o de ítems (si es basado en modelos, como ocurre en este proyecto) de características similares. Esta actividad forma parte de lo que conocemos como problemas de clasificación, y existen diversas técnicas, llamadas clasificadores, para resolver dicho problema. Uno de los clasificadores más difundidos es el algoritmo K-nn, que se utilizará en este proyecto para formar los grupos de ítems más similares para cada uno de los ítems de la base de datos.

Veamos con mayor detalle en qué consiste el algoritmo K-nn:

a) Funcionamiento

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

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

-Voto por la mayoría: Se clasifica el nuevo objeto según la clase predominante en los objetos de K .

-Voto compensado por la distancia: Se clasifica el objeto en base a su distancia ponderada con el resto de objetos de K .

b) Descripción del algoritmo

1. Se separan los datos en dos conjuntos disjuntos: entrenamiento (E) y test (T).

2. Aparece un nuevo objeto i_a .

3. Se obtienen los k objetos del conjunto E más cercanos a i_a

4. Se clasifica el objeto i_a de una de las dos formas antes mencionadas.¹

c) Principales características

-Es un algoritmo robusto frente al ruido cuando k es moderado ($k > 1$).

-Es muy eficaz cuando el número de clases posibles es alto y cuando los datos son heterogéneos o difusos.

-Tiene un orden de complejidad de $O(dn^2)$, siendo $O(d)$ la complejidad de la distancia de métrica empleada.

-El hecho de no utilizar modelos sino la base de datos al completo provoca que sea ineficiente en memoria.

¹ En este proyecto nos limitamos a utilizar el algoritmo K-nn para seleccionar los k vecinos más cercanos a cada uno de los objetos, por lo que no necesitamos implementar este último paso.

-Sirve tanto para realizar clasificación como para predicción numérica.

En nuestro proyecto se empleará el algoritmo K-nn para seleccionar los k ítems más similares a cada uno de los ítems que componen nuestra base de datos.

2.4.4.3. Medidas de Similitud

Para establecer la similitud entre objetos debemos definir una medida que nos permita evaluar el grado de parecido entre unos y otros.

Es importante resaltar que, en este proyecto, para calcular la similitud entre dos ítems x e y se tendrán en cuenta únicamente a aquellos usuarios que hayan evaluado a ambos ítems, no siendo tomados en consideración el resto.

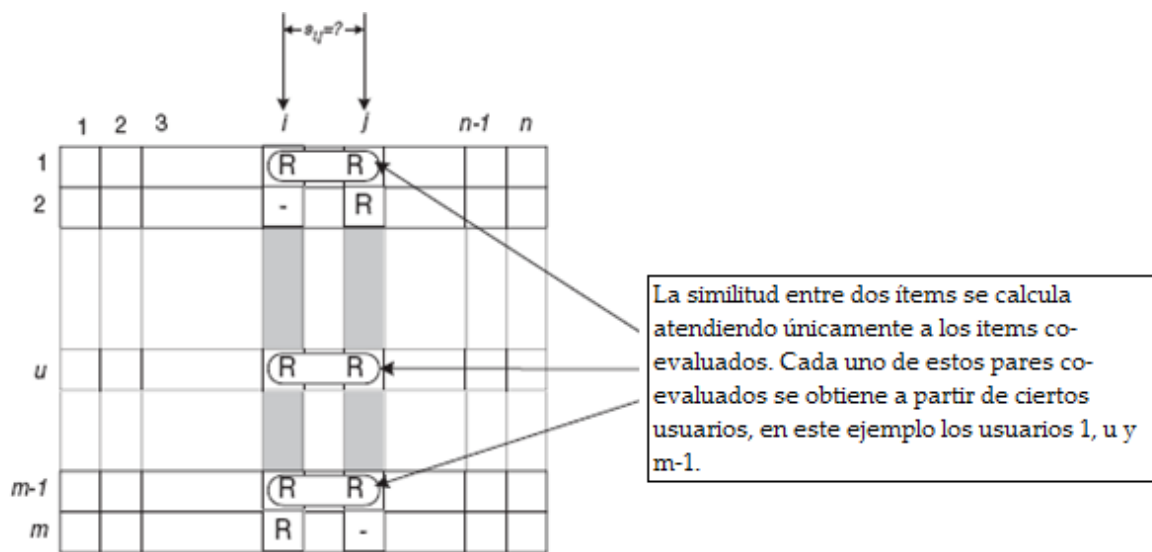


Figura 2.15: Cálculo de similitud basada solamente en ítems co-evaluados.

Existen en la literatura multitud de medidas de similitud [4][10], de entre todas, nosotros vamos a revisar dos de las más utilizadas.

1. Coeficiente de Correlación de Pearson

Este coeficiente es un índice que mide la relación lineal entre dos variables cuantitativas, siendo independiente de la escala de medida de dichas variables y dando un resultado dentro del intervalo $[-1,1]$. Se calcula mediante la siguiente expresión:

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

Siendo $\bar{x}$ e $\bar{y}$ la media de las valoraciones obtenidas para los ítems x e y .

2. Coeficiente Coseno

Este método supone que dos ítems x e y vienen representados por vectores en el espacio, por lo que la similitud entre ellos vendrá dada por el coseno del ángulo que forman. La expresión para su cálculo es la siguiente:

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

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

En este proyecto se ha elegido el coeficiente coseno como medida de similitud para la implementación del algoritmo de filtrado colaborativo.

2.4.4.4 Medidas de Predicción

Tras haber calculado el conjunto de vecinos para cada ítem, hemos de combinar las valoraciones de dicho conjunto para realizar la predicción del usuario sobre dicho ítem. Elegir la técnica adecuada para realizar la predicción supone el paso más crucial del filtrado colaborativo.

La selección de un algoritmo de predicción u otro depende de la naturaleza del conjunto de datos, ya que cada algoritmo se ajusta mejor a un conjunto de datos específico. En nuestro caso, utilizaremos la técnica llamada **weighted sum**. Este método calcula la predicción de un ítem i por parte de un usuario u_a como la suma de las evaluaciones del usuario u_a sobre ítems similares a i . Cada una de estas evaluaciones viene ponderada por la correspondiente similitud $s(i,j)$ entre los ítems i y j . Su expresión viene dada por:

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

Siendo k los k ítems más similares a i_a . Esta técnica intenta captar cómo el usuario activo evalúa a ítems similares al que quiere predecir. Para asegurarnos de que la predicción entra del rango previamente definido, es necesario ponderar estas evaluaciones con la similitud.

Ya hemos sentado las bases teóricas sobre las que se apoya nuestro proyecto para implementar nuestro algoritmo de filtrado colaborativo. En el siguiente apartado citaremos los principales problemas que han surgido de la utilización de Sistemas de Recomendación Colaborativos.

2.4.5. Problemas de los Sistemas de Recomendación Colaborativos

Como hemos mencionado con anterioridad, los Sistemas de Recomendación Colaborativos presentan una serie de problemas que se han ido poniendo de relieve a medida que su utilización se ha extendido: escasez, escalabilidad y problema del ítem nuevo. Para tener un mejor conocimiento de estos, los describimos brevemente a continuación:

- Escasez

Dada su naturaleza, los Sistemas de Recomendación Colaborativos requieren una gran cantidad de usuarios que realicen un igualmente grande número de puntuaciones sobre ítems similares, para poder así calcular los grupos de ítems cercanos y realizar recomendaciones de calidad. Si el número de usuarios registrados en el sistema es pequeño, o incluso si siendo elevado no han hecho suficientes puntuaciones, los cálculos de vecindad, predicción y recomendación serán deficientes y darán como consecuencia recomendaciones de poca calidad.

- Escalabilidad

Para obtener la similitud entre usuarios, los Sistemas de Recomendación Colaborativos utilizan algoritmos de cálculo del vecino más cercano (Knn). El problema de estos algoritmos es su elevado coste computacional, que crecerá de forma lineal respecto al número de elementos existentes en la base de datos.

- Problema del ítem nuevo

Este problema repercute sobre los dos elementos principales del Sistema de Recomendación, los usuarios y los ítems. En esencia, un ítem nuevo en el sistema apenas tendrá puntuaciones respecto a otros ítems ya existentes, por lo que no van a ser recomendados prácticamente nunca. En cuanto a los usuarios, un usuario nuevo en el

sistema habrá realizado pocas puntuaciones sobre los ítems existentes. Esto provoca que encuadrarlos en un grupo de vecinos adecuado sea una labor compleja, por lo que las recomendaciones que reciba serán pobres.

En los últimos tiempos se ha realizado una gran cantidad de experimentos, estudios e investigaciones de cara a reducir el impacto de estos problemas.

Para reducir el problema de la escasez se han utilizado técnicas de puntuaciones implícitas, correlación entre ítems y filtrado híbrido. Para tratar el problema de la escalabilidad se ha optado por una técnica muy empleada en el análisis de datos [7]: la reducción de la dimensionalidad, además de aproximaciones basadas en modelos. Finalmente, para solventar el problema del ítem nuevo se han propuesto técnicas de minería de datos Web, como por ejemplo las relativas a árboles de decisión.

Tras haber estudiado los principales problemas de los Sistemas de Recomendación Colaborativos, y las estrategias para reducirlos, pasamos a mostrar algunos ejemplos reales de SR Colaborativos existentes en la Web.

2.4.6. Sistemas de Recomendación Colaborativos en Internet

Tras estudiar los fundamentos teóricos de los Sistemas de Recomendación Colaborativos, pasamos a ver algunos ejemplos reales de sitios Web que emplean este tipo de sistemas.

2.4.6.1. Zagat



Figura 2.16: Portal de Zagat.com

Zagat Survey es una empresa norteamericana fundada en 1979, dedicada a la edición de todo tipo de guías de restaurantes, tiendas, hoteles y clubes de diferentes ciudades de los Estados Unidos y Canadá. El usuario registrado puede valorar hasta 30 aspectos diferentes del local referido, además de introducir breves comentarios dando su opinión o experiencia en el mismo. A partir de estas valoraciones, los responsables de la empresa asignan sus puntuaciones en sus guías anuales, que servirán para realizar las recomendaciones a los usuarios a través de la Web.

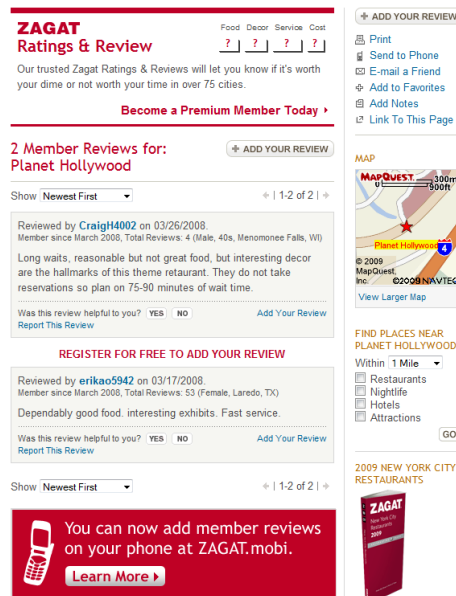


Figura 2.17: Opiniones de usuarios en Zagat.com

2.4.6.2. MovieLens

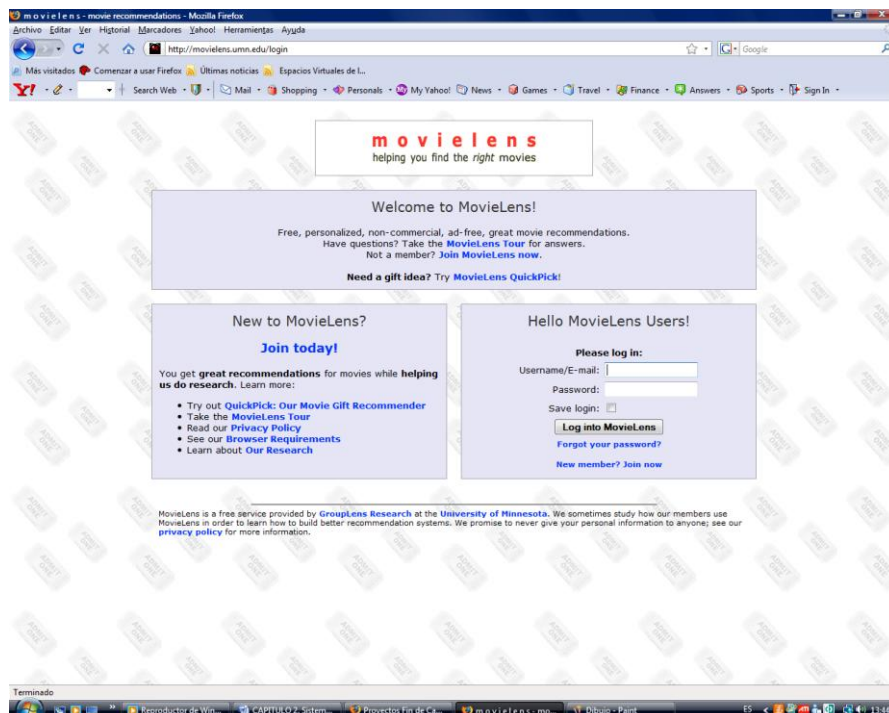


Figura 2.18: Portal de MovieLens

MovieLens (Figura 2.18) 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 recopilar puntuaciones de los usuarios sobre películas, y emplear esta información para agrupar a los usuarios de gustos similares. Así, 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.

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

CAPÍTULO 3

Radios Colaborativas en Internet

3.1. Introducción a la difusión de contenidos multimedia online

Hace algunos años, Internet no era más que un vago concepto para la mayoría de las personas. Hoy día, la Web tiene cientos de millones de usuarios habituales en todo el mundo, y la Web tal y como hoy la conocemos es muy diferente de lo que era hace apenas unos años. Los sitios Web que contienen únicamente texto e imágenes estáticas están siendo rápidamente reemplazados por un nuevo concepto de sitio Web [16], en el que podemos realizar acciones tan variadas como ver programas de televisión, hablar por video y audio-conferencias, escuchar clips de música, escuchar estaciones de radio en directo, e incluso asistir a una clase a distancia.

Así, el concepto de *Streaming multimedia* [17] está actualmente presente en la Web: nos referimos, por ejemplo, a emisoras de radio o televisión, retransmisión de conciertos, etc. Son cada vez más las compañías que proporcionan vídeo y audio a través de servidores multimedia distribuidos.

El gran impulso y aceptación que hasta el momento se le ha dado al contenido interactivo y dinámico de elementos multimedia transmitidos a través de la red está cambiando la forma en que las personas se comunican y acceden a la información. Este gran impulso es debido al gran desarrollo de las tecnologías de la información y la comunicación en los últimos años.

3.2. Música en la Web: Streaming

Streaming es un término utilizado para denominar la tarea de ver u oír un archivo multimedia (vídeo, audio...) de forma directa en una página Web, sin necesidad de descargar dicho archivo en el ordenador. Se podría describir como “hacer un clic y obtener”. En términos más específicos, puede definirse como sigue:

Una estrategia sobre demanda para la distribución de contenido multimedia a través de Internet.

Así, el Streaming de audio hace posible escuchar música a través de una página Web sin necesidad de descargar los archivos musicales a escuchar, ya que estos quedan almacenados en un buffer en el instante en que están siendo reproducidos.

Antes de que la tecnología Streaming apareciese en el año 1995, con el lanzamiento del sistema de transmisión de música online Real Audio 1.0, era necesario descargar el archivo de audio en el disco duro del usuario para la reproducción de música existente en Internet. Dado que estos archivos suelen tener un tamaño significativo, su descarga en el equipo podía suponer un proceso lento, especialmente en conexiones con límite de descarga no muy elevado.

Sin embargo, el Streaming permite descargar el archivo y reproducirlo al mismo tiempo, mientras aún está descargándose, consiguiendo así un tiempo de espera mínimo. Esta es la principal ventaja de esta tecnología, y lo que la ha hecho tan popular en los últimos años.

El funcionamiento del Streaming es el siguiente.

-En primer lugar, el ordenador del usuario (cliente) se conecta con el servidor (proveedor del sitio Web) y éste inicia el envío del fichero multimedia en cuestión.

-El cliente comienza a recibir el fichero y en ese preciso instante construye un buffer de memoria donde empieza a guardar la información.

-Cuando se ha llenado el buffer con una pequeña parte del archivo, el ordenador cliente empieza a reproducirlo, a la vez que continúa con la descarga. Normalmente el intervalo de tiempo necesario para ello es muy pequeño, casi nulo.

-El sistema está sincronizado para que el archivo se pueda reproducir mientras que el archivo se descarga, de modo que cuando el archivo acaba de descargarse el fichero ya lleva cierta cantidad de tiempo en reproducción.

-Si en algún momento la conexión sufre descensos de velocidad se utiliza la información que hay en el buffer, de modo que se pueda dar un margen temporal de tolerancia a dicho descenso en la conexión. Si la comunicación se cortase demasiado tiempo, el buffer se vaciaría y la ejecución el archivo se cortaría también hasta que se restaurase la conexión.

Hoy en día podemos encontrar multitud de sitios en Internet en los que se saca partido a la tecnología Streaming. Un ejemplo de ello lo tenemos en las llamadas emisoras de radio o **radios online**, concepto en torno al cual gira el objetivo de este proyecto. Se trata aplicaciones Web en las que el usuario tiene acceso a una colección de archivos musicales para escuchar sin necesidad de descargarlos en su ordenador, además de disponer de un perfil personal, crear sus propias listas de reproducción con aquellas canciones que desee, etc. Muchas de estas radios online trabajan incluso sobre la emisión de audio en vivo, como ocurre con las emisoras de radio clásicas.

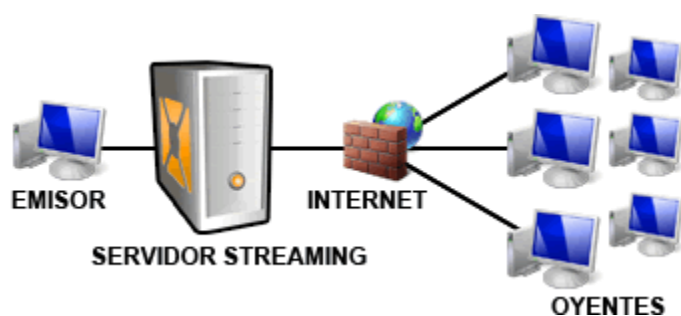


Figura 3.1: Ejemplo de estructura de audio-Streaming en directo.

En el siguiente epígrafe, veremos dos ejemplos de sitios radios online en Internet, provistas además de un Sistema de Recomendación Colaborativo para el usuario, según hemos estudiado en el capítulo 2.

3.3. Ejemplos de radios colaborativas en Internet

Este apartado presentamos dos radios online colaborativas ampliamente conocidas: Last.fm y Pandora.

3.3.1. Last.fm (www.lastfm.es)



Figura 3.2: Portal de Last.fm

Last.fm es una famosísima red social, una radio online y además un Sistema de Recomendación Colaborativo de música, que se basa en los datos de los usuarios registrados para construir perfiles y estadísticas de gustos musicales, para realizar las recomendaciones. El servicio de que dispone es de código abierto. Se originó a partir de un proyecto de informática en la Universidad de Southampton (Reino Unido).

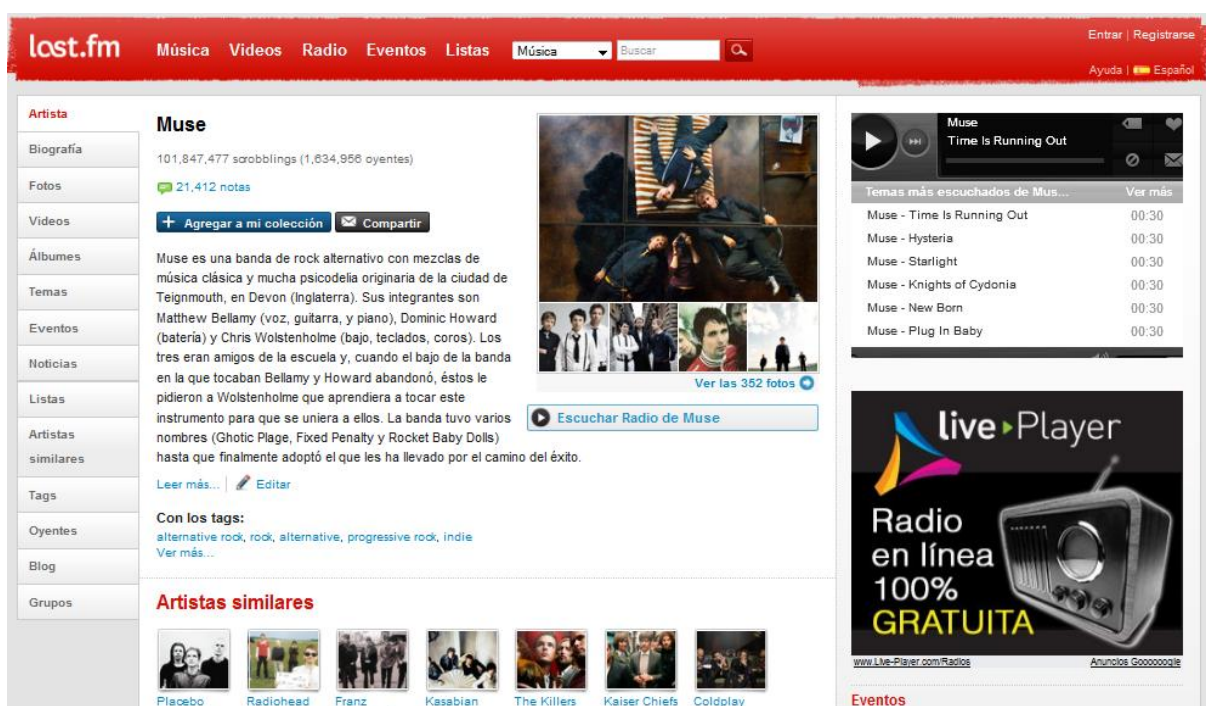


Figura 3.3: Sección de artistas y recomendaciones en Last.fm

Uno de los mayores atractivos de last.fm es la forma de medir el éxito de una canción, atendiendo al número de personas que reproducen la canción. Incluye además un reproductor descargable por el usuario, y un editor del perfil de usuario.

3.3.2. Pandora (www.pandora.com)



Figura 3.4: Portal de pandora.com

Pandora es una radio online que también se basa en un algoritmo de filtrado colaborativo, recomendando a los usuarios canciones que son del agrado de otros usuarios con gustos similares. Su gran punto fuerte respecto a otras radios es la interfaz, muy intuitiva y fácil de utilizar. El usuario se crea su propia emisora de radio, con la ayuda opcional de un asistente en la que posteriormente puede añadir y editar canciones escogidas por él, además de aquellas canciones que le sean recomendadas.

3.4 Licencias de libre distribución de contenidos.

Durante la realización de este proyecto ha sido necesario recopilar una muestra significativa de archivos musicales para formar el conjunto de objetos con los que trabaja nuestro Sistema de Recomendación. Un factor importante a la hora de seleccionar dichos archivos ha sido la licencia bajo la que se encontraban en Internet, siendo las licencias de libre distribución aquellas por las que hemos optado. Por esta razón, dedicaremos este último epígrafe del capítulo a un breve estudio de dichas licencias.

Una licencia de Software u otra obra intelectual consiste en la autorización, por parte del autor de dicha obra para utilizarla de una forma establecida y bajo unas restricciones y derechos legales sobre su uso. Un ejemplo de licencia podría ser aquella que restringe el territorio de utilización de un contenido de audio, la duración de dicha utilización o cualquier otra restricción que el autor del contenido estime oportuna.

En general, la persona que crea la obra puede autorizar o restringir el uso, modificación y/o difusión de la misma, adscribiéndose a un tipo de licencia determinado.

Nos centraremos en las licencias de libre distribución, que por lo general permiten la descarga, utilización y difusión de las obras adscritas a ella de forma gratuita y legal. En particular haremos un breve estudio de las siguientes dos licencias:

- *Copyleft*
- *Creative Commons*[18]

3.4.1. Copyleft



Figura 3.5: Logotipo de la licencia Copyleft

El término *Copyleft* se atribuye a una licencia o grupo de derechos de autor que se caracteriza por la eliminación de las limitaciones de modificación o distribución impuestas por la conocida licencia *Copyright*. La idea es garantizar al receptor de una obra la posibilidad de utilizar, modificar y difundir no sólo la propia obra, sino también posibles versiones derivadas de la misma, modificadas por el propio receptor. La única condición que *Copyleft* impone es que cualquier obra que derive de la modificación de una obra bajo licencia *Copyleft* siga manteniendo dicho régimen de derechos.

Existen varias licencias de tipo *Copyleft*, algunas de ellas son las siguientes:

- **GPL:** La GNU GPL (General Public License) es una licencia creada por la Free Software Foundation a mediados de los 80, y su principal objetivo es proteger la libre distribución, modificación y uso de software, declarando a todo software cubierto por esta licencia como Software Libre.
- **Licencia Arte Libre:** La razón de ser de esta licencia radica en promover y proteger prácticas artísticas liberadas de las reglas de la economía de mercado.
- **Coloriuris:** Esta licencia está destinada a los creadores de contenidos – literarios, musicales, audiovisuales y fotográficos – que utilizan la Web para su

difusión, publicación y/o puesta a disposición y que desean ceder los derechos patrimoniales de sus creaciones dentro y fuera de la red.

Hoy en día, existen producciones creativas de prácticamente cualquier tipo (Software, cultura, arte, obras científicas, etc.) sujetas a la licencia *Copyleft*. En la dirección Web <http://fundacioncopyleft.org/> encontramos toda la información relativa a Copyleft.

3.4.2. Creative Commons



Figura 3.6: Logotipo de la licencia Creative Commons

Para realizar este proyecto se ha optado por esta licencia para buscar los archivos musicales que componen nuestra base de datos de canciones. Por esta razón, merece una descripción detallada que pasamos a hacer a continuación.

Creative Commons es una organización no lucrativa, de origen americano, que establece un punto intermedio entre los derechos de propiedad intelectual de *Copyright* y la total eliminación de dichos derechos en licencias como *Copyleft*. Se basa en la idea de que algunos autores pueden no querer ejercer todos los derechos de propiedad intelectual que la ley les permite, ya que se ha comprobado que a veces el total derecho de copia no ayuda a conseguir toda la difusión que ellos desearían.

Muchos usuarios de Internet buscan compartir sus obras y poder reutilizar y modificar obras en común con otros usuarios. *Creative Commons* se ajusta a dicho propósito,

ofreciendo a su comunidad un conjunto de licencias sin coste alguno. Estas licencias permiten a los autores otorgar de forma voluntaria a otras personas la facultad de utilizar, copiar, distribuir, exhibir o modificar su obra, sujetos a una serie de condiciones.

Es evidente que Internet ha supuesto el mejor escaparate para extender el uso de las licencias *Creative Commons*, ya sea por medio de blogs, sitios Web multimedia con streamig, wiki, etc.

Podemos encontrar información bastante precisa sobre las licencias *Creative Commons* y el modo de utilizarlas en:

<http://es.creativecommons.org/proyecto/faq.php>

3.4.2.1. Creative Commons en España

Son varios los países que ya han incluido y adaptado el uso de las licencias *Creative Commons* en sus respectivas legislaciones, y muchos otros los que están en proceso. La traducción y adaptación de licencias *Creative Commons* fue presentada al régimen español de Derechos de Autor en octubre de 2004. Hasta esta fecha, la mayor parte de usuarios adaptaban sus obras a un modelo internacional de licenciamiento ajustado a las normas jurídicas norteamericanas, que son en muchos aspectos diferentes a las de España.

Las licencias *Creative Commons* vigentes en España varían según las cuatro condiciones en que el autor quiera compartir su obra:

Reconocimiento (*Attribution*): El material creado por un artista puede ser distribuido, copiado y exhibido por terceras personas si se muestra en los créditos.



No Comercial (*Non Commercial*): El material original y los trabajos derivados pueden ser distribuidos, copiados y exhibidos mientras su uso no sea comercial.



Sin Obra Derivada (*No Derivate Works*): El material creado por un artista puede ser distribuido, copiado y exhibido pero no se puede utilizar para crear un trabajo derivado del original.



Compartir Igual (*Share Alike*): El material creado por un artista puede ser modificado y distribuido pero bajo la misma licencia que el material original.



A partir de las cuatro combinaciones básicas se pueden generar las seis licencias que se pueden escoger. En ningún caso los autores trasladan la autoría de sus producciones, solamente autorizan a otras personas, mediante una serie de privilegios expresos, según las facultades o limitaciones establecidas. Las seis licencias disponibles son las siguientes:



Reconocimiento (by): Se permite cualquier explotación de la obra, incluyendo una finalidad comercial, así como la creación de obras derivadas, la distribución de las cuales también está permitida sin ninguna restricción.



Reconocimiento - NoComercial (by-nc): Se permite la generación de obras derivadas siempre que no se haga un uso comercial. Tampoco se puede utilizar la obra original con finalidades comerciales.



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.



Reconocimiento - NoComercial - SinObraDerivada (by-nc-nd): No se permite un uso comercial de la obra original ni la generación de obras derivadas.



Reconocimiento - CompartirIgual (by-sa): Se permite el uso comercial de la obra y de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.



