



ESCUELA TECNICA SUPERIOR DE INGENIERIA INFORMATICA
Ingeniería Informática Superior

Recomendador de gustos en redes sociales

Realizado por
Isabel García Orcajo

Dirigido por
José Ramón Portillo Fernández

Departamento
Matemática Aplicada I

Sevilla, 06/2011

Resumen

Esta documentación corresponde a un proyecto final de carrera basado en la creación de una aplicación para la red social FACEBOOK. Inicialmente consistía en el desarrollo de un **recomendador** de música acorde con las preferencias definidas en nuestro perfil de usuario de dicha red. Como se verá más adelante, el proyecto ha sido ampliado y no sólo nos limitaremos en él a plantear sugerencias sobre música, sino que haremos extensivas las recomendaciones a libros, películas, juegos o series y programas de televisión.

A lo largo de esta memoria se describe el procedimiento seguido para el desarrollo de la aplicación, partiendo de cero, para que pueda ser útil a otros alumnos que en un futuro acometan la ejecución de un proyecto similar.

Índice general.

Definición de objetivos.	1
Introducción.	3
2.1.- ¿Qué es Facebook?.....	3
2.2.- ¿Qué la hace tan popular?	4
Aplicaciones en Facebook.	7
3.1.- ¿Qué es una aplicación de Facebook?	7
3.2.- ¿Cómo funcionan?.....	7
Cómo crear una aplicación en Facebook.	9
4.1.- Registro de la aplicación paso a paso.	9
Nuestra aplicación.	15
5.1. – En qué consiste.	15
5.2. – Procedimientos para su diseño.....	15
5.2.1.- Primera idea.....	16
5.2.2.- Segunda idea.....	16
5.2.3.- Tercera idea	17
Criterios de recomendación.	21
6.1.- Criterio de selección con pesos.	21
6.2.- Criterio de selección.	22

6.3.- Criterio “invariable”	23
Accediendo a Facebook Developers.....	25
Documentándonos.....	27
8.1.- Apps on Facebook (aplicaciones en Facebook).	27
8.1.1.- Authorization (autorización).....	28
8.1.2.- Social channels (medios sociales).....	28
8.2.- Core concepts (conceptos básicos).....	29
8.2.1.- Social plugins (accesorios sociales).....	30
8.2.2.- Graph API (interfaz).	30
8.3.- Advanced topics (temas avanzados).....	30
8.4.- SDK & tools (SDK y herramientas).	31
8.4.1.- PHP SDK.	32
Estructurando nuestra aplicación.	33
9.1.- Servidor web.	33
9.2.- Base de datos.	35
9.3.- Código PHP de la aplicación.	39
Empezando a programar.	41
10.1.- Creación y gestión de sesión en Facebook.	42
10.2.- Conexión con la base de datos.	45
10.3.- Procesamiento del usuario.....	46
10.4.- Página principal de la aplicación.	50
10.5.- Recomendación de música.....	51
10.6.- Formulario.....	51
10.7.- Envío de la información.	55
10.8.- Procesando la información.	55
10.9.- Difundiendo la aplicación.....	56
Implementación de criterios.....	59
11.1.- Primer criterio – Criterio de selección con peso.	59

11.2.- Segundo criterio – Criterio de selección.....	61
11.3.- Tercer criterio – Criterio “invariable”.	63
Ampliación del proyecto.	65
12.1.- Recomendador de películas.	66
12.2.- Recomendador de televisión.....	67
12.3.- Recomendador de libros.....	68
12.4.- Recomendador de juegos.	68
Estética de la aplicación.	71
Análisis de antecedentes y aportación realizada.	77
Comparación con otras alternativas.....	79
15.1.- Primera idea	80
15.2.- Segunda idea	81
15.3.- Tercera idea	81
15.4.- Criterios de recomendación.	83
Análisis temporal y de costes de desarrollo.	85
16.1.- Aumento del tiempo de desarrollo.....	86
16.2.- Tiempo de desarrollo.....	87
Análisis de requisitos.	89
17.1.- Requisitos fundamentales.	89
Manual de usuario.....	91
Pruebas.....	97
19.1.- Registro de datos de usuario en nuestra BD.	97
19.2.- Modificando datos de usuario en nuestra BD.	98
19.3.- Pruebas con el criterio de selección con peso.	100
19.4.- Pruebas con el criterio de selección.	102
19.5.- Pruebas con el criterio “invariable”	103
19.6.- Comparación de criterios.	105
19.7.- Pruebas con errores.....	108

Conclusiones y desarrollos futuros.	109
20.1.- Conclusiones.	109
20.2.- Desarrollos futuros.	111
Bibliografía.....	113
Glosario.....	117

Índice de figuras.

Figura 3.1	8
Figura 4.1	9
Figura 4.2	10
Figura 5.1	18
Figura 9.1	35
Figura 9.2	36
Figura 9.3	36
Figura 9.4	37
Figura 9.5	38
Figura 10.1.....	43
Figura 10.2.....	44
Figura 10.3.....	45
Figura 10.4. Contenido del archivo mysql.php.	46
Figura 10.5.....	47
Figura 10.6.....	48
Figura 10.7. Obtención de los artistas almacenados en el perfil del usuario.	48
Figura 10.8.....	49
Figura 10.9.....	49
Figura 10.10. Formación tabla definida en Facebook.	50
Figura 10.11.....	50
Figura 10.12.....	52
Figura 10.13.....	53
Figura 10.14.....	53

Figura 10.15	54
Figura 10.16	54
Figura 10.17	55
Figura 10.18	55
Figura 10.19	56
Figura 10.20	57
Figura 10.21	57
Figura 10.22	58
Figura 11.1	60
Figura 11.2	60
Figura 11.3	60
Figura 11.4	61
Figura 11.5	62
Figura 11.6	63
Figura 11.7	64
Figura 12.1	66
Figura 12.2	66
Figura 12.3	67
Figura 12.4	67
Figura 12.5	68
Figura 12.6	68
Figura 12.7	68
Figura 12.8	69
Figura 12.9	69
Figura 13.1. Código de la tabla en FBML	72
Figura 13.2. Representación de la tabla	72
Figura 13.3. Página principal	72
Figura 13.4	73
Figura 13.5	74
Figura 13.6	74
Figura 13.7	75
Figura 13.8	75
Figura 16.1. Horas dedicadas a cada tarea del proyecto	87
Figura 18.1. Publicación en el muro de un amigo que ha usado el Recomendador	91
Figura 18.2. Solicitud de permisos	92
Figura 18.3	93

Figura 18.4. Página principal de la sección Música.	94
Figura18.5. Pantalla de resultados.	96
Figura 18.6. Publicación automática en nuestro muro.	96
Figura 19.1. Tabla fb_user_juego.	98
Figura 19.2.....	99
Figura 19.3.....	99
Figura 19.4.....	100
Figura 19.5. Recomendación obtenida mostrando cinco juegos.	101
Figura 19.6. Recomendación obtenida mostrando tres juegos.	101
Figura 19.7.....	102
Figura 19.8. Resultados obtenidos a partir del primer criterio.....	103
Figura 19.9. Sección libros sin ningún elemento seleccionado.....	104
Figura 19.10. Resultado tras la aplicación del tercer criterio.	104
Figura 19.11.....	105
Figura 19.12. Resultados tras la aplicación del primer criterio.....	106
Figura 19.13. Resultados tras la aplicación del segundo criterio.	107
Figura 19.14. Resultados tras la aplicación del tercer criterio.....	107
Figura 19.15.....	108

CAPÍTULO 1

Definición de objetivos.

Con el desarrollo del presente proyecto se pretende elaborar una aplicación para Facebook a través de la cual, los usuarios de esta red social que accedan a ella, puedan encontrar tanto música, como libros, programas de televisión, películas e incluso juegos afines a sus gustos y que desconozcan.

Los objetivos de este trabajo pueden dividirse en dos grandes bloques. En primer lugar destacamos los propios del desarrollador del proyecto, que son los siguientes:

- Familiarización con el entorno de desarrollo de Facebook.
- Aprendizaje de un nuevo lenguaje de programación interpretado, que en nuestro caso será PHP.
- Conocimiento y comprensión de los conceptos de servidor web, dominio, y hosting, así como su funcionamiento.
- Creación y gestión de una base de datos.

En segundo lugar, queremos subrayar que este trabajo está también orientado a todos aquellos alumnos que en un futuro opten por la realización de un proyecto fin de carrera de similares características al nuestro, de manera que puedan disponer así de una base desde la que partir, lo más accesible y clara posible. Ello permitirá a los interesados avanzar con mayor rapidez y profundizar más, no tanto en la creación de la aplicación en sí, como en el desarrollo y funcionalidad de ésta.

CAPÍTULO 2

Introducción.

La idea básica de nuestro proyecto es poder aprovechar las ventajas que nos ofrecen las redes sociales para relacionar nuestros gustos y aficiones con el de otras personas, conocidas o no, con el objetivo de poder descubrir nuevas canciones, películas, libros..., que puedan ser de nuestro interés.

Para ello vamos a hacer uso de la red social más importante en la actualidad: **Facebook**.

Aunque la inmensa mayoría de los usuarios de INTERNET conocen a grandes rasgos el funcionamiento de esta red social, vamos a subrayar sus principales características.

2.1.- ¿Qué es Facebook?

Facebook es, como hemos señalado anteriormente, una red social, gratuita y de gran popularidad. Aglutina las características más relevantes de los blogs, los foros y grupos de Internet y de los espacios dedicados a compartir imágenes, además de otras muchas funciones complementarias.

A través de las conexiones que crean los usuarios entre sí, podremos localizar fácilmente a viejos amigos o antiguos compañeros y ponernos en contacto con ellos, a nuevos clientes e incluso a personas a las que no conocemos pero que comparten nuestros mismos gustos e intereses.

2.2.- ¿Qué la hace tan popular?

En principio, puede parecer que Facebook no aporta grandes novedades frente a otras redes sociales desarrolladas con anterioridad como MySpace, por ejemplo. Sin embargo, la gran diferencia entre ambas está en las formas. Facebook incluye prestaciones muy parecidas a las que ofrece MySpace, pero de manera mucho más clara, controlada y profesional. Hoy, MySpace se ha convertido, prácticamente, en un espacio utilizado mayoritariamente por adolescentes y el número de sus usuarios es considerablemente menor que el de Facebook donde la edad de más de la mitad de sus miembros supera los 25 años. De acuerdo con estos datos puede afirmarse que esta red social ha dejado de ser una comunidad Web destinada sólo para universitarios, como en sus momentos iniciales, habiéndose extendido a todos los sectores de la sociedad.

A comienzos de 2011, más de 600 millones de personas utilizaban Facebook con regularidad en todo el mundo, cifra que sigue aumentando vertiginosamente, habiendo sido ya traducido a más de 70 idiomas.

En esta red social, como en la mayoría, podemos compartir fotos, enlaces, o enviar comentarios a nuestros amigos. Pero hay algo que la distingue de sus competidoras: su versatilidad, la variedad de tareas que podemos realizar con las herramientas y aplicaciones que nos facilita, lo que la ha convertido en una de los instrumentos más importantes para la comunicación interpersonal y grupal de nuestros días.

Facebook cuenta en la actualidad con unos 900 millones de objetos con los que podemos interactuar, entre los que se encuentran páginas, comunidades, grupos o eventos.

Después de las interacciones entre usuarios, las aplicaciones son la segunda razón de la creciente popularidad de Facebook. Esto se debe, básicamente, a la combinación de tres factores: sencillez, entretenimiento y utilidad. En este sentido, el usuario dispone de una gran cantidad de aplicaciones entre las que, a modo de ejemplo, encontramos ideas triviales como pintar con spray el muro de nuestros amigos o enviarles macetas con plantas virtuales que crecerán cada día, a otras que pueden resultar de gran interés, como crear un completo currículum profesional desde la propia red.

Otra de las grandes ventajas de Facebook, con respecto a sus competidoras, es que cualquier persona, empresa o grupo puede crear su propia página dentro de la red. Desde una marca comercial de prestigio internacional como Coca-Cola, hasta el último estreno cinematográfico, o la presentación de una novedad discográfica se hacen presentes en la red a través de Facebook. Esto les proporciona una rápida difusión y publicidad, permitiéndoles además el libre acceso a los datos de los usuarios que en su perfil han manifestado su interés por esos productos. Todo esto se traduce, finalmente, en una mayor rentabilidad económica de las propuestas comerciales que utilizan esta red social.

Según las cifras publicadas recientemente por eMarketer, los ingresos de Facebook el pasado año 2010, tan sólo por el concepto de publicidad, se elevaron a más de dos mil millones de dólares, mostrando alrededor de un billón de anuncios.

CAPÍTULO 3

Aplicaciones en Facebook.

3.1.- ¿Qué es una aplicación de Facebook?

El concepto de aplicación informática se define, según la RAE como un “programa preparado para realizar una aplicación específica”. Podríamos hacer extensiva esta definición al concepto de aplicación de Facebook.

Desde esta perspectiva, además de un programa preparado para realizar una acción específica, también debe entenderse como una plataforma para desarrolladores que ofrece un marco para interactuar con las características de esta red social. Servirá también para que los usuarios de la propia aplicación interactúen entre ellos o con la propia Facebook.

3.2.- ¿Cómo funcionan?

Lo primero que es preciso tener en cuenta es que las aplicaciones Facebook se alojan en un servidor web externo, es decir Facebook no nos proporciona espacio dentro de su sitio web para alojar la aplicación diseñada. Ante esta circunstancia puede optarse, bien por crear un servidor y alojarlo en nuestro ordenador, bien por usar los espacios que nos ofrecen ciertos portales especializados de Internet.

Dado que preferimos que la aplicación esté permanentemente disponible, optaremos por la segunda opción. En nuestro caso concreto, puesto que ya éramos poseedores de un dominio web, no hemos tenido problemas en lo referente a este aspecto.

El proceso de desarrollo y ejecución que sigue una aplicación de Facebook es idéntico al de cualquier otra aplicación web, con la diferencia de que el usuario accederá a ella a través de esta red social, y no mediante su página web. Es decir, la red actúa en este caso como un intermediario, un medio de acceso a nuestra aplicación, en vez de acceder directamente al espacio en el que está alojada.



Figura 3.1

CAPÍTULO 4

Cómo crear una aplicación en Facebook.

El primer paso que debemos dar a la hora de crear una aplicación, es formalizar su registro. Aunque en un principio parece un procedimiento sencillo y rutinario, hay ciertos detalles que pueden presentar mayor complejidad debido a que Facebook requiere ciertos conocimientos básicos que tal vez no poseamos.

4.1.- Registro de la aplicación paso a paso.

Para poder crear el “recomendador”, objeto de nuestro proyecto, tendremos que agregar a nuestro perfil la aplicación “Desarrolladores /Developers”. Este paso no presenta dificultad alguna. Para darnos de alta en este colectivo, abriremos el cuadro de diálogo de búsqueda y escribiremos en él la palabra “desarrollador”



Figura 4.1

Entre los resultados de la búsqueda aparecerá el recuadro “Aplicaciones” y como descriptor de su contenido el logo correspondiente de Facebook y, a su derecha, la palabra “Desarrollador” (figura 4.1). Cuando pulsemos sobre el nombre, la aplicación nos pedirá, como es habitual en esta red, una serie de permisos. Al aceptarlos habremos formalizado nuestro registro en la aplicación “Desarrolladores”.

Una vez en la “página de Desarrolladores”, observaremos que ésta se divide básicamente en dos zonas. La parte de la izquierda, mucho más ancha, funciona como un tablón de anuncios ya que es aquí donde periódicamente se publican noticias y actualizaciones de la propia red. La parte derecha presenta la relación de las distintas aplicaciones que hemos creado. En la parte superior de esta columna derecha figura el botón donde deberemos pulsar para crear la aplicación.

Pulsamos el botón “Set Up New App” (“crea una nueva aplicación”) y nos encontramos con un requisito que no se exigía si hubiésemos registrado nuestra aplicación tan sólo unos meses atrás.



Figura 4.2

La razón es que en Facebook, con objeto de prevenir el registro de determinadas aplicaciones cuya única función consiste en el envío masivo de “correo basura” o “mensajes basura”, (spam), se han implantado “controles de seguridad”.

La medida adoptada consiste en que, como paso previo al proceso de registro de cualquier nueva aplicación, deberemos introducir desde el teclado de nuestro ordenador un número de teléfono móvil o de una cuenta bancaria. Al interactuar de este modo, se pretende asegurar que el registro no se ha iniciado mediante sistemas automáticos.

Ante la disyuntiva de número de móvil o número de tarjeta de crédito, por simple cuestión de riesgos, nos inclinamos, en líneas generales, por la primera opción. Desde Facebook se nos enviará un mensaje al número facilitado con un código de confirmación. Este código será el que introduciremos cuando sea requerido durante el proceso de registro de nuestra aplicación.

Existe, no obstante, una importante observación que es preciso tener en cuenta llegados a este punto. Facebook tiene convenio con determinadas compañías telefónicas en ciertos países, entre las que, de momento, no están incluidas las que operan en España. Si transcurrido un plazo prudencial no hemos recibido el mensaje de confirmación, no nos quedará otra alternativa para registrarnos que la de recurrir a la opción de proporcionar un número de tarjeta de crédito. Antes de dar este paso es muy recomendable el asesoramiento por parte del personal de la entidad bancaria emisora de la tarjeta.

Una vez introducidos los datos correspondientes, podremos iniciar el proceso de registro de nuestra aplicación.

Como paso previo, Facebook nos muestra un formulario de aceptación de términos y condiciones. En síntesis, se hace referencia a una serie de normas de seguridad y control, en las que se exige el compromiso de que la aplicación desarrollada no sea usada para enviar virus, molestar a otros usuarios, etc. Se advierte además, que seremos los propietarios de lo que pueda publicarse en nuestra aplicación, si bien se concede a Facebook una licencia no exclusiva y transferible para el uso de cualquier contenido que publiquemos en esta red social, o en conexión con ella. De la misma manera se nos informa de que una vez que borremos nuestra aplicación, esta licencia se cancelará, aunque conservarán nuestra aplicación en copias de seguridad durante un tiempo que, sin más concreciones, denominan como “razonable”.

Aceptados los términos y condiciones planteados podemos empezar a cumplimentar los distintos campos de nuestra aplicación.

Lo primero que deberemos introducir es el título, que podremos modificar después fácilmente accediendo a ella a través de “Desarrolladores, y pulsando en “editar”.

En este punto debemos tener en cuenta que Facebook tiene reservados para uso exclusivo ciertos términos. Por esta razón, no podrán

formar parte del nombre de nuestra aplicación algunos de carácter genérico que son muy usuales en esta red como por ejemplo “Facebook”, “FB” o “muro”, por citar sólo algunos.

Tras introducir el nombre, se nos redirigirá a otra página, en la que tendremos que cumplimentar diversos campos de información básica sobre la aplicación. Como paso previo, otro “filtro de seguridad” nos requiere que tecleemos el captcha que nos aparece en la pantalla.

La información básica consta de una serie de campos, de los cuales explicaremos los más importantes a continuación:

- **Nombre de la aplicación:** es el título que acabamos de introducir en los primeros pasos.
- **Descripción:** aunque no es obligatorio cumplimentar este campo, resulta conveniente hacerlo ya que proporciona a los potenciales usuarios una breve descripción sobre los objetivos de nuestra aplicación.
- **Icono:** esta pequeña imagen será la que les aparezca a los usuarios al lado del nombre de nuestra aplicación si la agregan a sus menús de acceso rápido. Sería algo así como el icono de acceso directo.
- **Logotipo:** esta imagen será la que les aparezca a los usuarios cuando busquen en Facebook nuestra aplicación.
- **Idioma:** indica el idioma en el que está nuestra aplicación.
- **Manager users:** aquí indicaremos si queremos compartir la administración de nuestra aplicación con un amigo.

Además de la de información básica, hay otras cinco pestañas mediante las que podremos seguir documentando la aplicación. En este caso, únicamente nos interesará la titulada como “Facebook Integration”

Los campos importantes de este apartado son:

- **Canvas Page.** Contendrá la dirección para acceder a nuestra aplicación. Facebook la genera automáticamente, y simplemente consiste en: `http://apps.facebook.com/ + nombre de nuestra aplicación + /`.
- **Canvas URL.** Este campo es muy importante. Es la dirección de nuestro espacio web, donde está el contenido de la aplicación que Facebook cargará en su página.

- **Canvas Type.** Cuando empezamos a desarrollar este proyecto, podíamos escoger entre dos opciones, iFrame y FBML.

En este caso desarrollaremos una aplicación en modo FBML, mediante el cual quedará integrada en Facebook, desde donde se controlará su ejecución, demandando de nuestro servidor cuándo, cómo y qué debe mostrar en la pantalla del usuario. Esto difiere del modo iFrame en el que Facebook sólo crea un marco aparte (frame) en el que cargará nuestra web.

Recientemente Facebook decidió estandarizar el código de sus aplicaciones pretendiendo con ello “facilitar” el trabajo a los desarrolladores. Esta decisión trajo consigo la eliminación de la opción de crear aplicaciones basadas en FBML, un lenguaje propio de Facebook similar al HTML.

Dado que el desarrollo de nuestro proyecto se inició antes de que tuvieran lugar los cambios señalados, pudimos elegir entre las dos opciones mencionadas, decidiéndonos por la de FBML.

Los motivos que nos llevaron a ello fueron, entre otros, que la mayoría de ejemplos y tutoriales disponibles en la red hacían referencia a FBML y no a iFrame. Además, FBML nos permitía ciertas funcionalidades relativas a Facebook con más facilidad y rapidez, como, por ejemplo, extraer un cuadro para seleccionar amigos, o imprimir la lista de amigos de un usuario. Por otra parte, iFrame presentaba nuestra aplicación en una nueva ventana, mientras que nuestra pretensión era que estuviese integrada en la propia de Facebook, como la gran mayoría de las aplicaciones. Tuvimos en cuenta, además, que la mayor parte de las aplicaciones existentes para esta red están realizadas en este código y que en caso de plantearse dudas o problemas, el intercambio de experiencias con otros desarrolladores a través de distintos foros podría presentarnos menos problemas al ser los programadores en FBML más numerosos.

El campo Canvas URL puede presentarnos ciertos problemas, ya que se requieren determinados conocimientos previos que nos pueden resultar ajenos. Nociones como la necesidad de un servidor o qué es el alojamiento, son algunas cuestiones que, a modo de ejemplo, debemos conocer de antemano. A ellos nos dedicaremos más adelante.

Para terminar con el registro, tanto en esta pestaña como en la siguiente, quedan aún ciertos campos sin cumplimentar que, por el momento y para nuestro propósito, no necesitamos especificar. Por todo ello, llegados a ese punto pulsaremos la opción “guardar cambios” y ya tendremos registrada nuestra aplicación.

CAPÍTULO 5

Nuestra aplicación.

5.1. – En qué consiste.

El proyecto que emprendemos consiste, básicamente, en el desarrollo de una aplicación para recomendar música, es decir, de un sistema “recomendador”.

El objetivo principal que persigue es que los usuarios que accedan a la aplicación, tengan la posibilidad de descubrir nuevos grupos o intérpretes, canciones o estilos musicales que puedan estar relacionados con sus intereses y gustos personales, para que si lo estiman conveniente puedan incluirlos finalmente entre sus preferencias. Para ello, el sistema diseñado pretende aprovechar al máximo la información que proporciona Facebook del resto de usuarios.

5.2. – Procedimientos para su diseño.

Para la elaboración de una aplicación como la que se propone en este proyecto, pueden tomarse como punto de partida distintos enfoques. A continuación se esboza el proceso seguido en los momentos iniciales cuando se valoraron diversas opciones de acercamiento al problema y se exploraron las ventajas e inconvenientes de cada una de ellas, lo que nos permitió acotar y concretar la propuesta definitiva.

5.2.1.- Primera idea

En los momentos iniciales del diseño del “sistema recomendador” se pensó que cada usuario, al ejecutar la aplicación, introdujera los nombres de los intérpretes o grupos musicales y los títulos de sus canciones o temas preferidos. Junto a ello se solicitaría también definir determinadas características que permitieran completar la información como, por ejemplo, el estilo o tipo de música preferida, la ocasión o momento favorito para escucharla, su interpretación por un determinado artista, grupo, dueto, etc.

La adopción de este criterio, exigiría disponer de una base de datos, con el fin de que el usuario introdujera las características de intérprete y melodía una sola vez y no estuviese obligado a hacerlo en cada ocasión que la ejecutase.

Para su correcto funcionamiento, deberíamos tratar de conseguir que cuando un usuario hubiese grabado los datos de un determinado intérprete, por ejemplo, Ana Belén, el resto de usuarios que hubiese incluido a esta cantante entre sus preferidas debería hacerlo también con los mismos criterios que el anterior.

Para salvar esta dificultad se daría la oportunidad a los demás usuarios de elegir al intérprete directamente, sin más datos. Esta solución aparente nos llevaría directamente a otro asunto problemático, ya que es lógico pensar que cada usuario puede tener una opinión diferente respecto a las características que definen el estilo de un mismo artista.

Tras valorar estas circunstancias, descartamos esta idea no tanto por el posible conflicto descrito anteriormente, sino porque consideramos que una aplicación que exigiese a los usuarios cumplimentar para cada intérprete o canción no menos de tres o cuatro campos, resultaría poco atractiva y no tendría ningún éxito.

5.2.2.- Segunda idea

Tratando de aprovechar al máximo los datos que proporciona Facebook sobre las preferencias de los usuarios, la segunda idea que exploramos consistía en extraer patrones comunes.

La posibilidad de ahorrarnos la utilización de una base de datos implementando funciones que extraigan directamente la información de los usuarios y sus amigos, e incluso de los amigos de nuestros amigos, resultaba muy atractiva, pero, finalmente, esta opción también quedó descartada.

El método a seguir que requería este enfoque exigía que el usuario introdujese el nombre de uno o varios artistas o el de distintos temas o canciones. Junto a ello se precisaba también que se almacenase en distintas tablas los gustos y predilecciones de nuestros amigos y también las de sus amigos para, posteriormente, aplicarles nuestros criterios de recomendación y obtener así el resultado final.

El desarrollo de este procedimiento se abandonó finalmente debido a las limitaciones impuestas por Facebook, ya que aunque un usuario sí puede acceder a los datos sobre sus preferencias, en lo que respecta a la información que proporciona de sus amigos, únicamente le son accesibles los correspondientes al nombre y el id.

5.2.3.- Tercera idea

Tras la valoración de las opciones anteriores y de sus ventajas e inconvenientes, el método que finalmente se adoptó y que se desarrolla en este proyecto, es el resultante de una combinación de elementos de las dos anteriores propuestas.

Por un lado, pretendemos obtener el máximo provecho de los datos e información que nos ofrece Facebook, y vamos a tratar de franquear la dificultad de acceder a los gustos de nuestros amigos mediante una base de datos. Junto a todo ello, y para mejorar los resultados de la recomendación, ampliaremos los datos disponibles con los que vayan proporcionando el resto de usuarios que accedan a la aplicación, y no sólo los de nuestros amigos.

A este respecto, se ha optado por prescindir de los campos “intérprete” o “canción” para quedarnos únicamente con el de “música”, a pesar de que a la hora de manifestar nuestras preferencias musicales en Facebook habitualmente se mencionen intérpretes en general y no canciones en particular.

Por otro lado, el recomendador tampoco podrá seguir criterios tales como que nuestro artista interprete un determinado tipo de

música y, simplemente, se basará en la afinidad de gustos de unos usuarios con otros.

La idea principal es sencilla, aunque variará ligeramente según el criterio de recomendación que escojamos.. Para ofrecer un ejemplo práctico del funcionamiento del sistema, supongamos tres usuarios: Antonio, Beatriz y Carlos.

Carlos accede a la aplicación, donde ya se encuentran registrados Antonio y Beatriz, con el fin de encontrar artistas de características similares a dos de sus favoritos, Joaquín Sabina y Joan Manuel Serrat.

Tanto Beatriz como Antonio, tienen entre sus preferidos varios, entre ellos, los dos que Carlos usa como referencia de búsqueda. Además Antonio y Beatriz comparten uno en común, Ana Belén. Con estos antecedentes sería fácil suponer que a Carlos le puede gustar Ana Belén.

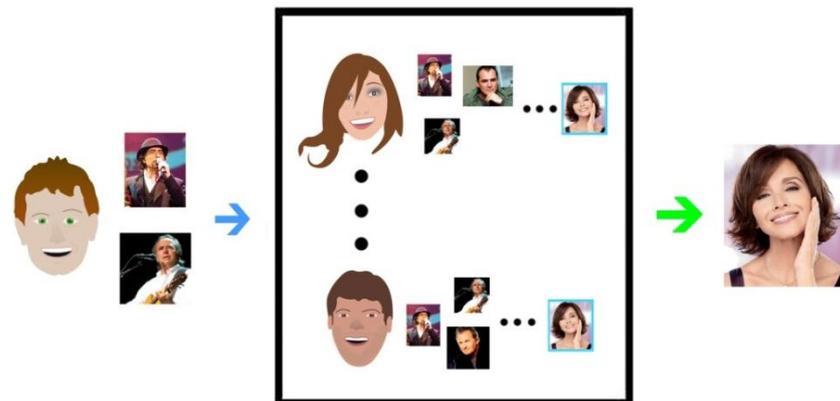


Figura 5.1

Ante la imposibilidad de acceder a los datos de nuestros amigos, como se ha señalado anteriormente, vamos a ayudarnos de una base de datos para salvar este problema.

Recordemos que, en relación a nuestros amigos, únicamente teníamos acceso a su nombre e id de usuario. Dado que a los gustos y preferencias de una persona sólo pueden consultarse desde la API si es el propio usuario de la aplicación, ¿por qué no acceder a la sesión del usuario cuando inicie nuestra aplicación? De esta manera, sí podremos llegar a conocer los gustos de cada uno, ya que, al ser uno mismo el que está accediendo a sus datos, no hay ninguna restricción hacia ellos.

Cuando se ejecute nuestra aplicación por primera vez, nuestros gustos quedarán registrados automáticamente en la base de datos mediante una serie de funciones. Este registro sobre nuestras preferencias, permanecerá inalterable salvo que el usuario, en posteriores accesos, modifique los datos en su perfil de Facebook.

Ya dentro de la aplicación, nos aparecerá un sencillo menú, en el que podremos elegir qué queremos que nos recomiende (en un principio sólo habíamos pensado en música, sin embargo, como veremos más adelante, la aplicación ha sido ampliada).

Una vez elegido el apartado de música, podremos proceder a la recomendación. Para ello, primero, deberemos seleccionar el criterio que preferimos. Dependiendo de nuestra elección, pasaremos, o no, a elegir entre los intérpretes que tenemos recogidos en nuestros gustos musicales en el perfil de Facebook, y con los que queremos que los seleccionados por nuestra aplicación guarden cierta similitud. Hecho esto pulsaremos en “enviar la información”, y se nos redirigirá automáticamente a una nueva página con el resultado de la búsqueda.

CAPÍTULO 6

Criterios de recomendación.

En la vida real, a la hora de recomendar a una persona algo que creamos que le pueda gustar, cada uno sigue un criterio. Es decir, no existe un método exacto para recomendar. La lógica nos dice que nos basemos en sus propios gustos, y, poniéndolos en común con los nuestros, saquemos ciertas conclusiones. Para nuestra aplicación, hemos implementado tres criterios distintos, con ciertas similitudes, pero con pequeñas diferencias que harán que los resultados no sean los mismos.

6.1.- Criterio de selección con pesos.

Es probable que nuestro primer criterio sea el más eficiente. Si buscamos algo concreto es el que deberíamos usar, ya que se supone que nos devolverá como resultado los intérpretes más parecidos a los que figuran en una relación previa que ya tenemos.

Si optamos por este criterio, lo primero que debemos hacer es seleccionar los artistas que han de servir de referencia de búsqueda. Una vez marcados, a enviaremos la información iniciando así el proceso.

Mediante un algoritmo que hemos implementado en nuestra aplicación, se seleccionará a los usuarios que incluyan en su perfil a alguno de estos artistas. Podríamos afirmar que los usuarios de la aplicación han pasado el “primer filtro”.

A estos elegidos, a quienes a partir de ahora llamaremos “recomendadores”, pasaremos a asignarles un peso.

Se podría afirmar que, si el número de coincidencias entre los artistas de un “recomendador” y los del usuario actual es muy elevado, ambos tendrán unos gustos muy similares, mientras que, al contrario, diferirán mucho si aquéllas son pocas o nulas. Por esta razón, podríamos considerar que no sería correcto que los artistas a recomendar de un “recomendador” que compartiera el 90% de los artistas de un potencial usuario, tuviesen el mismo peso que los artistas de otro “recomendador” que tan sólo compartiese con el usuario los artistas seleccionados al inicio de la aplicación. Para equilibrar esta desproporción consideramos una medida acertada el asignar un punto por cada una de las coincidencias, por lo que el peso final de cada “recomendador” vendrá determinado por la suma de esos puntos.

A continuación, una vez calculado el peso de cada “recomendador”, se le asignará también a cada artista otro peso. Éste se obtendrá sumando el valor de los pesos de cada uno de los “recomendadores” en cuya lista de preferencias aparezca.

Finalmente, mostraremos los dos primeros intérpretes con mayor peso.

6.2.- Criterio de selección.

Nuestro segundo criterio es similar al primero, aunque con ciertas diferencias.

Para empezar, debemos seguir el mismo procedimiento que en el anterior criterio, es decir, seleccionar artistas similares a los que queremos encontrar.

La diferencia aparece a la hora del resultado. En este caso, no tendremos en cuenta si un “recomendador” tiene más coincidencias que otro con el usuario de la aplicación, ni tampoco usaremos pesos, en un principio, para devolver a los artistas que recomendaremos.

Iniciada la búsqueda con los intérpretes de referencia, la aplicación selecciona a los usuarios cuya lista de preferencias contenga la totalidad de los referenciados asignándoles el papel de “recomendador”. Finalizada esta fase, se comprueba las coincidencias existentes entre ellos y se devuelve el número de artistas coincidentes que haya seleccionado el usuario.

Dado que en los momentos iniciales de la difusión de la aplicación aún no estará muy extendida, es posible que no haya “recomendadores” que cumplan los requisitos exigidos, por lo que añadiremos una modificación transitoria. Ésta consistirá en que si no existe ningún usuario que contenga a todos los artistas seleccionados, asignaremos el rol de recomendador a todos aquellos usuarios que tengan, todos los artistas seleccionados excepto uno, si tampoco existe, tomaremos aquellos usuarios que tengan todos menos dos, y así sucesivamente.

También es posible que nos encontremos con que, entre los recomendadores, no haya ningún artista coincidente. En caso de que no exista, la aplicación nos devolverá aquellos artistas que tengan en común más “recomendadores”.

6.3.- Criterio “invariable”.

Podríamos decir que éste es el criterio más estático de todos, ya que siempre nos devolverá el mismo resultado si no ha habido cambios en la aplicación (es decir, si desde que hicimos la última búsqueda no ha habido usuarios nuevos, ni ningún usuario ha cambiado nada de sus gustos, incluyéndonos a nosotros).

Ahora, no podremos seleccionar artistas sino que directamente obtendremos la recomendación conforme a todos ellos.

La recomendación se hará basada en pesos, igual que en el primer caso.

La asignación de pesos seguirá el mismo patrón que en el Criterio de selección y pesos, teniendo en cuenta el tanto por ciento que tienen en común los recomendadores con el usuario actual de la aplicación.

La única diferencia es que no haremos ningún filtrado previo, con lo cual no podremos concretar si queremos que la recomendación nos devuelva un usuario parecido a una serie de artistas específicos.

CAPÍTULO 7

Accediendo a Facebook Developers.

Una vez expuesta la idea, llega el momento de transformarla en una aplicación informática. Pero... ¿cómo hacerlo? ¿Qué lenguaje de programación utilizar?

Facebook nos indica que podemos desarrollar nuestra aplicación en los principales lenguajes de programación, como PHP, Python, Java o C# y, puesto que facilita sendas librerías en JavaScript y PHP, usaremos uno de estos dos.

Dado que, en principio, el nivel inicial de conocimientos que teníamos de ambos lenguajes era básico, se optó por PHP ya que al parecer es el utiliza la inmensa mayoría de programadores que desarrollan aplicaciones en Facebook actualmente. Una vez escogido el lenguaje de programación, y descargadas las herramientas que nos proporciona Facebook, todo estaba listo para comenzar esta fase del trabajo.

Como en todo proyecto que no cuenta con una base de partida y del que prácticamente se desconoce todo, sus inicios no son fáciles por lo que debemos documentarnos convenientemente. Esta fase inicial la podríamos resumir de manera muy breve: leer, leer y leer antes de escribir una sola línea de código.

La aplicación que Facebook pone a disposición de los desarrolladores, que en realidad es más bien una página web, cuenta con varias secciones.

Como hemos señalado anteriormente, la página se divide en dos grandes zonas, pero hay un pequeño espacio del que no hemos hablado, y que será el más importante para nuestro trabajo a partir de ahora.

En la parte superior izquierda de esta página encontramos tres enlaces: “Documentación”, “Foro” y “Política”.

En estos momentos iniciales del desarrollo el apartado de Documentación será el que capte preferentemente nuestra atención, aunque con frecuencia accederemos tanto a los foros como a la política de Facebook.

Antes de seguir, hay que hacer constar que toda la documentación de Facebook está en inglés, por lo que si no nos desenvolvemos con cierta soltura en este idioma, tendremos serios problemas para poder desarrollar ésta y cualquier otra aplicación.

Una vez accedemos a Documentación, se nos redirige a otra página web dividida en varias partes.

A la izquierda figura un panel con cuatro secciones desplegadas, en la que está básicamente toda la información, y donde, a nuestro entender, resulta fácil perderse entre sus contenidos.

El espacio central de la página es variable. Según pulsemos en cada una de las cuatro secciones de la parte izquierda, nos aparecerán desplegados los apartados que contienen cada una de ellas.

En el marco superior, se mostrará otro grupo de enlaces, entre los se incluyen nuevamente “Documentación” y “Foro” y, además, “Casos reales”, “Blog” y “Mis aplicaciones”.

En “Casos reales” aparecen referencias a páginas web que integran alguna utilidad de Facebook en sus portales, pero que, a pesar de su indudable interés y atractivo, carecen de utilidad para nuestros propósitos.

Algo parecido ocurre con la sección “Blog” aunque puede ser de interés realizar visitas ocasionales ya que es aquí donde en Facebook se publican las novedades referentes a actualizaciones y modificaciones y, dado el rápido avance tecnológico y la expansión continua de la red, la frecuencia de éstas va en progresivo aumento.

Por tanto, y de acuerdo con lo expuesto, vamos a centrarnos en el apartado de “Documentación”.

CAPÍTULO 8

Documentándonos.

Como hemos señalado en el capítulo anterior, el apartado de “Documentación” se divide en cuatro secciones: “Getting Started”, “Core Concepts”, “Advanced Topics” y “SDK & Tools”

Pulsaremos sobre “Getting Started” (“Para empezar”). Ésta se subdivide en otras cinco secciones y, dado que nuestra pretensión es desarrollar una aplicación en Facebook, nos dirigiremos a “Apps on Facebook”.

Conforme vayamos avanzando y profundizando en las secciones, sobre todo una vez inmersos en la programación del proyecto, nos daremos cuenta de que la API de Facebook no es tan completa como parece a primera vista y la falta de código o ejemplos en casi todos los apartados, puede dificultar la tarea a cualquier programador.

Una vez hecha esta aclaración, vamos a analizar cada sección, detallando las más importantes.

8.1.- Apps on Facebook (aplicaciones en Facebook).

En esta sección se nos exponen de manera básica conceptos generales sobre las aplicaciones de Facebook. Llegados a este punto, es necesario tener muy claros conceptos tales como “qué es una aplicación” o un “Canvas” (se traduciría al español como lienzo o página en blanco, aunque Facebook en algún sitio lo traduce como lona).

Conviene recordar que tanto “Canvas Page”, como “Canvas URL” son en realidad dos campos que tuvimos que cumplimentar al registrar nuestra aplicación, por lo que deberían resultarnos familiares. Debemos tener en cuenta que suelen ser fuente de bastantes problemas si no sabemos muy bien qué son exactamente.

El “Canvas Page” es, simplemente, la dirección donde se guarda nuestra aplicación en Facebook. En realidad es prácticamente un vínculo al “Canvas URL”, ya que, como hemos dicho anteriormente, Facebook no aloja ningún contenido de nuestra aplicación.

Así, “Canvas URL” es la “verdadera” dirección de nuestra aplicación, la dirección de una página dentro de nuestro dominio, donde estará alojada, aunque, si intentamos acceder a ella a través de esta dirección, obviamente no funcionará.

8.1.1.- Authorization (autorización).

A continuación, Facebook nos informa básicamente acerca de que los usuarios tienen que dar permisos a las aplicaciones y que, una vez que acceden a éstas, se puede obtener cierta información con objeto de llevar a cabo futuros proyectos más personalizados.

Los permisos que acepten nuestros usuarios limitarán en gran parte los datos que podremos obtener de ellos cuando accedan a nuestra aplicación. Cuantos más permisos requiera, más datos podremos obtener de los usuarios potenciales. Sin embargo, esto puede plantear algunos riesgos y, si nos excedemos en la solicitud de información, algunos podrían considerar que se vulnera su privacidad renunciando a su utilización. Por consiguiente, lo ideal es solicitar el número imprescindible de permisos para un correcto funcionamiento de nuestra aplicación.

En apartados posteriores analizaremos más profundamente los permisos que hemos incluido en nuestro proyecto y cómo lo hemos hecho.