Reconocimiento - SinObraDerivada (by-nd): Se permite el uso comercial de la obra pero no la generación de obras derivadas.

En este proyecto se han utilizado canciones bajo las licencias de *Reconocimiento - SinObraDerivada (by-nd)* y de *Reconocimiento - NoComercial - CompartirIgual (by-nc-sa)*.

CAPÍTULO 4

Desarrollo del Proyecto

4.1. Descripción del Proyecto

Una vez se han sentado las bases teóricas de los Sistemas de Recomendación, así como de las radios colaborativas existentes Internet, es el momento de entrar con profundidad en el desarrollo de este proyecto.

A grandes rasgos, el proyecto que se ha realizado consiste en el desarrollo de una radio por Internet, llamada OL-RADIOUJA, basada en un algoritmo de filtrado colaborativo, para recomendar canciones a los usuarios, siguiendo para todo ello una arquitectura cliente/servidor. Esta radio colaborativa consta de tres componentes fundamentales, que suponen los tres pilares básicos del desarrollo del proyecto:

- Una base de datos, que incluye todos los datos relativos a usuarios del sistema, canciones, puntuaciones de los usuarios sobre las canciones, predicciones...
- Una interfaz para la aplicación, en forma de portal Web, desde la que los usuarios accederán al sistema, escucharán y evaluarán canciones, y recibirán recomendaciones de música a modo de listas de reproducción.
- Un algoritmo de filtrado colaborativo, que se encarga de calcular las predicciones a partir de los datos de puntuaciones, usuarios y canciones existentes en la base de datos.

Tras haber expuesto una presentación y una visión general del proyecto, con su propósito y objetivos [11], una introducción teórica a los Sistemas de Recomendación y a las radios colaborativas en Internet, es el momento de detallar, en base a técnicas de Ingeniería del Software, el proyecto realizado. La razón de ello es sencilla: estamos ante un proyecto de desarrollo software y, como tal, hemos de seguir la metodología de la Ingeniería del Software.

No existe una definición única y estandarizada para la Ingeniería del Software. Pese a ello, las dos que pasamos a exponer a continuación resultarán perfectamente válidas para nuestro cometido.

La Ingeniería del Software consiste en la construcción de software de calidad con un presupuesto limitado y un plazo de entrega en contextos de cambio continuo.

La Ingeniería del Software es el establecimiento y uso de principios y métodos firmes de ingeniería, para obtener software económico que sea fiable y funcione de manera eficiente en máquinas reales.

La Ingeniería del Software se compone de las siguientes actividades [15]:

- Especificación de requerimientos: Se obtiene el propósito del sistema, así como las propiedades y restricciones impuestas sobre el mismo.
- Análisis del sistema: Se obtiene un modelo del sistema correcto, completo, consistente, claro y verificable.
- Diseño del sistema: Se establecen los objetivos del proyecto y las estrategias a seguir para conseguirlos.
- Implementación: Consiste en la traducción del modelo lógico del sistema a código fuente.
- Pruebas: Verificar y validar el sistema.

En adelante profundizaremos en cómo se han llevado las citadas etapas durante el desarrollo de nuestro proyecto.

4.2. Especificación de Requerimientos

El primer paso en el proceso de Ingeniería del Software debe ser determinar el propósito último del proyecto, las propiedades que debe satisfacer y las restricciones a las que ha sido impuesto. Este es, sin duda, un paso de vital importancia dentro del desarrollo de un proyecto software, ya que sin conocer el propósito del mismo ni establecer las diferentes limitaciones que debe afrontar, será realmente difícil realizar una aplicación software capaz de cumplir dicho propósito.

En un proyecto de ámbito comercial para una empresa real, la forma de determinar el propósito es mediante una serie de estudios como pueden ser: entrevista con los clientes, encuestas con posibles usuarios, observaciones en vivo de la actividad diaria en la empresa, estudios de viabilidad o análisis de la situación actual de la empresa. En nuestro caso, estamos ante un proyecto de ámbito académico, por lo que el propósito es conocido desde la concepción del mismo:

Creación de una radio colaborativa personalizada, mediante un algoritmo de filtrado colaborativo basado en ítem y provista de una interfaz Web. Dicha radio debe dar al usuario recomendaciones de música, en forma de lista de reproducción, basada en las preferencias musicales del mismo, y permitir en todo momento evaluar canciones, hayan sido o no previamente evaluadas.

Tras haber definido el propósito último del proyecto, el siguiente paso consiste en especificar los requerimientos del mismo. Los requerimientos de un proyecto software son el conjunto de propiedades o restricciones, definidas de forma totalmente precisa, que dicho proyecto ha de satisfacer. Existen dos tipos bien diferenciados de requerimientos:

- **Requerimientos funcionales:** Aquellos que se refieren expresamente al funcionamiento del sistema.

- **Requerimientos no funcionales:** Son todos aquellos requerimientos no referidos al estricto funcionamiento del sistema, sino a otros factores externos.

En los dos epígrafes siguientes, definiremos estos requerimientos (tanto funcionales como no funcionales), para el proyecto sobre el que versa la presente memoria.

4.2.1. Requerimientos funcionales

Como hemos dicho, los requerimientos funcionales de un sistema software son aquellos que describen las funcionalidades que dicho sistema debe proporcionar a los usuarios del mismo, para cumplir su objetivo.

Al estar ante un proyecto de tipo académico, no disponemos de cliente alguno para obtener los requerimientos, por lo que nos basamos en otros Sistemas de Recomendación de música existentes en el mercado, de reconocido éxito, para establecer los requerimientos.

A grandes rasgos, concluimos que las funcionalidades que el usuario potencial espera de un Sistema de Recomendación como el nuestro son las siguientes:

- Escuchar canciones.
- Puntuar aquellas canciones que ha escuchado y desea evaluar.
- Recibir recomendaciones de canciones, mediante la generación de una radio (lista de reproducción) personalizada, que incluya objetos aún no conocidos por el usuario.
- Editar el perfil del usuario. Esta funcionalidad supone:
 - Poder modificar las puntuaciones ya realizadas.
 - Consultar los datos personales y preferencias del usuario.
 - Modificar los datos personales y las preferencias del usuario.

- Disponer de información clara acerca de todas las canciones existentes en el sistema, así como las características de estas.

Una vez definidas las funcionalidades que los usuarios esperarían de nuestro sistema, pasaremos a caracterizar de manera más específica y formal cómo nuestro sistemas responderá a dichas funcionalidades:

1. Actualizar el modelo de recomendaciones.

El sistema debe actualizar su modelo de recomendación cada cierto tiempo, incorporando las nuevas puntuaciones que los usuarios hayan hecho sobre las canciones.

2. Recibir recomendaciones de nuevas canciones para el usuario.

El sistema debe ofrecer al usuario una serie de canciones que aún no ha escuchado y evaluado y que se estima sean de su agrado.

3. Actualizar las recomendaciones del usuario.

Una vez actualizado el modelo de recomendación, el sistema debe ser capaz de ofrecer nuevas recomendaciones a los usuarios, teniendo en cuenta la nueva información contenida en el modelo actualizado.

4. Darse de alta en el sistema.

El sistema debe proporcionar un mecanismo para que el usuario pueda registrarse en el sistema, de cara a ofrecerle un servicio personalizado.

5. Escuchar la radio.

El sistema debe permitir al usuario acceder a una lista de reproducción fácilmente utilizable para poder escuchar las canciones que desee.

6. Actualizar la radio.

En cualquier momento, el sistema debe brindar al usuario la posibilidad de re-lanzar la radio, ofreciéndole una nueva lista de reproducción.

7. Puntuar canciones.

El sistema ha de permitir al usuario evaluar numéricamente canciones, ya sean canciones nuevas para él, o bien canciones ya evaluadas cuya puntuación desee modificar.

8. Acceder a álbumes.

Además de la radio, el sistema debe permitir al usuario acceder a la música clasificada por álbumes, para escuchar los temas pertenecientes a dicho álbum y consultar la información del mismo.

9. Buscar música por género.

El sistema debe dar al usuario la posibilidad de buscar aquellos temas musicales cuyo género coincida con el género impuesto por este en su perfil de usuario.

10. Visualizar datos personales.

El sistema debe proporcionar la posibilidad de visualizar los datos personales del usuario.

11. Modificar datos personales.

El sistema debe brindar al usuario la capacidad de modificar sus datos personales cuando este lo desee.

12. Visualizar perfil del usuario.

El sistema debe permitir al usuario visualizar la información de su perfil, incluyendo sus preferencias musicales y la lista de puntuaciones realizadas previamente por él.

13. Modificar perfil del usuario.

El sistema ha de proporcionar al usuario la capacidad de modificar los datos y preferencias que figuran en su perfil.

14. Acceder al libro de visitas.

De forma adicional, el sistema debe dar al usuario la posibilidad de entrar a un libro de visitas, en el que los usuarios dejan comentarios.

4.2.2. Requerimientos No Funcionales

Los requerimientos no funcionales son las restricciones impuestas a los requerimientos funcionales del sistema. Pese a que en principio puedan no parecer demasiado relevantes, son tan importantes como los requerimientos funcionales y, en muchos casos, pueden incluso llegar a ser críticos para la aceptación del sistema. Normalmente, estos requerimientos especifican propiedades del sistema software en sí (velocidad, rendimiento...) y de la interfaz de usuario, además de todas las restricciones impuestas por la organización (plazo, legalidad vigente, política empresarial...).

Obviamente, al no estar ante un proyecto de tipo empresarial o comercial, no hay necesidad de someterse a restricciones organizacionales. Así, los requerimientos no funcionales deben obtenerse y analizarse a partir de las necesidades hardware y software de los equipos informáticos, para dar al usuario la funcionalidad requerida de forma eficiente, y de la interfaz gráfica entre el usuario y la aplicación.

- **Requerimientos del Equipo Informático**

Dado que nuestro proyecto se sustenta sobre una arquitectura cliente/servidor, será necesario distinguir entre los requerimientos del equipo cliente y los requerimientos del servidor.

Los **requerimientos del Equipo Informático del cliente** son bastante sencillos. Tan sólo es necesario un equipo con conexión a Internet (preferiblemente de banda ancha, para optimizar el tiempo de descarga en el cliente de la música a escuchar), que disponga de un navegador adecuado para visualizar la aplicación correctamente (se recomienda Internet Explorer 6 o superior, o Mozilla Firefox). Dicho navegador debe poder permitir la reproducción de contenido Flash (normalmente el navegador avisa e invita al usuario a instalar el plugin en caso de no existir) y, a ser posible, con el bloqueo de pop-ups desactivado. Además, es necesario disponer de altavoces y tarjeta de sonido.

Los **requerimientos del Equipo Informático del servidor**, son bastante más ambiciosos y detallados, por lo que los dividiremos en dos grupos: requerimientos de hardware y requerimientos de software.

1) **Hardware del Equipo Servidor**

-Tipo de Procesador: Intel Xeon X3320 Quad Core a 2,50 GHz

-Número de procesadores: 1

-Núcleo del procesador disponible: Quad

-Memoria caché interna: 6 MB de caché de nivel 2.

-Bus del sistema: Bus frontal de 1333 MHz

-Memoria de serie: 4 GB de memoria estandar.

- Unidad de disco duro interna:** 1 unidad SATA 1TB, 3,5'' sin conexión en caliente.
- Velocidad de la unidad de disco duro:** 7.200 rpm
- Almacenamiento masivo interno:** 1 TB
- Interfaz de red:** Adaptador de servidor NC105i Gigabit incorporado.
- Puertos de E/S externos:** Serie: 1, USB 2.0: 5, Gráficos:1, Teclado:1,Ratón:1, Conector de red RJ-45.

2) Software del Equipo Servidor

- Sistema Operativo Instalado:** Mandriva v.Spring 2009, Sistema Operativo de libre distribución de la familia Linux.
- Máquina virtual:** Java v1.6.
- PHP:** Versión 5.0.
- Navegador:** Última versión del navegador Mozilla Firefox.
- Sistema Gestor de Bases de datos:** MySql Server 5.1.
- Gestor de Contenidos (CMS):** Drupal v6.9, se trata de la aplicación sobre la que se monta el prototipo de nuestro proyecto.
- Servidor Web:** Apache 2.
- Entorno de desarrollo Java:** NetBeans 6.5 con JDBC.

Parte del software necesario será proporcionado al administrador de la aplicación, el cual dispone de un manual para su instalación en el Anexo A.

4.2.3. Requerimientos de la Interfaz

Los requerimientos de la interfaz gráfica entre la aplicación y el usuario están estrechamente ligados a la usabilidad y sus principios [13]. La usabilidad se puede definir de varias formas:

Usabilidad se define como la medida en que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico[19].

Usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.

La usabilidad puede definirse de forma coloquial como facilidad de uso, ya sea de una página Web, una aplicación informática o cualquier otro sistema que interactúe con el usuario.

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

- **Facilidad de aprendizaje:** Se refiere a aquellas características de la interfaz que permiten a los usuarios noveles comprender cómo usarla inicialmente y cómo obtener un máximo grado de productividad. Para que un sistema sea fácil de aprender debe reunir las siguientes características.
 - *Predecibilidad:* El conocimiento de la historia de una interacción debe ser suficiente para determinar el resultado de una interacción futura.
 - *Síntesis:* El usuario debe poder captar fácilmente los cambios de estado del sistema.

- *Familiaridad*: Es importante diseñar el sistema de forma que los elementos de su interfaz resulten conocidos y familiares para el usuario.

- *Generalización*: Las tareas semejantes deben ser resueltas de modo parecido.

- *Consistencia*: Todos los mecanismos que se utilizan deben ser usados siempre de la misma manera, sea cual sea el momento en que se utilicen. Esto se consigue, por ejemplo, siguiendo guías de estilo y diseñando con un *look & feel* común.

- **Flexibilidad**: Este principio de usabilidad establece que debe haber varias formas en que el sistema y el usuario intercambian la información. Las características que hacen a una interfaz flexible son:

- *Control del diálogo por parte del usuario*: Hay que proporcionar al usuario la capacidad, siempre que sea posible, para decidir cuándo empezar u acabar las operaciones.

- *Migración de tareas*: Tanto el usuario como el sistema deben poder realizar una tarea en exclusiva, o pasarla al otro, o realizarla de forma conjunta.

- *Capacidad de sustitución*: La capacidad de sustitución permite que valores equivalente puedan ser sustituidos los unos por los otros. Un ejemplo de ello es el del ancho de página en un procesador de textos, que puede expresarse en centímetros o en pulgadas.

- *Capacidad de adaptación*: La interfaz debe poder adaptarse automáticamente a las necesidades del usuario actual.

- **Robustez**: Es el nivel de fiabilidad del sistema, o el grado en que el sistema es capaz de tolerar fallos durante la interacción. Las siguientes características hacen posible la robustez:

- *Navegabilidad*: El usuario debe poder observar el estado del sistema sin que esta observación repercuta de forma negativa en él.

- *Uso de valores por defecto:* Ayudan al usuario mediante recuerdo pasivo. Mostrar un valor por defecto ayuda a que el usuario sepa qué tipo de valor debe introducir.
- *Persistencia:* Un sistema persistente es aquel en el que las notificaciones al usuario permanecen como objetos manipulables después de su presentación.
- *Recuperación de información:* La aplicación debe permitir el regreso a un estado anterior, tras algún error en una interacción previa.
- *Tiempo de respuesta:* Es el tiempo que necesita el sistema para reflejar los cambios realizados por el usuario.

4.3. Análisis del Sistema

Tras haber conocido el propósito del proyecto software, las propiedades de este y las restricciones a las que debe someterse, es el momento de analizar el sistema y crear un modelo del mismo que sea correcto, completo, consistente, claro y verificable. Para ello, se definirán los casos de uso según los requerimientos previamente obtenidos y, acto seguido, se describirán los principales escenarios y flujos de eventos de dichos casos de uso [15].

4.3.1. Modelo de Casos de Uso

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

Los casos de uso son tareas con significado, coherentes y con cierta independencia, que los actores realizan de forma cotidiana al utilizar el sistema. En un caso de uso pueden participar uno o varios actores.

En definitiva, los casos de uso explican cómo se realiza una tarea de forma precisa, y constan de los siguientes elementos:

- Nombre único e unívoco del caso de uso.
- Actores participantes.
- Condiciones de entrada.
- Flujo de eventos.
- Condiciones de salida.
- Requerimientos especiales.

Por lo tanto, la primera cuestión es la de determinar cuáles son los actores participantes en cada uno de los casos de uso.

Un actor modela una entidad externa que interacciona con el sistema, es decir, es un tipo de usuario del sistema. Al igual que ocurre con un caso de uso, un actor debe tener un nombre único y adecuado, que puede ir acompañado de una descripción del mismo.

En nuestro sistema contamos con los actores siguientes:

Usuario_registrado: Se corresponde con cada uno de los usuarios que se han registrado en el sistema para poder realizar tareas propias de ellos como escuchar una radio personalizada, evaluar canciones, modificar sus datos personales y de perfil, etc.

Usuario_anónimo: Se refiere a cada uno de los usuarios, no registradas en el sistema, que pueden utilizar los servicios básicos proporcionados por este, tales como escuchar una radio global y visualizar los álbumes existentes.

Administrador: Hace referencia al responsable de la aplicación, es el que se encarga de administrar los datos alojados, y de calcular las recomendaciones para los usuarios de forma periódica en base a las nuevas puntuaciones realizadas por estos.

BBDD (Base de Datos): Es la base de datos que almacena y proporciona los datos necesarios para el sistema (usuarios registrados y datos de su perfil, canciones, puntuaciones de usuarios sobre canciones, predicciones para los usuarios sobre canciones, listas de reproducción personalizadas y datos del sitio Web).

Una vez definidos cuáles van a ser los actores del sistema, llega el momento de crear los diferentes casos de uso. Para ello, es importante que cada uno de los requerimientos funcionales anteriormente definidos sean cubiertos por al menos uno de los casos de uso aunque, por otro lado, puede haber casos de uso nuevos, en los que no aparezca ninguno de los requerimientos, ya que nos encontramos en una fase de refinamiento del sistema en la que pretendemos construir un modelo detallado del mismo.

El primer paso para la realizar el modelo de casos de uso de nuestro sistema es la obtención de los diversos diagramas de casos de uso de nuestro sistema. El primero de ellos es un diagrama frontera, es decir, un diagrama que describe completamente la funcionalidad de un sistema.

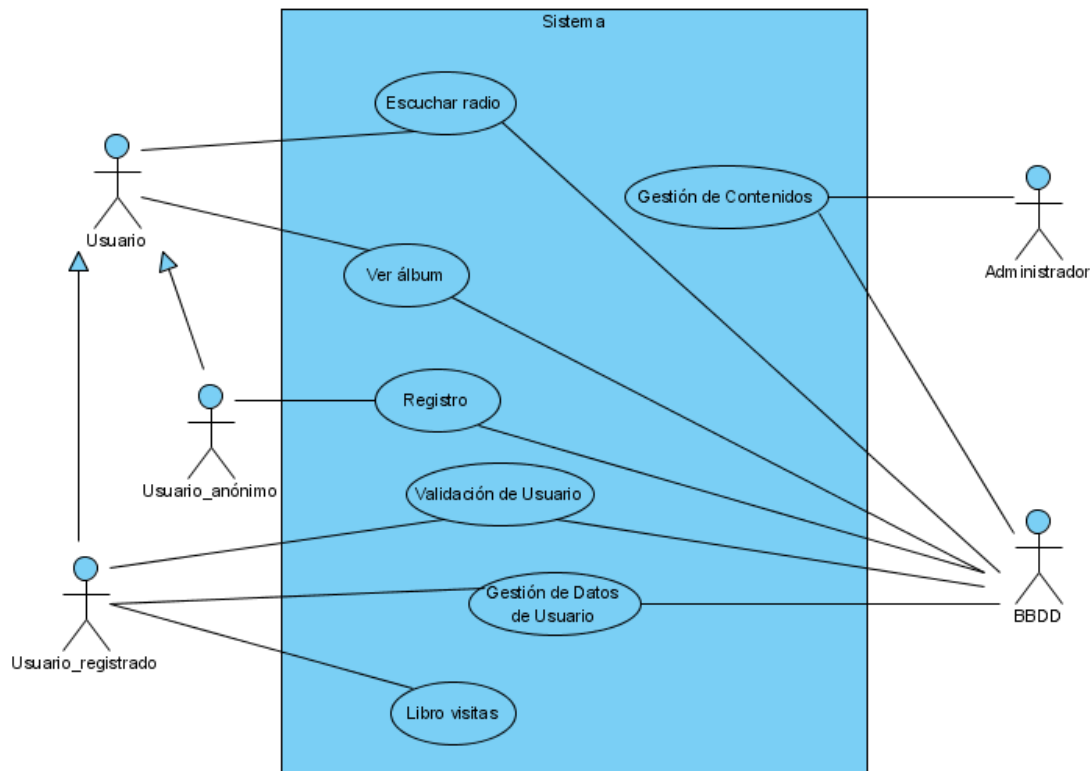


Figura 4.1: Diagrama Frontera del sistema

Los casos de uso reflejados en un diagrama frontera pueden ser lo suficientemente precisos o, por el contrario, necesitar ser explicados en mayor detalle. A la hora de detallar un caso de uso se pueden emplear dos tipos de relaciones.

- **<<extend>>**: Se trata de una relación cuyo sentido es hacia el caso de uso a detallar, que representa comportamientos excepcionales del caso de uso.
- **<<include>>**: Es una relación cuyo sentido es contrario al de la relación <<extend>>, que representa un comportamiento común del caso de uso.

En el caso de nuestro sistema nos encontramos ante varios casos de uso que deben ser descritos con mayor profundidad.

A continuación, describimos detalladamente cada uno de los casos de uso mostrados en las figuras anteriores, y exponemos los diagramas que se identifican con aquellos casos de uso que necesitan un mayor nivel de detalle en su descripción.

▪ **Caso de Uso 1:** Registro

Actores participantes: Usuario_anónimo, BBDD.

Condiciones de entrada: Existe usuario que pretende darse de alta en el sistema.

Flujo de eventos:

1. El usuario inicia la aplicación.
2. El sistema muestra un enlace para el registro de usuarios.
3. El usuario accede a la página de registro.
4. El sistema muestra un formulario de entrada.
5. El usuario introduce un identificador de usuario, dirección de correo-e e información para su perfil.
6. El sistema comprueba que los datos introducidos son válidos (E-1)(E-2).
7. Los datos del nuevo usuario quedan almacenados en la BBDD.
8. El usuario recibe un e-mail con la contraseña para validarse en el sistema.

Condiciones de salida: El nuevo usuario ha sido registrado en la BBDD del sistema.

Excepciones

E-1: El identificador de usuario ya está asignado. El sistema informa al usuario de dicha situación. El usuario puede introducir otro identificador de usuario diferente o salir del caso de uso.

E-2: La dirección de correo-e no es válida. El sistema informa al usuario de dicha situación. El usuario puede introducir otra dirección de correo-e diferente o salir del caso de uso.

▪ **Caso de Uso 2:** Validación de usuario.

Actores participantes: Usuario_registrado, BBDD.

Condiciones de entrada: Existen cuentas de usuarios registradas en la aplicación.

Flujo de eventos:

1. El usuario inicia la aplicación
2. El sistema muestra una sección con un formulario de inicio de sesión.
3. El usuario introduce su identificador y su contraseña.
4. El sistema accede a la BBDD y comprueba que el identificador y la contraseña son válidos (E-1).
5. El sistema entra al sistema con los privilegios de usuario registrado.

Condiciones de salida: El nombre de usuario y la contraseña han sido aceptados y el usuario accede al sistema con privilegios de usuario registrado.

Excepciones:

E-1: El identificador o la contraseña introducidos por el usuario no son válidos. El sistema muestra un aviso para informar al usuario de la situación. El usuario puede introducir un identificador y/o contraseña válidos o salir del caso de uso.

▪ **Caso de Uso 3:** Ver álbum

Actores participantes: Usuario, BBDD.

Condiciones de entrada: El usuario ha entrado en el sistema.

Flujo de eventos:

1. El usuario selecciona la opción 'Álbumes'.
2. El sistema muestra una relación de álbumes musicales de diferentes géneros.
3. El usuario selecciona el álbum de su interés.
4. El sistema se comunica con la base de datos (E-1) para obtener los datos del álbum.
5. El sistema muestra la información y canciones del álbum seleccionado.

Condiciones de salida: El usuario es capaz de visualizar la información del álbum solicitado.

Excepciones:

E-1: Ha habido un error en la comunicación entre el sistema y la base de datos. El sistema informa al usuario de dicha situación. El caso de uso se reanuda por el paso 3.

▪ **Caso de Uso 4:** Libro Visitas

Actores participantes: Usuario_registrado

Condiciones de entrada: El usuario se ha registrado y validado en el sistema.

Flujo de eventos:

1. El usuario selecciona la opción 'Libro de visitas'.
2. El sistema muestra los comentarios de otros usuarios y da al usuario la opción de añadir un nuevo comentario.
3. El usuario selecciona la opción añadir un nuevo comentario.
4. El sistema muestra una caja de texto para que el usuario escriba su comentario.
5. El usuario escribe su comentario y lo envía.

Condiciones de salida: El nuevo comentario del usuario queda publicado y almacenado en la aplicación.

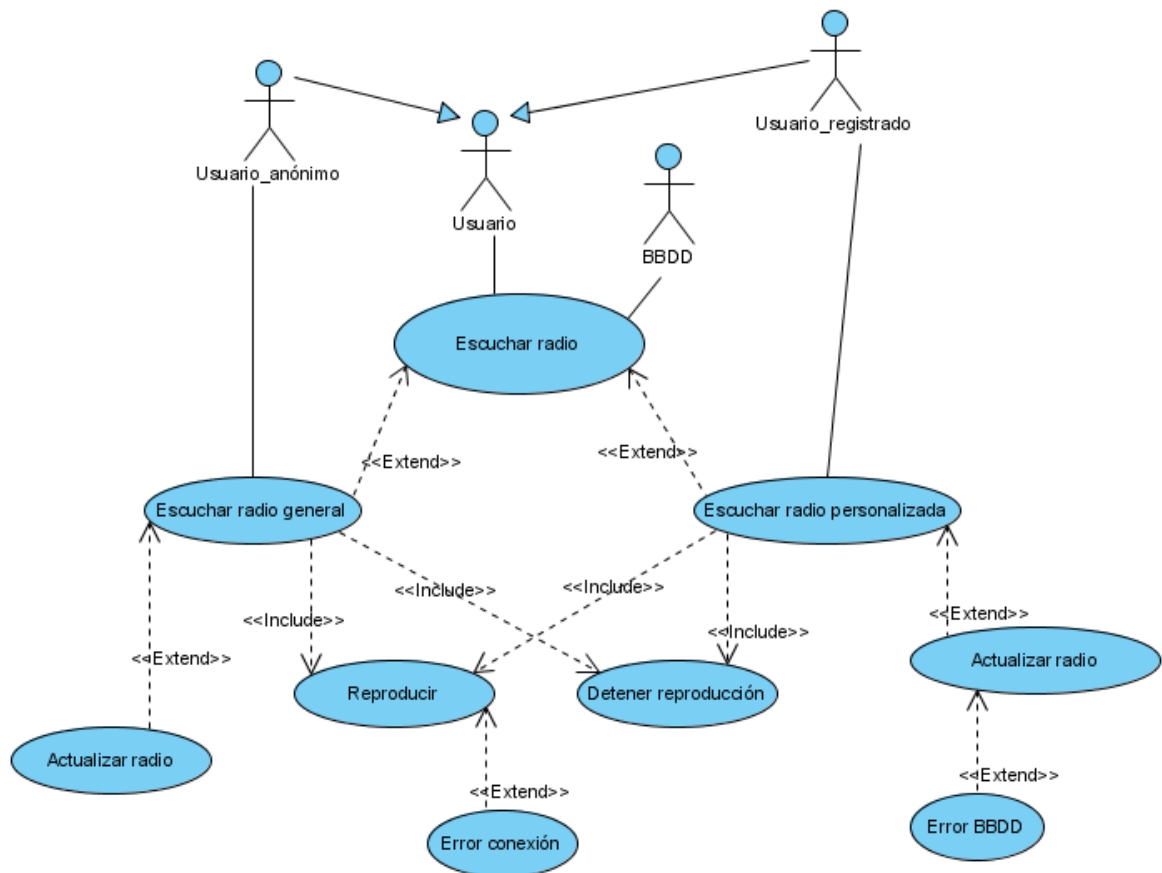


Figura 4.2: Caso de uso 'Escuchar radio'

▪ Caso de Uso 5: Escuchar radio

Actores participantes: Usuario, BBDD.

Condiciones de entrada: El usuario ha accedido al sistema.

Flujo de eventos:

1. El usuario accede a la radio de la aplicación.
2. Si el usuario es un usuario anónimo, se realiza S-1.
3. Si el usuario se validado en el sistema como un usuario registrado, se realiza S-2.

Subflujos de eventos:-S1: Escuchar radio general

1.1. El sistema se comunica con la base de datos (E-1) para obtener los datos de una serie de canciones de forma aleatoria.