8.1.2.- Social channels (medios sociales).

Facebook pone a nuestra disposición numerosas herramientas para que nuestra aplicación se difunda a través de su red y crezca rápidamente en número de usuarios.

Cuando alguien ejecuta nuestra aplicación (aunque sólo acceda la primera vez), Facebook crea un enlace directo a ella en su página de inicio. Además, el usuario publicará automáticamente en su tablón que ha comenzado a usarla.

A parte de esto, también nos proporciona herramientas para que los usuarios publiquen en su tablón los resultados obtenidos; es decir, en nuestro caso, qué intérpretes o artistas les ha recomendado.

Si pretendemos que nuestra aplicación se difunda con mayor rapidez deberemos incluir la herramienta que permite a los usuarios enviar una petición a sus amigos para que accedan a ella.

Además de todas estas funciones, también podremos crear una página de nuestra aplicación, donde los usuarios podrán expresar su opinión sobre ella o sobre cualquier otro tema.

Una vez leído todo el contenido de la primera sección (Getting Started), vamos tomando conciencia del funcionamiento de esta red.

Las otras tres secciones del menú, las resumiremos también brevemente, deteniéndonos más adelante en ciertos detalles, que consideramos relevantes para nuestro proyecto, cuando los vayamos necesitando.

8.2.- Core concepts (conceptos básicos).

Al desarrollar nuestra aplicación, probablemente hayamos iniciado la programación mucho antes de leernos todos y cada uno de los apartados, ya que es una tarea que requiere mucho tiempo y puede resultar, a primera vista, algo tediosa. Sin embargo, se aconseja realizar estas lecturas, estudiando los contenidos de las distintas secciones con objeto de evitar después continuas revisiones y consultas que no producirán otro resultado que el de retrasar el desarrollo del proyecto.

Como su nombre indica, en esta parte se encuentran los conceptos básicos, las bases de nuestra aplicación y, en general, de todo lo que hagamos en Facebook.

Este apartado se vuelve a subdividir en otras cinco secciones.

8.2.1.- Social plugins (accesorios sociales).

Básicamente estas herramientas nos permitirán saber lo que les gusta a nuestros amigos, lo que comentan o comparten en la web, etc.

En nuestro caso, esta sección ofrece poco interés, ya que su uso está destinado más bien a páginas webs que integran herramientas que interactúan con Facebook, como el famoso botón de “Me gusta”, entre otras muchas utilidades.

8.2.2.- Graph API (interfaz).

Para la mayoría de las aplicaciones y creaciones de Facebook en general, esta será la parte más importante puesto que, como hemos señalado en otro capítulo, contiene todos los objetos con los que podemos interactuar así como una descripción de sus características principales.

En este caso, debido a las peculiaridades de nuestra aplicación, tan sólo hemos usado dos objetos, el objeto “User”, y el objeto “FriendList”, por lo que no profundizaremos en ella.

Los dos siguientes apartados son “Social Channels” y “Authentication”, que prácticamente incluyen los mismos contenidos que la primera en parte, aunque ahondando algo más en algunos detalles, así como incluyendo partes de código, que nos podrían ser útiles.

El último apartado, “Open Graph protocol”, contiene información parecida a la de “Social Plugins”, y también nos proporciona herramientas para incluir aún más accesorios de Facebook en páginas webs. Sin embargo, como hemos señalado, nuestro interés se centra de manera exclusiva en las aplicaciones de Facebook por lo que, de momento, no los utilizamos.

8.3.- Advanced topics (temas avanzados).

Aunque por el título parezca que esta sección es para desarrolladores avanzados, en realidad, es aquí donde se empieza a tratar realmente todo lo relacionado con la programación.

A pesar de lo amplia que es esta sección, nos centraremos principalmente en el apartado de “Legacy FBML”.

Un problema que nos hemos encontrado al empezar la aplicación es que Facebook, como cualquier otra tecnología hoy en día, está en constante transformación. Estos cambios se deben a la aparición de nuevos equipos más evolucionados y a las constantes mejoras introducidas tanto en los lenguajes de programación como en la propia aplicación de la red.

Puede afirmarse que, en la actualidad, Facebook se encuentra en una época de transición, en la que está eliminándose el lenguaje FBML y la API también ha sido renovada recientemente.

A pesar de todo, hemos seguido incluyendo el lenguaje FBML, ya que nos facilita ciertas funciones que de otra manera serían inaccesibles o difíciles de implementar. En nuestra decisión también ha influido el hecho de que el número de tutoriales y ayudas sobre FBML es mucho mayor que sobre HTML en cuanto a funciones específicas de Facebook se refiere.

Por otra parte, utilizamos la API actual de Facebook y no la antigua a pesar de estar mucho más documentada al llevar más tiempo en uso y ser más permisiva en ciertos aspectos. Con ello nos adaptamos a las herramientas más actuales de Facebook.

8.4.- SDK & tools (SDK y herramientas).

El último apartado se subdivide en cuatro secciones. Con sólo observar el título de las tres primeras, comprobaremos que no son útiles para nuestra aplicación puesto que hacen referencia a los SDK de otros lenguajes y no del que hemos elegido.

La última sección, llamada “tools” (herramientas), reclama sin embargo nuestra atención, ya que las herramientas que se nos proporcionan nos ayudan a depurar ciertos errores, así como algunas otras funcionalidades.

Sin embargo, no debemos centrarnos en este caso sólo en las secciones en las que se subdivide SDKs & Tools. Si nos fijamos en la parte central, observamos que hay una subsección más, que no aparece en el desplegable.

Ésta es precisamente la que nos interesa.

8.4.1.- PHP SDK.

Llegados a este punto podemos plantearnos: ¿por qué al SDK de PHP no se le dedica una subsección como al resto?

Al comienzo de la creación de esta aplicación, el SDK de PHP figuraba en esta sección pero, como hemos señalado, Facebook está en constante cambio, y parece ser que ahora se prefiere que los desarrolladores utilicen JavaScript también para programar las aplicaciones. Por nuestra parte, como comenzamos programándola en PHP y tenemos documentación y manuales sobre este lenguaje, seguiremos adelante con él.

Pulsamos sobre PHP SDK y nos redirige a otra página de descargas.

En ella encontramos cuatro archivos distintos para descargarnos. El único que necesitamos realmente es el `src/`, pero nos puede ser también de interés la descarga de los ejemplos y, si lo consideramos conveniente, también de los test. Hay muchos sitios donde se nos habla de testear nuestra aplicación, sin embargo no hemos necesitado hacer uso de ellos.

Una vez finalizada la lectura inicial de la documentación, lo que nos ofrece una perspectiva general de sus contenidos, ya podemos empezar a estructurar el proyecto.

CAPÍTULO 9

Estructurando nuestra aplicación.

Antes de empezar la programación, y sabiendo que, conforme vayamos avanzando en el desarrollo de la aplicación nos encontraremos con problemas que forzarán cambios en nuestro diseño inicial, es conveniente contar con un breve guión o estructura de la misma. Para poder abarcar todo el proyecto, y facilitar la tarea, se han planteado los siguientes módulos:

- Servidor web.
- Base de datos.
- Código PHP de la aplicación.

En los siguientes capítulos iremos desgranando cada sección, con una atención especial a la última, que será la que presente mayor complejidad. Veamos ahora los aspectos más relevantes de cada una de ellas.

9.1.- Servidor web.

Como se ha expuesto en otro capítulo, existen distintos procedimientos para obtener un servidor web. La primera opción, más laboriosa pero sin coste económico alguno, es la de convertir nuestro propio ordenador en un servidor web. Aunque en un principio barajamos esta posibilidad, quedó descartada por las dificultades para mantener un equipo conectado de manera permanente.

Ya que queríamos que la aplicación estuviese siempre operativa, optamos por usar un dominio que proporcionase también un servidor web. Puesto que ya disponíamos del acceso a uno, esto no supuso mayores problemas.

Para acceder a este servidor necesitamos algún gestor que nos permita hacerlo fácilmente. En nuestro caso usamos FileZilla, un programa gratuito y fácil de usar.

La figura 9.1 representa una captura de pantalla del programa donde señalamos las partes más importantes.

- Número 1: este botón es un desplegable. En él deberemos introducir los datos que nos proporcione nuestro dominio para conectarnos a nuestro servidor. Podremos guardar la conexión para no tener que introducir cada vez el nombre del servidor y la contraseña. Una vez conectados, nos aparecerá el contenido alojado en nuestro servidor en la zona 3.
- Número 2: esta sección es el árbol de carpetas de nuestro PC. Aquí podremos navegar por ellas. Su contenido completo aparecerá en la zona 4.
- Número 3: árbol de carpetas de nuestro servidor, cuyo contenido completo aparecerá en la zona 5.
- Número 4: archivos contenidos en la carpeta de nuestro ordenador a la que hemos accedido a través de la zona 2. Los archivos que figuren aquí serán los que queremos subir a nuestro servidor. Para ello, tan sólo tendremos que seleccionar y arrastrar hacia la zona 5.
- Número 5: archivos contenidos en nuestro servidor. En este espacio es hacia donde arrastraremos los archivos que queremos subir a éste, así que debemos asegurarnos que estamos en la carpeta correcta.
- Número 6: esta zona nos avisa simplemente si la transferencia de archivos de nuestro ordenador al servidor se ha realizado con éxito.

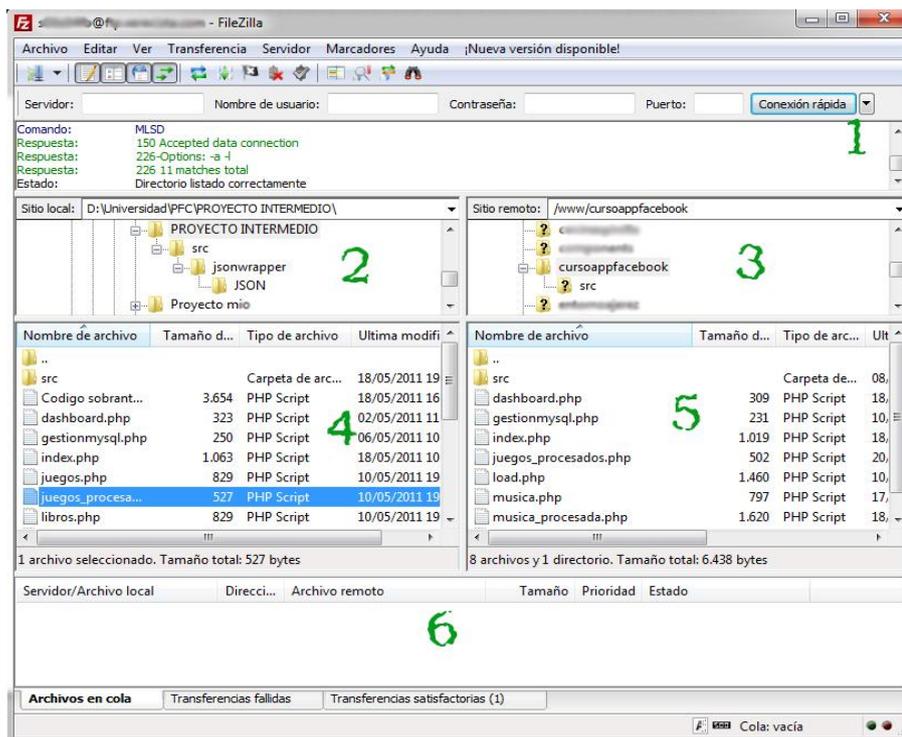


Figura 9.1

La carpeta que contendrá los archivos de nuestra aplicación ha de situarse de manera que tenga como directorio padre al “www”, y su nombre debe coincidir con el que le hemos dado al directorio en la ruta de “Canvas URL”.

Una vez explicado cómo funciona brevemente el gestor del servidor, pasaremos a ver cómo gestionar la base de datos. .

9.2.- Base de datos.

Al igual que lo comentado en el apartado sobre el servidor web, lo primero que debemos saber es cómo acceder y gestionar la base de datos.

Debemos volver a los datos del registro de nuestro dominio, para obtener la dirección del panel de control de éste.

Una vez dentro, tendremos que acceder al apartado de base de datos. En el caso de nuestro dominio, ésta se gestionaba mediante phpMy Admin.

Este gestor online nos permitirá hacer prácticamente cualquier tipo de operación sobre una base de datos.



Figura 9.2

Como nuestro servidor puede alojar más de una base de datos y puede haber más de un usuario con posibilidad de acceder a éste, debemos crearnos un usuario (en el caso de que no lo tengamos), y una base de datos para nuestra aplicación accediendo a la sección “MySQL® Base de Datos”. Al usuario le daremos los permisos que creamos oportunos.

Las mencionadas operaciones pueden hacerse de manera más sencilla pulsando en la sección “asistente de MySQL®”.

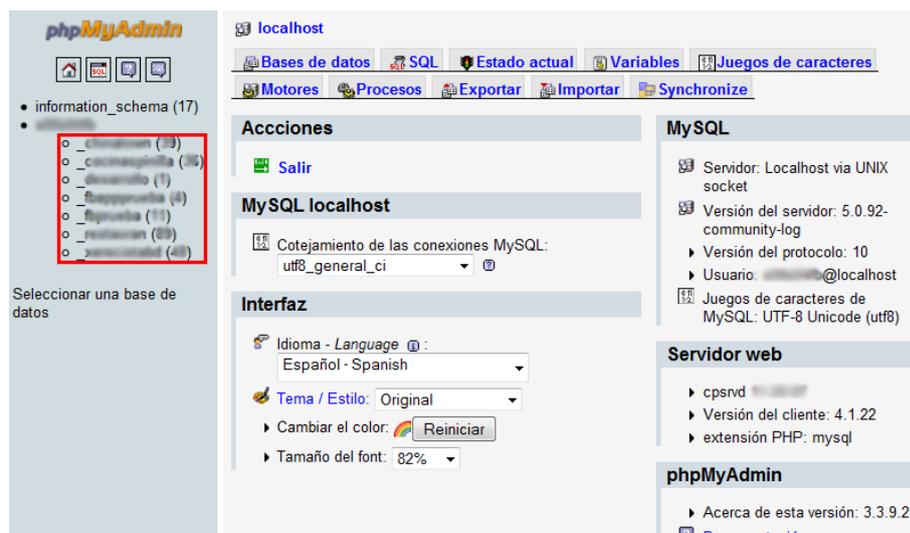


Figura 9.3

Una vez creada y teniendo los permisos necesarios, ya podemos proceder a diseñar las tablas que contendrán nuestra base de datos. Para ello accedemos ahora a la sección phpMy Admin, donde nos encontraremos en una página web con varias herramientas.

Como indicamos en la figura 9.3, en la parte señalada por el recuadro rojo se encuentran las distintas bases de datos que hay en nuestro servidor entre las que se encontrará la que acabamos de crear en el paso

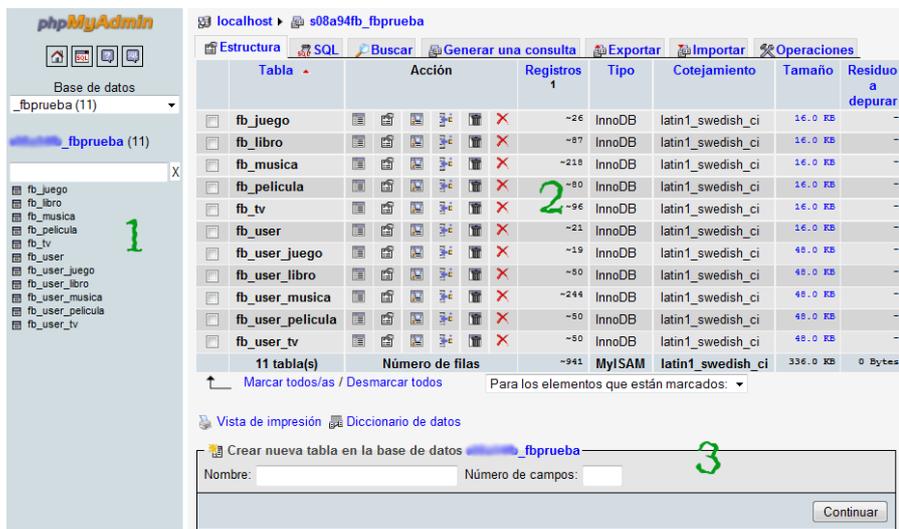


Figura 9.4

anterior. Pulsamos sobre ella y accedemos a una nueva página.

Dentro de cada base de datos, podemos crear el número de tablas que nos interesen.

- Número 1: en este apartado aparecerán cada una de las tablas creadas dentro de nuestra base de datos. Para acceder a ellas tan sólo pulsaremos sobre el nombre de la tabla, y se nos mostrará el contenido de ésta.
- Número 2: aquí también aparecen cada una de las tablas de la base de datos, pero con cierta información, y con la posibilidad de realizar algunas operaciones sobre ellas, como eliminarlas, vaciar su contenido o ver el número de registros que contienen.
- Número 3: en esta sección, podremos crear una nueva tabla dentro de nuestra base de datos.

En nuestro caso, necesitamos una primera tabla para almacenar los datos de cada usuario que acceda a la aplicación, a la que denominaremos “**fb_user**”. Contendrá un campo con el nombre del usuario y otro con el **id_usuario** que asigna Facebook automáticamente. Los campos **id_** debemos declararlos de tipo double, ya que si lo declaramos de tipo int, por mucha longitud que le demos, tendrá problemas con el tipo de Facebook. Este último campo es el que marcaremos además como **Primary Key** de la tabla.

Crearemos, además, una segunda tabla en la que se registrarán los datos de los grupos escuchados, que también contendrán un **id_artista**, y un nombre de artista. Como en la anterior, usaremos como **Primary Key** el **id_artista**.

Por último, necesitaremos una tercera tabla que relacione a los usuarios con los artistas que les gustan, la denominaremos **fb_user_musica**. Hay que tener en cuenta que cada usuario puede tener muchos artistas asignados y viceversa. Esta tabla contendrá tanto el campo **id_artista**, como el **id_usuario**. Para que la tabla **fb_user_musica** esté relacionada con las tablas **fb_user** y **fb_musica**, tendremos que declarar en la tabla **fb_user_musica** que el campo **id_user** sea clave foránea de la **fb_user**, y el campo **id_artista** sea clave foránea de la tabla **fb_musica**.

Tenemos que tener cuidado al crear las tablas puesto que, como motor de almacenamiento predefinido, **phpMy Admin** usa **MyISAM**. Este formato no nos deja hacer ciertas operaciones con las tablas, entre ellas, crear claves foráneas. Al declarar las tablas, pondremos como motor de almacenamiento **InnoDB**, que sí nos lo permitirá, así como el resto de operaciones que necesitaremos hacer a lo largo del proyecto.

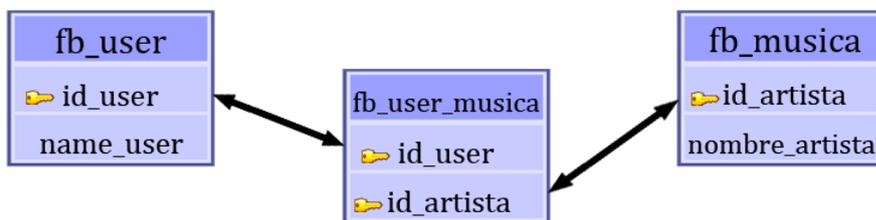


Figura 9.5

9.3.- Código PHP de la aplicación.

El código de la aplicación se estructurará en los siguientes apartados:

- Creación de objeto de Facebook y comprobación de sesión.
- Conexión con la base de datos.
- Procesamiento del usuario.
- Selección de criterio y transmisión de datos.
- Resultado final.

Además, en todas las aplicaciones de Facebook podemos hablar de parte común y parte específica. Obviamente, la parte común es la más sencilla de programar, puesto que hay ejemplos que podremos tomar como base.

CAPÍTULO 10

Empezando a programar.

Antes de empezar, debemos aclarar que este documento no es un manual de PHP, y que, por lo tanto, no se explicará con detalle la estructura del lenguaje, sin embargo, sí nos ocuparemos de unas nociones básicas para aclarar algunas dudas.

Es probable, que el programador que quiera desarrollar una aplicación de Facebook haya tenido cierto contacto con lenguajes interpretados, además de los compilados. Si no es así, quien empiece a usarlos, comprobará pequeñas diferencias entre ellos.

Centrándonos en el lenguaje en el que está realizado nuestro proyecto, PHP, hay dos cosas básicas que debemos saber si no estamos aún familiarizados con él.

Por un lado, tenemos que tener en cuenta lo dicho anteriormente, es decir, PHP es un lenguaje interpretado, por lo que, para ir comprobando los cambios que hacemos en nuestro código no podremos hacerlo directamente en el compilador, si no que tendremos que subir nuestro código al servidor, y probarlo desde la página de nuestra aplicación directamente.

Esta tarea resulta a veces un poco tediosa, puesto que al mínimo cambio, para comprobar sus efectos, tenemos que: guardar el documento, actualizar el FileZilla, subir el archivo al servidor y, finalmente, actualizar la página donde hemos hecho el cambio. En cada uno de los pasos anteriores no se tarda más de unos segundos, sin embargo, cuando hablamos de un

proyecto como este, donde los cambios y errores pueden ser frecuentes, el tiempo invertido en ello puede ser considerable.

Por otro lado, debemos saber cómo se ejecuta nuestro código, es decir, si todo debe estar en un mismo archivo, si podemos crear varios archivos que se llamen unos a otros, cómo o dónde empieza el código a ejecutarse, etc. En el caso de PHP, todo programa empieza a ejecutarse a partir de un archivo principal, que siempre recibe el nombre de `index.php`. No es necesario que todo el código esté dentro del archivo, sino que podremos crear distintos archivos `.php` que serán llamados a través de nuestro archivo principal, `index.php`, o de otros que sean llamados por éste.

Despejadas las dudas acerca del funcionamiento básico del lenguaje, y habiéndonos documentado a través de distintos manuales sobre su estructura y funciones, llega el momento de la codificación. A la hora de estructurarlo, conforme vayamos avanzando en el desarrollo, iremos reagrupando código y creando nuevos archivos que tengan características en común, o que se refieran a una sola página de nuestra aplicación.

El archivo `index.php` estará prácticamente vacío, y tan sólo será el punto inicial de nuestra aplicación, a través del cual llamaremos a distintos ficheros que hagan realmente las funciones de la aplicación.

10.1.- Creación y gestión de sesión en Facebook.

Si estamos familiarizados con lenguajes orientados a objetos, podríamos decir que, lo primero que debe hacer cualquier programador de aplicaciones para Facebook es crear un objeto de la clase Facebook.

Si hemos seguido los pasos anteriores señalados de este documento, en este momento tendremos en nuestro ordenador el **SDK** que proporciona Facebook para PHP. Es una carpeta con el nombre de **src**, que, en el momento de su descarga, contiene tres archivos: una carpeta llamada **jsonwrapper**, un archivo con el nombre de **fb_ca_chain_bundle.crt** y otro archivo denominado **facebook.php**. Esto puede no ser así, ya que es posible que Facebook cambie el contenido del SDK en un futuro.

El archivo que deberemos incluir en nuestro código, para que podamos hacer uso de las funciones proporcionadas, será el llamado **facebook.php**.

Vamos a ir creándonos, como hemos dicho anteriormente, un archivo **.php** por cada función bien diferenciada de Facebook que elaboremos.

Puesto que vamos a gestionar una nueva sesión de Facebook, donde se cargarán datos del usuario, crearemos un nuevo archivo con el nombre de **load.php**. Este archivo no sólo contendrá el inicio de sesión de Facebook, sino también la conexión con la base de datos, así como la información básica de nuestra aplicación necesaria a lo largo del proyecto.

Tras incluir nuestro nuevo fichero, **load.php**, debemos crear la variable `$facebook`, a través de la cual ya podremos obtener los datos de sesión. Para crearla, debemos escribir el siguiente código:

```
$facebook = new Facebook(array(  
    'appId' => $appId,  
    'secret' => $appSecret,  
    'cookie' => true,  
));
```

Figura 10.1

Los valores que introduciremos en 'appId' y en 'secret' son, respectivamente, el número de identificación y el número secreto de identificación de nuestra aplicación. Para acceder a estos datos, debemos dirigirnos a la aplicación de **Desarrolladores**. Una vez allí, iremos a nuestras aplicaciones, y pulsaremos sobre la que estemos desarrollando. Accederemos a una pantalla como la representada en la figura 10.2, donde aparecerá la información que necesitamos.

Las variables `$appId` y `$appSecret` contendrán, por tanto, el valor App ID y el App Secret respectivamente.

Ahora, podremos obtener información de la sesión, así como del usuario que la ha iniciado. Con ello conseguimos controlar los accesos a nuestra aplicación, a la que sólo podrán acceder aquellos que hayan iniciado sesión en Facebook. En caso de que algún usuario intente acceder a la aplicación sin haber iniciado sesión, se producirá un error y será redirigido a una página para que acceda mediante sus datos a Facebook.

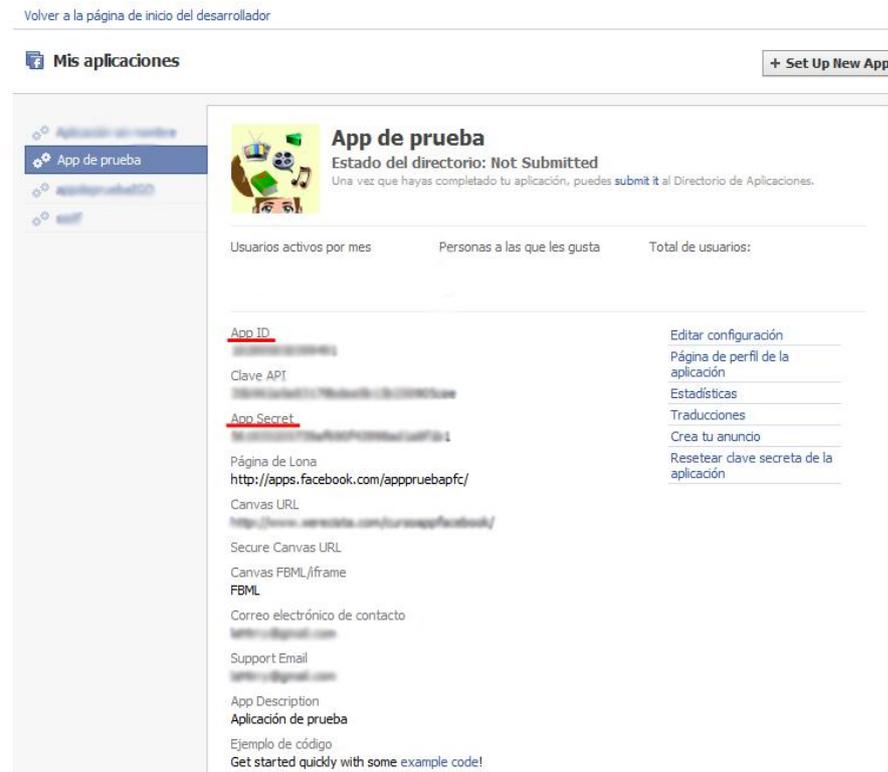


Figura 10.2

El código utilizado (figura 10.3), hace uso del **SDK** de facebook para obtener los datos de sesión, así como del usuario que accede a la aplicación.

Además, aquí haremos uso ya del lenguaje específico de Facebook **FBML**, que como dijimos al principio del documento, se puede combinar con PHP, al igual que se hace con **HTML**. El tag `fb:redirect url`, nos redirigirá, en caso de que el usuario que esté accediendo a la aplicación no seamos nosotros, a la dirección que le indiquemos.

La dirección a la que queremos redirigir al usuario en caso de que no haya iniciado sesión será una página que contenga dicho inicio.

```
$facebook = new Facebook(array(
    'appId' => $appId,
    'secret' => $appSecret,
    'cookie' => true,
));

$session = $facebook->getSession();
$access_tok = $facebook->getAccessToken();
$me = null;

// Session based API call.
if ($session) {
    try {
        $uid = $facebook->getUser();
        $me = $facebook->api('/me');
    } catch (FacebookApiException $e) {
        error_log($e);
    }
}
?>

<?php if (!$me): ?>
<fb:redirect url="<?php echo $loginUrl; ?>" />
<?php endif; ?>
```

Figura 10.3

Con esto, ya habremos conseguido que el usuario acceda a nuestra aplicación. Sin embargo, como dijimos en un principio, necesitamos extraer la información de cada uno de los usuarios que acceda a la aplicación, por lo que es el momento de realizar la conexión con la base de datos.

10.2.- Conexión con la base de datos.

Realizar una conexión con una base de datos en PHP es una tarea bastante sencilla; el procedimiento está suficientemente explicado en numerosos tutoriales y páginas webs. Sin embargo, también podemos encontrarnos con pequeñas dificultades, sobre todo al identificar correctamente cada uno de los parámetros de las funciones.

Para hacer el código lo más limpio posible, crearemos un nuevo archivo, llamado **mysql.php**, que contendrá las funciones que gestionen la conexión con la base de datos. Estas funciones serán llamadas a través del archivo creado anteriormente, **load.php**, que como ya hemos señalado, contiene la información básica de nuestra aplicación, y entre ella, la requerida para realizar la conexión.

Este patrón es común a cualquier conexión a una base de datos **mysql**.

Vamos a analizar a continuación cada uno de los parámetros de la función.

- `$db_user`: nombre del usuario que ha creado, o se encarga de gestionar la base de datos que vamos a usar.
- `$db_pass`: contraseña del usuario con acceso a la base de datos, es decir, del `$db_user`.
- `$db_name`: nombre de la base de datos que vamos a utilizar para almacenar la información de nuestra aplicación.

```
function conexion ( $db_user, $db_pass, $db_name )
{
    $link = mysql_connect('localhost', $db_user, $db_pass);
    if (!$link)
    {
        die('No pudo conectarse: ' . mysql_error());
    }
    select ($db_name,$link);
}

function select ($db_name,$link)
{
    if (mysql_select_db( $db_name,$link))
    {
        echo "Acierto al conectar con la base de datos.<br>";
    }
}
```

Figura 10.4. Contenido del archivo `mysql.php`.

Ya tenemos listo tanto el acceso del usuario a la aplicación, como la conexión con la base de datos. El siguiente paso que deberemos realizar será el de proporcionar datos a nuestra aplicación para que sea funcional.

10.3.- Procesamiento del usuario.

Cada vez que un usuario accede a la aplicación, debemos comprobar si es la primera vez que accede a ésta o, por el contrario, lo ha hecho en ocasiones anteriores

En caso de ser la primera vez, se le solicitarán una serie de permisos que son necesarios para que los desarrolladores puedan acceder a cierta

información restringida de los usuarios (figura 10.5). La lista de permisos se encuentra en la sección de “**Authentication**”, dentro del apartado de la “**Documentación Core Concepts**”.

Para solicitar los permisos de la aplicación, Facebook hace uso del protocolo **OAuth 2.0**.



Figura 10.5

Una vez aceptados los permisos, se registrarán automáticamente sus datos, tanto su nombre como su número de identificador. Estos datos son campos del objeto `$me`. Además, se registrarán los artistas o intérpretes que aparezcan en sus gustos sobre música. Para ello nos crearemos dos tablas. Por un lado la tabla usuario, que contendrá tanto el número de identificación del usuario, como su nombre.

Aunque el campo “nombre” no tiene ninguna utilidad a la hora de añadir funcionalidad a nuestra aplicación, lo incorporaremos para comprobar qué usuarios son los que han entrado en ella con mayor facilidad.

Todos los datos almacenados se obtendrán, como hemos dicho antes, a partir del objeto `$me`. Para saber qué atributos tiene y cuáles vamos a necesitar debemos acceder nuevamente a la aplicación de Facebook **Desarrolladores**, y una vez allí, a la **API**, que se encuentra en la sección de **Documentación**, en el apartado **Core Concepts**. Seleccionaremos el objeto **user**, y obtendremos la información buscada. La ruta de acceso es la siguiente: <http://developers.facebook.com/docs/reference/api/user/>

Observamos que podemos extraer cualquier información que se nos ocurra de un usuario, siempre y cuando tengamos los permisos necesarios, pero como para nuestra aplicación sólo nos interesa algo muy concreto, vamos a centrarnos en ello.

Como hemos dicho, de cada usuario nos interesan su nombre y su id. Para obtenerlos accederemos a los campos `"id"` y `"name"` almacenados en el objeto `$me` (figura 10.6).

```
$sql = "INSERT INTO `fb_user` (`id_user`, `name_user`)
VALUES ('$me[id]', '$me[name]');";
$query = mysql_query($sql);
```

Figura 10.6

Si seguimos examinando los atributos del objeto usuario, veremos que dentro de la sección **Conexiones** está el atributo **music**. Éste contiene un **array** con el id, el nombre, la categoría y la fecha de creación de su contenido.

Accederemos a estos datos de manera similar a como hicimos con el objeto `$me`, y los almacenaremos tanto en la tabla **user_musica**, como en la tabla **música**, para que queden registrados todos los artistas con su nombre y su respectivo id. En este caso, la consulta será un poco distinta, puesto que no se trata de un solo elemento, sino de un **array** de elementos con sus respectivas características.

Para almacenarlos, simplemente nos tendremos que crear un pequeño bucle que recorra dicho **array** y acceda a cada uno de los campos del artista o intérprete determinado (figuras 10.7, 10.8 y 10.9).

```
$artists = $facebook->api('/me/music?fields=id,name');
```

Figura 10.7. Obtención de los artistas almacenados en el perfil del usuario.

```
foreach ($artists[data] as $artista) {  
    introduceartista($artista, $me);  
}
```

Figura 10.8

```
function introduceartista($artista,$me) {  
    $sql = "INSERT INTO `s08a94fb_fbprueba`.`fb_musica` (`id_artista`, `nombre_artista`)  
VALUES (`$artista[id]`, `$artista[name]`)";  
    $sql2 = "INSERT INTO `s08a94fb_fbprueba`.`fb_user_musica` (`id_artista`, `id_user`)  
VALUES (`$artista[id]`, `$me[id]`)";  
    $query = mysql_query($sql);  
    $query2 = mysql_query($sql2);  
}
```

Figura 10.9

En caso de no ser un nuevo usuario, la aplicación actualizará los datos. En realidad, no es una actualización como tal, sino que eliminamos datos existentes y los volvemos a introducir en la base de datos.

En principio, puede parecer que el reinscribir los gustos de un usuario cada vez que acceda a la aplicación es un gasto de recursos innecesario. Se podría pensar que es mejor comprobar primero si el usuario ha cambiado algo, y si es así, entonces actualizarlo.

Para realizar esta comprobación, tendríamos que recorrer la tabla de **usuarios_musica**, ir analizando una a una las entradas, y, si se han producido cambios, eliminar los datos y volverlos a introducir. Es muy probable que el hecho de recorrer la tabla ya suponga por sí solo un consumo de recursos similar al empleado en eliminar y volver a escribir, sin considerar el coste del proceso de actualización en el caso de que el usuario haya modificado sus preferencias.

Otra alternativa sería la de dar al usuario la oportunidad de actualizar sus gustos mediante la correspondiente opción desde la propia aplicación. Sin embargo, un usuario normal de Facebook probablemente no sepa con seguridad si ha cambiado o no sus preferencias, ya que en muchas ocasiones, con pulsar el botón “Me gusta” en algunas sugerencias aleatorias que hace Facebook, ya se añaden nuevos gustos al usuario en cuestión.

Una vez registrados los datos del usuario, sea por primera vez o por actualización, se accederá a la página principal de la aplicación.

10.4.- Página principal de la aplicación.

Para estructurar la página principal de la aplicación, hemos hecho uso de uno de los accesorios que nos proporciona Facebook, las tablas.

Las tablas de Facebook no son más que una serie de botones, a los que se podrá asignar una acción cuando los pulsemos. En nuestro caso, nos servirán para diferenciar la página principal de las de recomendación. Están realizadas con etiquetas **FBML**, y su implementación básica sería la mostrada en la figura 10.10.

```
$theTabs = <<<tabs
<fb:tabs>
  <fb:tab-item href="index.php" title='Bienvenidos' />
  <fb:tab-item href="pagina1.php" title='Botón 1' />
  <fb:tab-item href="pagina2.php" title='Botón 2' />
  <fb:tab-item href="pagina3.php" title='Botón 3' />
</fb:tabs>
tabs;
echo $theTabs;
```

Figura 10.10. Formación tabla definida en Facebook.

Físicamente, el código anterior nos mostraría una aplicación como la representada en la figura 10.11. Cada uno de los botones de la tabla tendrá asociada otra página, la referenciada en el campo **href** de cada etiqueta `<fb:tab-item />`, a la que nos redirigirá cuando pulsemos sobre el botón.

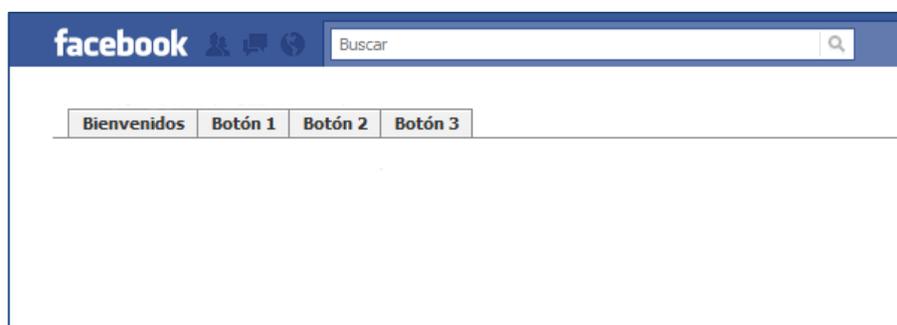


Figura 10.11

10.5.- Recomendación de música.

Tras acceder a la aplicación, debemos decidir qué queremos que nos recomiende, y cómo queremos que lo haga. Es decir, si queremos recomendaciones sobre música, libros, películas, programas y series de televisión o juegos.

Una vez decido, debemos acceder al apartado correspondiente, así, si queremos que se nos recomiende un artista musical, deberemos elegir el apartado “Música”, o si, por ejemplo, lo que queremos es descubrir un nuevo libro deberíamos acceder al apartado “¿Qué leo?”

Con respecto al cómo queremos que la aplicación nos recomiende lo escogido anteriormente, se refiere en realidad, a qué criterio queremos que siga la aplicación a la hora de obtener sus resultados.

Los criterios en los que se basa la aplicación, son tres:

- Criterio de Selección con Pesos.
- Criterio de Selección.
- Criterio “Invariable”.

Estos tres criterios han sido detalladamente descritos en la sección 6 de este mismo documento (*véase página 20*).

Si el usuario ha elegido como criterio el “Criterio invariable”, en el que no es necesario que escoja ningún intérprete, deberá seleccionar la opción “Criterio 3” y será redirigido a una nueva página cuyo funcionamiento se explicará más adelante.

En el caso de que el usuario haya escogido el “Criterio de Selección”, o el “Criterio de Selección con Pesos”, se le mostrará la lista de artistas musicales que tiene en su perfil. Podrá seleccionar el número que quiera de ellos, según desee que la búsqueda sea más o menos precisa, y una vez seleccionados, enviará la información para que se realice la recomendación.

10.6.- Formulario.

Puesto que necesitamos que el usuario seleccione una serie de artistas de una lista y luego envíe la información correspondiente, necesitamos usar un formulario.

Los formularios en **FBML** tienen prácticamente la misma sintaxis que los que se usan en **HTML**, así que nos resultarán bastante sencillos si tenemos ciertas nociones de cómo hacerlo en **HTML**.

Para saber cómo hacer un formulario en **FBML** podemos acceder a la API de FBML, denominada ya “Legacy FBML”. Aunque vienen explicados cada uno de los posibles tags y funciones de FBML, para crear un formulario nos será más sencillo si buscamos tutoriales dentro del propio foro de Facebook Developers debido a la falta de código de ejemplo dentro de la propia API.

Este foro nos resultará de mucha ayuda, ya que normalmente obtendremos respuesta a las dudas que planteemos, siempre en inglés, o que, en caso de no resolvérnoslas directamente, nos redirijan a manuales y tutoriales que sí puedan hacerlo.

Hay muchas formas de enviar la información a través de formularios, así que en primer lugar hay que decidir qué tipo de formulario queremos hacer. En nuestro caso optaremos por una serie de botones circulares para marcar todos los artistas que queramos y nos dé la posibilidad de seleccionar, que en este caso será cada uno de los que tenemos agregados a nuestros gustos. Al lado de cada botón, queremos que aparezca el artista al que se hace referencia, con su imagen, para que sea más fácil para el usuario y a la vez más atractivo. Debemos puntualizar que no siempre será posible que aparezca una imagen correcta del elemento que estemos seleccionando, puesto que no es Facebook quien rellena esta información, si no el primer usuario que registró dicho artista en sus gustos musicales, y creó automáticamente una página sobre él.

El tipo de botón que vamos a usar en nuestra aplicación no viene definido directamente en FBML, por lo que no vamos a poder usar el tag `<fb:editor-... />`. Vamos a crear una pequeña clase que funcionará igual que aquella, pero que nos permitirá usar botones radiales (figura 10.10).

```
<tr><th class="label"><label>
<?php echo "<fb:profile-pic uid=\"\$artist[id]\" width=\"50\"
height=\"50\" /><fb:name uid=\"\$artist[id]\" />";?>
</label></th><td>
<input type="radio" name="<?php echo "field".$i?>"
value="<?php echo \$artist[id]?> " />
</td></tr>
```

Figura 10.12

Puesto que debemos asignar un botón a cada uno de los elementos (intérpretes, libros, etc.) de nuestra lista, simplemente alojaremos el código de la figura 10.12 en un bucle obteniendo el resultado deseado. En un principio, sin añadirle ningún tipo de característica para alinearlos, obtenemos lo representado en la figura 10.13.