1.2. El usuario inicia la reproducción de la radio obtenida.

1.3. El sistema descarga las canciones en el navegador (E-2) y las reproduce en la lista de reproducción.

1.4. El usuario detiene la reproducción.

1.5. Si el usuario desea una lista de reproducción diferente, elige actualizar radio, y se realiza S-3.

-S2: Escuchar radio personalizada

2.1. El sistema se comunica con la base de datos (E-1) para obtener los datos de una serie de canciones recomendadas para el usuario.

2.2. El usuario inicia la reproducción de la radio obtenida.

2.3. El sistema descarga las canciones en el navegador (E-2) y las reproduce en la lista de reproducción.

2.4. El usuario detiene la reproducción.

2.5. Si desea obtener una lista de reproducción diferente, elige actualizar radio, y se realiza S-3.

-S-3: Actualizar radio

3.1. El sistema selecciona un nuevo conjunto de canciones para el usuario (E-1).

3.2. El usuario dispone de una nueva lista de reproducción.

Condiciones de salida: El usuario ha utilizado la radio.

Excepciones:

E-1: Ha habido un error en la comunicación entre el sistema y la base de datos. El sistema informa al usuario de dicha situación. El caso de uso se reanuda desde su comienzo.

E-2: Error de conexión, durante la descarga de las canciones en el navegador. El sistema se recupera una vez establecida la conexión de nuevo. El caso de uso se reanuda.

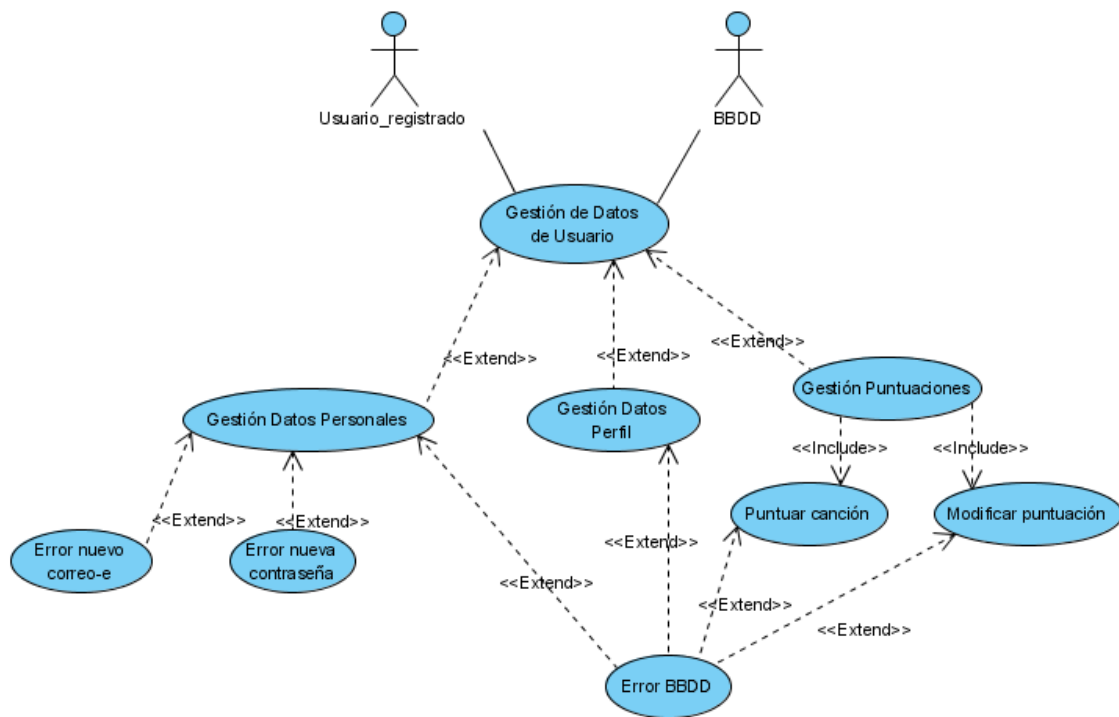


Figura 4.3: Caso de uso 'Gestión de Datos de Usuario'

▪ Caso de Uso 6: Gestión de Datos del Usuario

Actores participantes: Usuario registrado, BBDD.

Condiciones de entrada: El usuario ha entrado en el sistema y se ha validado como usuario registrado.

Flujo de eventos:

1. El usuario desea gestionar sus datos personales, se realiza S-1.
2. El usuario desea gestionar los datos de su perfil, se realiza S-2
3. El usuario desea gestionar sus puntuaciones, se realiza S-3.

Subflujos de eventos:

-S1: Gestión Datos Personales

- 1.1. El usuario elige la opción Mi Cuenta del menú principal.
- 1.2. El sistema se comunica con la base de datos (E-1) para obtener los datos personales del usuario.
- 1.3. El sistema muestra los datos personales del usuario.
- 1.4. El usuario, si lo desea, modifica su dirección de correo electrónico (E-2) o su contraseña (E-3).
- 1.5. El sistema guarda los cambios efectuados.

-S2: Gestión Datos Perfil

- 2.1. El usuario elige la opción Perfil de Usuario del menú principal.
- 2.2. El sistema se comunica con la base de datos (E-1) para obtener los datos del perfil del usuario.
- 2.3. El sistema muestra los datos de perfil al usuario.
- 2.4. El usuario, si lo desea, modifica los datos de su perfil (género musical).
- 2.5. El sistema guarda los cambios efectuados.

-S-3: Gestión Puntuaciones

- 3.1. El usuario accede a una canción (radio o álbumes).
- 3.2. El sistema muestra la canción al usuario.

3.3. El usuario elige puntuar la canción.

3.4. El sistema muestra un panel de puntuación asociado a la canción.

3.5. El usuario asigna una puntuación numérica a la canción.

3.6. El sistema se comunica con la base de datos (E-1) para guardar la puntuación.

3.7. El usuario elige la opción Mi Perfil.

3.8. El sistema se comunica con la base de datos (E-1) para obtener los datos de las puntuaciones hechas por el usuario.

3.9. El usuario modifica una de las puntuaciones anteriormente realizadas.

3.10. El sistema se comunica con la base de datos (E-1) para guardar los cambios.

Condiciones de salida: El usuario ha realizado las gestiones oportunas sobre sus datos.

Excepciones:

E-1: Ha habido un error en la comunicación entre el sistema y la base de datos. El sistema informa al usuario de dicha situación. El caso de uso se reanuda desde el punto en que se quedó.

E-2: Error en el correo electrónico nuevo. El sistema informa al usuario de dicha situación. El usuario puede introducir una nueva dirección o bien terminar el caso de uso.

E-3: Error en la nueva contraseña. El sistema informa al usuario de dicha situación. El usuario puede introducir una nueva contraseña o bien terminar el caso de uso.

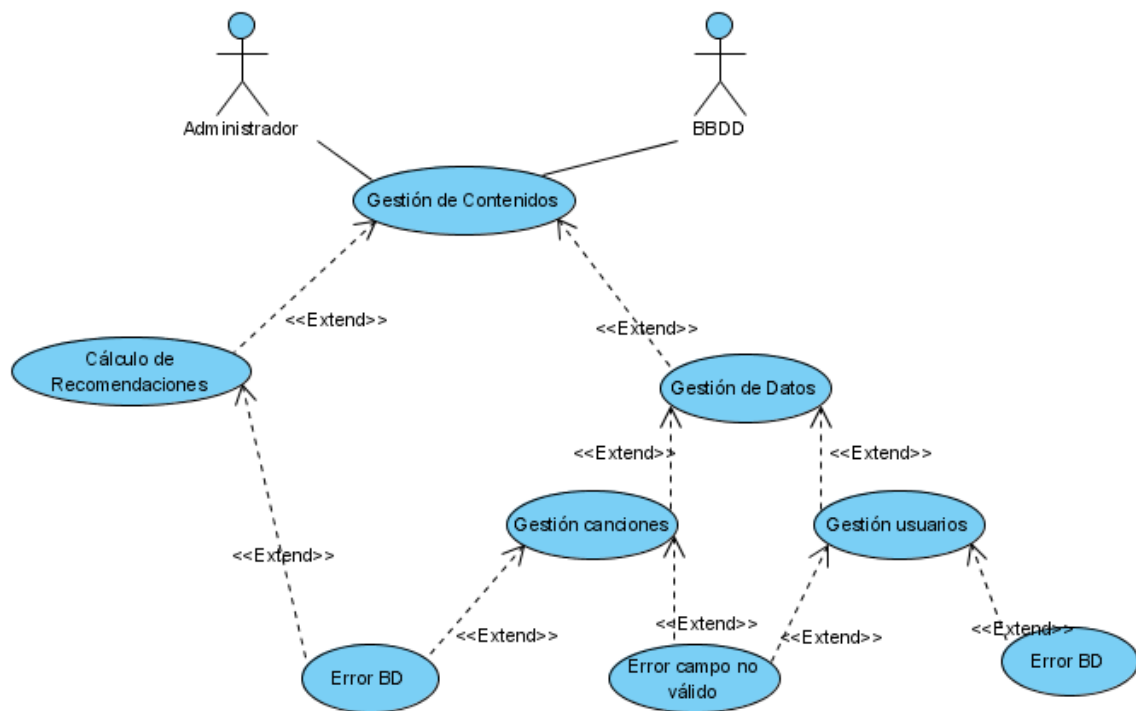


Figura 4.4: Caso de uso 'Gestión de contenidos'

▪ **Caso de Uso 7: Gestión de Contenidos**

Actores participantes: Administrador, BBDD.

Condiciones de entrada: El Administrador accede a la base de datos de la aplicación.

Flujo de eventos:

1. Si se efectúa el cálculo de las recomendaciones, se realiza S-1.
2. Si el administrador desea gestionar los datos de la aplicación, se realiza S-2.

Subflujos de eventos:-S1: Cálculo de recomendaciones

1.1. El sistema ejecuta, de forma periódica, el algoritmo de filtrado para generar las nuevas predicciones.

1.2. El sistema guarda los nuevos datos en la base de datos (E-1).

-S2: Gestión de Datos²

2.1. El usuario elige Gestión de Datos en el sistema.

2.2. Si el usuario desea gestiones sobre las canciones, se realiza la Gestión de canciones (E-2).

2.3. Si el usuario desea gestiones sobre los usuarios, se realiza la Gestión de usuarios (E-2).

Condiciones de salida: El usuario ha escuchado canciones en la aplicación.

Excepciones:

E-1: Ha habido un error en la comunicación entre el sistema y la base de datos. El sistema informa al usuario de dicha situación. El caso de uso se reanuda.

E-2: Error al insertar o modificar algún campo de datos. El sistema informa al usuario de dicha situación. El usuario puede volver a introducir el valor, o bien terminar el caso de uso.

² Este sub-flujo de eventos tendría más sub-flujos a un nivel inferior, para la inserción, modificación y borrado de usuarios en canciones, pero no entraremos en mayor detalle para estos casos de uso al no haberlo hecho en los anteriores diagramas.

4.3.2. Escenarios

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

Los escenarios son historias ficticias que describen posibles interacciones de un caso de uso. Son de gran utilidad porque permiten a los diseñadores anticiparse a los problemas. A pesar de ser historias ficticias, deben desarrollarse con el mayor nivel de detalle posible. Así, por ejemplo, los personajes deben tener nombre, motivaciones para usar la aplicación, deben ser situados en un contexto real con las restricciones que ello supone, etc. De esta manera es mucho más sencillo para los diseñadores discutir sobre la interfaz ya que a las personas les cuesta más trabajo discutir las posteriores decisiones de diseño sobre una situación abstracta.

Esta forma de proceder, además, fuerza a los diseñadores a tener en cuenta el rango de usuarios que va a usar el sistema y el rango de actividades para las que lo van a usar. Los escenarios permiten hacer diferentes combinaciones de usuarios y actividades de forma que se tengan en cuenta todas las posibilidades.

Un escenario está formado por los siguientes elementos:

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

Como se ha indicado, para cada caso de uso puede haber varios escenarios. Para nuestro proyecto se han creado y descrito gran cantidad de casos de uso, por lo que nosotros vamos a definir solo algunos de los escenarios para los principales casos de uso, así se tendrá un ejemplo de las principales funcionalidades del sistema: registro de usuario, validación de usuario, escuchar radio personalizada, realizar puntuaciones en la radio, ver álbum.

Escenario 1: Registro de usuario

Nombre: Registrar usuario.

Descripción: El usuario Paco entra en el sistema y decide darse de alta en el mismo para disfrutar de los servicios que esta le ofrece.

Actores participantes: Usuario anónimo y Base de Datos.

Flujo de eventos:

1. Paco entra como invitado en el sistema y pulsa sobre la opción 'Regístrate'.
2. El sistema muestra un formulario con todos los datos requeridos para el registro.
3. El usuario rellena los siguientes campos:
 - a. Usuario: Paco.
 - b. E-mail: Paco@hotmail.es.
 - c. Género musical favorito: Acústica.
4. Paco pulsa sobre el botón 'Registrarse' en la parte inferior del formulario.
5. El sistema almacena los datos de Paco en la BBDD, muestra un mensaje informando de que el registro ha sido llevado a cabo con éxito, y envía a la dirección de correo de Paco un mensaje que contiene la contraseña y el enlace que le permitirá activar la cuenta.
6. Paco accede a su buzón de correo, abre el correo enviado por el sistema y pulsa sobre el enlace que aparece en el e-mail.
7. El sistema activa la cuenta de Paco, para que este pueda acceder al sistema como usuario registrado.

Escenario 2: Validación de usuarios en el sistema

Nombre: Validación del usuario Paco.

Descripción: El usuario registrado como Paco inicia sesión en el sistema como usuario identificado para gozar de la máxima funcionalidad.

Actores participantes: Paco y Base de Datos.

Flujo de eventos:

1. Paco entra en el sistema.
2. El sistema muestra el formulario para iniciar sesión como usuario registrado.
3. El usuario introduce Paco en el campo de texto correspondiente al identificador de usuario, y Paco123 como contraseña en el campo de texto correspondiente a la contraseña.
4. El usuario Paco pulsa el botón 'Iniciar sesión'.
5. El sistema comprueba los datos introducidos por Paco en la BBDD, valida correctamente al usuario y este entra en la aplicación como usuario registrado.
6. El sistema muestra, donde antes estaba el formulario de inicio de sesión, un mensaje de bienvenida a Paco, y los nuevos elementos de menú: 'Mi cuenta' y 'Terminar sesión'.

Escenario 3: Escuchar radio personalizada

Nombre: Escuchar radio personalizada de Tosco.

Descripción: El usuario registrado en el sistema con el nombre Tosco desea escuchar su radio personalizada, compuesta por canciones de su agrado, además de canciones que el sistema le ha recomendado en base a sus gustos musicales.

Actores participantes: Tosco y Base de Datos.

Flujo de eventos:

1. Tosco entra en el sistema y se valida de forma correcta.
2. El sistema se comunica con la base de datos para obtener las puntuaciones y las predicciones de canciones calculadas por el algoritmo de filtrado para Tosco y obtiene una lista de reproducción para él.
3. Tosco se encuentra en la pantalla principal del sistema, con su radio personalizada lista para escuchar, y pulsa sobre el botón de iniciar reproducción.
4. El sistema descarga las canciones asociadas a la lista de reproducción en el navegador y las reproduce.
5. Tras escuchar un par de canciones, Tosco detiene la reproducción.
6. Tosco decide actualizar la radio, para obtener una lista de reproducción diferente, pulsando sobre la opción 'cambiar lista'.
7. El sistema se comunica nuevamente la base de datos y actualiza la lista de reproducción.

Escenario 4: Realizar puntuaciones sobre la radio

Nombre: Puntuar canción 'The Mission'.

Descripción: El usuario registrado en el sistema con el nombre Tosco quiere evaluar la canción 'The Mission' de Josh Woodward como Muy Buena.

Actores participantes: Tosco y Base de Datos

Flujo de eventos:

1. El usuario entra en el sistema y se valida de forma correcta como Tosco.
2. El sistema se comunica con la base de datos para obtener las puntuaciones y las predicciones hechas para Tosco y obtiene una lista de reproducción para él.
3. Tosco se encuentra en la pantalla principal del sistema, con su radio personalizada lista para escuchar, y pulsa sobre la cuarta canción de su radio, 'The Mission', para comenzar a reproducirla.
4. El sistema descarga la canción asociada a la lista de reproducción en el navegador y la reproduce.
5. Tosco decide evaluar la canción, por lo que pulsa el botón 'Evaluar canción'.
6. El sistema muestra una ventana emergente con un panel de puntuaciones, en el que aparecen los datos de la canción 'The Mission', junto con la carátula del disco 'The Simple Life' de Josh Woodward, y un formulario para puntuar con 5 valores posibles del 1 al 5 (1: Muy Mala, 2: Mala, 3: Normal, 4: Buena, 5: Muy Buena).
7. Tosco escoge el valor 5 (Muy buena) y pulsa sobre el botón Puntuar.
8. El sistema guarda la nueva puntuación en la base de datos, e informa a Tosco de que la puntuación ha sido registrada.

Escenario 5: Ver álbum

Nombre: Ver el álbum 'Summertime Releases 2008'.

Descripción: El usuario registrado en el sistema con el nombre Webis quiere ver información sobre el álbum 'Summertime Releases 2008' para saber cuántas canciones de dicho álbum hay en el sistema.

Actores participantes: Webis y Base de Datos.

Flujo de eventos:

1. El usuario entra en el sistema y se valida de forma correcta como Webis.
2. El sistema se comunica con la base de datos para obtener las puntuaciones y las predicciones hechas para Webis y obtiene una lista de reproducción para él.
3. Webis ignora por el momento su radio y escoge la opción 'Álbumes' del menú principal.
4. El sistema se comunica con la base de datos para obtener y mostrar al usuario una lista de los álbumes disponibles en la aplicación.
5. Webis decide hacer un filtrado de los álbumes, escoge ver los álbumes del género House y pulsa el botón Aceptar.
6. El sistema muestra a Webis los dos álbumes correspondientes al género House:
 - a. 'Summertime Releases 2008' de Kinetic Beatz.
 - b. 'Summer 2007' de DJ iPep's.
7. Webis selecciona el álbum 'Summertime Releases 2008'.
8. El sistema se comunica de nuevo con la base de datos para obtener detalles del álbum y la información de aquellas canciones que pertenecen a dicho álbum.
9. El sistema muestra una ficha con los principales datos del álbum, el número de canciones disponibles del mismo, y una miniatura de la carátula, y a continuación ofrece a Webis una lista con las canciones ordenadas, y listas para escuchar y puntuar.

4.4. Diseño del sistema

No cabe duda de que realizar de forma adecuada cada una de las actividades que conlleva la Ingeniería del Software es imprescindible para la realización de un proyecto software de calidad. Por ello, no se puede afirmar que ninguna de estas actividades sea más importante que el resto. Sin embargo, sí podemos decir que la actividad de diseño es la más delicada y la más laboriosa de realizar.

- Es delicada porque, de no llevarse a cabo correctamente, se hace imposible el codificar de manera correcta en la actividad de implementación el modelo obtenido anteriormente en la fase de análisis del sistema. Esto puede repercutir en hacer inútil todo el esfuerzo invertido en las primeras actividades de la Ingeniería del Software.
- Es laboriosa porque las estrategias a seguir para que la traducción entre modelo y código se lleve a cabo correctamente son muy diversas y bastante complejas.

Se puede decir, pues, que el diseño del sistema es la actividad de la Ingeniería del Software en la que se identifican los objetivos finales del sistema y se plantean las diversas estrategias para alcanzarlos en la actividad de implementación [14].

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

4.4.1. Diseño de los datos

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

- Las **canciones**, de las que conocemos el nombre de la canción, el artista o intérprete de la canción, el álbum al que pertenece, el género musical al que pertenece la canción y la dirección en el servidor en la que se aloja el archivo de sonido asociado a dicha canción.
- Los **usuarios**, de los que conocemos su identificador o nombre de usuario, contraseña, dirección de e-mail, fecha de registro, idioma utilizado y género musical favorito.
- Las **puntuaciones**, de las que sabemos el usuario que realiza la puntuación, la canción que ha sido puntuada y el valor de dicha puntuación.
- Las **predicciones**, de las que se conocen el usuario sobre el que se ha calculado la predicción, la canción para la que se ha hecho dicha predicción, y el propio valor de la predicción.
- Las **listas de reproducción**, de las que conocemos el usuario cuya lista se tiene almacenada, la canción que forma parte de la lista y un valor numérico que determina su posición dentro de la lista.

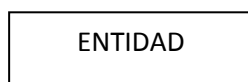
Una vez determinados cuales son los elementos de información del sistema, se deben obtener sus representaciones en forma de tablas de una base de datos [24]. Para ello, se debe realizar en primer lugar un diseño conceptual de la base de datos para, posteriormente, obtener las tablas requeridas. Para realizar este diseño conceptual se utilizará el modelo Entidad-Relación (E-R).

Modelo Entidad-Relación

El modelo Entidad-Relación (también conocido por sus iniciales, E-R) es una técnica de modelado de datos que emplea diagramas entidad-relación. No es la única técnica de modelado pero sí es, con diferencia, la más extendida y utilizada.

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

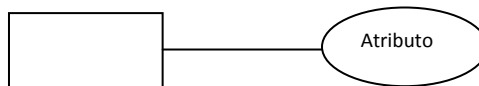
- *Entidades*: Son objetos (cosas, conceptos o personas) sobre los que se tiene información. Se representan mediante rectángulos etiquetados con un nombre en su interior. Llamamos instancia a cualquier ejemplar concreto de una entidad.



- *Relaciones*: Interdependencias entre una o más entidades. Se representan mediante rombos etiquetados en su interior con un verbo. Si la relación es entre una entidad consigo misma se denomina reflexiva, si es entre dos entidades se denomina binaria, si es entre tres ternaria y múltiple si es entre cuatro o más entidades (muy poco común).



- *Atributos*: Describen características propias de una entidad o relación. Se representan mediante elipses etiquetadas con un nombre en su interior.



En los diagramas entidad-relación también hay que tener en cuenta otros aspectos como los que indicamos a continuación:

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

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

1. Convertir el enunciado del problema (o, como es nuestro caso, los elementos del sistema software) en un Esquema Conceptual del mismo.

2. Convertir este Esquema Conceptual (o EC) en uno más refinado, conocido como Esquema Conceptual Modificado (ECM).

3. Obtener las tablas de la base de datos a partir del Esquema Conceptual Modificado.

Normalización en el modelo Entidad-Relación

La normalización es un proceso que consiste en imponer a las tablas ciertas restricciones mediante una serie de transformaciones consecutivas. Con ello nos aseguramos de que las tablas contengan los atributos necesarios y suficientes para describir la realidad de la entidad que representan, separando aquellos que pueden contener información cuya relevancia permite la creación de otra nueva tabla.

Para asegurar la normalización, Codd estableció tres formas normales, las cuales hacen que toda base de datos que las cumple se considere normalizada.

Estas formas normales son:

- *Primera forma Normal (FN1)*: Una tabla está en FN1 si todos los atributos no clave dependen funcionalmente de la clave, o lo que es lo mismo, no existen grupos repetitivos para un valor de clave.
- *Segunda forma Normal (FN2)*: Una tabla está en FN2 si está en FN1 y además todos los atributos que no pertenecen a la clave dependen funcionalmente de ella de forma completa. De esta definición se puede concluir que una tabla en FN1 y cuya clave está compuesta por un único atributo ya está en FN2.
- *Tercera forma Normal (FN3)*: Una tabla está en FN3 si está en FN2 y además no existen atributos no clave que dependan transitivamente de la clave.

Dada la no muy elevada complejidad de las tablas que obtendremos para el modelo de datos de este proyecto, no realizaremos el proceso de normalización. Por tanto, las entidades obtenidas en el Esquema Conceptual Modificado constituirán las tablas de nuestra base de datos.

4.4.1.1. Esquema conceptual

Necesitamos convertir nuestros elementos de información en entidades o relaciones. En nuestro caso, Canciones y Usuarios pasarán a convertirse en entidades de nuestro esquema conceptual, y Puntuaciones, Predicciones y Listas de reproducción se convierten en relaciones que unen la entidad Canciones con Usuarios. Así, nuestro esquema conceptual quedaría como podemos ver en la Figura 4.5:

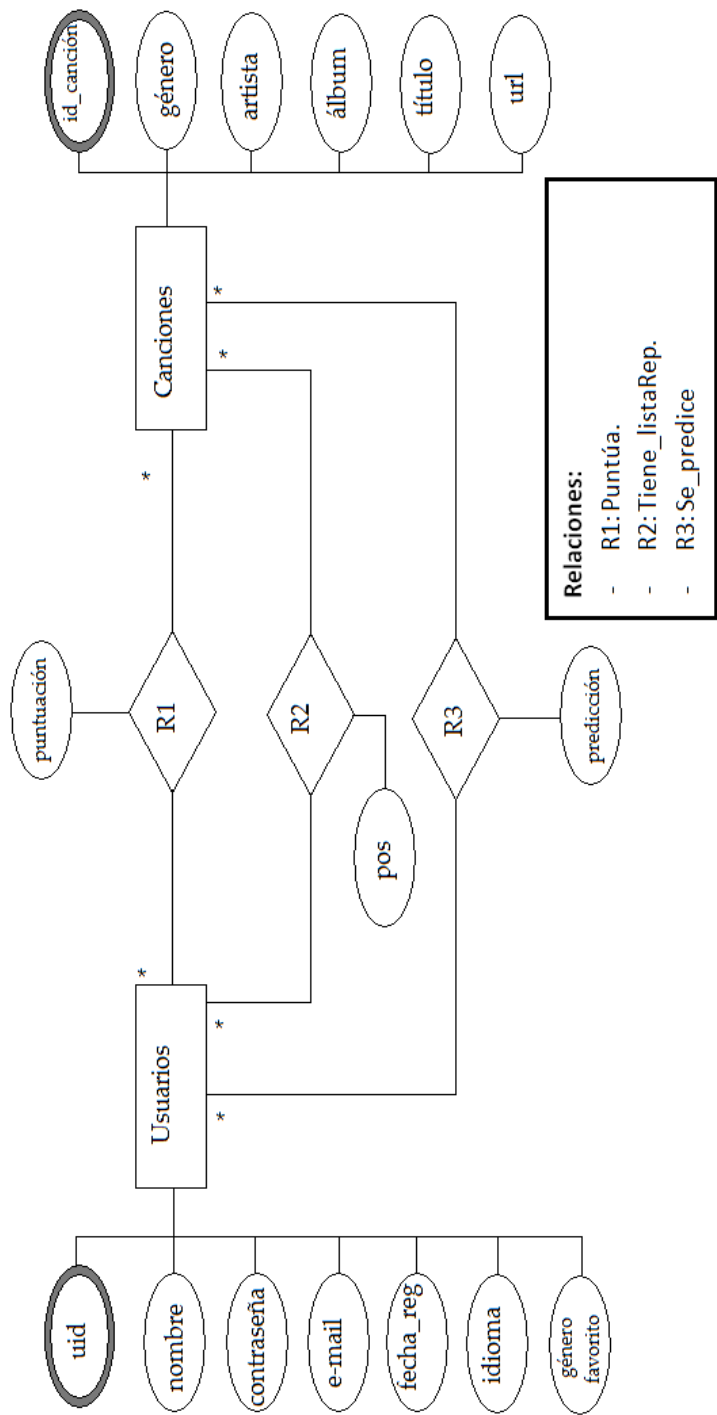


Figura 4.5: Esquema Conceptual de la aplicación

4.4.1.2. Esquema Conceptual Modificado

Para la obtención del Esquema Conceptual Modificado a partir del Esquema Conceptual se deben hacer los cambios que enunciamos a continuación:

- Eliminar todas las entidades débiles.
- Eliminar las relaciones de muchos a muchos.
- Eliminar las relaciones con atributos existentes en el Esquema Conceptual.

En nuestro caso, no es necesaria la eliminación de entidades débiles, ya que no las hay. Realizando las modificaciones oportunas, Obtendríamos tres nuevas entidades: Puntuaciones, Predicciones y Listas_rep, derivadas de las tres relaciones muchos a muchos existentes en el EC. Nuestro Esquema Conceptual Modificado (ECM) quedará como muestra la Figura 4.6.