Figura 10.13

Tras seleccionar los artistas, en el supuesto de los dos primeros criterios, o sin seleccionar ninguno, si escogemos el último, debemos dirigirnos al botón desplegable de “seleccionar criterio”.

Criterios :

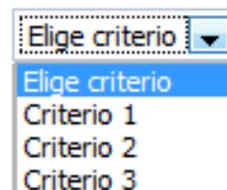


Figura 10.14

En esta ocasión hemos escogido un cuadro desplegable para enviar la información. La recepción de información será prácticamente igual que la de los botones radiales, aunque hay una pequeña diferencia.

En el caso de la selección de artistas, podíamos decidir no elegir ninguno, siempre y cuando el criterio a aplicar fuese el último, sin embargo aquí no es posible, ya que necesitamos saber qué criterio aplicar a la hora de presentar la recomendación al usuario.

En el supuesto de que el usuario actual, intencionadamente o por descuido, no cumplimentase la selección de criterio a seguir, (figura 10.15), escogeremos como predeterminado el “Criterio invariable” ya que, para la recomendación, no hace uso de ningún otro parámetro.

```
switch ($criterio) {
  case 1:
    $devueltos = criterio1($array_artistas_seleccionados, $me, $numartdev, "musica");
    break;
  case 2:
    $devueltos = criterio2($array_artistas_seleccionados, $me, $numartdev, "musica");
    break;
  case 3:
    $devueltos = criterio3($me, $numartdev, "musica");
    break;
  default:
    $devueltos = criterio3($me, $numartdev, "musica");
}
```

Figura 10.15

También hemos añadido la opción de que el usuario seleccione el número de artistas que quiere que le sean recomendados, hasta un máximo de cinco (figura 10.16). En el supuesto de que el usuario no escoja ningún valor, el número de artistas devueltos será uno. La recepción de los valores enviados será igual que en los dos anteriores casos.

Es posible que, aunque el usuario haya seleccionado un determinado número de artistas, la aplicación le devuelva un número distinto, siempre menor que el seleccionado. Esto puede deberse a que no existen artistas o intérpretes suficientes que cumplan con el criterio seleccionado por el usuario.

Número de artistas devueltos:

Selecciona número de artistas ▾

Selecciona número de artistas

1

2

3

4

5

Figura 10.16

10.7.- Envío de la información.

Para que la aplicación ejecute el algoritmo de recomendación es necesario remitir, de algún modo, la información proporcionada por el usuario. Recurriremos para ello a un método muy parecido al usado en HTML.

```
<fb:editor-buttonset>
  <fb:editor-button value="Submit"/>
  <fb:editor-cancel value="Cancel" href="input_form.php"/>
</fb:editor-buttonset>
```

Figura 10.17

Una vez hecha toda la selección, incluyendo el criterio y el número de artistas devueltos, tan sólo pulsaremos en el botón “realizar consulta” y enviaremos la información a una nueva página que la procesará. Para ello deberemos indicar, mediante código FBML (figura 10.17), que cuando se realice la acción de **submit**, se redirija a la página deseada (figura 10.18).

```
<fb:editor
  action="musica_procesada.php?"
  labelwidth="150">
```

Figura 10.18

Crearemos una nueva página, llamada “musica_procesada.php”, donde recogeremos la información enviada, y aplicaremos el algoritmo de criterio según haya escogido el usuario de la aplicación.

10.8.- Procesando la información.

Como hemos dicho, FBML es muy similar a HTML, y el modo de extraer la información de los formularios también.

En la anterior página, se nos redirigía, con la información seleccionada, a “musica_procesada.php”. Dependiendo de si el usuario ha escogido el “Criterio Invariable” o alguno de los otros dos, tendremos que proceder de forma distinta.

Vamos a explicar primero el más sencillo, el “Criterio invariable”. En este caso, no tenemos que recoger información de los artistas seleccionados por el usuario, puesto que no hay. Los únicos campos que

tiene que marcar son el de “Criterio” y el de “Número de artistas devuelto”.

Para obtener esta información, usaremos las funciones representadas en la siguiente figura.

```
if (isset($_REQUEST['crit'])){
    $criterio = html_entity_decode( $_REQUEST['crit'], ENT_QUOTES);
}

if ($criterio == 1 || $criterio == 2) {
    for ($i = 1; $i <=$numero; $i += 1) {
        if (isset($_REQUEST['field'.$i])) {
            $array_artistas_seleccionados[$j] =
                html_entity_decode( $_REQUEST['field'.$i], ENT_QUOTES);
            $j +=1;
        }
    }
}

if (isset($_REQUEST['numart'])){
    $numartdev = html_entity_decode( $_REQUEST['numart'], ENT_QUOTES);
    if($numartdev == 0)
        $numartdev = 1;
}
```

Figura 10.19

En caso de que el usuario haya escogido el Criterio 1 o el Criterio 2, tendremos que almacenar en un *array* los valores que se nos han enviado desde la página anterior, en este caso, “musica.php”.

Para ello, debemos alojar en un bucle la comprobación de cada campo de los botones, si se han enviado o no, almacenar los datos (figura 10.19), y pasárselos como entrada al criterio seleccionado (figura 10.15).

Llegados a este punto, ya tendremos toda la información necesaria para ejecutar el algoritmo correspondiente a cada criterio, pero antes de analizar más detenidamente cada uno de ellos, vamos a explicar otra función que se ejecuta en “musica_procesada.php”.

10.9.- Difundiendo la aplicación.

Para dar a conocer nuestra aplicación a través de Facebook, usaremos una función llamada “publicaMuro” que se ejecutará cada vez que obtengamos los resultados de la recomendación. Ésta publicará automáticamente en nuestro tablón que estamos usando la aplicación,

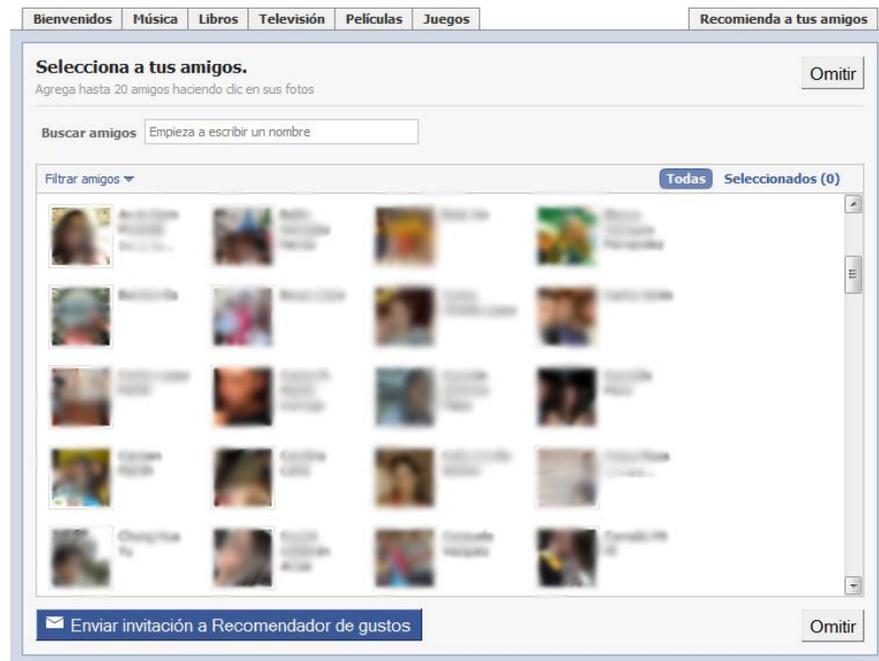


Figura 10.22

CAPÍTULO 11

Implementación de criterios.

11.1.- Primer criterio – Criterio de selección con peso.

El primer criterio, se basa, como hemos explicado en el apartado 6.1 más detalladamente, en que el usuario de la aplicación selecciona una serie de artistas o intérpretes musicales que quiere tomar como base para la recomendación.

Seguidamente, se selecciona a los usuarios que incluyan en su perfil a alguno de los artistas elegidos por quien solicita la recomendación. Tras esto, se adjudica un peso a cada uno de los usuarios seleccionados según el porcentaje de gustos que comparta con el solicitante. Una vez asignados los pesos a los “recomendadores”, y basándonos en ellos, pasamos a adjudicarle peso a los artistas que se encuentran entre las preferencias de los “recomendadores”. Por cada “recomendador” al que le guste un determinado intérprete, se le sumará a dicho artista el peso del recomendador. Una vez acabado con todos los intérpretes, escogeremos a los de más peso, quienes compondrán la recomendación.

Para diseñar este algoritmo hemos combinado PHP y SQL. Pretendemos realizar todo el cálculo en una sola consulta de SQL, puesto que, además de intentar optimizar el resultado, también tratamos de conseguir que el tiempo que invierta en obtenerlo sea mínimo. Puesto que lo que más suele tardar en estos casos son los accesos a la base de datos, procuraremos que sean los menos posibles.

Vamos a explicar por partes la primera consulta, que nos devolverá como resultado los intérpretes que les gustan a los “recomendadores” ya ordenados por el peso adjudicado según los criterios explicados en los párrafos anteriores.

```

$sql = "SELECT t4.`id_artista` , mus.`nombre_artista` ,
SUM( T_RECOMENDADORES.`PESO_RECOMENDADOR` ) AS PESO_ARTISTA
FROM `s08a94fb_fbprueba`.`fb_user_musica` t4, `s08a94fb_fbprueba`.`fb_musica` mus,

```

Figura 11.1

Mediante el código representado en la figura 11.1, seleccionamos los **id_artista** de la tabla que relaciona a los usuarios, y el nombre de los artistas de la tabla **id_musica** que cumplen una característica que describiremos a continuación, realizando, además, la suma de los pesos adjudicados a los “recomendadores” para calcular el peso final de los artistas.

```

(SELECT t1.`id_user` , COUNT( t1.`id_user` ) AS PESO_RECOMENDADOR
FROM `s08a94fb_fbprueba`.`fb_user_musica` t1,
`s08a94fb_fbprueba`.`fb_user_musica` t2,

```

Figura 11.2

El código representado en la figura 11.2, es el inicio de la subconsulta que devuelve los pesos de los “recomendadores”, según el número de artistas coincidentes con el usuario buscador.

```

(SELECT t3.`id_user` FROM `s08a94fb_fbprueba`.`fb_user_musica` t3 WHERE";
for($i=0;$i<count($array_artistas_seleccionados);$i++){
    if ($i == 0)
        $sql.=" t3.`id_artista`= $array_artistas_seleccionados[$i]";
    else
        $sql.=" OR t3.`id_artista`= $array_artistas_seleccionados[$i]";
}

$sql.= " GROUP BY t3.`id_user`)T_AUX WHERE t2.`id_user` = $me[id] AND
t1.`id_user` <> t2.`id_user` AND t2.`id_artista` = t1.`id_artista`
AND t1.`id_user` = T_AUX.`id_user`
GROUP BY t1.`id_user`
ORDER BY t1.`id_user`
)T_RECOMENDADORES
WHERE t4.`id_user` = T_RECOMENDADORES.`id_user`
AND t4.`id_artista` = mus.`id_artista`
GROUP BY t4.`id_artista`
ORDER BY `PESO_ARTISTA` DESC";

```

Figura 11.3

En el código representado en la figura 11.3, se inicia la subconsulta que devuelve los “recomendadores” que tienen al menos uno de los artistas. Puesto que es una consulta dinámica, ya que el número de intérpretes que puede seleccionar el usuario es un parámetro variable, debemos combinar código PHP, para añadir una restricción en la consulta por cada artista que se haya seleccionado.

Además, al final de la consulta, ordenaremos los intérpretes por peso, para que nos sea mucho más fácil quedarnos con el número que queramos de ellos.

Una vez terminada la consulta, hemos obtenido una lista con los artistas preferidos por nuestros “recomendadores”, ordenados de mayor a menor según su peso. Tan sólo tendremos que comprobar que los intérpretes que queremos devolver no se encuentran ya entre nuestros gustos, y tendremos finalizado el primer criterio. Para verificarlo, tendremos que hacer otra consulta y comprobar si el **id_artista** que vamos a recomendar coincide con alguno de los que ya se encuentran relacionados con nuestro **id_user** en la tabla **fb_user_musica** (figura 11.4).

```
while (($row = mysql_fetch_array($result, MYSQL_NUM)) & $numartdev > 0) {

    $sql2 = mysql_query ("SELECT * FROM `s08a94fb_fbprueba`.`fb_user_musica`
    WHERE `fb_user_musica`.`id_user`= $me[id] AND
    `fb_user_musica`.`id_artista` = $row[0]");

    if (mysql_num_rows($sql2) == 0) {
        echo "<fb:profile-pic uid=\"\$row[0]\" width=\"50\" height=\"50\" />
        <fb:name uid=\"\$row[0]\" />";
        $numartdev -=1;
    }
}
}
```

Figura 11.4

11.2.- Segundo criterio – Criterio de selección.

El segundo criterio, explicado más detalladamente en el punto 6.2, parece, en principio, muy similar al primero. Al comienzo debemos proceder en ambos de igual modo, es decir, seleccionando una serie de artistas a los que queremos que se parezca el resultado final. Sin embargo su funcionamiento es distinto.

En el anterior criterio escogíamos todos los usuarios que tuvieran alguno de los artistas seleccionados entre sus gustos; en este caso los seleccionados serán quienes tengan entre sus preferencias a todos los elegidos por el usuario actual.

Una vez obtenida la lista de los recomendadores, debemos buscar entre ellos aquellos artistas que tengan en común y, en caso de que los haya, devolverlos como resultado.

Debido a que esta aplicación es completamente dependiente del número de usuarios que haya accedido a ella, y por tanto estén registrados en la base de datos, este criterio ha tenido que ser ampliado. Por una parte, es posible que no existan muchos usuarios que tengan entre sus gustos a todos los artistas que el petionario haya seleccionado, sobre todo si esta recopilación es muy numerosa. Por lo tanto en vez de escoger sólo a los usuarios que contengan a todos los artistas seleccionados, tomaremos gradualmente a aquellos que tengan el máximo número de ellos.

Dado que existe la posibilidad de no haya ningún artista en común entre todos los “recomendadores”, modificaremos la selección en el sentido de escoger aquellos artistas que aparezcan con más frecuencia.

A diferencia del primer criterio, ahora no se tiene en cuenta si los gustos de un “recomendador” se parecen mucho o poco a los del usuario de la aplicación.

Al igual que en el primero, hemos realizado este criterio combinando PHP y SQL.

La consulta que hemos realizado nos devolverá como resultado todos los artistas de los mejores recomendadores, ordenados según el número de recomendadores a quienes les gustan.

```
mysql = "SELECT `US_MUS`.`id_artista`, COUNT( `US_MUS`.`id_artista` ) as N_COINCIDENCIAS
FROM `s08a94fb_fbprueba`.`fb_user_musica` US_MUS,
{"
```

Figura 11.5

La figura 11.5 representa el comienzo de la consulta, donde se seleccionarán los **id_artista** de aquellos intérpretes que nos hayan devuelto las siguientes subconsultas y que sean los más populares entre los “mejores recomendadores” (figura 11.6).

El **select** más interno de esta consulta selecciona el número de artistas coincidentes con los seleccionados por el usuario que tienen los “recomendadores”.

El **select** que lo engloba, selecciona a los “recomendadores” cuyo número de artistas coincidentes con los indicados es igual al número máximo de los datos devueltos por el **select** interno.

```

SELECT t.`id_user` , COUNT( t.`id_user` ) AS NUM_ARTISTAS
FROM `s08a94fb_fbprueba`.`fb_user_musica` t
WHERE t.`id_user` != $me[id] AND t.`id_artista` IN ( ";

for($i=0;$i<count($array_artistas_seleccionados);$i++){
  if ($i == 0)
    $sql.=" $array_artistas_seleccionados[$i]";
  else
    $sql.=",$array_artistas_seleccionados[$i] ";
}

$sql.=") GROUP BY t.`id_user`
        HAVING NUM_ARTISTAS >= ALL(
        SELECT COUNT( t2.`id_user` ) AS NUM_ARTISTAS
        FROM `s08a94fb_fbprueba`.`fb_user_musica` t2
        WHERE t2.`id_artista` IN ( ";

for($i=0;$i<count($array_artistas_seleccionados);$i++){
  if ($i == 0)
    $sql.=" $array_artistas_seleccionados[$i]";
  else
    $sql.=",$array_artistas_seleccionados[$i] ";
}

$sql.=") GROUP BY t2.`id_user` )) T_MEJORES_RECOMENDADORES
WHERE `US_MUS`.`id_user` = `T_MEJORES_RECOMENDADORES`.`id_user`
GROUP BY `US_MUS`.`id_artista`
ORDER BY N_COINCIDENCIAS DESC";

```

Figura 11.6

Una vez obtenido el resultado de la consulta, que nos devolverá los artistas más frecuentes de los mejores “recomendadores” según este criterio, tendremos que volver a comprobar, como en el anterior, si los artistas que vamos a recomendar ya se encuentran entre nuestros gustos. Si no es así, los tomaremos como recomendación (figura 11.4).

11.3.- Tercer criterio – Criterio “invariable”.

El último criterio, cuya explicación detallada se encuentra en el punto 6.2 de este mismo documento, es muy parecido al primero. Se podría decir que es una versión más general de aquél.

En este caso, el usuario de la aplicación no tendrá que realizar selección alguna. La recomendación se basará en sus gustos en general y no en unos artistas seleccionados previamente con los que se quiera obtener algún parecido.

La recomendación seguirá el modo de adjudicación de pesos adoptado en el primer criterio. Por tanto, los “recomendadores”, que en este caso serán todos los usuarios de la aplicación, tendrán asignado un peso, dependiendo del número de artistas coincidentes con los del usuario actual de la aplicación. Posteriormente, se les asignará a los intérpretes un peso, que será el resultado de sumar el valor de los pesos de los “recomendadores” a los que le guste dicho artista. Una vez acabado con todos ellos, escogeremos a los de mayor peso, y ese será el resultado final.

El algoritmo de extracción de los pesos de los artistas según la base de datos es prácticamente igual al del primer criterio, con la diferencia de que no se exige a los “recomendadores” que tengan a ningún artista en concreto entre sus gustos. La consulta quedaría por tanto como la hemos representado en la figura 11.7.

```
mysql = "SELECT t4.`id_artista` , SUM( T_RECOMENDADORES.`PESO_RECOMENDADOR` )
AS PESO_ARTISTA
FROM `fb_user_musica` t4, (

SELECT t1.`id_user` , COUNT( t1.`id_user` ) AS PESO_RECOMENDADOR
FROM `fb_user_musica` t1, `fb_user_musica` t2

WHERE t2.`id_user` = $me[id]
AND t1.`id_user` <> t2.`id_user`
AND t2.`id_artista` = t1.`id_artista`
GROUP BY t1.`id_user`
ORDER BY t1.`id_user`
)T_RECOMENDADORES
WHERE t4.`id_user` = T_RECOMENDADORES.`id_user`
GROUP BY t4.`id_artista`
ORDER BY `PESO_ARTISTA` DESC";
```

Figura 11.7

CAPÍTULO 12

Ampliación del proyecto.

La idea principal de este proyecto era la de desarrollar un recomendador de música basada en las preferencias de nuestros amigos. Poco a poco, y conforme íbamos profundizando en las posibilidades que nos ofrecía Facebook, el proyecto ha ido cambiando ligeramente. No sólo ha habido que restringir ciertas partes, debido a los límites que imponía la propia red con respecto a la privacidad de los usuarios, sino que también hemos podido ampliar otras.

En nuestro perfil de Facebook podemos contar infinidad de datos acerca de nosotros mismos. Entre esa información personal, podemos configurar, como hemos visto durante todo este documento, nuestros gustos musicales, dentro de una serie de límites.

A las preferencias en el campo de la música de cada usuario se accede mediante un atributo del objeto **user**, como hemos podido observar en los capítulos anteriores. Sin embargo, algo que no hemos tenido en cuenta hasta ahora es que, además de los atributos mencionados anteriormente del objeto **user**, hay otros que hacen referencia a preferencias distintas de la música.

Entre otras cosas, el usuario puede dar datos sobre lo que le gusta en los deportes, equipos preferidos, creencias personales, gustos artísticos y de ocio. Entre estos últimos se encuentran sus predilecciones musicales. Probablemente no tenga mucho sentido ampliar el proyecto a la mayoría de los campos, como el de los deportes o las creencias filosóficas, sin

embargo sí podríamos usar nuestra aplicación para extenderla a los intereses sobre arte y ocio.

En esta sección, además de hablar de nuestros gustos musicales, podemos añadir nuestras películas favoritas, libros, juegos e incluso programas y series de televisión.

¿Por qué no hacer un proyecto más amplio, sólo con algo más de esfuerzo? Eso es lo que hemos pretendido al incluir los siguientes apartados.