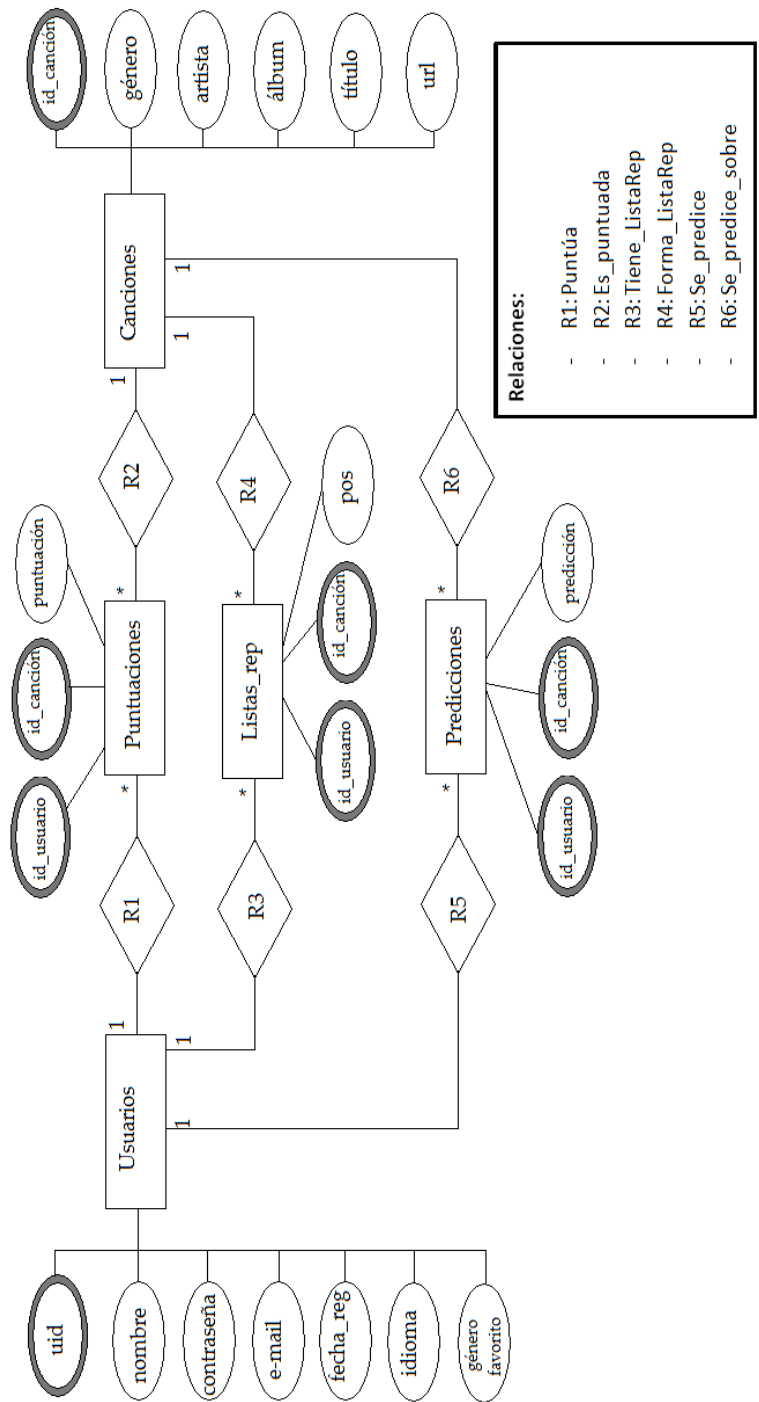


Figura 4.6: Esquema Conceptual Modificado de la aplicación

4.4.1.3. Tablas de la aplicación

Basándonos en el ECM obtenido previamente, podemos determinar un total de 5 tablas en la base de datos, teniendo en cuenta que:

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

Por tanto, según el ECM, obtendríamos las siguientes tablas:

- La entidad Canciones da lugar a la tabla CANCIONES.
- La entidad Usuarios pasa a ser la tabla USUARIOS.
- La entidad Puntuaciones se convierte en la tabla PUNTUACIONES.
- La entidad Predicciones da lugar a la tabla PREDICCIONES.
- La entidad Listas_rep se convierte en la tabla LISTAS_REP.

Sin embargo, nos permitiremos un par de pequeñas excepciones. Dado que estamos empleando un Gestor de Contenidos (CMS) Drupal para alojar nuestra aplicación Web, ya disponemos a priori de una base de datos con toda la información necesaria para dicho CMS.³ En particular, existe una tabla USERS que podemos emplear como la tabla USUARIOS. Además, el atributo 'género favorito', al referirse a información del perfil del usuario, es almacenado en otra tabla especial, denominada PROFILE_VALUES.

Existen muchas más tablas en la base de datos por defecto de Drupal, pero para el desarrollo de nuestro proyecto solamente nos interesan las dos tablas citadas.

³ Véase documentación acerca de la base de datos del CMS Drupal para más información.

De esta forma, nos quedarían las siguientes tablas:

- CANCIONES
- USERS
- PROFILE_VALUES
- PREDICCIONES
- PUNTUACIONES
- LISTAS_REP

CANCIONES

Tabla de 197 filas con los siguientes atributos cada una:

- ID_CANCION: Entero. Llave primaria. Identificador numérico y unívoco de la canción.
- ARTISTA: Cadena de 40 caracteres. Autor de la canción.
- ÁLBUM: Cadena de 40 caracteres. Nombre del álbum al que pertenece la canción.
- TÍTULO: Cadena de 40 caracteres. Título de la canción.
- URL: Campo de texto. Enlace en el servidor al archivo correspondiente a la canción.
- GÉNERO: Conjunto de valores. Género al que la canción pertenece. Puede tomar los siguientes valores.

- ACÚSTICA
- ROCK
- TECHNO
- INSTRUMENTAL
- POP
- HOUSE

- PSICODELIC
- AMBIENTAL
- BSO

			id_cancion	genero	artista	album	url	titulo
☐		☒	1	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/101-HeritagePlac...	Heritage Place
☐		☒	2	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/102-Go.mp3	Go
☐		☒	3	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/103-Saturday.mp3	Saturday
☐		☒	4	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/104-TheParade.mp...	The Parade
☐		☒	5	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/105-RevolutionNo...	Revolution Now
☐		☒	6	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/106-Flypaper.mp3	Flypaper
☐		☒	7	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/107-ImLettingGo....	Im letting go
☐		☒	8	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/108-MorningBlue....	Morning blue
☐		☒	9	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/109-TheBottom.mp...	The bottom
☐		☒	10	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/110-RogueNation....	Rogue nation
☐		☒	11	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/111-HistoryRepea...	History repeats
☐		☒	12	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/201-Ships.mp3	Ships
☐		☒	13	acustica	Josh Woodward	The Simple Life	musica/JoshWoodward-TheSimpleLife/202-FlutterByBut...	Flutter by butterfly

Figura 4.7: Ejemplo de tuplas en la tabla CANCIONES

USUARIOS

Se trata de una tabla que contiene tantas filas como usuarios hay registrados en el portal Web de la radio.

La tabla 'users' guarda un amplio conjunto de atributos, los que realmente interesan para llevar a cabo la gestión de puntuaciones sobre canciones son:

- UID: Entero. Llave primaria. Identificador numérico único para el usuario.
 - Existe un usuario especial, el usuario anónimo, cuyo UID es 0. Este dato es útil para condicionar ciertos aspectos del sitio Web según el usuario esté registrado o sólo sea un invitado.
- NAME: Cadena de 60 caracteres. Nombre del usuario.
- PASS: Cadena de 32 caracteres. Contraseña del usuario codificada mediante un algoritmo de cifrado.

- MAIL: Cadena de 64 caracteres. Dirección de correo-e del usuario.
- LANGUAGE: Cadena de 12 caracteres. Código del idioma empleado por el usuario.
- ACCESS: Entero. Tiempo transcurrido desde el último acceso del usuario a la aplicación.

PUNTUACIONES

Tabla que contiene los siguientes campos:

- ID_USUARIO: Entero. Llave primaria. Llave externa. Identificador del usuario.
- ID_CANCION: Entero. Llave primaria. Llave externa. Identificador de la canción.
- PUNTUACIÓN: Entero de longitud 1. Puntuación (1, 2, 3, 4 ó 5) del usuario sobre la canción.

			id_usuario	id_cancion	puntuacion
☐			1	189	3
☐			7	19	1
☐			1	124	2
☐			5	25	3
☐			1	10	3
☐			1	31	2
☐			1	156	4
☐			8	142	1
☐			8	143	2
☐			8	144	3
☐			8	145	4
☐			8	146	5
☐			8	147	1
☐			8	148	2
☐			8	149	4

Figura 4.8: Ejemplo de tuplas en la tabla PUNTUACIONES

PREDICCIONES

Tabla que contiene los siguientes campos:

- ID_USUARIO: Entero. Llave primaria. Llave externa. Identificador del usuario.
- ID_CANCION: Entero. Llave primaria. Llave externa. Identificador de la canción.
- PREDICCIÓN: Real. Valor entre 0 y 5.25. Predicción para el usuario sobre la canción.

LISTAS_REP

Tabla que contiene los siguientes campos:

- ID_USUARIO: Entero. Llave primaria. Llave externa. Identificador del usuario.
- ID_CANCION: Entero. Llave primaria. Llave externa. Identificador de la canción.
- POS: Real. Valor que determina la posición de la canción en la lista de reproducción del usuario.

PROFILE_VALUES⁴

- FID: Entero. Llave primaria. Identificador del campo de perfil de usuario.
- UID: Entero. Llave primaria. Llave externa. Identificador del usuario.
- VALUE: Conjunto de valores. Género favorito del usuario. Puede tomar los valores siguientes:
 - ACÚSTICA
 - ROCK
 - TECHNO
 - INSTRUMENTAL

⁴ Esta tabla es generada por el sistema de gestión de contenidos para almacenar valores de campos de perfil para los usuarios. En nuestro diseño de datos original, formaría parte de la tabla Usuarios.

- POP
- HOUSE
- PSICODELIC
- AMBIENTAL
- BSO

4.4.2. Diseño de la Interfaz

En esta etapa del diseño del sistema software se define cual va a ser la apariencia visual de la aplicación, es decir, se define la interfaz visual entre el usuario y la aplicación. Sin duda, llevar a cabo un buen diseño de la interfaz resulta decisivo debido a que esta debe ser atractiva para el usuario de la aplicación pero al mismo tiempo debe resultar fácil de entender y de trabajar sobre ella [19].

Esta importancia se acrecienta aun más en el caso de nuestro proyecto ya que la interfaz de nuestro proyecto es una interfaz Web y la usabilidad Web es un tema candente, sujeto a cierta controversia [13]. Esto es así porque para las aplicaciones con interfaces Web no existe una guía de estilo estándar como existe, por ejemplo, para el desarrollo de interfaces para aplicaciones de escritorio en sistemas operativos como Windows XP/Vista y que resulten atractivas y familiares al mismo tiempo. Cada programador, desarrollador o diseñador Web debe definir, pues, su propia guía de estilo y procurar que, en base a ella, la interfaz resultante alcance un nivel digno de atractivo visual, familiaridad y facilidad de uso.

En este apartado definiremos nuestra guía de estilo de la aplicación, describiremos y analizaremos las metáforas y revisaremos el diseño de pantallas y los caminos de navegación (storyboards) empleados para el desarrollo de la interfaz de nuestro prototipo de aplicación software.

4.4.2.1. Guía de Estilo

Antes de entrar de lleno en el diseño de la interfaz de usuario, es conveniente definir el estilo de la misma. Esto es de suma importancia cuando el diseño va a ser compartido entre varios diseñadores, ya que ayuda a mantener la coherencia interna de la interfaz.

Sin embargo, a pesar de lo que pueda parecer en un principio, también es de gran utilidad definir una guía de estilo cuando sólo hay un diseñador encargado de la interfaz. Esto es debido a varias razones:

- En ocasiones es posible que mantener la coherencia y consistencia de una interfaz, si esta es muy grande o muy ambiciosa, sea complicado incluso si sólo hay un diseñador.
- El diseñador primitivo puede, por las más diversas razones, abandonar el diseño. Por tanto, es de gran utilidad para sus sustitutos contar con una guía de estilo predefinida para no tener que empezar desde cero de nuevo. Esto es también aplicable cuando no es el diseñador original el que se encarga de la actualización o el mantenimiento de la interfaz.

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

Fuentes

Estándar

- Tipo: Arial
- Color: #666666
- Tamaño: 0.85 em.

Estándar (nombres de campo).

- Tipo: Arial (Bold).
- Color: #666666
- Tamaño: 0.85 em.

Menús

- Tipo: Arial
- Color: #0062A0
- Tamaño: 0.85 em.

Información listas de reproducción

- Tipo: Arial
- Color: #666666
- Tamaño: 0.75 em.

Cabeceras

- Tipo: Arial
- Color: #666666
- Tamaño: 0.75 em.

Enlaces

Color del enlace azul, mismo código empleado para la fuente en menús. Texto no subrayado por defecto, subrayado al pasar por encima el cursor. El destino de un enlace se abre en la misma pestaña/ventana, ya que son enlaces a páginas internas del sitio. Para enlaces de elementos de menú, el enlace o elemento de menú activo aparecerá en negrita.

Colores

El color predominante será el blanco marfil, acompañado de un gris suave (25%) para los menús y un gris muy oscuro para la cabecera del sitio.

Alineación

Texto y contenido alineado a la izquierda, salvo tablas cuya alineación es centrada.

4.4.2.2. Metáforas

Una metáfora es el empleo de un objeto con un significado o dentro de un contexto diferente al habitual. En el diseño de una interfaz gráfica, la utilización de metáforas resulta muy útil ya que permiten al usuario, mediante la comparación con otro objeto o concepto, comprender de forma más intuitiva las diversas tareas que la interfaz permite desarrollar. Las metáforas juegan un papel muy importante en el éxito o fracaso de una interfaz, por tanto deben diseñarse con mucho cuidado.

Al igual que pasa en el ámbito de la literatura, para que una metáfora cumpla con su cometido, el desarrollador de la aplicación y el usuario final de la misma deben tener una base social y cultural similar. Es muy posible que el uso de un icono de manera metafórica sea entendido de una manera por el usuario occidental y de otra bien distinta por un usuario oriental. Pensemos, por ejemplo, en el icono de una mano con el pulgar levantado: para un usuario occidental es muy probable que signifique que la tarea se ha realizado con éxito. Sin

embargo, un usuario natural de Turquía, por ejemplo, consideraría este gesto como un acto ofensivo.

Es necesario, pues, tener en cuenta la base cultural de los usuarios para diseñar una buena metáfora. En definitiva, hay que intentar que las metáforas empleadas sean lo más universales posibles, a fin de que sean comprendidas a la perfección por la mayor parte del público potencial.

Las aplicaciones de escritorio de Windows actuales suelen seguir la Guía de Estilo Vista y utilizan una serie de metáforas con las que el usuario está plenamente familiarizado (por ejemplo, una papelera con el símbolo de la triple flecha de reciclado indica el lugar al que provisionalmente se destinan aquellos archivos que han sido borrados). En el mundo de las aplicaciones Web también existe una gran cantidad de metáforas de amplia difusión como, por ejemplo, el conocido icono del carrito de la compra que emplean casi todos los comercios online.

Pero las metáforas no solo dependen del tipo de aplicación (escritorio o Web) sino también del ámbito de la misma. Por ejemplo, el carrito de la compra es una metáfora conocida por todos, pero si nuestra aplicación no va a vender producto alguno al usuario no conviene utilizarla ya que puede dar lugar a confusión.

En nuestra aplicación hemos utilizado las siguientes metáforas:

-  Esta metáfora aparece a la izquierda de los paneles de puntuaciones, junto a la puntuación de menor valor, e indica que si elegimos una puntuación próxima a dicha metáfora significa que la canción no es del agrado del usuario.
-  Esta metáfora aparece a la derecha de los paneles de puntuaciones, junto a la puntuación de mayor valor, e indica que si elegimos una puntuación próxima a dicha metáfora significa que la canción es del agrado del usuario.

Para entender mejor el funcionamiento de estas dos metáforas en la aplicación, veamos el diseño del formulario para puntuar canciones:

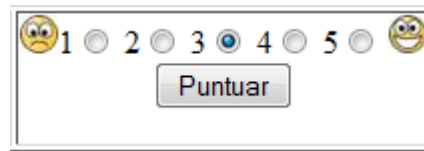


Figura 4.9: Panel de puntuaciones de la aplicación.



El propio panel de controles del reproductor de la radio es una metáfora de los radios, equipos de música y demás aparatos de reproducción de audio y vídeo en general. Incluye botones para pasar a la anterior y siguiente canción, comenzar, detener o reanudar la reproducción, ajustar el volumen y desplazarse dentro de la canción que está siendo reproducida en el momento.

- La disposición de los diferentes álbumes, de forma matricial, en la sección “álbumes” recuerda a cómo los discos de música son colocados en un escaparate real.

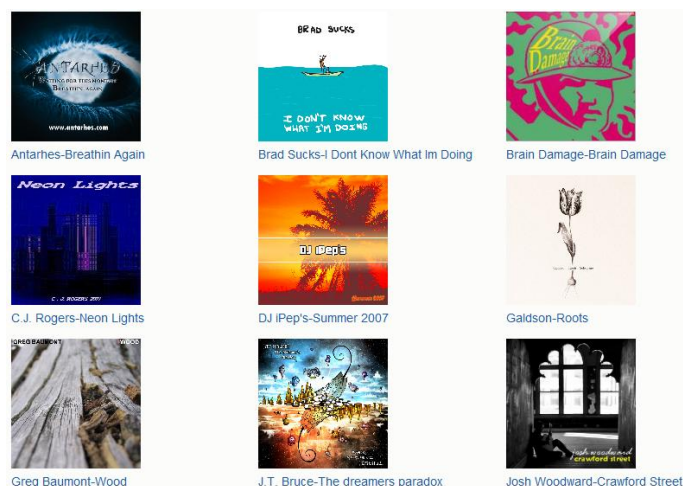


Figura 4.10: Disposición de la lista de álbumes.

-  Esta metáfora aparece en el libro de visitas e indica al usuario a dónde debe dirigirse para escribir o responder a un comentario.
-  Esta metáfora aparece en el libro de visitas e indica al usuario a dónde debe dirigirse para editar un comentario anteriormente publicado por él.

4.4.2.3. Prototipos de la aplicación

En este apartado, vamos a definir la estructura de nuestra interfaz con el usuario, ya que la parte del servidor no tendrá interfaz ninguna pues se ejecutará automáticamente sin intervención manual.

Mediante el diseño de prototipos se pretende tener un esbozo de lo que será la interfaz de usuario de nuestra aplicación. Dichos prototipos no expresan el diseño final, simplemente dan una idea de lo que será nuestro sistema para el usuario final. Estos prototipos serán, por tanto, susceptibles de cambio durante el proceso de implementación.

En nuestro caso, optamos por el **diseño de pantallas** como técnica de definición de prototipos de la interfaz. La mejor herramienta para ello es, simplemente, usar dibujos hechos a mano con lápiz y papel. La razón es que estos diseños de pantalla son un vehículo para poder analizar la calidad de nuestro diseño, por ello no es adecuado perder tiempo ni dinero realizando estos diseño previos con alguna herramienta especializada.

Pantalla principal/radio (Usuario anónimo).

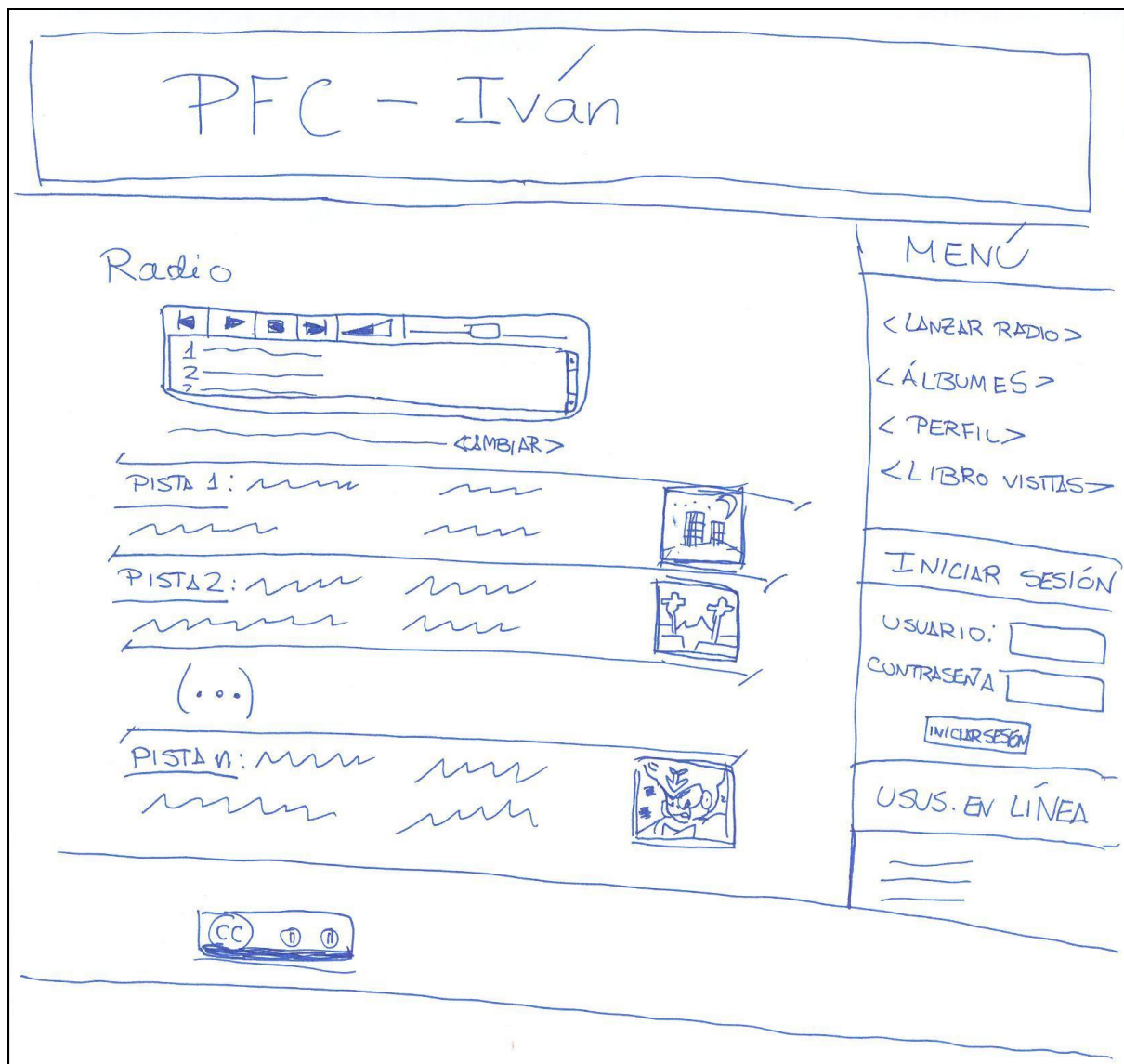


Figura 4.11: Pantalla principal/radio para el usuario anónimo

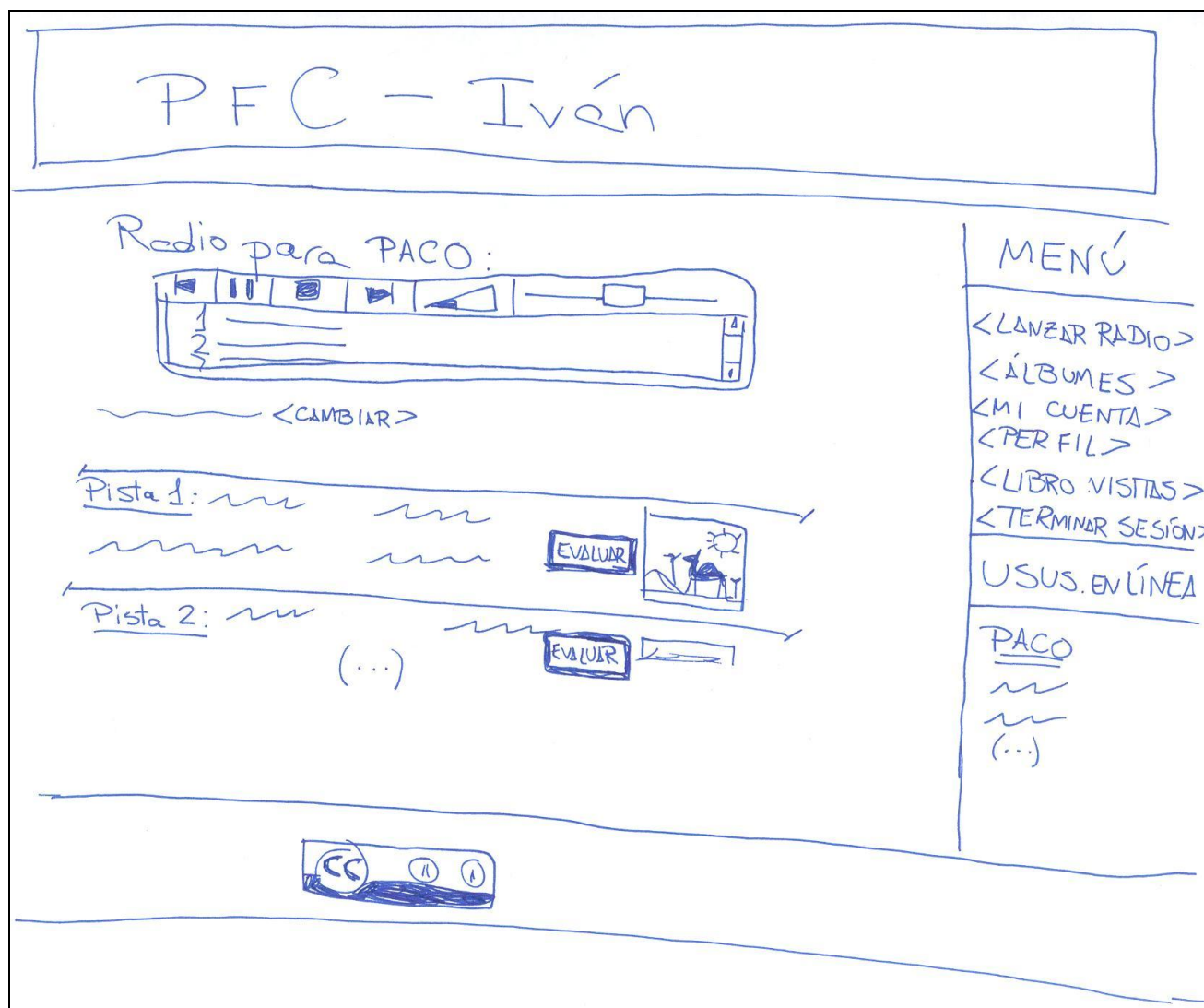
Pantalla principal/radio (Usuario registrado)