12.1.- Recomendador de películas.

En el caso de que el usuario quiera que se le recomienden películas en lugar de música, tan sólo tendrá que acceder a la sección “Películas”.



Figura 12.1

El procedimiento y los criterios seguidos son similares a los expuestos anteriormente para el apartado de música. En este caso, en lugar de seleccionar artistas, se elegirán títulos de películas.

Para acceder a la información sobre los gustos en cuanto a películas del usuario actual, tendremos que hacerlo mediante el código que se muestra en la figura 12.2.

```
$peliculas = $facebook->api('/me/movies?fields=id,name');
```

Figura 12.2

Deberemos también ampliar nuestra base de datos para almacenar la nueva información. Las nuevas tablas que crearemos serán las de **fb_user_pelicula** y **fb_pelicula**. Ambas seguirán la misma estructura que las creadas al comienzo para almacenar los datos relacionados con música (figura 9.5), es decir, **fb_pelicula** contiene los campos **id_pelicula** y **nombre_pelicula**, mientras que la tabla **fb_user_pelicula** contiene los

campos **id_user** e **id_pelicula**, y relacionará la tabla **fb_pelicula** con la tabla **fb_user**.

Ahora, crearemos una función que siga el mismo comportamiento que la representada en la figura 10.7. Aunque, en este caso, la entrada y el lugar de almacenamiento de los datos serán distintos (figura 12.3).

```
function introducepelicula($pelicula, $me) {
    $sql = "INSERT INTO `s08a94fb_fbprueba`.`fb_pelicula` (`id_pelicula`, `nombre_peli`)
    VALUES ('$pelicula[id]', '$pelicula[name]')";
    $query = mysql_query($sql);
    $sql2 = "INSERT INTO `s08a94fb_fbprueba`.`fb_user_pelicula` (`id_pelicula`, `id_user`)
    VALUES ('$pelicula[id]', '$me[id]')";
    $query2 = mysql_query($sql2);
}
```

Figura 12.3

12.2.- Recomendador de televisión.

Si en lugar de artistas musicales o películas, el usuario de la aplicación quiere que se le recomienden programas o series de televisión, tendremos que acceder al apartado “Televisión”.

Para extraer los gustos sobre televisión del usuario actual, tendremos que acceder a su sección de gustos sobre televisión dentro de su perfil de Facebook, esto lo haremos mediante la siguiente función:

```
$tvs = $facebook->api('/me/television?fields=id,name');
```

Figura 12.4

Tenemos que proceder ahora igual que en los anteriores casos y almacenar la información con respecto a los gustos televisivos del usuario actual de la aplicación en nuestra base de datos.

El algoritmo que seguiremos será idéntico a los usados anteriormente y tan solo variará el contenido al que se accede y el lugar donde se almacena. Debemos crear por tanto otras dos tablas, **fb_tv** y **fb_user_tv**, que contendrán los mismos campos y funcionarán del mismo modo que las tablas creadas anteriormente para almacenar los datos del usuario con respecto a sus gustos musicales (figura 9.5).

```
function introducetv($tv, $me) {
    $sql = "INSERT INTO `s08a94fb_fbprueba`.`fb_tv` (`id_tv`, `nombre_tv`)
    VALUES ('$tv[id]', '$tv[name]');";
    $query = mysql_query($sql);
    $sql2 = "INSERT INTO `s08a94fb_fbprueba`.`fb_user_tv` (`id_tv`, `id_user`)
    VALUES ('$tv[id]', '$me[id]');";
    $query2 = mysql_query($sql2);
}
```

Figura 12.5

12.3.- Recomendador de libros.

Para la recomendación de libros, la manera de operar es similar a la empleada en los anteriores casos. Debemos crearnos dos tablas en la base de datos para almacenar la información de los usuarios con respecto a sus gustos sobre los libros, que se denominaran **fb_libro** y **fb_user_libro**, y también seguirán la misma estructura que las anteriores tablas. Extraeremos la información del usuario de la siguiente manera:

```
$libros = $facebook->api('/me/books?fields=id,name');
```

Figura 12.6

Para la inserción de los datos en las tablas, usaremos también una función muy similar a las anteriores, pero que en este caso use la información del usuario sobre sus gustos de libros (figura 12.7).

```
function introducelibro($libro, $me) {
    $sql = "INSERT INTO `s08a94fb_fbprueba`.`fb_libro` (`id_libro`, `nombre_libro`)
    VALUES ('$libro[id]', '$libro[name]');";
    $sql2 = "INSERT INTO `s08a94fb_fbprueba`.`fb_user_libro` (`id_libro`, `id_user`)
    VALUES ('$libro[id]', '$me[id]');";
    $query = mysql_query($sql);
    $query2 = mysql_query($sql2);
}
```

Figura 12.7

12.4.- Recomendador de juegos.

Por último, para recomendar al usuario nuevos juegos, procederemos, igualmente, como en el resto de los anteriores apartados. Crearemos otras dos tablas en la base de datos, **fb_juego** y **fb_user_juego**. Su estructura será igual que la de todas las tablas anteriores (ver fig. 9.5).

La extracción de información del usuario actual con respecto a sus gustos en juegos, así como la posterior inserción de información en la base de datos, se realizará mediante el código representado en las figuras 12.8 y 12.9.

```
$juegos = $facebook->api('/me/games?fields=id,name');
```

Figura 12.8

```
function introducejuego($juego, $me) {
    $sql = "INSERT INTO `s08a94fb_fbprueba`.`fb_juego` (`id_juego`, `nombre_juego`)
    VALUES (`$juego[id]`, `$juego[name]`)";
    $query = mysql_query($sql);
    $sql2 = "INSERT INTO `s08a94fb_fbprueba`.`fb_user_juego` (`id_juego`, `id_user`)
    VALUES (`$juego[id]`, `$me[id]`)";
    $query2 = mysql_query($sql2);
}
```

Figura 12.9

CAPÍTULO 13

Estética de la aplicación.

La estética de nuestra aplicación, es probablemente la parte más sencilla que hemos desarrollado. Esto se debe a que el tema principal de este proyecto no es tanto el aspecto físico como la aplicación en sí. Aun así, hemos intentado darle a la aplicación una apariencia trabajada y no directamente dejando los datos como nos los devolvía Facebook.

Para ello nos hemos valido de hojas de estilo embebidas, así como de imágenes creadas mediante Photoshop y de ciertas herramientas de FBML.

Con respecto a la organización de las secciones, usaremos FBML. Mediante la etiqueta `<fb:tabs>`, FBML crea unas “tablas”. Cada una de las pestañas será en realidad una nueva página web, a la que nos redirigirá cuando pulsemos sobre ellas. Dentro de la estructura de la tabla, crearemos `<fb:tab-item>` por cada una de las pestañas que queramos crear. Este elemento tendrá una serie de atributos, que podremos consultar en la documentación de Facebook, en el apartado de “Legacy FBML” dentro de la API. Los dos más importantes serán `href` y `title`. El primero representa la dirección a la que se nos redirigirá en caso de seleccionar dicha pestaña. El segundo hace referencia al nombre que se mostrará sobre la pestaña.

Añadiremos una pestaña por cada una de las secciones de nuestro recomendador, es decir, para música, televisión, películas, libros y juegos. Además, añadiremos otra que nos dirigirá a la página principal, y una

última para que los usuarios puedan recomendar la aplicación a sus amigos (figuras 13.1 y 13.2).

```
$theTabs = <<<tabs
<fb:tabs >
  <fb:tab-item href="index.php" title='Bienvenidos' align="center"/>
  <fb:tab-item href="musica.php" title='Música' align="center"/>
  <fb:tab-item href="libros.php" title='Libros' align="center"/>
  <fb:tab-item href="television.php" title='Televisión' align="center"/>
  <fb:tab-item href="peliculas.php" title='Películas' align="center"/>
  <fb:tab-item href="juegos.php" title='Juegos' align="center"/>
  <fb:tab-item href="invita_amigos.php" title='Recomienda a tus amigos' align="right"/>
</fb:tabs>
tabs;
echo $theTabs;
```

Figura 13.1. Código de la tabla en FBML.

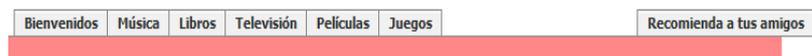


Figura 13.2. Representación de la tabla.

La página principal de nuestra aplicación contendrá una imagen, donde se dará una breve información sobre el funcionamiento de la aplicación, describiendo cada uno de los criterios superficialmente.

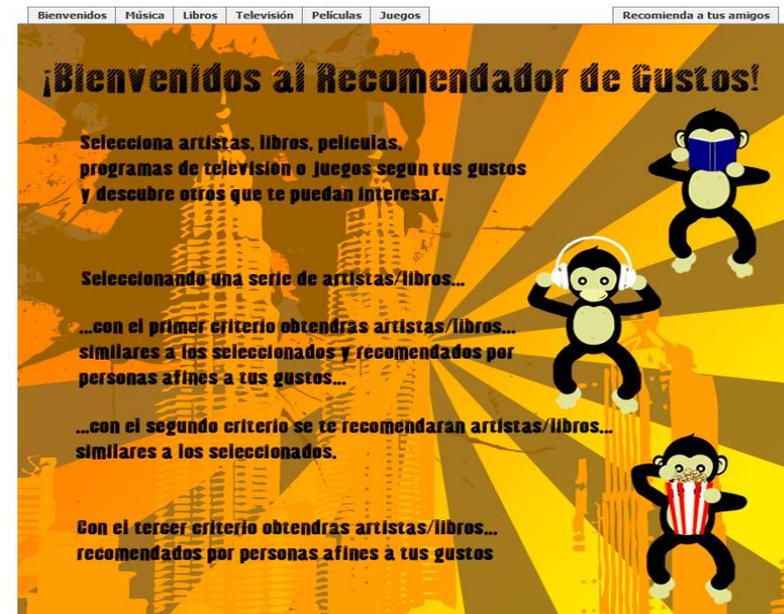


Figura 13.3. Página principal.

Hemos añadido a nuestra página principal un pequeño código CSS embebido que consiste en lo siguiente:

```
<style>
.container
{
  background-image:url('http://www.101.com/cdnimg/stock/10100002.jpg');
  background-repeat:no-repeat;
  width:760px;
  height:700px;
}
</style>
```

Figura 13.4

Aunque FBML permite CSS, presenta ciertas restricciones con respecto a HTML. Por ejemplo, no está permitida la etiqueta `<body>`, con lo que tendremos que crearnos “clases” con nombre para cada sección a la que queramos aplicar un diseño.

Cada una de las secciones de la aplicación, exceptuando la página principal y la página de invitar a nuestros amigos a usar la aplicación, llevarán la misma configuración estética. En la página donde deberemos seleccionar artistas, libros, juegos, etc., construiremos una tabla donde iremos insertando cada una de las opciones que podremos seleccionar a la hora de ejecutar la aplicación. Si no insertamos la tabla, la página nos quedaría como la representada en la figura 10.13. Usaremos una tabla con cuatro columnas y el número de filas dependerá del número de artistas, libros, etc., con el que el usuario cuente en su perfil. En este caso, usaremos dos clases distintas para organizar el contenido, la primera, hará referencia al diseño de la tabla, la segunda lo hará al resto de la página (figuras 13.5 y 13.6).

```
<style>
.tabla
{
padding-left:30px;
background-image:url('http://www.lets.com/images/logo.png');
background-repeat:no-repeat;
width:700px;
}

.container
{
width:760px;
height:700px;
border-left:20px;
}
</style>
```

Figura 13.5

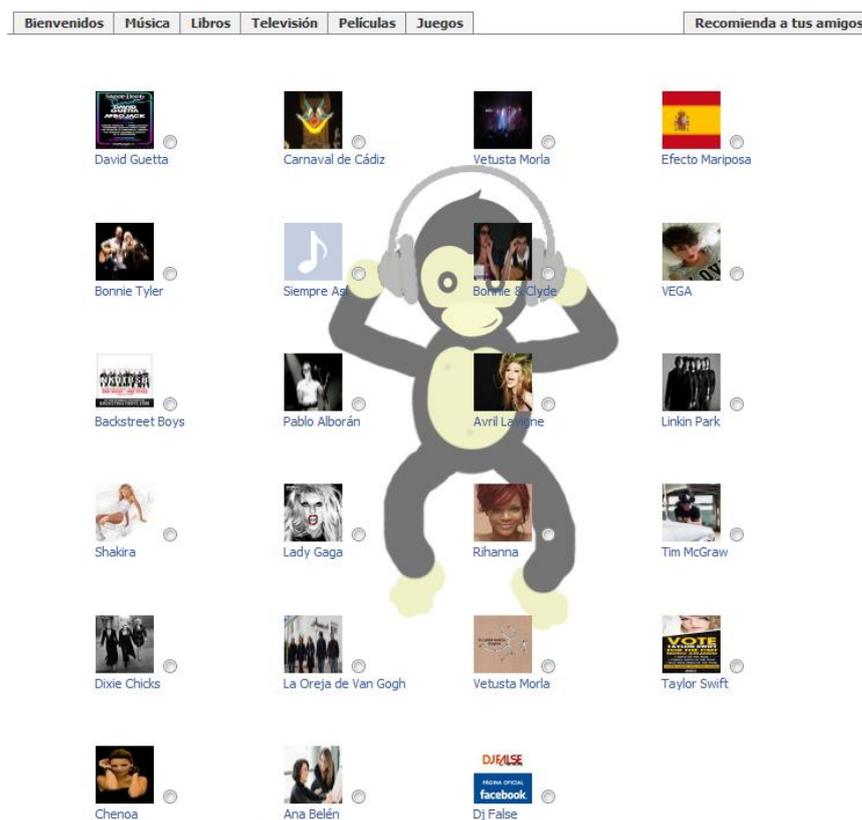


Figura 13.6

Para las páginas de muestra de resultado, haremos algo similar a la anterior, con la diferencia de que éstas serán de dimensiones menores, ya que el número de artistas, libros, etc., que aparecerán será como mucho de cinco (figuras 13.7 y 13.8).

```

<style>
.container
{
width:760px;
height:700px;
border-left:20px;
}

.tabla
{
padding-left:80px;
background-image:url('http://www.ora.com/resources/images/monkey.png');
background-repeat:no-repeat;
width:700px;
height:500px;
}
</style>

```

Figura 13.7

Bienvenidos Música Libros Televisión Películas Juegos Recomienda a tus amigos



Figura 13.8

CAPÍTULO 14

Análisis de antecedentes y aportación realizada.

El desarrollo y creación de una aplicación para la red social Facebook como Proyecto Fin de Carrera, creemos que puede ser de interés a la vez que aportar una línea de trabajo novedosa. En este sentido, conviene señalar que a pesar de existir infinidad de aplicaciones para Facebook, la mayoría de ellas no están documentadas, excepción hecha de las aplicaciones de carácter básico en las que apenas encontraremos ayuda para el desarrollo de la que es objeto de nuestro proyecto. De la misma manera, tras explorar una amplia relación de aplicaciones no hemos encontrado referencias de las que pudiéramos extraer ideas para la que hemos desarrollado.

Por tanto, podríamos afirmar que no apenas existen antecedentes sobre la temática que se aborda en el presente trabajo, debido en buena medida, a que la red social para la que hemos desarrollado la aplicación no ha cobrado importancia a nivel mundial hasta hace apenas un par de años.

Entendemos que, por esta razón, el proyecto que presentamos podrá cumplir también un papel de orientación para otros alumnos que, en los próximos años, elijan como Proyecto Fin de Carrera la creación de otras posibles aplicaciones para la red social Facebook.

Aunque nuestra aplicación haya tenido como principal objetivo su carácter práctico y sus posibilidades de implementación como herramienta de utilidad para los usuarios de la citada red, creemos que

puede aportar también referencias metodológicas para todos aquellos que, en el futuro, quieran profundizar de manera más detenida en los aspectos de programación de la propia aplicación en sí.

CAPÍTULO 15

Comparación con otras alternativas.

El desarrollo de este proyecto podría haber sido ligeramente diferente según las decisiones tomadas a lo largo de su elaboración.

La primera elección que hemos tenido que realizar ha sido el lenguaje de programación y el modo de la aplicación. Con respecto al lenguaje, disponíamos de un amplio abanico de posibilidades, aunque Facebook proporcionaba SDKs sólo para Javascript y PHP. Puesto que necesitábamos interactuar directamente con Facebook, debíamos elegir uno de los dos anteriores. Al no contar con conocimientos previos de ninguno de los dos, nos decantamos por PHP, ya que es el más usado por la mayoría de los desarrolladores de aplicaciones, y, a la hora de realizar consultas en los foros, podríamos obtener más ayudas que usando Javascript.

El modo de la aplicación, que puede ser tanto FBML como iFrame, fue elegido prácticamente con el mismo criterio que el lenguaje de programación. Además de que la mayoría de aplicaciones hasta ahora estaban hechas usando FBML, este modo incluye un lenguaje específico (llamado también FBML) que nos permite realizar ciertas acciones de una manera mucho más sencilla. Por otro lado, el modo iFrame ejecuta la aplicación en una ventana aparte, y esto tampoco nos interesaba, puesto que queríamos que nuestra aplicación estuviera directamente integrada en la propia página de Facebook.

Las dos decisiones anteriores han quedado actualmente anticuadas, ya que Facebook ha decidido eliminar la posibilidad de crear nuestra

aplicación en modo FBML, para compatibilizar y estandarizar sus aplicaciones. Ahora, el usuario deberá realizar las funciones que nos proporciona FBML de una manera bastante sencilla, a partir de los SDK de los lenguajes de programación, combinados con código HTML.

Para la elaboración de una aplicación como la que se propone en este proyecto, pueden tomarse como punto de partida distintos enfoques. Comentaremos a continuación las diversas opciones que valoramos a la hora de realizar la aplicación, detallando las ventajas e inconvenientes de cada una de ellas, lo que nos permitió acotar y concretar la propuesta definitiva.

15.1.- Primera idea

En los momentos iniciales del diseño del “sistema recomendador” se pensó que cada usuario, al ejecutar la aplicación, introdujera los nombres de los intérpretes o grupos musicales y los títulos de sus canciones o temas preferidos. Junto a ello se solicitaría también definir determinadas características que permitieran completar la información como, por ejemplo, el estilo o tipo de música preferida, la ocasión o momento favorito para escucharla, su interpretación por un determinado artista, grupo, dueto, etc.

La adopción de este criterio, exigiría disponer de una base de datos, con el fin de que el usuario introdujera las características de intérprete y melodía una sola vez y no estuviese obligado a hacerlo en cada ocasión que la ejecutase.

Para su correcto funcionamiento, deberíamos tratar de conseguir que cuando un usuario hubiese grabado los datos de un determinado intérprete, por ejemplo, Ana Belén, el resto de usuarios que hubiese incluido a esta cantante entre sus preferidas debería hacerlo también con los mismos criterios que el anterior.

Para salvar esta dificultad se daría la oportunidad a los demás usuarios de elegir al intérprete directamente, sin más datos. Esta solución aparente nos llevaría directamente a otro asunto problemático, ya que es lógico pensar que cada usuario puede tener una opinión diferente respecto a las características que definen el estilo de un mismo artista.

Tras valorar estas circunstancias, descartamos esta idea no tanto por el posible conflicto descrito anteriormente, sino porque consideramos que una aplicación que exigiese a los usuarios cumplimentar para cada intérprete o canción no menos de tres o cuatro campos, resultaría poco atractiva y no tendría ningún éxito.

15.2.- Segunda idea

Tratando de aprovechar al máximo los datos que proporciona Facebook sobre las preferencias de los usuarios, la segunda idea que exploramos consistía en extraer patrones comunes.

La posibilidad de ahorrarnos la utilización de una base de datos implementando funciones que extraigan directamente la información de los usuarios y sus amigos, e incluso de los amigos de nuestros amigos, resultaba muy atractiva, pero, finalmente, esta opción también quedó descartada.

El método a seguir que requería este enfoque exigía que el usuario introdujese el nombre de uno o varios artistas o el de distintos temas o canciones. Junto a ello se precisaba también que se almacenase en distintas tablas los gustos y predilecciones de nuestros amigos y también las de sus amigos para, posteriormente, aplicarles nuestros criterios de recomendación y obtener así el resultado final.

El desarrollo de este procedimiento se abandonó finalmente debido a las limitaciones impuestas por Facebook, ya que aunque un usuario sí puede acceder a los datos sobre sus preferencias, en lo que respecta a la información que proporciona de sus amigos, únicamente le son accesibles los correspondientes al nombre y el id.

15.3.- Tercera idea

Tras la valoración de las opciones anteriores y de sus ventajas e inconvenientes, el método que finalmente se adoptó y que se desarrolla en este proyecto, es el resultante de una combinación de elementos de las dos anteriores propuestas.

Por un lado, pretendemos obtener el máximo provecho de los datos e información que nos ofrece Facebook, y vamos a tratar de franquear la dificultad de acceder a los gustos de nuestros amigos mediante una base

de datos. Junto a todo ello, y para mejorar los resultados de la recomendación, ampliaremos los datos disponibles con los que vayan proporcionando el resto de usuarios que accedan a la aplicación, y no sólo los de nuestros amigos.

A este respecto, se ha optado por prescindir de los campos “intérprete” o “canción” para quedarnos únicamente con el de “música”, a pesar de que a la hora de manifestar nuestras preferencias musicales en Facebook habitualmente se mencionen intérpretes en general y no canciones en particular.

Por otro lado, el recomendador tampoco podrá seguir criterios tales como que nuestro artista interprete un determinado tipo de música y, simplemente, se basará en la afinidad de gustos de unos usuarios con otros.

La idea principal es sencilla, aunque variará ligeramente según el criterio de recomendación que escojamos. Para ofrecer un ejemplo práctico del funcionamiento del sistema, supongamos tres usuarios: Antonio, Beatriz y Carlos.

Carlos accede a la aplicación, donde ya se encuentran registrados Antonio y Beatriz, con el fin de encontrar artistas de características similares a dos de sus favoritos, Joaquín Sabina y Joan Manuel Serrat.

Tanto Beatriz como Antonio, tienen entre sus preferidos varios, entre ellos, los dos que Carlos usa como referencia de búsqueda. Además, Antonio y Beatriz comparten uno en común, Ana Belén. Con estos antecedentes sería fácil suponer que a Carlos le puede gustar Ana Belén.

Ante la imposibilidad de acceder a los datos de nuestros amigos, como se ha señalado anteriormente, vamos a ayudarnos de una base de datos para salvar este problema.

Recordemos que, en relación a nuestros amigos, únicamente teníamos acceso a su nombre e id de usuario. Dado que a los gustos y preferencias de una persona sólo pueden consultarse desde la API si es el propio usuario de la aplicación, ¿por qué no acceder a la sesión del usuario cuando inicie nuestra aplicación? De esta manera, sí podremos llegar a conocer los gustos de cada uno, ya que, al ser uno mismo el que está accediendo a sus datos, no hay ninguna restricción hacia ellos.

Cuando se ejecute nuestra aplicación por primera vez, nuestros gustos quedarán registrados automáticamente en la base de datos

mediante una serie de funciones. Este registro sobre nuestras preferencias, permanecerá inalterable salvo que el usuario, en posteriores accesos, modifique los datos en su perfil de Facebook.

Ya dentro de la aplicación, nos aparecerá un sencillo menú, en el que podremos elegir qué queremos que nos recomiende (en un principio sólo habíamos pensado en música, sin embargo, como veremos más adelante, la aplicación ha sido ampliada).

Una vez elegido el apartado de música, podremos proceder a la recomendación. Para ello, primero, deberemos seleccionar el criterio que preferimos. Dependiendo de nuestra elección, pasaremos, o no, a elegir entre los intérpretes que tenemos recogidos en nuestros gustos musicales en el perfil de Facebook, y con los que queremos que los seleccionados por nuestra aplicación guarden cierta similitud. Hecho esto pulsaremos en “enviar la información”, y se nos redirigirá automáticamente a una nueva página con el resultado de la búsqueda.

15.4.- Criterios de recomendación.

Puesto que no existe un criterio exacto que nos asegure que recomendamos al usuario lo que éste espera, desarrollamos tres criterios diferentes, detallados en los capítulos seis y once de este mismo documento, que proporcionarán resultados distintos aun partiendo de los mismos datos.

A pesar de existir innumerables maneras de realizar una recomendación, la elección de estos tres criterios se debe a que, a nuestro parecer, serán bastante eficaces con respecto a los resultados que buscamos en esta aplicación. Sin embargo, esto no significa que no existan otros criterios que podamos aplicar, y probablemente, a cada persona que le preguntásemos se le ocurriría uno diferente.

CAPÍTULO 16

Análisis temporal y de costes de desarrollo.

Uno de los aspectos fundamentales en la elaboración de cualquier proyecto es el relacionado con la valoración de los costes del mismo, una estimación realista de éstos, tanto de los temporales como de los económicos, determinará, a priori, si es o no rentable llevar a cabo el proyecto e implementarlo en la práctica.

Para que dicha estimación resulte lo más acertada posible es necesaria la confluencia de varios factores, entre los que destacamos la experiencia del equipo que realiza la valoración de los costes, las dimensiones del proyecto o la experiencia del personal que lo desarrolla.

En el caso de nuestro proyecto, la estimación temporal la determina la propia Universidad de Sevilla, ya que en sus programas académicos considera que la elaboración del correspondiente a la titulación de Ingeniería en Informática durará aproximadamente 540 horas.

En relación al proyecto que describe este documento, hemos encontrado algunas dificultades para estimar, en principio, el tiempo necesario para su desarrollo, toda vez que no existían antecedentes de proyectos similares que pudieran servirnos de orientación o referencia. En este sentido, los cálculos realizados inicialmente podrían no ajustarse a los costes temporales que finalmente ha tenido.

De acuerdo al tiempo realmente empleado en todas las fases de elaboración del presente Proyecto de Fin de Carrera, que estimamos en

745 horas, se ha invertido casi un 38% más de lo previsto en un principio para su realización.

16.1.- Aumento del tiempo de desarrollo.

El aumento de coste temporal asignado por la Escuela Técnica Superior de Ingeniería Informática a los proyectos de fin de carrera, ha podido verse incrementado en el desarrollo del presente proyecto por diversos motivos.

Una de las razones puede deberse a que es el primer año en el que se realiza un trabajo con estas características. En este sentido, la ausencia de antecedentes trae también como consecuencia cierta falta de información más concreta sobre este tipo de aplicaciones y de otros trabajos de similares características que hubiesen podido servirnos de orientación y apoyo.

Puede pensarse que Facebook contiene una API lo suficientemente clara para que el hecho de no contar con proyectos de referencia no plantee ningún problema. Sin embargo, como se mencionado en el capítulo 8 del presente proyecto, la documentación sobre aspectos concretos de Facebook no es tan abundante como pueda suponerse.

Otro motivo por el que ha podido verse incrementado el tiempo de desarrollo del proyecto ha sido la ausencia de bibliografía específica que nos sirviera de ayuda a la hora de realizar las distintas consultas que para el desarrollo de la aplicación hemos tenido que realizar. En este sentido, las referencias son escasas y nos hemos visto obligados a localizar la información complementaria a través de otros canales que nos han exigido una mayor dedicación.

Los costes temporales reales de desarrollo del proyecto se recogen en el siguiente apartado, donde se realiza un desglose de las horas dedicadas a cada tarea dentro del análisis, diseño e implementación de la aplicación, así como a la formación y a la redacción de la memoria y de su presentación.

En base a este análisis temporal y a otros factores como las retribuciones económicas por las horas de trabajo del programador o del tutor del proyecto, se podría estimar también de manera realista, el coste económico del mismo.

16.2.- Tiempo de desarrollo.

El siguiente gráfico muestra las horas dedicadas a cada etapa del desarrollo del proyecto. En total han sido necesarias unas 630 horas.

- Análisis de requisitos: 80 h.
- Estudio de PHP: 35 h.
- Estudio de MySQL: 30 h.
- Diseño e implementación de la aplicación: 353 h.
- Redacción de la memoria: 170 h.
- Creación de la presentación: 35 h.
- Entrevistas con el tutor: 3 h.

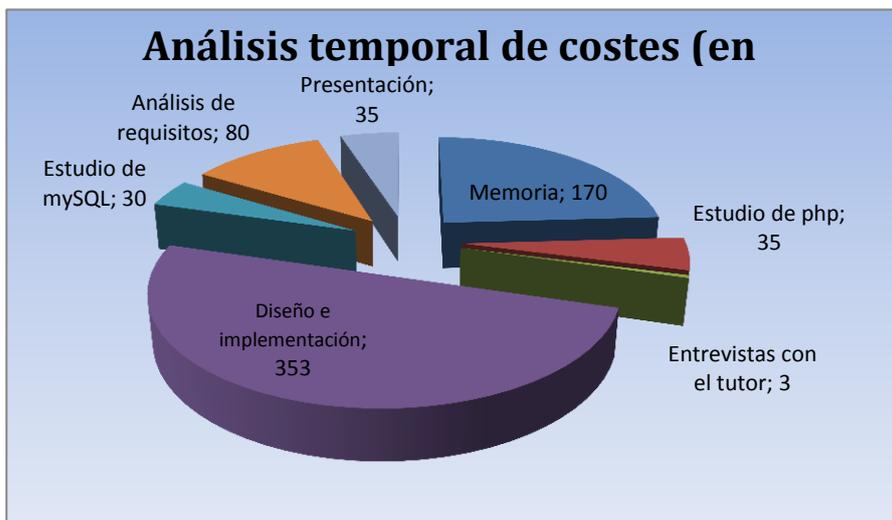


Figura 16.1. Horas dedicadas a cada tarea del proyecto.

Puede comprobarse en el gráfico anterior que la mayor parte del tiempo se ha empleado en el diseño y la implementación de la propia aplicación ya que, en diferentes ocasiones, hemos tenido que volver a replantear conceptos y revisar material para poder continuar con su desarrollo, lo que justifica el incremento del tiempo empleado.

CAPÍTULO 17

Análisis de requisitos.

17.1.- Requisitos fundamentales.

La aplicación objeto de este proyecto, que se va a implementar, debe cumplir ciertos requisitos funcionales. Así, como cuestiones más relevantes señalamos que deberá:

- Funcionar a través de la plataforma para aplicaciones de Facebook.
- Ser accesible para cualquier usuario que disponga de una cuenta Facebook.
- Almacenar los gustos o preferencias de los usuarios registrados en lo que se refiere a música, películas, televisión, juegos y libros.
- Mostrar distintos resultados conforme a la configuración escogida por el usuario.

Para cumplir con todos los requisitos, deberemos realizar una documentación exhaustiva sobre Facebook, con respecto a cómo funcionan sus aplicaciones, en qué lenguaje se pueden programar, qué formatos soportan, etc...

Con respecto a la accesibilidad de los usuarios a nuestra aplicación, deberemos consultar los permisos y políticas de Facebook, así como su forma de solicitarlos.

A la hora de registrar a los usuarios, así como almacenar sus preferencias en lo que se refiere a música, películas, televisión, juegos y libros, deberemos aprender la creación y gestión de una base de datos, de forma que sepamos introducir, borrar o modificar registros desde nuestra aplicación.

A la hora del manejo de la aplicación, ofreceremos al usuario distintos tipos de configuraciones. Podremos elegir el tipo de criterio que se usará para realizar la recomendación, así como el número de artistas que queremos que se nos muestre como resultado final. También podremos, en caso de que hayamos escogido alguno de los criterios que lo requieren, escoger una serie de artistas que se encuentran entre nuestras preferencias musicales dentro del perfil de Facebook. Esta selección afectará directamente a la recomendación, obteniendo resultados que guardan cierta similitud con los artistas seleccionados.

CAPÍTULO 18

Manual de usuario.

El manejo de esta aplicación es muy simple, debido en gran parte a que las acciones permitidas dentro de ésta son muy limitadas.

Para acceder a nuestra aplicación, así como al resto de las aplicaciones de Facebook, tenemos tres posibilidades:

- Un amigo nos ha mandado una invitación a la aplicación.
- Un amigo ha publicado en su muro información sobre la publicación.
- Buscar la aplicación en el repositorio de aplicaciones.

Para las dos primeras opciones, tan sólo debemos pulsar el enlace que nos lleve a la aplicación.



Figura 18.1. Publicación en el muro de un amigo que ha usado el Recomendador.

Si, por el contrario, optamos por buscar la aplicación en el repositorio, escribiremos su nombre en el cuadrado de búsqueda de Facebook. Una vez obtenidos los resultados de la búsqueda, pulsaremos entonces sobre el botón “Ir a la aplicación” y se nos redirigirá inmediatamente a ella.

La primera vez que accedamos a la aplicación visualizaremos la siguiente pantalla, que nos solicitará que aceptemos los permisos que se requieren para poder usarla ella. Si estamos de acuerdo con éstos, pulsamos sobre el botón “Permitir” y accederemos a la página principal.



Figura 18.2. Solicitud de permisos.

Una vez aceptados los permisos, se nos redirigirá automáticamente a la página principal de la aplicación (figura 18.3)

En ésta se describen muy brevemente los criterios, ya que al usuario realmente lo que le interesará es el resultado, y no cómo funciona nuestra aplicación.

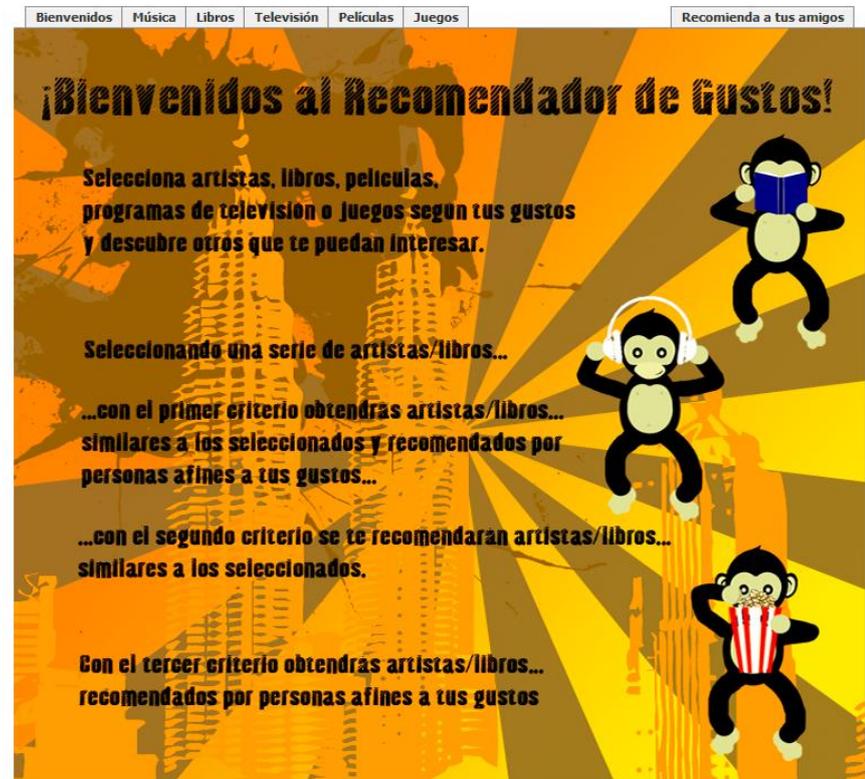


Figura 18.3

Dependiendo de lo que queramos que nos sea recomendado en ese momento, accederemos a cada uno de los diferentes apartados de la aplicación.

Puesto que el funcionamiento de cada una de las secciones es idéntico al resto, procederemos solamente a la explicación del apartado “Música”.

Tras seleccionar la pestaña música, se nos dirigirá a una nueva página, donde aparecerán todos los artistas que poseamos en nuestro perfil, para, si es necesario en el criterio elegido, seleccionar alguno de ellos.

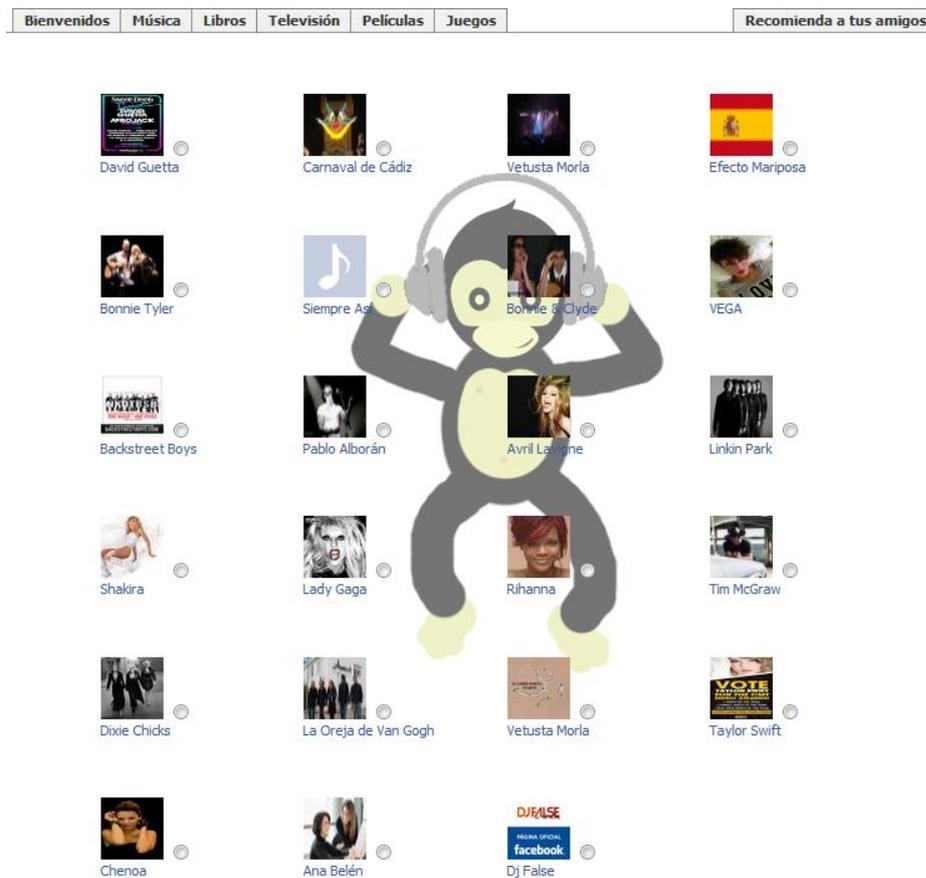


Figura 18.4. Página principal de la sección Música.

Si vamos a escoger el primer o el segundo criterio para obtener la recomendación final, deberemos seleccionar una serie de artistas similares a lo que queramos encontrar.

En el caso de que queramos usar como criterio de recomendación el tercero, nos saltaremos el paso anterior, ya que éste se basa en nuestros gustos en general y no en ciertos artistas en particular.

El siguiente paso que debemos realizar para cualquiera de los criterios es indicarle a la aplicación el criterio escogido. En el caso de que se nos olvide seleccionar algún criterio, la aplicación usará el tercer criterio como forma de recomendación.

A continuación debemos seleccionar el número de artistas que queremos que nos recomiende la aplicación. Hemos puesto un límite de cinco artistas como máximo a devolver. Aunque podríamos poner el límite que quisiéramos, cuantos más resultados queramos que se nos muestren, más alejados estarán de lo que buscábamos obtener. En caso de que no seleccionemos ningún número, sólo se mostrará el primer resultado de la recomendación.

Una vez configurado todo, pulsaremos el botón “Submit” para obtener los resultados.

Seremos redirigidos automáticamente a una nueva página, en la que se nos mostrarán los resultados obtenidos acorde a la configuración realizada de la aplicación.

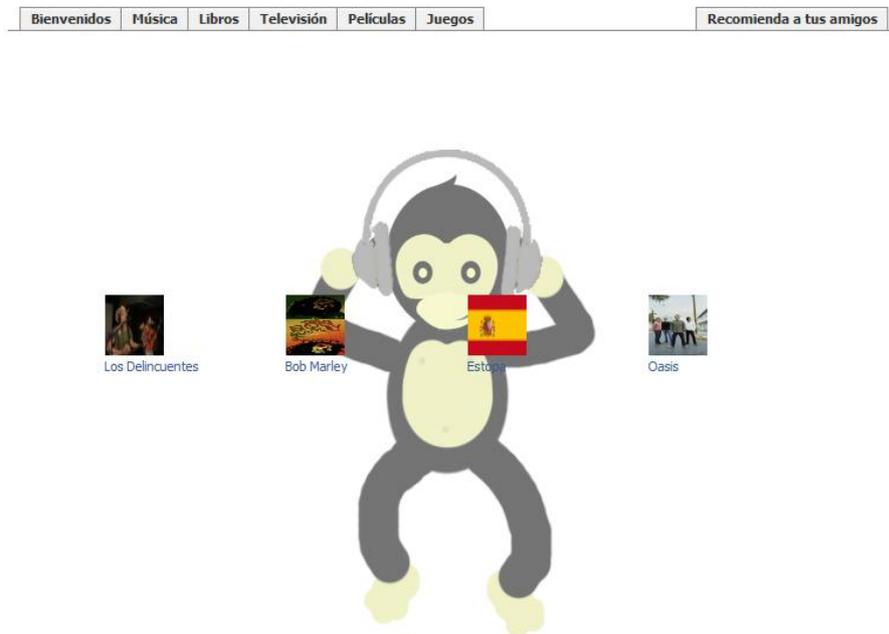


Figura18.5. Pantalla de resultados.

Además, cuando hayamos obtenido los resultados, se publicará automáticamente en nuestro tablón un anuncio avisando al resto de usuarios que estamos usando la aplicación “Recomendador de gustos” (figura 18.6). Aunque pueda parecer un poco abusivo que el usuario no pueda elegir si quiere o no que el resto de sus amigos vean que está usando la aplicación, ésta es una práctica muy común, ya que es prácticamente el único medio que tienen las aplicaciones de darse a conocer.