Figura 4.12: Pantalla principal/radio para el usuario registrado

Pantalla Panel de puntuaciones



Figura 4.13: Panel de puntuaciones

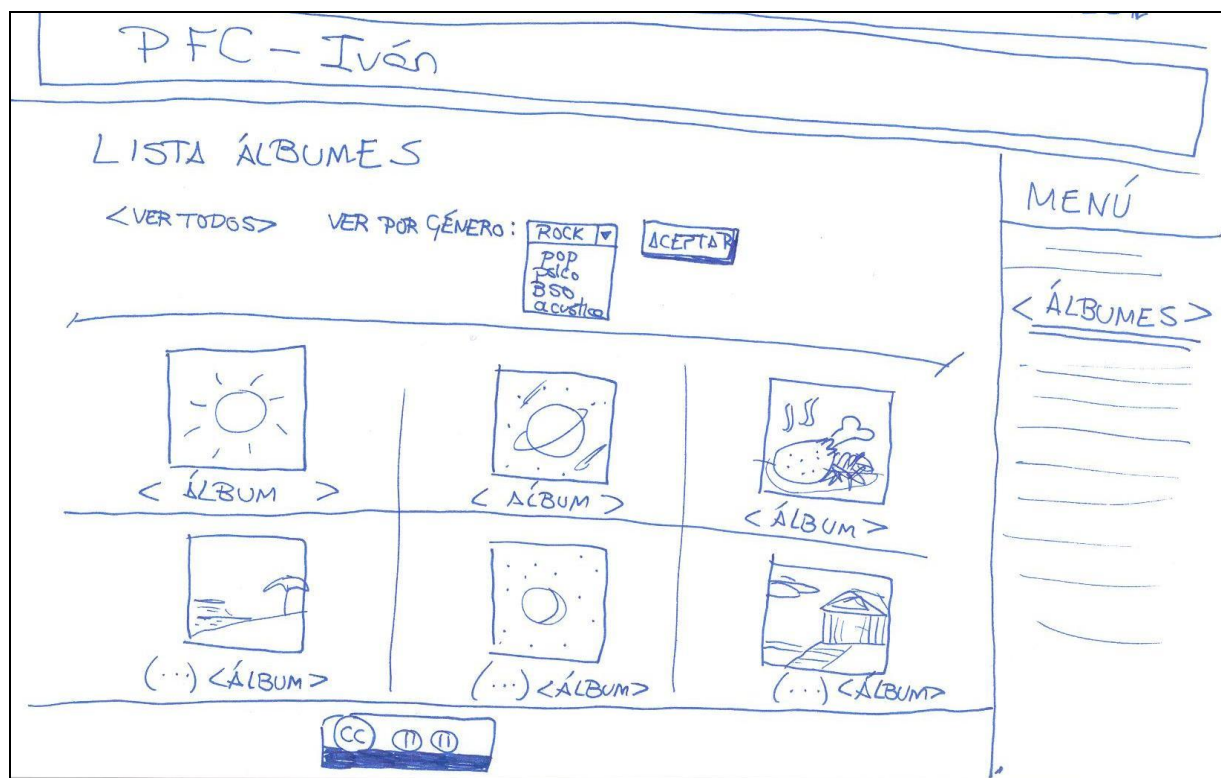
Pantalla Lista de Álbumes

Figura 4.14: Pantalla Lista de Álbumes

Pantalla Álbum

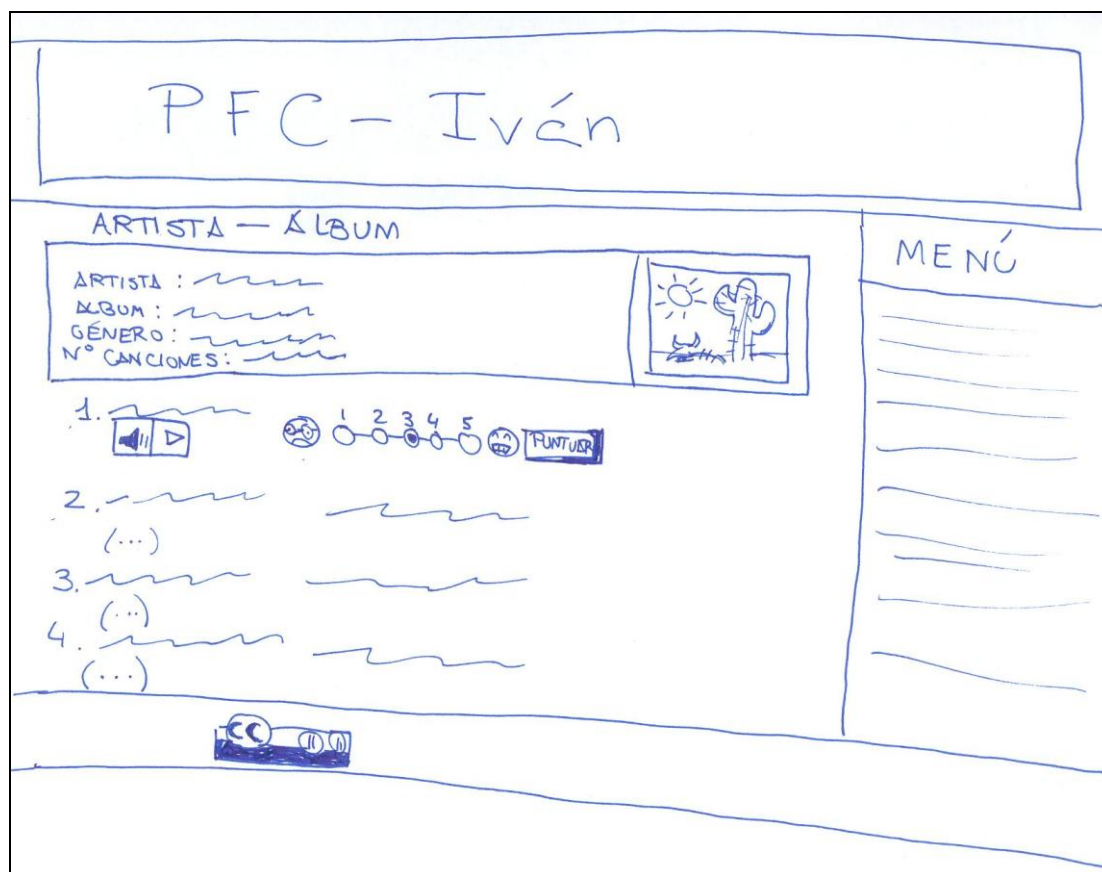


Figura 4.15: Pantalla álbum

Pantalla Perfil del Usuario

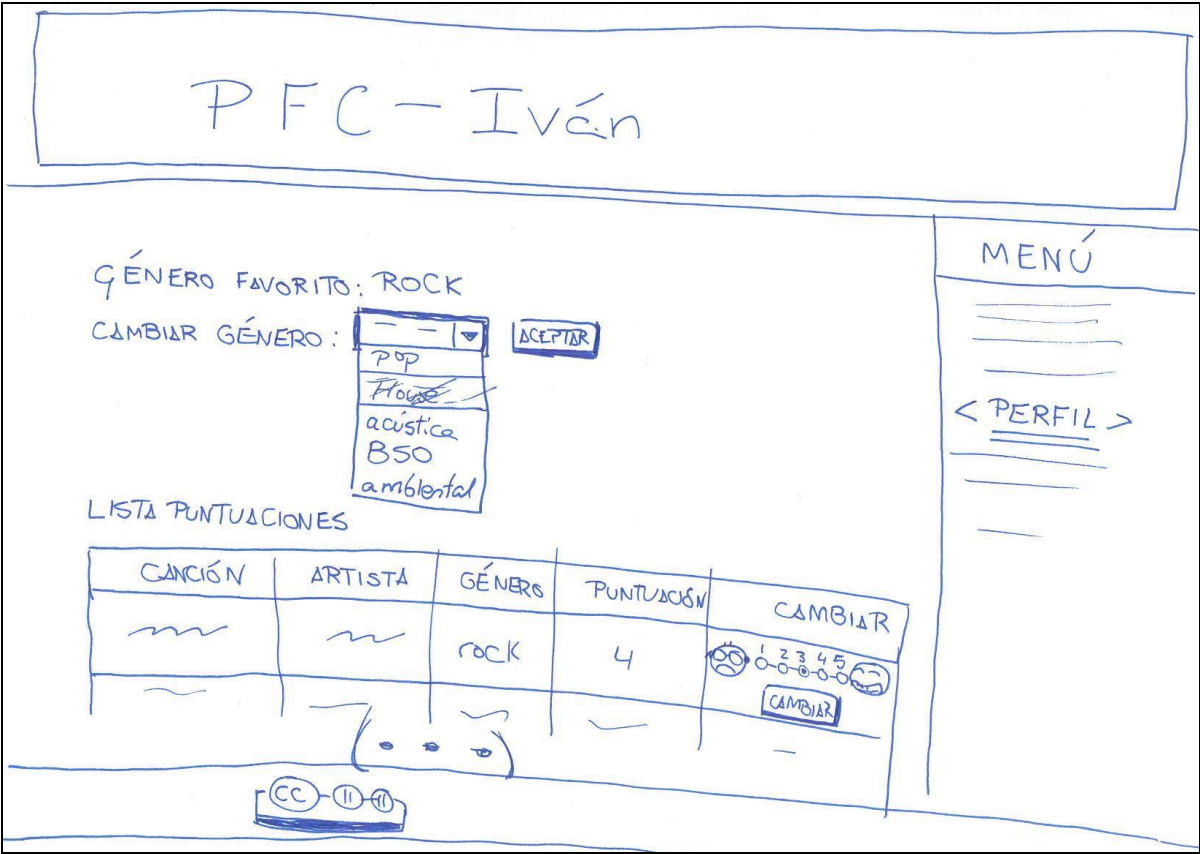


Figura 4.16: Pantalla Perfil del Usuario

Pantalla Libro de Visitas



Figura 4.17: Pantalla Libro de Visitas

4.4.2.4. Caminos de navegación

Hasta este momento tenemos un diseño visual de la interfaz estática, es decir, cada pantalla diseñada individualmente, pero no tenemos una idea de si en el conjunto de la interacción, la acción va a transcurrir de forma fluida y comprensible para el usuario. Para ello vamos a diseñar la interfaz en movimiento y comprobar que es usable.

Para estudiar los caminos de navegación se empleará una herramienta llamada **storyboard**, que consiste en mostrar, a modo de secuencia, las diferentes pantallas por las que se va pasando al realizar el usuario una determinada acción sobre la aplicación.

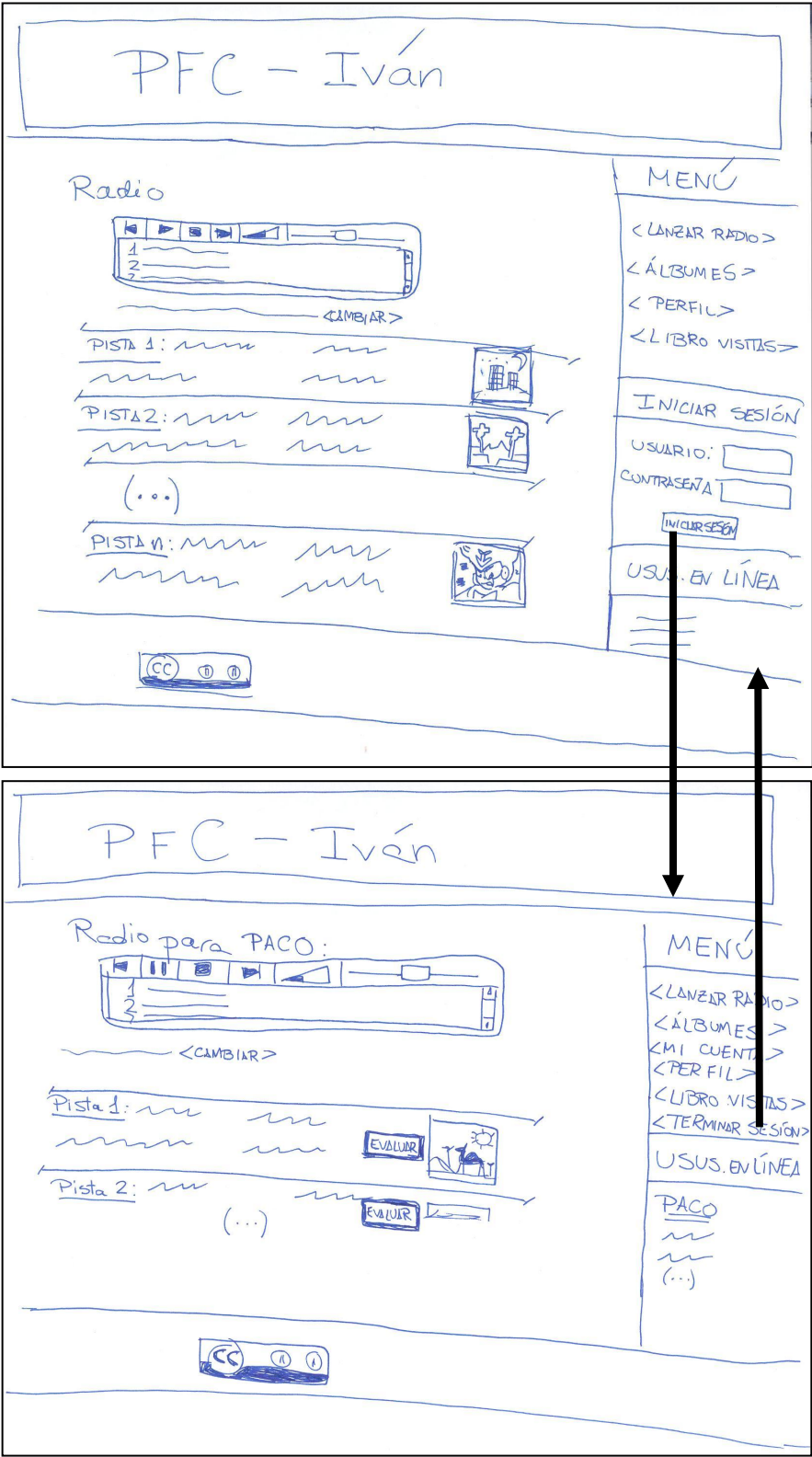
El procedimiento es el siguiente: se sitúan capturas de las pantallas de la interfaz unidas mediante flechas para indicar el camino que sigue la interacción. La posición de origen de las flechas debe ayudar a entender cuál es el elemento que ha desencadenado el paso de una pantalla a otra. Los storyboards también están muy ligados a los escenarios anteriormente vistos.

El storyboard sirve de prototipo para ser evaluado por el usuario y poder introducir correcciones en fases tempranas, ya que cuanto más tiempo se tarde en validar una interfaz, más coste de tiempo y trabajo acarreará.

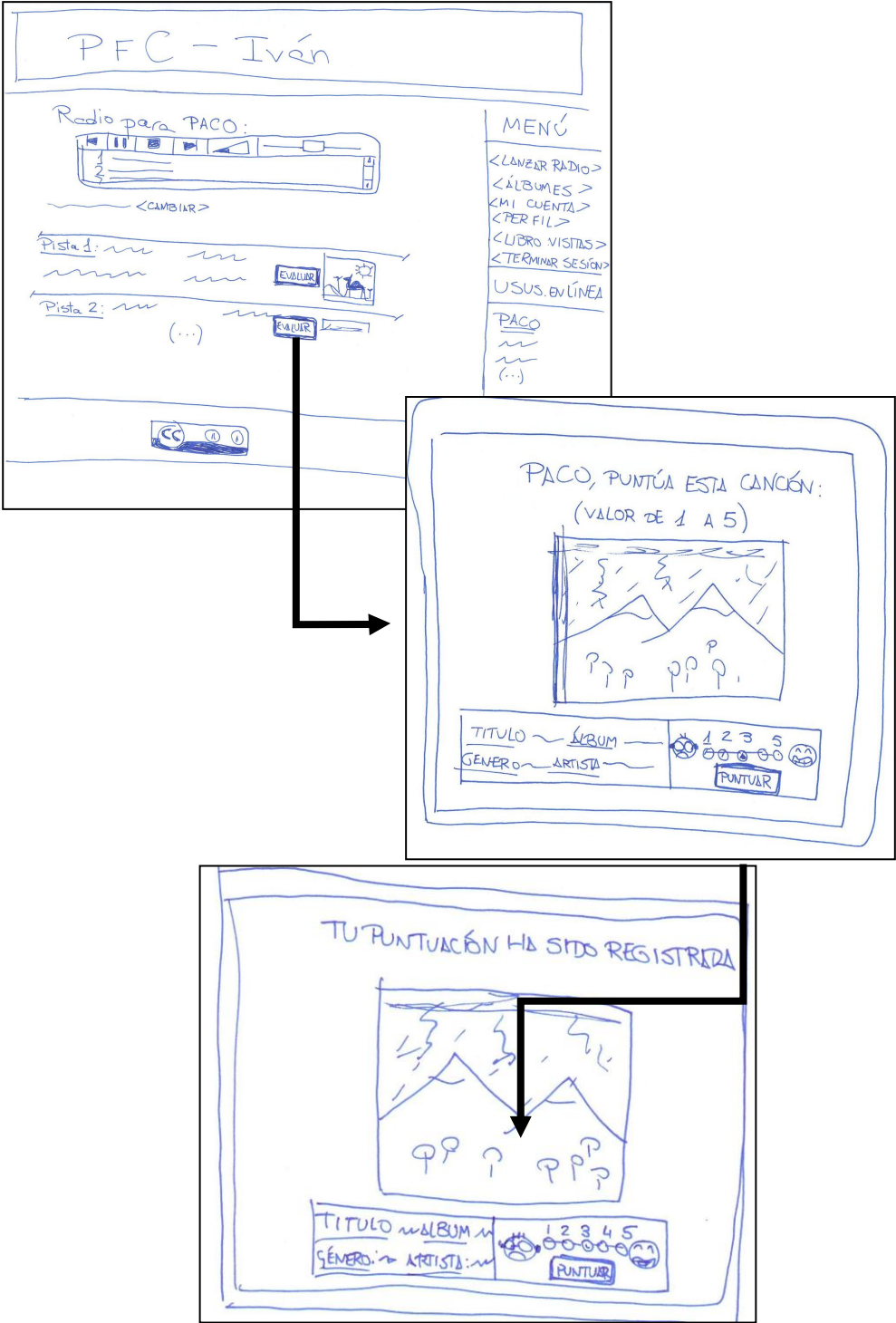
A continuación, mostramos los storyboards para las acciones más importantes que se pueden llevar a cabo en nuestro sistema.

- Storyboard validación de usuario/ cierre de sesión.
- Storyboard: Puntuar canción.
- Storyboard Ver álbum de música.
- Storyboard Modificar perfil.
- Storyboard escuchar radio partiendo de una pantalla diferente.

Storyboard Validación de usuario/cierre de sesión



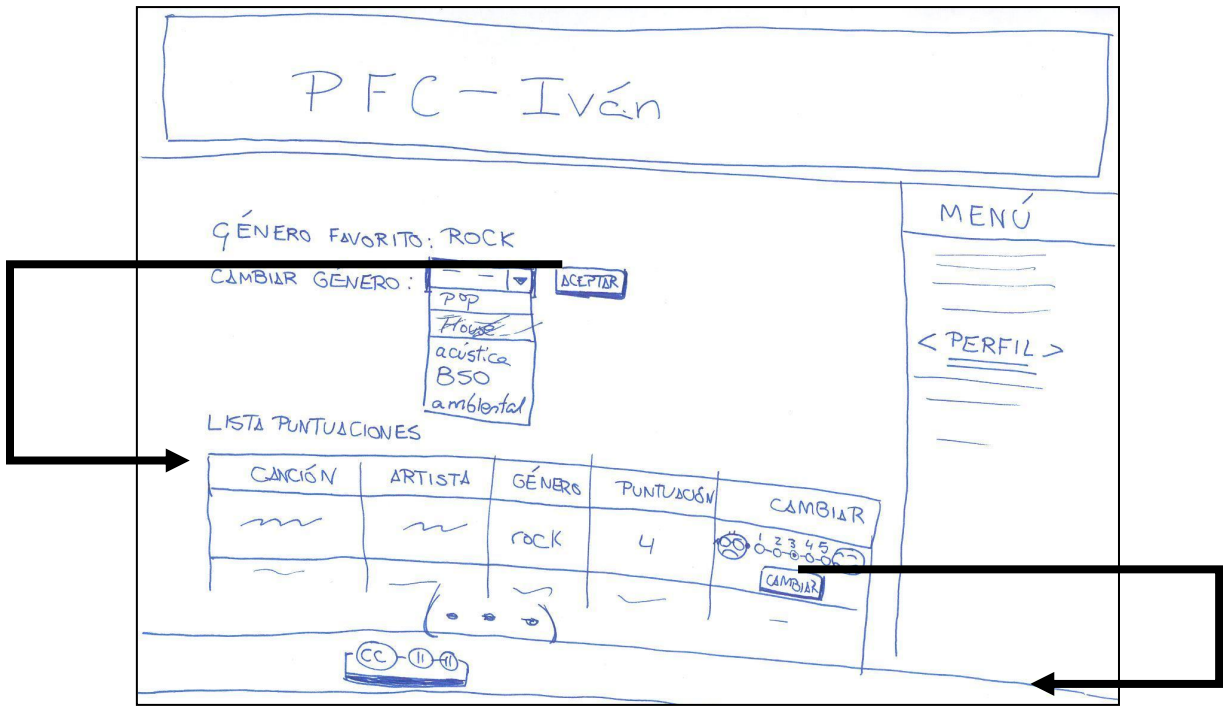
Storyboard Puntuar canción



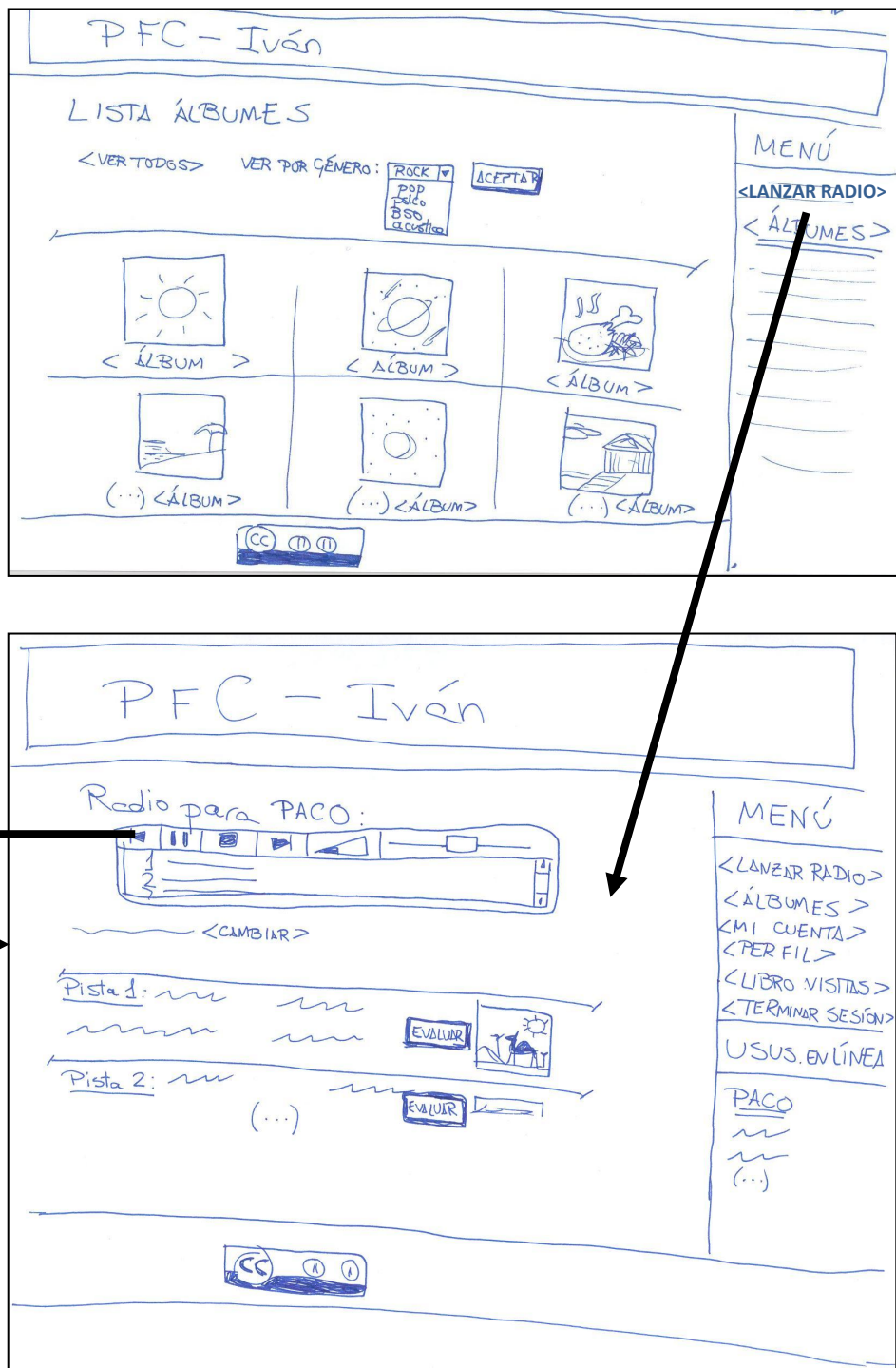
Storyboard Ver álbum de música



Storyboard Modificar perfil



Storyboard Escuchar radio (partiendo de una pantalla diferente)



4.5. Implementación

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

4.5.1. Tipo de arquitectura de la aplicación

En el caso de nuestra aplicación, desarrollaremos un Sistema de Recomendación con una arquitectura cliente/servidor y una interfaz Web de comunicación con el usuario. El funcionamiento de este tipo de arquitectura es sencillo: 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.

Una arquitectura cliente/servidor (véase Figura 4.18) elimina del usuario final de la aplicación la tarea de tener que instalar dicha aplicación en su máquina, consiguiendo además que cada usuario acceda únicamente a la información que le corresponde. Además, dado su diseño modular, esta arquitectura es fácilmente escalable y ampliable tanto en nuevos clientes como en servidores añadidos.

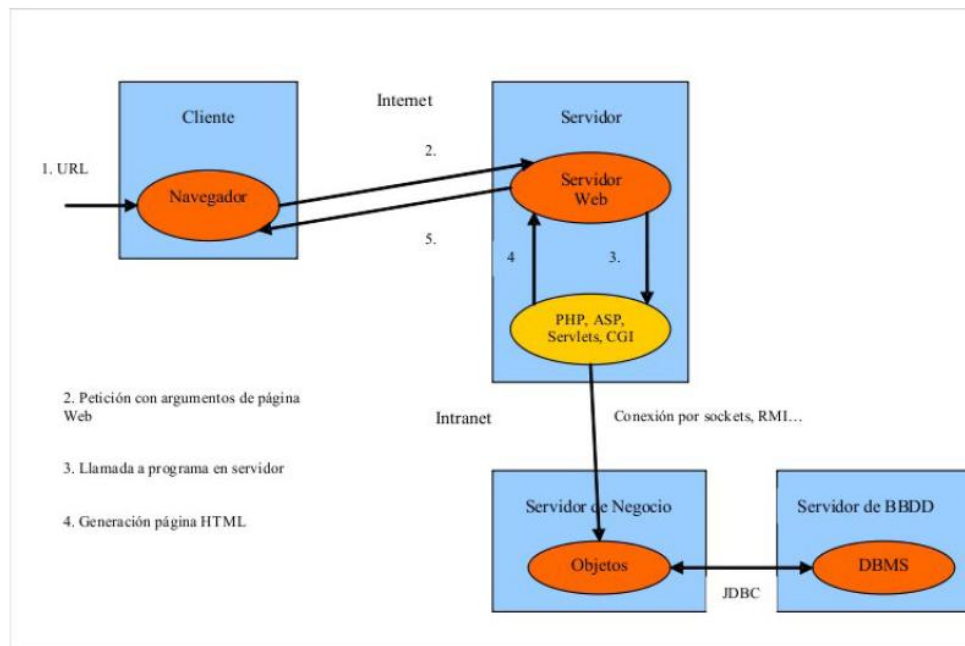


Figura 4.18: Arquitectura Cliente/Servidor

4.5.2. Lenguajes de programación utilizados

Parte Cliente

Ante la arquitectura y el funcionamiento previsto de nuestra aplicación, resulta evidente que el uso de HTML simple y plano no es adecuado, sino que precisa de otros lenguajes capaces de generar contenido dinámico desde el servidor, de manera transparente al usuario final. Para tal fin, existen varias alternativas, desde Perl a ASP (Active Server Pages) pasando por el uso de CGIs (Common Gateway Interfaces) o JSP (Java Server Pages). Nosotros hemos optado por utilizar el lenguaje PHP [12].

PHP, acrónimo recursivo de Hypertext Preprocessor, es un lenguaje de programación interpretado, que se ejecuta del lado del servidor y genera contenido dinámico a petición del cliente.

Se trata de un lenguaje con una serie de importantes ventajas frente a otros lenguajes que realizan funciones parecidas, como son las siguientes:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL (de gran utilidad en la implementación de este proyecto).
- Lectura y manipulación de datos desde diversas fuentes, incluyendo datos que pueden introducir los usuarios desde formularios HTML.
- Capacidad de expandir su potenciales gracias al enorme abanico de módulos disponibles (llamados extensiones o ext's).
- Posee una amplia documentación en su página oficial, entre la cual cabe destacar la explicación de todas las funciones del sistema, mediante ejemplos y archivos de ayuda.
- No precisa de la declaración de variables.
- Es libre, por lo que se presenta como una alternativa fácilmente accesible para todos.
- Permite las técnicas de programación Orientada a Objetos.
- Permite la creación de formularios para la Web.
- Amplia biblioteca nativa de funciones incluida. No requiere definición de tipos de variables ni un manejo avanzado a bajo nivel.

Estas características hacen de PHP el lenguaje ideal para nuestros propósitos:

- El cliente solicita cualquier funcionalidad.
- El servidor, mediante PHP, conecta con nuestra base de datos MySQL para obtener en cada momento los datos pertinentes.
- También en PHP se realizan los cálculos y acciones sobre los datos recién obtenidos.
- Finalmente, genera el código xHTML adecuado y se lo presenta al cliente de manera totalmente transparente.

Con PHP es suficiente para satisfacer las funcionalidades que debe presentar la aplicación a sus usuarios. Sin embargo, para realizar una adecuada implementación de la interfaz Web se hace necesario el uso de otro lenguaje: CSS.

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

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

Parte Servidor

La implementación de esta parte se ha realizado usando Java y sentencias sql (MySQL) mediante el empleo de la API conocida como JDBC.

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los 90. El lenguaje en sí toma buena parte de su sintaxis de los lenguajes C y C++, pero tiene un modelo de objetos más sencillo y elimina además herramientas de bajo nivel que pueden inducir a errores, como la manipulación directa de punteros.

Las aplicaciones Java están normalmente compiladas en un bytecode⁵, aunque la compilación en código máquina nativo es también posible. En tiempo de ejecución, el bytecode suele ser interpretado o compilado nuevamente, a código nativo para su ejecución, aunque también es posible la ejecución directa del bytecode vía hardware.

La implementación original y de referencia del compilador, la máquina virtual y las librerías de clases de Java fueron desarrolladas por Sun Microsystems en el año 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje por medio del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas de las cuales se encuentran bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó buena parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de manera que actualmente todo el Java de Sun es software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre).

⁵ Un tipo de código intermedio utilizado por Java, de mayor abstracción que el código máquina, que suele ser tratado como un fichero binario ejecutable.

4.5.3. Herramientas de desarrollo

Una vez descritos los lenguajes de programación empleados para este proyecto, pasamos a especificar las herramientas empleadas para el desarrollo de la aplicación a través de dichos lenguajes.

Para generar código Java se requiere de una herramienta o entorno de desarrollo. En nuestro caso, hemos optado por utilizar el entorno de desarrollo integrado (IDE) **NetBeans 6**. Se trata de una plataforma para el desarrollo de aplicaciones de escritorio usando Java.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes software llamados módulos. Un módulo es un archivo que contiene varias clases de Java escritas para interactuar con las APIs de NetBeans y un archivo especial (archivo manifest) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas mediante la agregación de nuevos módulos. Dado que los módulos pueden ser desarrollados de forma independiente, las aplicaciones basadas en NetBeans pueden ser fácilmente extendidas por otros desarrolladores de software.

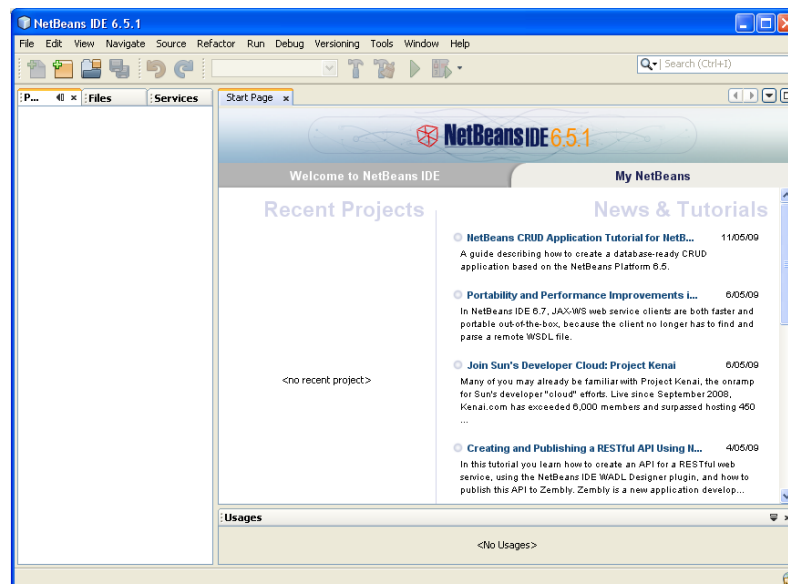


Figura 4.19: Panel de bienvenida de NetBeans 6

NetBeans es un proyecto de código abierto de gran éxito y una enorme cantidad de usuarios en la actualidad, una comunidad en continuo crecimiento y en torno a 100 socios en todo el mundo. Fue fundado como proyecto de código abierto por Sun Microsystems en junio de 2000 y continua siendo el principal patrocinador de los proyectos.

Para la generación de código CSS y xHTML no se requiere una herramienta o entorno de desarrollo en particular. Con un simple editor de textos se pueden escribir las sentencias y etiquetas y guardar el resultado, especificando la extensión correspondiente. El desarrollo de aplicaciones Web mediante estos editores puede ser bastante delicado a veces, ya que son demasiado genéricos y no son capaces de reconocer la sintaxis. Una posible solución es el empleo de editores de textos avanzados como Notepad++.

Notepad++ (Figura 4.20) es un editor de código fuente gratuito, que soporta varios lenguajes de programación y se ejecuta bajo sistemas operativos Windows. Algunas de las características más destacables de Notepad++ son:

- Sintaxis coloreada y envoltura de sintaxis.
- Autocompletado.
- Soporte para buscar/reemplazar expresiones regulares.
- Resaltado de paréntesis y sangrías.
- Grabación y reproducción de macros.
- Multi-Vista.
- Multi-Documento.

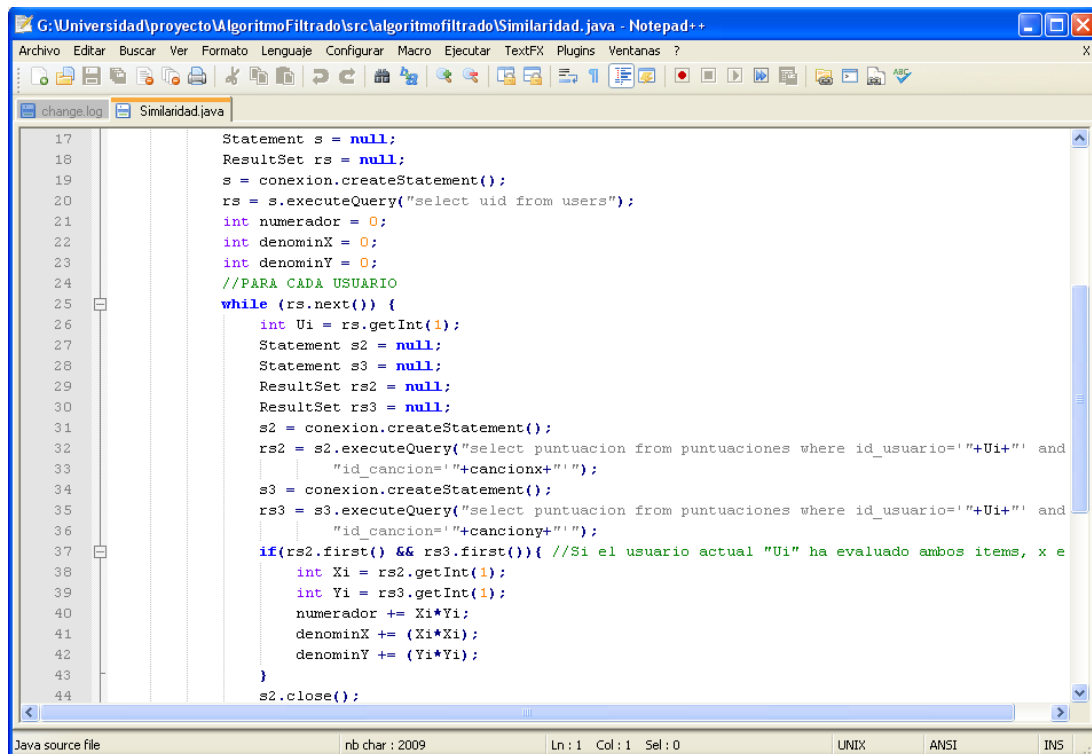


Figura 4.20: Ejemplo de funcionamiento de NotePad++

Para llevar a cabo la funcionalidad PHP y el desarrollo en general de la interfaz de la aplicación se han utilizado dos herramientas perfectamente compatibles:

- Para la construcción del sitio Web se ha utilizado Drupal (Figura 4.21).
- Para el manejo de la base de datos MySQL se ha usado la herramienta phpMyAdmin.

Drupal es un Sistema de Gestión de Contenidos (CMS) ampliamente reconocido, con estructura modular y altamente configurable, que permite la construcción de sitios Web en línea potentes.

Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una amplia comunidad de usuarios. Destaca por la calidad de

su código y de las páginas generadas, el respeto por los estándares Web y una especial atención hacia la usabilidad y consistencia de todo el sistema.

El diseño de Drupal es particularmente apto para la construcción y gestión de grandes comunidades en Internet. No obstante, posee una gran flexibilidad y adaptabilidad, lo que unido a la enorme cantidad de módulos adicionales disponibles, lo hace idóneo para diferentes tipos de sitios Web. Algunos ejemplos de módulos disponibles son:

- **Calendar:** Para la creación de calendarios de eventos.
- **CCK:** Permite crear tipos de contenido y formularios de forma muy flexible y altamente configurable.
- **Views:** Módulo empleado para la publicación de contenidos sintetizados con un formato común, a modo de vistas.
- **Forum:** Para la creación de foros de debate.
- **Profile:** Permite añadir y modificar campos para el perfil de los usuarios.



Figura 4.21: Panel de administración de Drupal

Miles de usuarios de todo el mundo participan activamente en drupal.org, el sitio oficial de desarrollo y coordinación de Drupal.

PhpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas Web, utilizando Internet. Actualmente permite la creación y eliminación de bases de datos, creación, eliminación y modificación de tablas, borrar, editar y añadir campos, ejecutar sentencias SQL, administrar claves en campos, administrar privilegios, y exportar e importar datos en varios formatos. Está disponible en 50 idiomas, y bajo licencia GPL.

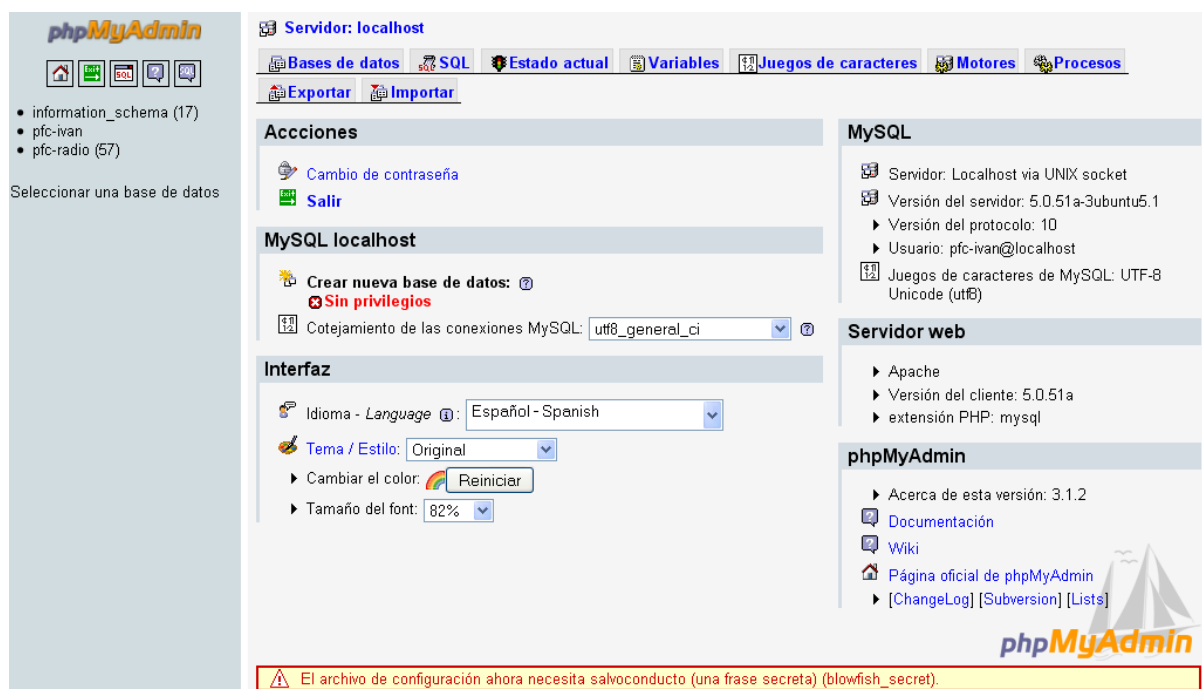


Figura 4.22: Pantalla principal de phpMyAdmin

4.5.4. Actualización del algoritmo de filtrado

La base principal para desarrollar un Sistema de Recomendación Colaborativo es contar con un buen algoritmo de filtrado colaborativo. Para ello es necesario refinarlo conforme los clientes vayan mejorando sus perfiles al puntuar nuevas canciones.

Para realizar esto, el servidor se encarga de lanzar de forma periódica una aplicación Java con el objetivo de actualizar el algoritmo de filtrado, siguiendo los pasos descritos en el capítulo 2 para el funcionamiento de un algoritmo de filtrado colaborativo basado en ítem. La aplicación está programada para su ejecución mediante un temporizador (usaremos el cron). Esta actualización requiere de un tiempo considerable para realizarse, ya que se deben volver a calcular los grupos de vecinos para todas las canciones de la base de datos.

CAPÍTULO 5

Conclusiones

5.1. Conclusiones

Actualmente vivimos en una sociedad en la que la cantidad de información que los seres humanos recibimos para realizar prácticamente cualquier tarea es cada vez mayor, llegando a ser casi imposible de procesar de forma natural. El fuerte desarrollo de la Sociedad de la Información se debe, en buena parte, a la mejora de las comunicaciones en los últimos años, el aumento de los medios de almacenamiento de información y, sobretodo, a la rápida y reciente explosión de Internet.

En muchas de las operaciones que los usuarios realizan con frecuencia en Internet, se hace presente el problema de la sobrecarga de información, por culpa del cual en muchas ocasiones el usuario que realiza una búsqueda en Internet pueda llegar a sentirse agobiado y confundido ante una gran cantidad de información difícil de asimilar, e incluso abandonar su intento de búsqueda.

Para solucionar este problema, aparecieron los Sistemas de Recomendación, que ayuda al usuario a encontrar aquella información de su interés mediante recomendaciones basadas en información proveniente del propio usuario, de otros usuarios o de expertos en la materia. Así, esta herramienta abre las puertas a los usuarios a un mundo donde únicamente reciben la información que realmente quieren recibir. Es por esta razón que, en un corto intervalo de tiempo, los sistemas de recomendación han alcanzado una gran popularidad, que se estima que aumente en los próximos años.

En este proyecto se ha desarrollado una radio online capaz de proporcionar a sus usuarios canciones de su agrado, además de otras canciones que, sin haber sido evaluadas por el usuario, se estima que sean de su gusto. Para ello, la radio se ha desarrollado en torno a un Sistema de Recomendación Colaborativo, siguiendo todo el sistema en su conjunto la arquitectura cliente/servidor.

Desde los primeros momentos de la concepción del proyecto, la intención era alojarlo en un sitio Web, para que diferentes usuarios se registren en él y lo visiten con mayor o menor frecuencia, escuchen diferentes canciones de distintos artistas y géneros musicales y realicen puntuaciones sobre las canciones. Basándose en estas puntuaciones, el Sistema de Recomendación subyacente a la radio crea un perfil de usuario y ofrece al usuario una serie de canciones recomendadas, de acuerdo con los gustos del propio usuario y de otros usuarios de gustos similares a él.

Para la realización del proyecto hemos recopilado un conjunto de datos musicales. Para ello, hemos optado por la obtención de una base de datos de álbumes musicales reales, todos ellos bajo algunas licencia de libre distribución del tipo *Creative Commons*, de manera que la versión prototipo de nuestro sistema cuenta con 197 canciones, agrupadas en 26 álbumes de 9 géneros musicales diferentes. Todos estos archivos musicales han sido alojados en el servidor utilizado para nuestro proyecto.

Además, hemos realizado un estudio de los diferentes tipos de Sistemas de Recomendación, el funcionamiento de los distintos algoritmos de filtrado y, basándonos en el estudio previo realizado en el proyecto “Sistema de Recomendación de Alquiler de Películas” [21], hemos implementado un algoritmo de filtrado colaborativo basado en ítem y, más concretamente, en el vecino más cercano (K-nn). De forma paralela, hemos desarrollado la parte cliente del proyecto, con el análisis, diseño e implementación de una interfaz Web usable para la comunicación con el usuario, donde el usuario puede escuchar y evaluar canciones, modificar su perfil de usuario y obtener recomendaciones.

Para el desarrollo de este proyecto hemos seguido todas las etapas de la Ingeniería del Software. En primer lugar, se determinaron las propiedades que el sistema debía satisfacer, así como las restricciones a las que se encuentra sometido. Seguidamente se ha

creado un modelo del sistema correcto, completo, consistente, claro y verificable. Finalmente se ha codificado este modelo en una versión prototipo y se ha instalado en el servidor.

Por último, añadir que para el autor del proyecto ha supuesto una enorme satisfacción el haber conseguido darle forma a una idea que, desde bastante tiempo atrás, habían plasmado tanto él como los tutores del proyecto; y el ver cómo poco a poco, y a pesar de las pequeñas adversidades que inevitablemente siempre van surgiendo durante el desarrollo de cualquier aplicación, mi aplicación ha conseguido llegar a buen puerto. Además, este proyecto me ha aportado conocimientos, a veces totalmente nuevos y otras veces para profundizar, en campos como el de los Sistemas de Recomendación, las tecnologías Web, el diseño y administración de bases de datos y la propia Ingeniería del Software. Esto ha supuesto para mí un verdadero reto desde los primeros momentos de su realización, pero a la vez y sobretodo, ha sido una experiencia muy positiva y enriquecedora.

ANEXO A

Manual de instalación del Servidor

Este manual recoge la instalación del servidor necesario para montar nuestra aplicación. En él, se describirá la instalación de elementos necesarios como el servidor Apache 2, php, mysql, la interfaz phpMyAdmin, la máquina virtual Java y NetBeans.

Consideraciones previas

Es posible, y de hecho normalmente se realiza así, la instalación de cada uno de los citados elementos por separado. No obstante, hemos optado por la instalación de **AppServ**, un paquete software de código abierto que integra todas las aplicaciones, a excepción del entorno de desarrollo y la máquina virtual de Java.

Hemos supuesto para este manual que la unidad principal de disco duro es C:, y que la unidad principal de CD/DVD es D:. Toda la instalación y configuración descrita en este manual se refiere a un servidor montado sobre un Sistema Operativo Windows.

Todo el material necesario para instalar y poner en funcionamiento el servidor se encuentra disponible en el disco que acompaña a esta memoria.

A.1. Instalación de AppServ

AppServ permite, de una forma fácil y rápida, la instalación de los siguientes elementos:

- Apache
- PHP
- MySql
- PhpMyAdmin.

Se puede descargar en la dirección <http://www.appservnetwork.com>, siendo la versión 2.5.10 la última versión estable en el momento de la realización de este manual, incluida además en el disco que acompaña a esta memoria. Dicha versión incluye las siguientes herramientas (ver Figura A.1):

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

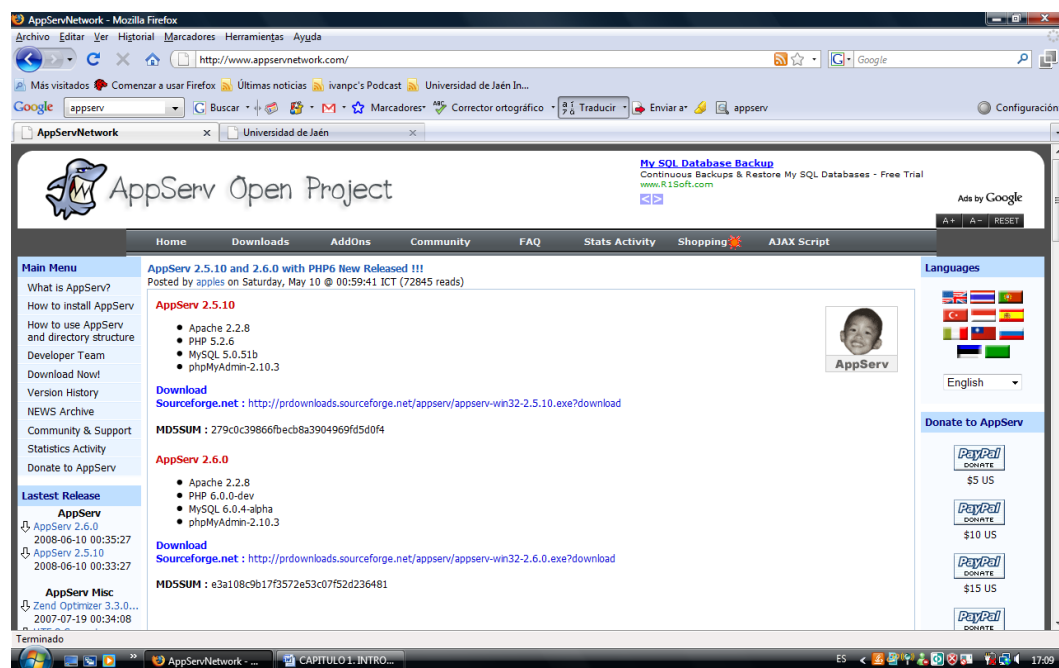


Figura A.1: Portal deAppservNetwork.com

Para descargar AppServ, basta con hacer clic sobre el enlace de descarga y elegir un directorio de destino en nuestro equipo para el instalador. Una vez descargado ejecutaremos el instalador haciendo clic sobre el archivo descargado *appserv-win32-x.x.x.exe* y nos aparecerá la siguiente ventana (Figura A.2).



Figura A.2: Instalador de AppServ

Al pulsar el botón 'Next', aparecerá el acuerdo de licencia que, tras leer con detenimiento, deberá ser aceptado, mediante la opción 'I Agree' (Figura A.3), para continuar.

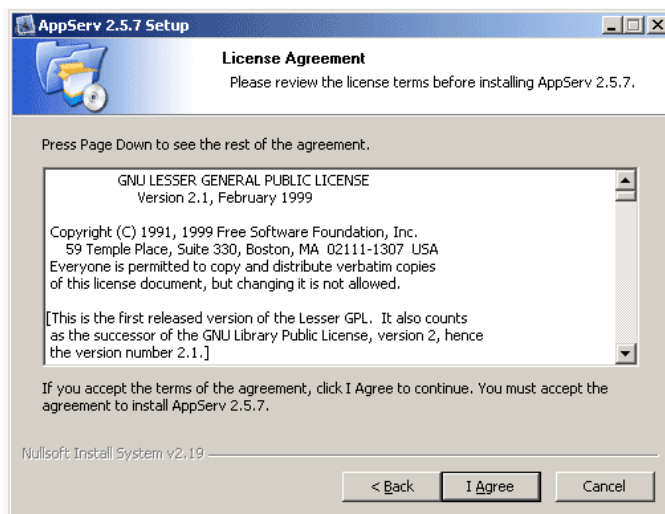


Figura A.3: Acuerdo de licencia de AppServ

A continuación el usuario debe elegir un directorio para la instalación de la aplicación, (normalmente aparecerá la ruta por defecto C:\AppServ) y seguidamente pulsar 'Next' (Figura A.4).

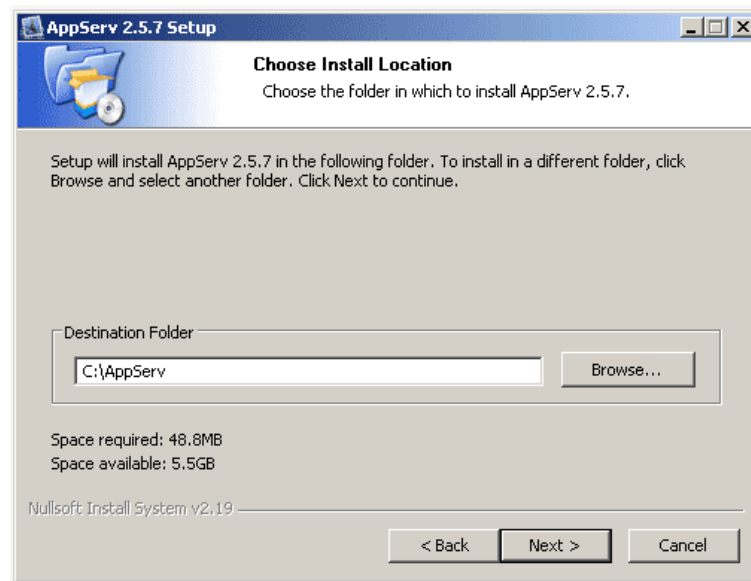


Figura A.4: Selección del directorio de instalación

La siguiente pantalla que aparece es la pantalla de selección de componentes a instalar (Figura A.5). Por defecto aparecen todos los componentes seleccionados para ser instalados. En nuestro caso, utilizaremos esta opción, por lo que solo habrá que pulsar nuevamente sobre 'Next'.

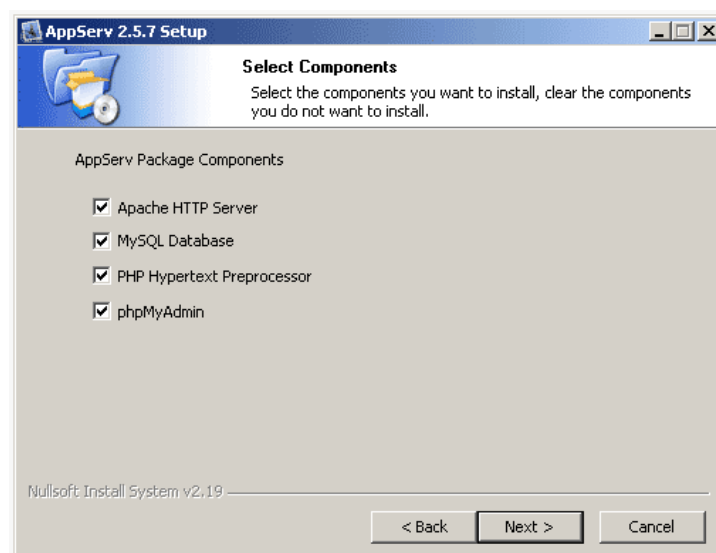


Figura A.5: Pantalla de selección de componentes

El siguiente paso es la configuración del servidor Apache (Figura A.6). AppServ se encarga por sí solo de realizar casi todas las configuraciones de forma automática, por lo que el usuario solo debe preocuparse de introducir los siguientes campos:

- Server Name (Nombre del servidor): Normalmente será 'localhost'.
- Administrator's email adress: Una dirección de correo válida del administrador del servidor.
- Apache HTTP Port: Puerto de escucha del servidor, dejar el puerto 80 asignado por defecto.

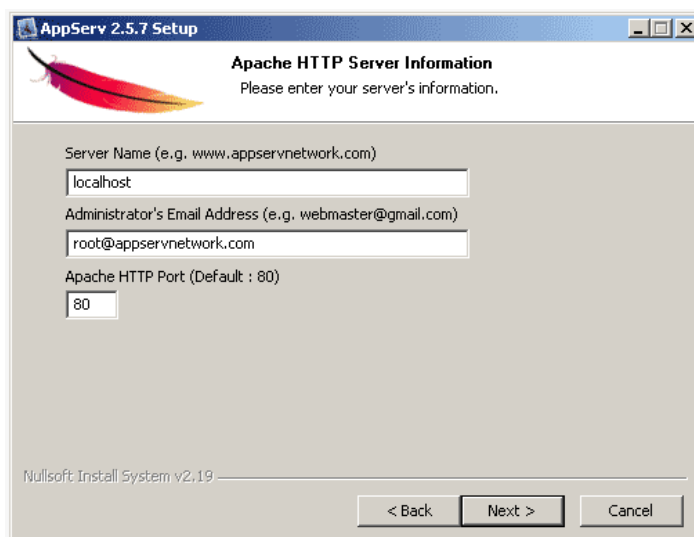


Figura A.6: Configuración del Servidor Apache

Una vez ha quedado definida la configuración de nuestro servidor Apache, el siguiente paso consiste en configurar el componente MySQL Server. Nos aparecerá la ventana mostrada en la Figura A.7:

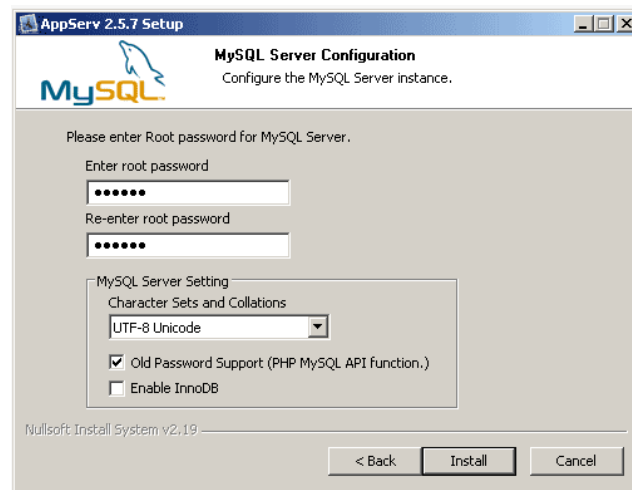


Figura A.7: Configuración de MySQL Server

En nuestro caso, basta con introducir dos veces la contraseña para nuestra cuenta de usuario administrador 'root', que es el usuario por defecto que AppServ crea para MySQL. La contraseña de la base con la que nuestra aplicación viene preparada para funcionar es 'administrador'. El resto de opciones se puede dejar con los valores asignados por defecto, por lo que una vez introducida una contraseña pulsaremos sobre 'Next'.

Tras finalizar la instalación (Figura A.8), se nos preguntará si deseamos comenzar la ejecución de Apache y MySQL de inmediato, aceptamos y pulsamos sobre el botón 'Finish'.

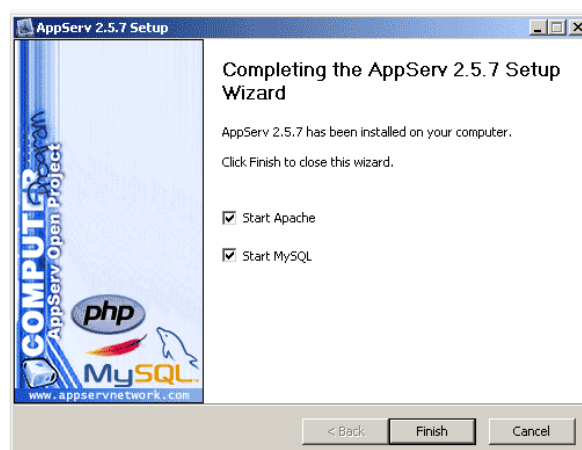
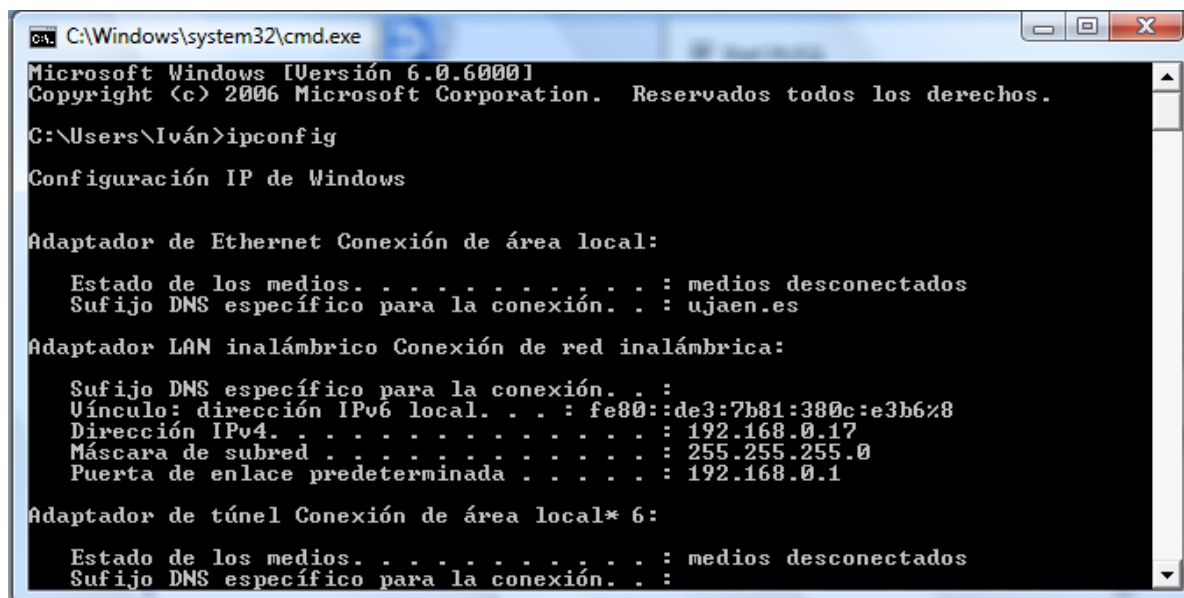


Figura A.8: Pantalla fin de instalación de AppServ

Una vez instalado AppServ y comenzado la ejecución del servidor Apache y MySQL, es importante saber cómo acceder a nuestro servidor desde el navegador.

Para ello basta con escribir nuestra dirección IP en el navegador. En caso de no saberla, se puede averiguar fácilmente ejecutando la línea de comandos de Windows (cmd), tecleando la orden 'ipconfig'. El campo 'DirecciónIPv4' nos da el número de nuestra dirección IP (Figura A.9).



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.0.6000]
Copyright (c) 2006 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Iván>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Conexión de área local:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . : ujaen.es

Adaptador LAN inalámbrico Conexión de red inalámbrica:

    Sufijo DNS específico para la conexión. . :
    Vínculo: dirección IPv6 local. . . : fe80::de3:7b81:380c:e3b6%8
    Dirección IPv4. . . . . : 192.168.0.17
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . : 192.168.0.1

Adaptador de túnel Conexión de área local* 6:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :
```

Figura A.9: Obtención de la dirección IP

Tras introducir la dirección IP en el navegador, si todo ha ido correctamente, debería aparecernos una pantalla similar a la mostrada en la Figura A.10.

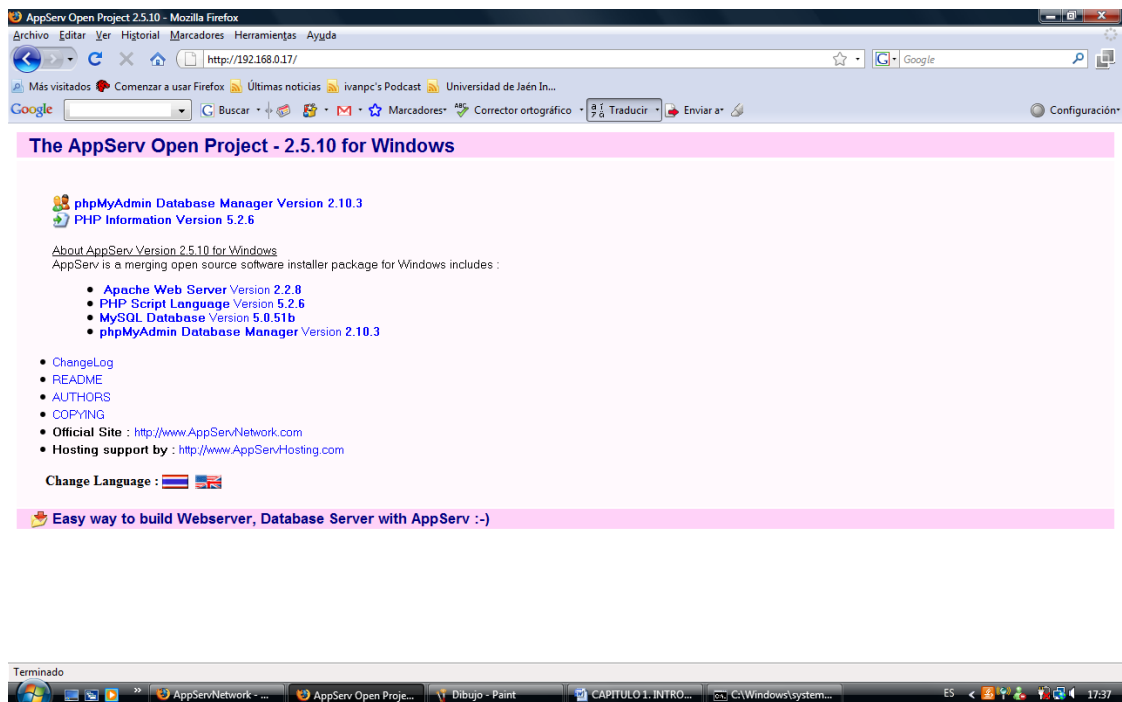


Figura A.10: Pantalla inicial del servidor AppServ

Desde aquí se puede acceder a las principales funcionalidades de los componentes instalados. En particular, nos interesa el acceso a la base de datos, para lo cual hacemos clic sobre el enlace 'phpMyAdmin Database Manager Version X.X.X.'. Se nos pedirá usuario y contraseña para continuar (Figura A.11). La contraseña es la que introducimos durante la instalación de MySQL (administrador), y el usuario es 'root'.

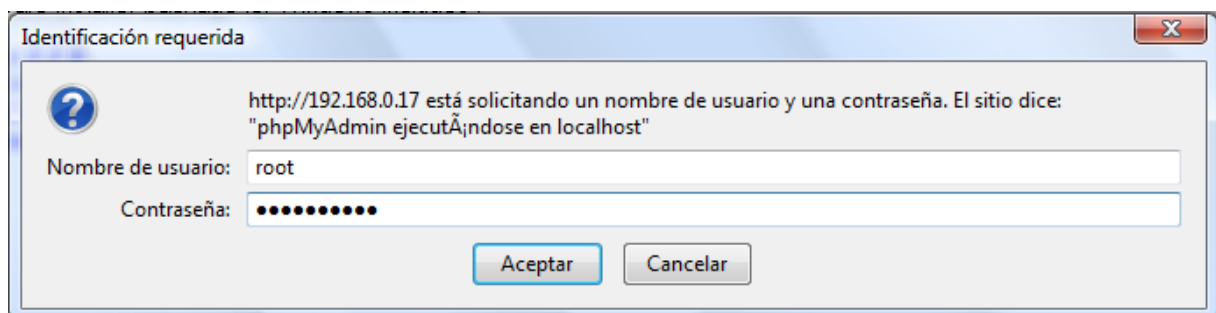


Figura A.11: Control de acceso a phpMyAdmin

Una vez identificado, aparecerá la pantalla principal de la interfaz phpMyAdmin, desde la que se puede administrar la base de datos.

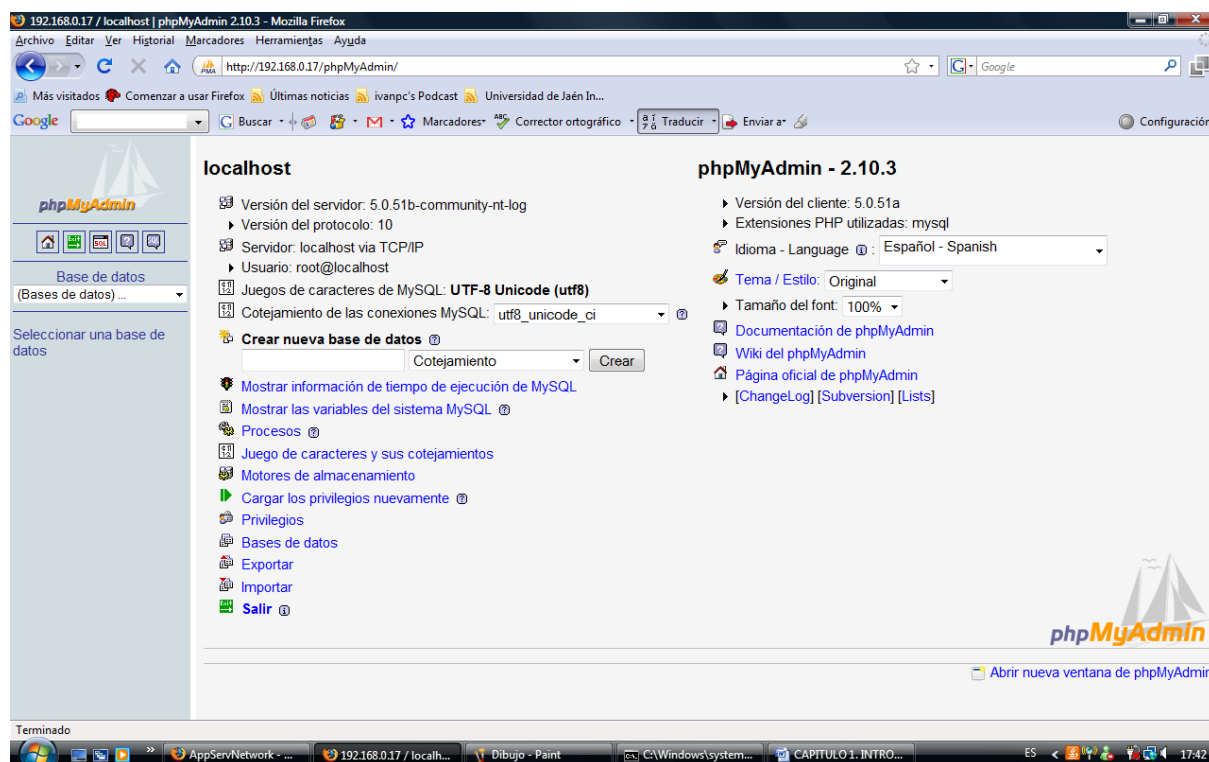


Figura A.12: Pantalla de inicio de phpMyAdmin en el servidor

A.2. Alojamiento del contenido de la aplicación en el servidor

Una vez que hemos terminado con la instalación del servidor, tendremos que alojar nuestra aplicación dentro del mismo.

En la configuración estándar de AppServ el directorio de instalación se encuentra en C:\AppServ, la ubicación que equivale a la dirección <http://localhost> o [http://\[DirecciónIP\]](http://[DirecciónIP]) es C:\AppServ\www (donde reemplazaremos [DirecciónIP] por la dirección de red de nuestro equipo). Por tanto, habremos de copiar la carpeta OL-RadioUJA dentro de C:\AppServ\www.

A.3. Importar la base de datos y activar Drupal

Para importar el archivo con la base de datos, 'pfc-radio.sql', debemos acceder a nuestra interfaz de phpMyAdmin y, en la pantalla principal, seleccionamos la opción *Crear nueva base de datos*. En la casilla de texto escribiremos el mismo nombre del archivo, 'pfc-radio', y pulsamos el botón Crear (ver Figura A.12). Seguidamente, seleccionamos la opción 'importar' situada en la parte inferior de la pantalla inicial.

Importar

Archivo a importar

Localización del archivo de texto: C:\Users\Iván\Desktop\pfc-radio.sql (Tamaño máximo: 81,920KB)

Juego de caracteres del archivo: utf8

La compresión escogida para el archivo a importar se detectará automáticamente de: Ninguna, gzip, zip

Importación parcial

☒ Permite la interrupción de la importación en el caso de que el script detecte que se ha acercado a su límite de tiempo. Esto podría ser un buen método para importar archivos grandes; sin embargo, puede dañar las transacciones.

Número de registros (consultas) a saltarse desde el inicio: 0

Formato del archivo importado

☒ SQL

Opciones SQL

Modalidad compatible con SQL: NONE

Continuar

[Abrir nueva ventana de phpMyAdmin](#)

Figura A.13: Importación de la base de datos.

Para cargar el archivo solamente tenemos que buscarlo mediante el botón Examinar y, acto seguido, pulsar el botón Continuar situado en la esquina inferior derecha (Figura A.13).

La instalación de Drupal queda realizada con sólo copiar la carpeta OL-RadioUJA en el directorio del servidor C:\AppSer\www. Esto es así porque hemos incluido en el material todos los archivos de Drupal de forma que el sitio Web quede intacto al migrarlo de un equipo a otro.

Para terminar, teclearemos la siguiente dirección en el navegador, sustituyendo [DirecciónIP] por la dirección IP de nuestro equipo.

[http://\[DirecciónIP\]/OL-RadioUJA](http://[DirecciónIP]/OL-RadioUJA)

Si todo ha ido bien, debería aparecernos la página principal de la aplicación.



Figura A.14: Página principal de la aplicación

A fin de que el usuario pueda disfrutar de inmediato de la aplicación como usuario registrado, la base de datos incluye por defecto un usuario administrador del sitio web, así como otro usuario de prueba. Las credenciales para cada uno de ellos son:

Identificador administrador: root. **Contraseña administrador:** administrador.

Identificador usuario: usuario. **Contraseña usuario:** usuario.

A.4. Instalar la máquina virtual Java

La instalación de la máquina virtual Java (JVM) permite ejecutar todo tipo de aplicaciones realizadas sobre esta tecnología. El proceso es muy sencillo: diríjase a la dirección Web <http://www.java.com/es/download/> para descargar la última versión de Java, en nuestro caso la versión 6, como muestra la Figura A.15.



Figura A.15: Descarga de Java 6.

Una vez descargado el instalador, ejecútelo para comenzar con la instalación. El proceso comenzará con la descarga de una serie de archivos necesarios (Figura A.16).



Figura A.16: Instalador de Java 6.

Tras aceptar el acuerdo de licencia, dará comienzo la instalación (Figura A.17).



Figura A.17: Instalación de Java 6.

Tras unos minutos, la instalación de Java habrá finalizado.

A.5. Instalar NetBeans y JDBC para la ejecución del Algoritmo de Filtrado

Para ejecutar el algoritmo de filtrado y actualizar el modelo de recomendaciones de la aplicación, es necesario previamente instalar un entorno de desarrollo de Java con capacidad para acceder a bases de datos MySQL. Por ello, en este epígrafe explicaremos cómo instalar el entorno de desarrollo NetBeans con JDK (Java Development Kit) y la librería de acceso de bases de datos JDBC.

En el disco que acompaña a esta memoria podemos encontrar el instalador de la versión 6.5.1 de NetBeans con JDK 5 incluido. También es posible descargar la última versión en la dirección <http://java.sun.com/j2se/1.5.0/download-netbeans.html> (ver Figura A.18).

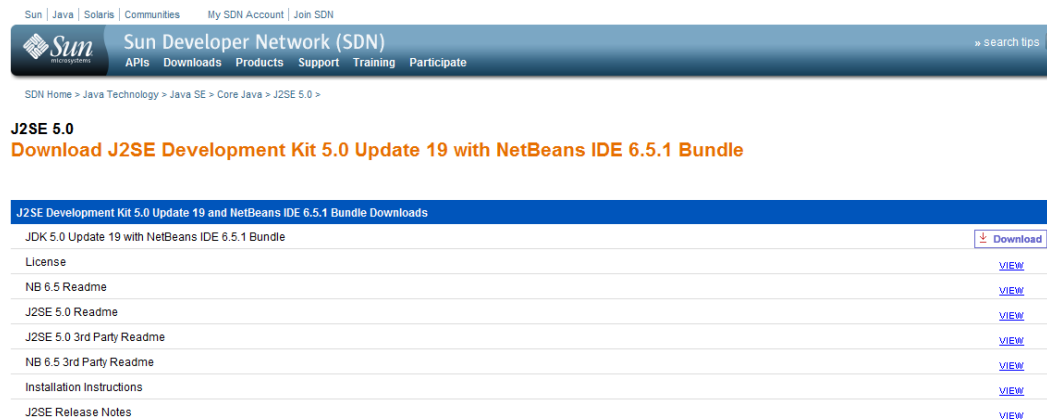


Figura A.18: Descarga de NetBeans con JDK

La instalación de la herramienta es muy sencilla y similar a la descrita en el anterior apartado para Java 6. Por defecto, el directorio de instalación de JDK es un subdirectorio de la ubicación en la que se instaló Java, en nuestro caso C:\Archivos de Programa\Java.

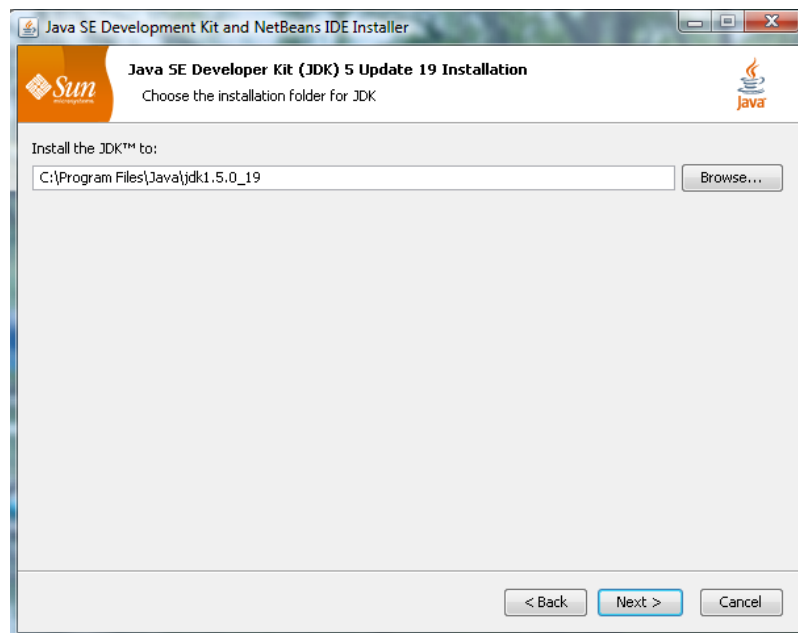


Figura A.19: Directorio de instalación de JDK

Una vez instalados NetBeans y JDK, debemos incluir la librería JDBC de forma que sea accesible por el entorno de desarrollo. Para ello, copiamos el archivo 'mysql-connector-java-5.1.7-bin' incluido en el disco que acompaña a la memoria, y lo pegamos en la siguiente ruta, existente dentro del directorio de instalación del JDK antes mencionado (Figura A.20):

C:\Archivos de Programa\jdkX.X.X_X\jre\lib\ext

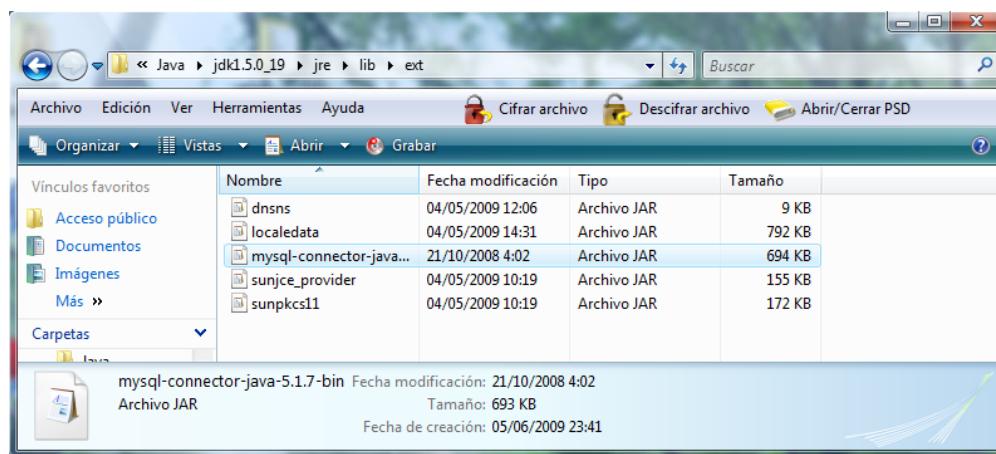


Figura A.20: Directorio de colocación de JDBC

Si se han realizado los pasos anteriores correctamente, nuestro equipo ya está listo para ejecutar el algoritmo de filtrado. Para ello, abrimos la aplicación NetBeans 6.5.1 y, en el menú 'File', elegimos 'Open Project' (véase Figura A.21). El proyecto Java que debemos abrir para ejecutar el algoritmo se encuentra en C:\AppServ\www\OL-RadioUJA\AlgoritmoFiltrado. Tras abrir el algoritmo, lo ejecutamos pulsando F6 o mediante el menú 'Run → Run Main Project'.

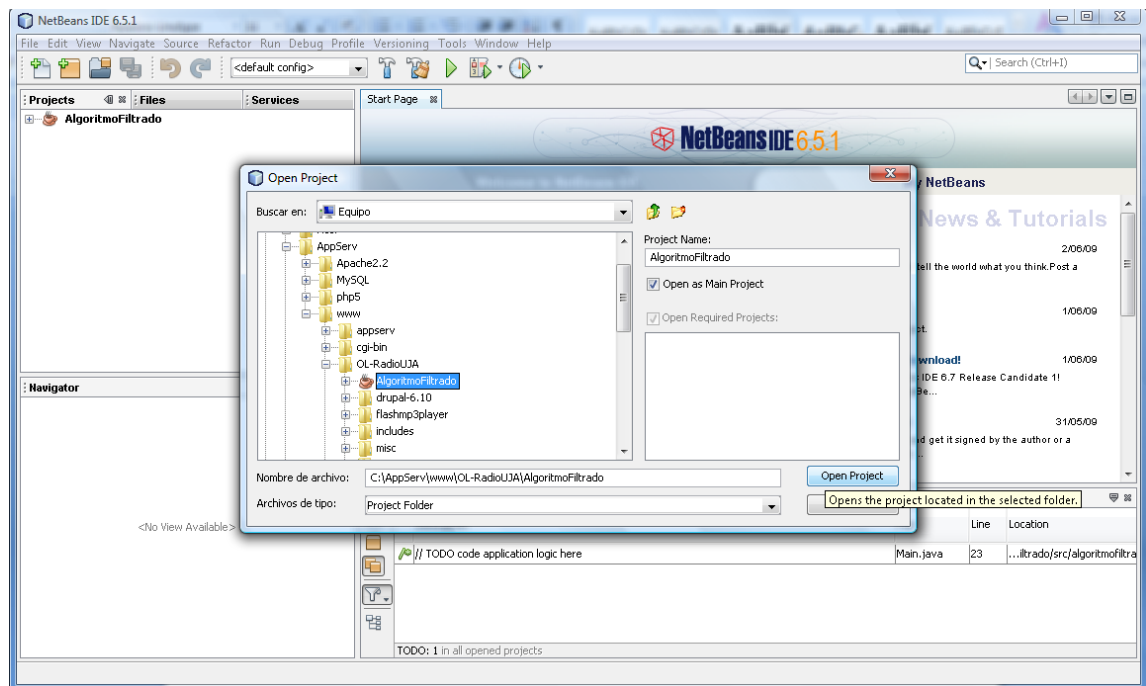


Figura A.21: Abrir proyectos en NetBeans.

Es importante destacar que un algoritmo de filtrado tiene un coste computacional considerable que aumenta conforme la base de datos de usuarios, canciones y puntuaciones se hace mayor, por ello, no es de extrañar que su ejecución pueda suponer varios minutos.

ANEXO B

Manual de Usuario

En este anexo mostramos, a modo de visita guiada, un manual de la aplicación de utilidad para el usuario. Se pretende, así, que el usuario pueda aprender la utilización sacar el máximo partido de la aplicación de la forma más sencilla posible.

No obstante, es conveniente en primer lugar dejar claros algunos aspectos importantes para el usuario:

- La aplicación de radio online OL-RadioUJA es una aplicación Web recomendada especialmente para las últimas versiones existentes de Mozilla Firefox (que se puede descargar en <http://www.mozilla-europe.org/es/>) y de Internet Explorer. Se recomienda una resolución de 1024x768 píxeles o superior, para evitar posibles fallos de visualización. A pesar de estos posibles fallos en la presentación del sitio Web, la funcionalidad no se verá en absoluto afectada.
- Es importante tener instalado el plugin de reproducción de contenido Flash en el navegador, para poder visualizar la radio. En caso de no disponer de él, la mayoría de versiones tanto de IE como de Firefox muestran una barra emergente en la parte superior de la ventana, bajo las barras de tareas, para brindar al usuario la posibilidad de instalarlo.
- Por último, téngase en cuenta que, pese a que el sitio está hecho para mostrar los paneles de puntuaciones de la radio como ventanas emergentes, se han encontrado versiones de Firefox en las que en lugar de ventanas emergentes aparecen los paneles de puntuación como nuevas pestañas de navegación.

Para empezar, el usuario debe acceder a través de su navegador a la siguiente dirección: <http://trantor.ujaen.es/~pfc-ivan>⁶. Al hacerlo, aparecerá la página de inicio de la aplicación, como podemos observar en la Figura B.1.

⁶ Dirección Web en la que la aplicación ha quedado alojada en el momento de la realización de esta memoria. En caso de haber instalado la aplicación según el Anexo A, la dirección Web será la especificada en dicho anexo.



Figura B.1: Página de inicio de OL-RadioUJA

El usuario deberá dirigirse a la sección ‘Inicio de sesión’, situada en la parte derecha de la aplicación, introducir su identificador de usuario y contraseña, y pulsar el botón ‘Iniciar sesión’ (Figura B.2). Llegados a este punto, pueden ocurrir dos cosas. Si el identificador y contraseña son correctos, aparecerá la pantalla principal con privilegios de usuario registrado, que corresponde a la radio personalizada del usuario (Figura B.3).

Figura B.2: Formulario de inicio de sesión



Figura B.3: Pantalla principal para usuario autenticado.

Si, por el contrario, el identificador de usuario o contraseña introducidos son incorrectos, aparecerá un mensaje de error en la pantalla inicial, tal y como muestra la Figura B.4.



Figura B.4: Error de validación en OL-RadioUJA

Si el usuario no dispone de un identificador para validarse en la aplicación, es posible registrarse en tiempo de ejecución, mediante la opción 'Regístrate' situada en la parte inferior del formulario de inicio de sesión.

Al hacer clic sobre dicha opción, aparecerá la siguiente pantalla:

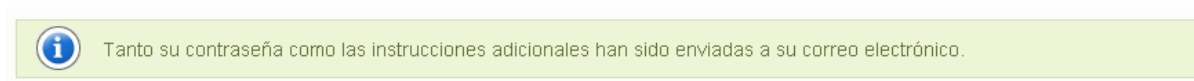


The screenshot shows a web form titled "Cuenta de usuario". At the top, there are three buttons: "Regístrate" (highlighted), "Iniciar sesión", and "Solicitar una nueva contraseña". Below this is a section titled "Información de la cuenta" containing two input fields. The first is labeled "Usuario: *" and contains the text "paco". Below it is a note: "Se permite la utilización de espacios; los signos de puntuación no están permitidos a excepción de puntos, guiones altos y guiones bajos." The second field is labeled "Dirección de correo electrónico: *" and contains "paco23@hotmail.es". Below it is a note: "Una dirección de correo electrónico válida. Todos los correos del sistema se enviarán a esta dirección. La dirección de correo no es pública y solamente será usada para recibir una contraseña nueva o para el envío de ciertas noticias y notificaciones." Below this section is another titled "Información perfil" with a dropdown menu labeled "Género musical favorito: *" showing "Instrumental". Below the dropdown is a note: "Por favor, selecciona el género musical que más te guste." At the bottom left of the form is a "Regístrate" button.

Figura B.5: Registro de usuario.

Para registrarse sólo es necesario introducir un nombre de usuario, una dirección de correo electrónico válida, y seleccionar el género musical favorito del usuario de entre un conjunto de géneros disponibles.

Si los datos han sido introducidos correctamente, aparecerá la pantalla de inicio con el siguiente mensaje en la parte superior.



Lo único que resta es que el nuevo usuario acceda a su bandeja de correo electrónico para abrir el mensaje que habrá recibido de la aplicación y consultar su contraseña, que será la que utilizará para validarse y entrar como usuario autenticado.

Una vez validados en la aplicación, es hora de utilizarla, para ello nos encontramos con un menú principal (como muestra la Figura B.6) en la parte derecha de la pantalla, con varias opciones, que pasamos a explicar a continuación:



Figura B.6: Menú principal de OL-RadioUJA

Radio

Como se ha comentado, la radio se encuentra en la pantalla principal de la aplicación. Además, es accesible desde la opción de menú 'Lanzar radio', que también sirve para actualizar la lista de reproducción en caso de que el usuario ya se encuentre en esta pantalla.



Figura B.7: Detalles de la pantalla de radio.

Esta sección permite al usuario escuchar una lista de canciones que le han sido recomendadas, por ser similares a otras canciones que este ha valorado positivamente con anterioridad.

Escuchar la radio es tan sencillo como utilizar los botones existentes en la lista de reproducción para iniciar, detener o pausar la reproducción, pasar a la canción anterior o siguiente, y ajustar el nivel de volumen, como podemos ver en la siguiente Figura.

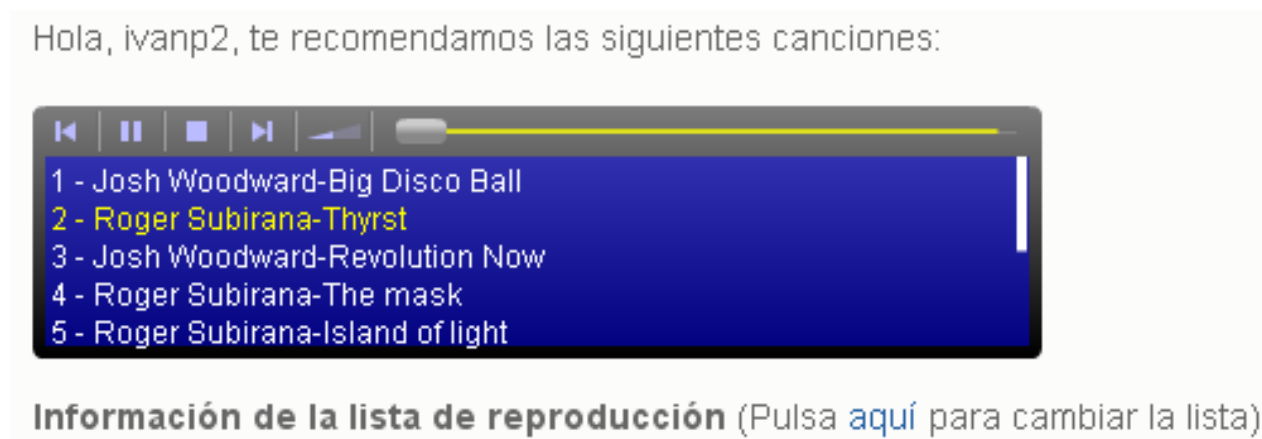


Figura B.8: Lista de reproducción (radio).

El usuario también puede pulsar el enlace situado bajo la radio para cambiar la lista y obtener un nuevo conjunto de canciones recomendadas.

Debajo de la radio aparecen los detalles de la lista de reproducción, donde se puede ver, entre otros datos, el nombre de la canción, género o carátula del álbum al que pertenece.



Figura B.9: Detalles de canción en la radio del usuario.

Si el usuario lo desea, puede evaluar la canción, que consiste en asignarle una puntuación de 1 a 5, según la siguiente escala de valores:

- 1: Muy Mala.
- 2: Mala
- 3: Normal
- 4: Buena
- 5: Muy Buena.

Al pulsar sobre 'Evaluar canción', aparecerá una ventana emergente⁷ en la que el usuario puede asignar la puntuación deseada a la canción (Figura B.10). Basta con elegir uno de los 5 valores y pulsar sobre el botón 'Puntuar'.

⁷ O pestaña, en algunas versiones de Mozilla Firefox

ivanp2, evalúa esta canción.

Escoge un valor de 1 a 5



ivanp2, evalúa esta canción.

Tu puntuación ha sido registrada. Gracias.(Puntuación: 4)



Figura B.10: Pantalla de puntuación de canciones.

Todo este proceso de puntuación se puede hacer cómodamente mientras el usuario sigue escuchando su radio, sin que la reproducción de esta se vea interrumpida. Una vez puntuada la canción, se puede cerrar la ventana de puntuaciones y continuar disfrutando de la aplicación.

El usuario anónimo también puede disfrutar de la radio de la aplicación, con la peculiaridad de que no podrá evaluar canciones, y de que la lista de reproducción lanzada estará compuesta por canciones escogidas de forma totalmente aleatoria.

Álbumes

Esta opción del menú permite acceder a la colección de álbumes disponible en la aplicación. Por defecto aparece una vista global de todos los álbumes en orden alfabético de artista (Figura B.11).

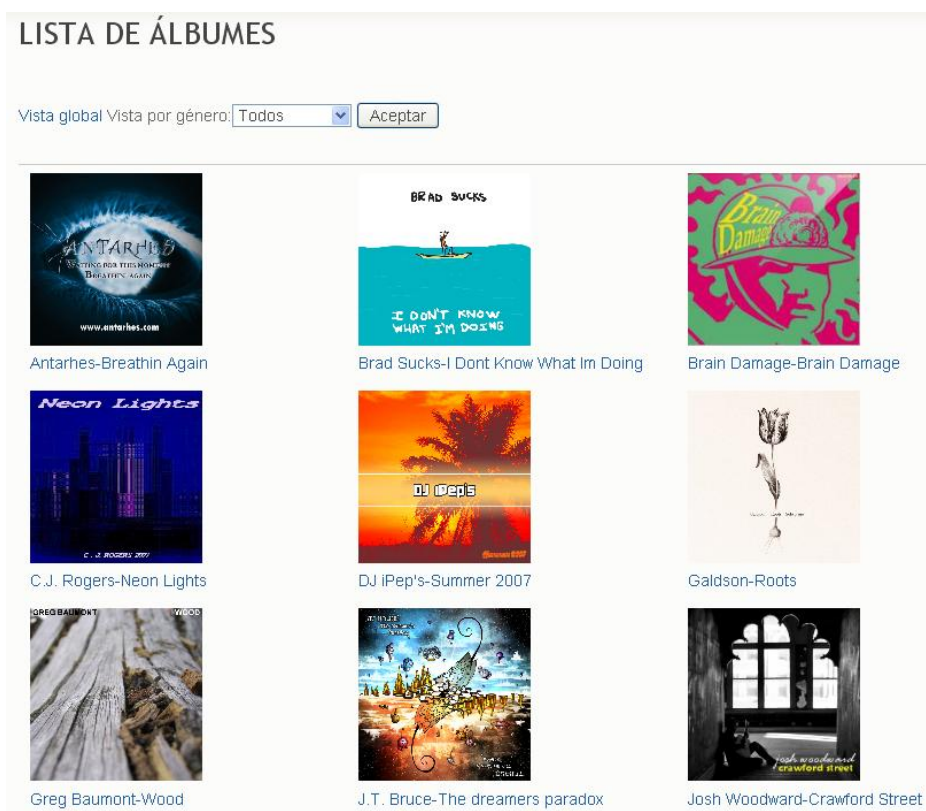


Figura B.11: Vista principal de la sección ‘Álbumes’

También es posible seleccionar un género musical para mostrar únicamente aquellos álbumes que pertenecen a dicho género (ver Figura B.12), para ello, nos vamos a ‘Vista por género’, en la parte superior de la lista de álbumes, seleccionamos un género y pulsamos ‘Aceptar’.

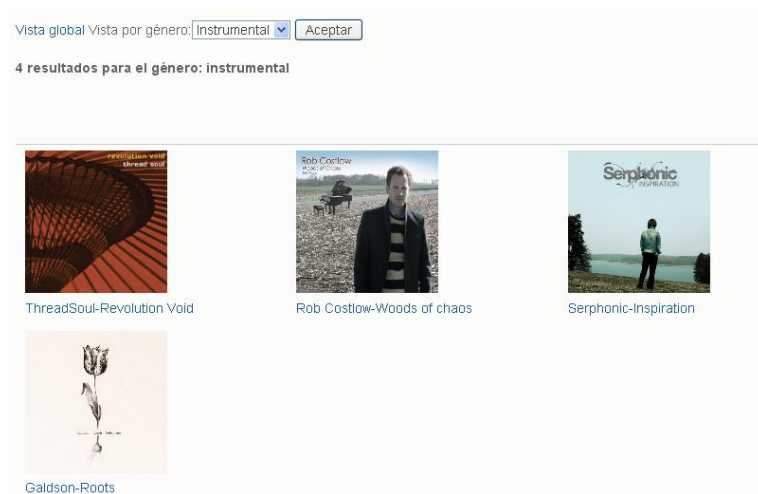


Figura B.12: Vista de los álbumes por género

Para ver la información de un álbum, tan solo hay que hacer clic sobre su nombre o carátula. Nos aparecerá así la pantalla de detalles del álbum (Figura B.13), con datos del mismo, y una lista de canciones reproducibles para cada una de las canciones del álbum. Tan solo hay que pulsar el botón PLAY sobre la canción deseada para comenzar a escucharla.

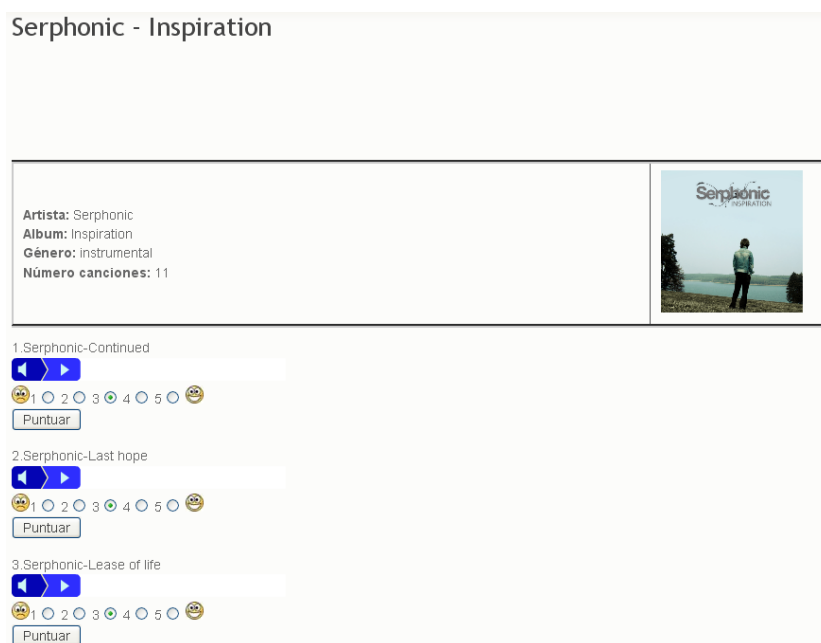


Figura B.13: Página de información de álbum

Además, también se ofrece desde aquí la posibilidad de evaluar las canciones que forman parte del álbum.

El usuario anónimo también acceder al álbum que desee, con la limitación de no poder evaluar canciones.

Mi cuenta

Desde aquí, el usuario puede modificar sus datos personales: dirección de correo electrónico y contraseña, además de opciones de idioma y franja horaria, si lo desea. Solamente hay que hacer clic sobre la opción 'Editar', modificar el campo deseado y pulsar el botón 'Guardar'. En caso de introducir un dato erróneo, el sistema nos mostrará un mensaje de forma similar a como se explicó en el registro de usuarios.

Perfil de usuario

Esta sección es como una ficha personal de las preferencias musicales del usuario, donde aparece no sólo su género favorito, sino también el listado de la canciones que han sido evaluadas por él, ordenadas de mayor a menos puntuación.

Perfil de Usuario

Género favorito: acustica

Cambiar Género: --ELEGIR-- Aceptar

Mis puntuaciones

Cancelar

Rock
Acústica
Techno
Instrumental
Pop
House
Psicodélico
Ambiental
BSO

	Artista	Género	Puntuación	Modificar puntuación
Heritage	Josh Woodward	acustica	5	😊 1 2 3 4 5 😞 Modificar
Cant't take our love away	Josh Woodward	acustica	5	😊 1 2 3 4 5 😞 Modificar
The spirit world	Josh Woodward	acustica	5	😊 1 2 3 4 5 😞 Modificar
Twelve monkeys	Josh Woodward	acustica	5	😊 1 2 3 4 5 😞 Modificar

ivanp2

- Lanzar radio
- Álbumes
- Mi cuenta
- Perfil de usuario**
- Libro de visitas
- Terminar sesión

Bienvenida

¡Bienvenido, ivanp2!

Quién está en línea

Actualmente hay 1 usuario y 0 invitados en línea.

Usuarios en línea

- ivanp2

Figura B.14: Perfil del usuario en OL-RadioUJA

El usuario puede, en cualquier momento, acceder a este perfil para modificar su género musical favorito, seleccionando un género en la lista desplegable y pulsando 'Aceptar'. También puede modificar una puntuación previamente dada a una canción, cambiando el valor numérico en la fila perteneciente a dicha canción y pulsando el correspondiente botón 'Modificar'.

Libro de visitas

En esta última sección el usuario puede leer comentarios o firmas que otros usuarios han escrito, así como escribir el suyo propio, haciendo clic sobre 'Añadir nuevo comentario'.

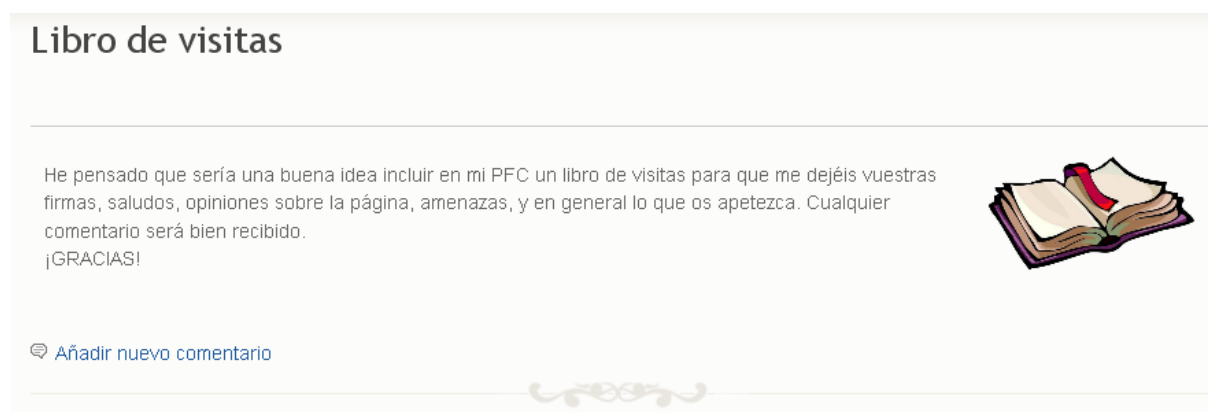


Figura B.15: Libro de visitas

Terminar sesión

Para terminar, mediante esta opción del menú se nos permite cerrar nuestra sesión en la aplicación cada vez que queramos terminar de utilizarla.

ANEXO C

Código fuente del Algoritmo de filtrado colaborativo

En este tercer anexo incluimos la implementación en lenguaje Java del algoritmo de filtrado colaborativo que se ha desarrollado para este proyecto. Se trata de un algoritmo basado en ítem, del tipo ‘vecino más cercano’ (K-nn), que emplea el coeficiente coseno como medida de similitud entre canciones, y el algoritmo weighted-sum para calcular las predicciones de los usuarios sobre las canciones.

Hemos considerado útil dedicar un anexo a este código fuente, ya que consideramos que de esta forma el lector interesado podrá encontrar una cómoda referencia a la hora de implementar un algoritmo de filtrado colaborativo o, simplemente, para aquel lector que desee profundizar en el funcionamiento de dicho algoritmo.

Contenido del fichero Main.java:

```
package algoritmofiltrado;

import java.sql.*;
import java.util.Vector;
import java.math.*;

/**
 * La clase Main es la clase principal o clase de Aplicación del algoritmo. Incluye el
 * establecimiento de la conexión con la base de datos y las llamadas a los distintos
 * métodos para llevar a cabo el algoritmo de filtrado.
 * @author Iván
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here

        //Preparación de las clases de la API JDBC para conectar con la Base de Datos
        try {
            Class.forName("org.gjt.mm.mysql.Driver"); //IP DE TRANTOR:150.214.178.128 Puerto:3306
            Connection conexion = DriverManager.getConnection("jdbc:mysql://127.0.0.1/pfc-radio", "root", "pfc-
ivan008");

            System.out.println("Conexion realizada");

            //Ejecución de una sentencia SQL: consulta de todas las canciones de la BD.
            //Volcado de los identificadores de las canciones en un vector.
            Statement s = conexion.createStatement();
            Vector<Integer> id_canciones = new Vector<Integer>();
            ResultSet rs = s.executeQuery("select id_cancion from canciones");
            while (rs.next()) {
                id_canciones.add(rs.getInt(1));
            }
        }
    }
}
```

```
}  
  
//Para cada par de canciones, se calcula la similitud entre ellas.  
//Guardamos, para cada canción, las similitudes con el resto de canciones.  
//De esta forma podremos, mas adelante, seleccionar los k vecinos más cercanos a cada objeto.  
  
Vector<Vector<Double>> similitudes = new Vector<Vector<Double>>();  
for (int i = 0; i < id_canciones.size(); i++) {  
    Vector<Double> sims = new Vector<Double>();  
    for (int j = 0; j < id_canciones.size(); j++) {  
        //Llamada al método coeficiente_coseno de la clase Similitud  
        double sim = Similitud.coeficiente_coseno(id_canciones.get(i), id_canciones.get(j), conexion);  
        System.out.println("sim(" + id_canciones.get(i) + "," + id_canciones.get(j) + ") " +  
            "=" + sim);  
        sims.add(sim);  
    }  
    similitudes.add(sims);  
}  
  
//Realizar las predicciones de cada usuario, en base a los k vecinos más cercanos a cada canción  
//Inserción a actualización de la predicción usuario-cancion en la BBDD  
Statement s2 = null;  
ResultSet rs2 = null;  
s2 = conexion.createStatement();  
rs2 = s.executeQuery("select uid from users");  
while (rs2.next()) { //PARA CADA USUARIO  
    int uid = rs2.getInt(1);  
    Statement su = null;  
    ResultSet rsu = null;  
    su = conexion.createStatement();  
    rsu = su.executeQuery("select value from profile_values where fid='2'and uid='" + uid + "'");  
    String genero = null;  
    if (rsu.first()) {  
        genero = rsu.getString(1);  
    } else {  
        genero = "rock";  
    }  
}
```

```

    }
    for (int x = 0; x < id_canciones.size(); x++) { //PARA CADA CANCION
        int idc = id_canciones.get(x);
        Vector<Double> aux = similitudes.get(x);
        //Llamada al método Weighted_sum de la clase Predicción
        double pred = Prediccion.weighted_sum(uid, idc, conexion, aux, id_canciones, 10);
        Statement su2 = null;
        ResultSet rsu2 = null;
        su2 = conexion.createStatement();
        rsu2 = su2.executeQuery("select genero from canciones where id_cancion=" + idc + "");
        if (rsu2.first()) {
            String genero2 = rsu2.getString(1);
            if (genero.equals(genero2)) {
                pred += 0.25;
            } //Plus de 0.25 por ser del genero favorito del usuario
        }
        su2.close();
        rsu2.close();
        System.out.println("PREDICCION DE USUARIO " + uid + " SOBRE CANCION " + idc + ": " + pred);
        Statement ss = null;
        ResultSet rss = null;
        ss = conexion.createStatement();
        rss = ss.executeQuery("select prediccion from predicciones where id_usuario=" + uid +
            "" and id_cancion=" + idc + "");
        if (rss.first()) {
            //Ya existia Pred(u,i) en la BBDD: ACTUALIZACIÓN
            Statement stmt = conexion.createStatement();
            stmt.executeUpdate("UPDATE predicciones SET prediccion=" + pred + " WHERE id_usuario=" +
uid +
            "" and id_cancion=" + idc + "");
            stmt.close();
        } else {
            //Pred(u,i) no existia: INSERCIÓN

```

```
        PreparedStatement stmt = conexion.prepareStatement("INSERT INTO predicciones VALUES  
(?,?,?)");  
  
        stmt.setInt(1, uid);  
        stmt.setInt(2, idc);  
        stmt.setDouble(3, pred);  
        stmt.executeUpdate();  
        stmt.close();  
        ss.close();  
        rss.close();  
        su.close();  
        rsu.close();  
    }  
    s2.close();  
    rs2.close();  
    conexion.close();  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}  
}
```

Contenido del fichero Similitud.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package algoritmofiltrado;

import java.sql.*;

/**
 * Esta clase contiene la operación que calcula la similitud entre dos canciones, por
 * medio del coeficiente coseno
 * @author ivanp
 */
public class Similitud {

    /**
     * Cálculo de la similitud entre 2 canciones con el coeficiente coseno
     * @param cancionx Identificador de la primera de las 2 canciones
     * @param canciony Identificador de la segunda de las 2 canciones
     * @param conexion La variable de conexión MySQL
     * @return El valor de la similitud, un número real entre 0 y 1.
     */
    public static double coeficiente_coseno(int cancionx, int canciony, Connection conexion) {
        try {
            Statement s = null;
            ResultSet rs = null;
            s = conexion.createStatement();
            rs = s.executeQuery("select uid from users");
            int numerador = 0;
            int denominX = 0;
            int denominY = 0;
            //PARA CADA USUARIO
            while (rs.next()) {
```



```
int Ui = rs.getInt(1);
Statement s2 = null;
Statement s3 = null;
ResultSet rs2 = null;
ResultSet rs3 = null;
s2 = conexion.createStatement();
//Ver si el usuario "Ui2 ha evaluado la cancion x
rs2 = s2.executeQuery("select puntuacion from puntuaciones where id_usuario='"+Ui+"' and " +
    "id_cancion='"+cancionx+"'");
s3 = conexion.createStatement();
//Ver si el usuario "Ui2 ha evaluado la cancion y
rs3 = s3.executeQuery("select puntuacion from puntuaciones where id_usuario='"+Ui+"' and " +
    "id_cancion='"+canciony+"'");
//Si el usuario actual "Ui" ha evaluado ambos items, x e y
if(rs2.first() && rs3.first()){
    //Se tiene en cuenta en el calculo del coeficiente coseno
    int Xi = rs2.getInt(1);
    int Yi = rs3.getInt(1);
    numerador += Xi*Yi;
    denominX += (Xi*Xi);
    denominY += (Yi*Yi);
}
s2.close();
s3.close();
rs2.close();
rs3.close();

}
s.close();
rs.close();
double denominador = Math.sqrt(denominX*denominY);
if (denominador==0) return 0;
return numerador/denominador;
```

```
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return 0;  
}  
}
```

Contenido del fichero Predicción.java

```
/**
 * Esta clase incluye el método que realiza la prediccion para un usuario de una canción,
 * mediante la medida llamada weighted_sum
 * @author ivanp
 */
public class Prediccion {
    /**
     * Calcula la predicción mediante la medida de weighted_sum
     * @param Ua El usuario sobre el cual se realiza la predicción
     * @param Ia El ítem sobre el que se realiza la predicción
     * @param conexion La variable de conexión a MySQL
     * @param simIa Conjunto de similitudes del ítem Ia con el resto
     * @param id_canciones Identificadores de las canciones similares
     * @param k El número de vecinos más cercanos a la canción Ia que se deben tener en cuenta
     * @return El valor de la prediccion, un real que se mueve inicialmente entre 0 y 5
     */
    public static double weighted_sum(int Ua, int Ia, Connection conexion, Vector<Double> simIa,
        Vector<Integer> id_canciones, int k) {
        Vector<Double> sim_Ia = new Vector<Double>(simIa);
        Vector<Double> mayor_sims = new Vector<Double>();
        Vector<Integer> mayor_items = new Vector<Integer>();
        double denominador = 0;
        double numerador = 0;
        //PASO 1: OBTENER LOS IDs DE LOS K ITEMS MAS SIMILARES A Ia, Y LAS PROPIAS SIMILITUDES
        for (int i = 0; i < k; i++) { //se extraen 50 similitudes
            double mayorsim = 0;
            int mayorpos = 0;
            for (int j = 0; j < sim_Ia.size(); j++) { //para extraer cada una se recorre el vector en busca del mayor
                if (sim_Ia.get(j) >= mayorsim) {
                    mayorsim = sim_Ia.get(j);
                    mayorpos = j;
                }
            }
        }
    }
}
```

```

    }
    sim_Ia.set(mayorpos, -1.0); //se limpia la posicion del mayor valor encontrado esta vez
    //System.out.println("Cancion " + id_canciones.get(mayorpos) + " : " + mayorsim);
    mayor_sims.add(mayorsim);
    mayor_items.add(id_canciones.get(mayorpos)); //en mayor_items ya se guardan los IDs de los items
}
//FIN PASO 1
//PASO 2: CALCULAR EL NUMERADOR Y EL DENOMINADOR
try {
    for (int i = 0; i < k; i++) {
        Statement s = null;
        ResultSet rs = null;
        s = conexion.createStatement();
        rs = s.executeQuery("select puntuacion from puntuaciones where id_usuario=" + Ua + " and " +
            "id_cancion=" + mayor_items.get(i) + "");
        //Se obtiene asi la puntuacion R(Ua,Ih)
        if(rs.first()){ //Si el item Ih ha sido puntuado por el usuario, se tiene en cuenta
            //System.out.println("cancion " + mayor_items.get(i) + " puntuada por el usuario: " + rs.getInt(1));
            numerador += (mayor_sims.get(i)*rs.getInt(1)); //SUM[sim(Ia,Ih)*RU(Ua,Ih)]
            denominador += mayor_sims.get(i); //SUM[|sim(Ia,Ih)|]
        }
        s.close();
        rs.close();
    }
    //System.out.println("NUMERADOR: " + numerador);
    //System.out.println("DENOMINADOR: " + denominador);
    if(denominador==0) return 0;
    else return (numerador/denominador);
} catch (Exception e) {
    e.printStackTrace();
}
return 0;
}
}

```


Bibliografía y Referencias

[1]Balabanovic, M. y Shoham, Y. (1997), "*Content-based, collaborative recommendation*".

Communications of the ACM, Vol. 40

[2]Cho, Y.H., J. Kyeong Kim, "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*

[3]Herlocker, J., Konstan, J., Terveen, L., Riedl, J. (2004), "*Evaluating Collaborative Filtering Recommender Systems*". ACM Vol. Transactions of Information Systems, vol. 22, pp. 5-53.

[4]Breese, J., Heckerman, D., Kadie, C., "*Empirical Analysis of Predictive Algorithms for Collaborative Filtering*", Technical Report MSR-TR-98-12, May 2008.

[5]Vozalis, E., Margaritis, K., "*Analysis of Recommender Systems Algorithms*".

[6] Jung, K.Y., Choi, J.H., Rim, K.W., 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.

[7] Yu, K., Xu, X., Tao, J., Ester, M., Kriegel H.P., "*Instance Selection Techniques for Memory-Based Collaborative Filtering*", 12th international workshop on database and expert systems applications, proceedings, 2001.

- [8] Papagelis, M., Plexousakis, D., Rousidis, I., Theoharopoulos, E., *“Qualitative Analysis of User-based and Item-based Algorithms for Recommendation Systems”*, Institute of Computer Science, Foundation for Research and Technology – Hellas.
- [9] Sarwar, B., Karypis, G., Konstan, J., Riedl, J., *“Item- based Collaborative Filtering Recommendation Algorithms”*, WWW10, May 1-5, 2001.
- [10] Shardanand, U., *“Social Information Filtering for Music Recommendation”*, Massachusetts Institute of Technology, Sept. 1994.
- [11] Dawson, C. W. y Martin, G. (2002), *“El proyecto fin de carrera en Ingeniería Informática: Una guía para el estudiante”* Ed. Prentice Hall
- [12] Gutierrez, A., Bravo, G. *“PHP 5 a través de ejemplos”*, Ed. RA-MA
- [13] Krug, S. *“No me hagas pensar: una aproximación a la Usabilidad en la Web”*. Ed. Prentice Hall
- [14] Freeman, E., Sierra, K., Bert Bates, *“Head First Design Patterns”*. Ed. O'Reilly
- [15] Brett D. McLaughli, Pollice, G., West, D. *“Head First Object-Oriented Analysis and Design”*. Ed. O'Reilly
- [16] Reynolds, P.A., Mason, R., Eaton, K.A., *“Webcasting: casting the Web more widely”*, British Dental Journal, Vo. 204, Issue 3, pp 145-149, 2008.
- [17] López, H., *“Multimedios en el Web: Streaming media”*, Universidad de Las Américas Puebla, (http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/lopez_a_h/capitulo3.pdf)
- [18] Creative Commons España, (<http://es.creativecommons.org/>)

[19] Conde, F., *“Interacción Persona-Ordenador: ¿Pero qué narices es una buena interfaz de usuario”*. Apuntes de la asignatura.

[20] Castellano, E.J., *“Evaluación del Uso de Algoritmo colaborativos para orientar académicamente al alumnado de Bachillerato”*, Memoria investigadora del D.E.A., Universidad de Jaén, Depto. De Informática, Sept.2007.

[21] Siles Fernández, F. (2007) *“Sistema de Recomendación colaborativo de alquiler de películas”*. Proyecto Fin de Carrera. Escuela Politécnica Superior (Universidad de Jaén).

[22] Martínez Mimbrera, F. (2008) *“Sistema de Recomendación actualizable y con Gestión de Usuarios”*. Proyecto Fin de Carrera. Escuela Politécnica Superior (Universidad de Jaén).

[23] Araque Ibáñez, A. (2008) *“Sistema de Recomendación de restaurantes Georreferenciados”*. Proyecto Fin de Carrera. Escuela Politécnica Superior (Universidad de Jaén).

[24] Feito Higuera, F., Ruiz de Miras, J., Molina Aguilar, A. (1996), *“Análisis y Gestión de Datos”*, Editorial Universidad de Jaén.