Figura 18.6. Publicación automática en nuestro muro.

CAPÍTULO 19

Pruebas.

Puesto que nuestra aplicación es totalmente dependiente de la información contenida en nuestra base de datos, los resultados obtenidos en las pruebas realizadas, variarán dependiendo de si hay nuevos usuarios que hayan accedido a la aplicación posteriormente a la realización de dichas pruebas.

19.1.- Registro de datos de usuario en nuestra BD.

La primera vez que accedamos a la aplicación, deben quedar registrados en nuestra base de datos, tanto nuestro nombre, como nuestros gustos musicales. Para comprobar el correcto registro de nuestros datos, deberemos acceder a la dirección donde se encuentre alojada, y comprobar que se haya creado una nueva entrada en cada una de las tablas. Mostraremos solamente la tabla de registros de **fb_user_juego**, ya que para el resto de las tablas será similar (figura 18.1). Podremos comparar sabiendo nuestro id, y el id de los juegos que tenemos asociados a nuestro perfil, si el registro se ha hecho correctamente o no (para consultar los ids tanto del usuario como de los juegos, accederemos a las tablas **fb_user** y **fb_juego** donde se relaciona el nombre con el id del objeto)

Opciones			id_user	id_juego
<input type="checkbox"/>			6005	12663804871
<input type="checkbox"/>			6005	78883723363
<input type="checkbox"/>			6005	94930628299
<input type="checkbox"/>			6005	137978073729
<input type="checkbox"/>			6005	201120755306
<input type="checkbox"/>			6005	107768035913377
<input type="checkbox"/>			6005	112537385423860
<input type="checkbox"/>			6005	113337792084757
<input type="checkbox"/>			6005	131208100273632
<input type="checkbox"/>			6005	161960623848142
<input type="checkbox"/>			6005	180355198656633

Figura 19.1. Tabla fb_user_juego.

A continuación vamos a realizar una serie de pruebas con cada uno de los criterios, así como de las distintas secciones de la aplicación.

19.2.- Modificando datos de usuario en nuestra BD.

Vamos a comprobar que la aplicación es capaz de modificar datos de usuario, es decir, si un usuario agrega nuevos gustos o los elimina, deberán quedar reflejados en la base de datos. Vamos a mostrar el antes y después del estado de la tabla **fb_user_tv** de un usuario cuando ha eliminado de su perfil un programa de televisión y ha añadido otros dos.

Eliminaremos de nuestros gustos la serie “The Wire”, cuyo **id_tv** es 5991693871, y añadiremos “Perdidos” y “House”, cuyos **id_tvs** son 109788019061149 y 7608631709 respectivamente.

Antes de que el usuario acceda a la aplicación tras haber realizado los cambios en su perfil, el estado de la tabla sería el mostrado en la figura 19.2.

			id_user ▲	id_tv
<input type="checkbox"/>			600	5991693871
<input type="checkbox"/>			600	7807422276
<input type="checkbox"/>			600	8811587900
<input type="checkbox"/>			600	16737238291
<input type="checkbox"/>			600	22577904575
<input type="checkbox"/>			600	22934684677
<input type="checkbox"/>			600	24007603179
<input type="checkbox"/>			600	48692682008
<input type="checkbox"/>			600	48801603687
<input type="checkbox"/>			600	50812608637
<input type="checkbox"/>			600	103496387381
<input type="checkbox"/>			600	136198000763
<input type="checkbox"/>			600	147812530863
<input type="checkbox"/>			600	188304921840
<input type="checkbox"/>			600	291824160232
<input type="checkbox"/>			600	141091539263448

Figura 19.2

Tras acceder a la aplicación, se deberá reflejar en la base de datos los cambios realizados (figura 19.3).

			id_user ▲	id_tv
<input type="checkbox"/>			600	7608631709
<input type="checkbox"/>			600	7807422276
<input type="checkbox"/>			600	8811587900
<input type="checkbox"/>			600	16737238291
<input type="checkbox"/>			600	22577904575
<input type="checkbox"/>			600	22934684677
<input type="checkbox"/>			600	24007603179
<input type="checkbox"/>			600	48692682008
<input type="checkbox"/>			600	48801603687
<input type="checkbox"/>			600	50812608637
<input type="checkbox"/>			600	103496387381
<input type="checkbox"/>			600	136198000763
<input type="checkbox"/>			600	147812530863
<input type="checkbox"/>			600	188304921840
<input type="checkbox"/>			600	291824160232
<input type="checkbox"/>			600	109788019061149
<input type="checkbox"/>			600	141091539263448

Figura 19.3

19.3.- Pruebas con el criterio de selección con peso.

La primera prueba que vamos a realizar es dentro de la sección juegos. Seleccionaremos tres juegos al azar, el Ahorcado, el Trivial Pursuit y Los Colonos de Catán.

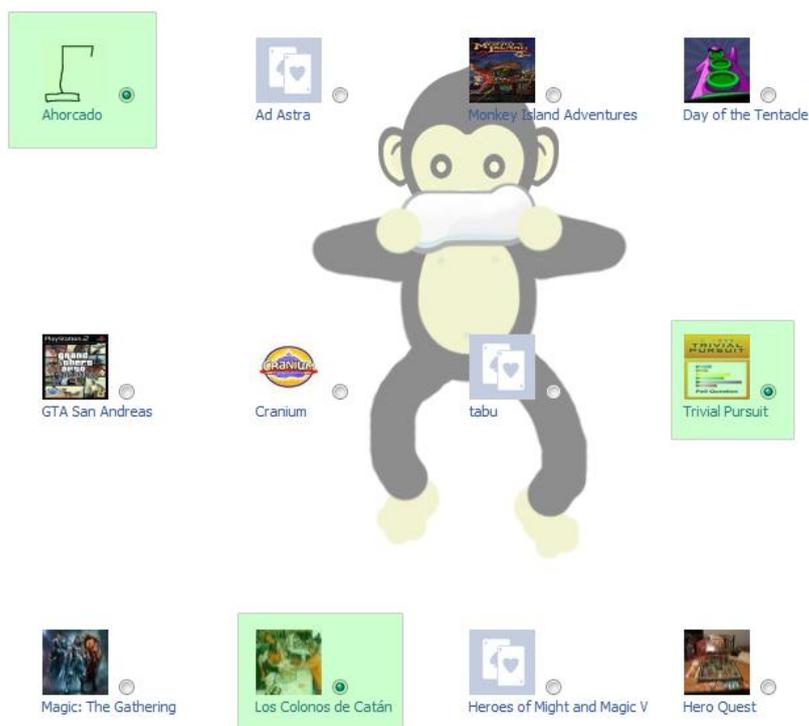


Figura 19.4

Seleccionaremos en la aplicación la opción de que nos muestre cinco resultados. La recomendación obtenida será la mostrada en la figura 19.5.

Con los mismos juegos seleccionados que en la figura 19.4, escogeremos esta vez la opción de que sólo nos muestre tres juegos como resultado. Estos tres juegos deberán ser los primeros que aparecían en la figura 19.5. Los resultados se muestran en la figura 19.6



Figura 19.5. Recomendación obtenida mostrando cinco juegos.

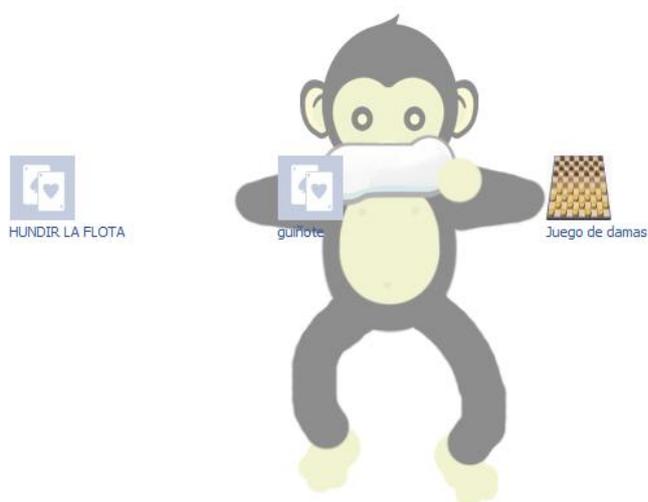


Figura 19.6. Recomendación obtenida mostrando tres juegos.

19.4.- Pruebas con el criterio de selección.

Para comprobar el funcionamiento del segundo criterio, haremos una batería de pruebas similar a la usada en el anterior apartado, aunque esta vez usaremos otra de las secciones de la aplicación.

En este apartado usaremos la sección “películas” para realizar las pruebas.

Seleccionamos dos películas, por ejemplo El Señor de los Anillos y Harry Potter y configuramos la aplicación para que nos devuelva cuatro resultados.



Figura 19.7

Seleccionamos dos películas, por ejemplo El Señor de los Anillos y Harry Potter y configuramos la aplicación para que nos devuelva cuatro resultados. Puesto que hemos seleccionado dos películas que guardan cierta similitud, cabría esperar que la recomendación que obtengamos tenga cierto parecido con éstas. Sin embargo, es probable que no lo sea, debido en gran a que la base de datos no es lo suficientemente grande como para precisar demasiado en los resultados. A pesar de que

tuviéramos una base de datos muy amplia, también se podría dar el caso de que los resultados obtenidos no guardaran similitud con los seleccionados, ya que en realidad, sólo nos basamos en los gustos de los usuarios y no en ninguna característica específica de la película.



Figura 19.8. Resultados obtenidos a partir del primer criterio.

Como observamos en la figura anterior, las películas recomendadas no guardan prácticamente similitud entre las películas inicialmente seleccionadas.

19.5.- Pruebas con el criterio “invariable”.

Para comprobar el funcionamiento del tercer criterio, haremos una batería de pruebas similar a las de los anteriores apartados, aunque esta vez usaremos otra de la sección “libros”.

Puesto que este criterio no requiere la selección de ningún elemento, nos saltaremos este paso y realizaremos directamente la configuración del número de resultados que queremos obtener. En este caso pediremos a la aplicación que nos devuelva dos libros como recomendación.



Figura 19.9. Sección libros sin ningún elemento seleccionado.

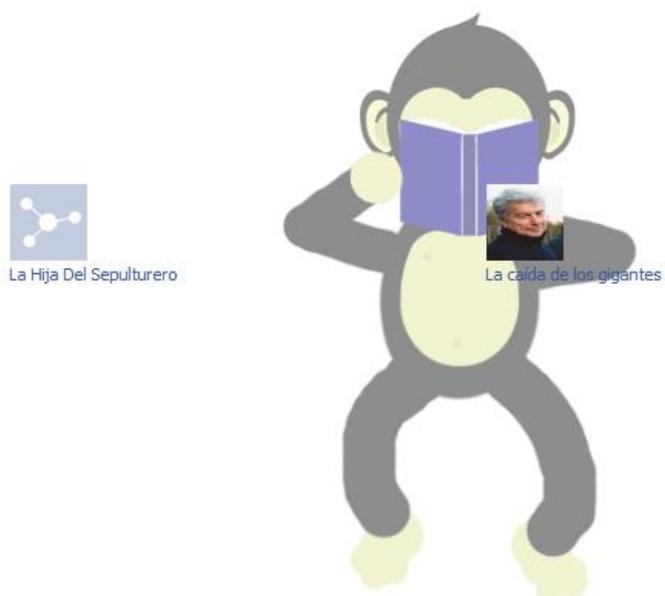


Figura 19.10. Resultado tras la aplicación del tercer criterio.

Por último, comprobaremos que, en el caso de que el usuario no seleccione ningún criterio, el que se aplicará será el tercero.

Realizamos la prueba en esta misma sección para comparar resultados con lo obtenido en la anterior prueba y comprobamos que los resultados coinciden con los mostrados en la figura 19.11.

19.6.- Comparación de criterios.

Vamos a realizar por último una comparación de los resultados obtenidos por los tres criterios, configurándolos de igual manera, en la medida de lo posible.

Vamos a seleccionar tres artistas, Rihanna, Lady Gaga y Shakira. A continuación pediremos que se nos recomienden dos artistas. Posteriormente aplicaremos cada uno de los criterios para compararlos entre sí.

En el caso del tercer criterio, no tendrá ningún sentido seleccionar artistas, pero comprobaremos también los resultados obtenidos, para observar que funcionan de distinta forma.

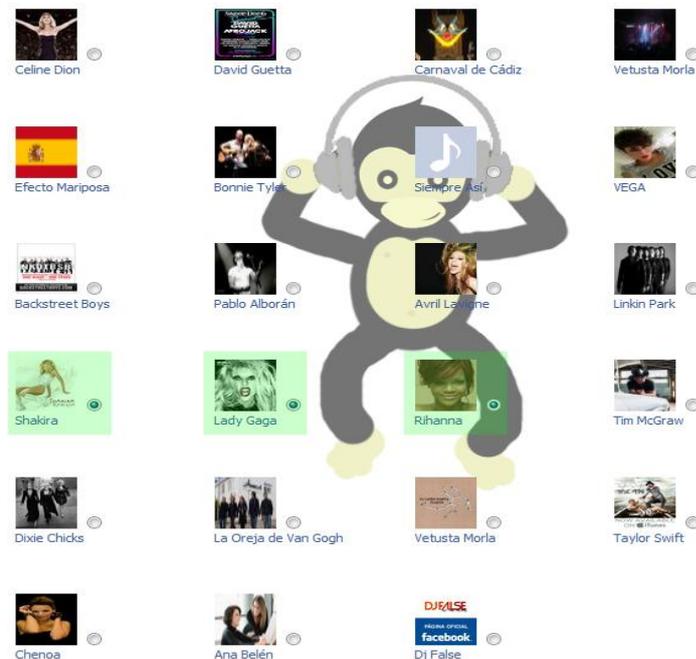


Figura 19.11

En el caso del criterio uno, los artistas que se nos han recomendado son “Los Delincuentes” y Oasis.



Figura 19.12. Resultados tras la aplicación del primer criterio.

Los artistas recomendados por el tercer criterio son “Katy Perry” y “Oasis” (figura 19.4). Como vemos, el resultado obtenido ha sido distinto pero no demasiado, ya que de los dos artistas obtenidos, uno de ellos es el mismo. En este caso podríamos decir que uno de los resultados si guarda cierta similitud con los artistas seleccionados en el comienzo.

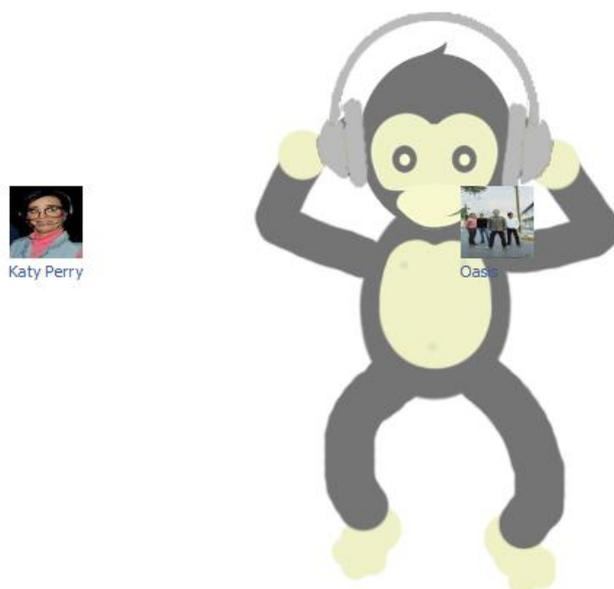


Figura 19.13. Resultados tras la aplicación del segundo criterio.

Los resultados obtenidos tras la aplicación del tercer criterio, como hemos dicho anteriormente, se basan en nuestros gustos en general. Como resultado hemos obtenido “Los Delincuentes” y “Estopa”. En este caso, vemos que uno de los artistas también coincide con otro de los recomendados en el primer criterio.



Figura 19.14. Resultados tras la aplicación del tercer criterio.

19.7.- Pruebas con errores.

Es posible que el usuario de la aplicación, usando el primero segundo criterio, pase por alto seleccionar algún artista, libro, etc. En este caso, al usuario simplemente se le avisará de que no ha seleccionado ningún elemento, en una pantalla de error.



¡ERROR! Recuerda que debes seleccionar algún elemento si quieres usar el segundo criterio para obtener tu recomendación.



Figura 19.15

CAPÍTULO 20

Conclusiones y desarrollos futuros.

20.1.- Conclusiones.

El desarrollo de este proyecto ha sido largo en el tiempo pero a la vez muy ameno. Hemos profundizado en una serie de temas muy actuales, que abarcan una amplia gama de conocimientos y por tanto resulta difícil tratarlos en su totalidad en los estudios de nuestra carrera.

Durante su elaboración nos hemos familiarizado con tecnologías referentes al funcionamiento de aplicaciones, así como de páginas webs, lo que nos resultará muy útil en un futuro para acometer cualquier tipo de proyecto relacionado con Internet.

Sobre Facebook, cabe decir que es una plataforma que ofrece infinidad de posibilidades a los desarrolladores, y probablemente, este trabajo tan solo plantea algunas de las muchas líneas que se pueden explorar. Esto se debe en gran parte a que es un proyecto experimental sobre la creación de una aplicación y que desconocíamos de antemano a qué nos enfrentábamos, por lo que gran parte del tiempo ha sido invertido en realizar la base de la aplicación.

La mayor parte del proyecto ha sido programado en PHP. Aunque carecíamos de conocimientos previos de este lenguaje, su aprendizaje no ha entrañado grandes dificultades por su gran similitud con otros ya estudiados a lo largo de la carrera. Además, dada su creciente utilización en lo que a programación web se refiere, a buen seguro que la experiencia y los conocimientos adquiridos en el desarrollo de este proyecto, nos serán de gran utilidad en un futuro próximo.

A pesar de que Facebook posee una amplia documentación, en muchos casos hemos tenido dificultades a la hora de usar el lenguaje propio de Facebook, FBML, para realizar determinadas funciones. Esto se debe, fundamentalmente, a que la documentación carece totalmente de ejemplos, por lo que, a pesar del nivel de detalle en el contenido de algunas funciones, desgraciadamente no se hace extensivo a cómo usarlas. Para suplir esta falta de información hemos recurrido a los diferentes foros internos de desarrolladores de Facebook, que en más de una ocasión nos han resuelto los problemas que hemos ido encontrando en su diseño.

Aunque Facebook ha dejado de incluir FBML para la creación de sus aplicaciones, dada su semejanza con HTML podremos hacer uso en un futuro de gran parte de los conceptos asimilados en el desarrollo de este proyecto.

Además, hemos aprendido a crear y gestionar bases de datos, refrescando conocimientos sobre SQL y adquiriendo otros nuevos. Esta sección también será de gran importancia en un futuro, puesto que prácticamente cualquier página web o aplicación que podamos diseñar posee una base de datos asociada a ella.

También hemos usado, para la estética de la aplicación, CSS embebido combinado con ciertas funciones básicas de FBML. El CSS utilizado ha sido muy básico, puesto que el fin de este proyecto no estaba centrado tanto la estética, como en conseguir la creación de una aplicación y en su funcionalidad.

Por último, no queremos dejar de subrayar que consideramos de gran interés, de cara a nuestro futuro profesional, haber profundizado en el conocimiento de la elaboración de aplicaciones para Facebook, la red social más extendida y con mayor proyección. No cabe duda de que en este espacio convergen los intereses económicos y de negocio de numerosas empresas y que, por esta razón, se precisan en nuestro país profesionales de la informática que se especialicen en la elaboración de aplicaciones específicas para Facebook, en las que –con las limitaciones propias de un trabajo de este tipo- he pretendido iniciarme con la elaboración de este proyecto fin de carrera.

20.2.- Desarrollos futuros.

En lo que respecta a nuestro proyecto, podríamos decir que nuestra aplicación está prácticamente concluida, aunque siempre estaría abierta a incorporar ciertas funcionalidades como algún otro criterio de búsqueda o a mejorar el aspecto visual.

Dado su carácter experimental, creemos también que podrá servir de gran ayuda a otros alumnos que, en un futuro, escojan como proyecto fin de carrera la creación de aplicaciones en Facebook.

En estas páginas encontrarán una guía minuciosamente detallada sobre la creación de una aplicación, desde el comienzo, que en buena medida será común al de otras aplicaciones que pudieran diseñarse, especialmente hasta que se aborda la parte específica de este proyecto, donde también podrán encontrar algunas indicaciones que les resultarán de utilidad.

Debido a los constantes cambios que se llevan a cabo en la plataforma de desarrollo de Facebook, es posible que ciertos apartados de este documento queden superados a corto plazo con la incorporación de nuevos planteamientos. Esto ha ocurrido, como ya se ha indicado, durante la elaboración de nuestro propio proyecto, al modificar e incluso suprimir ciertas funcionalidades, como el uso del lenguaje FBML.

A buen seguro que el futuro desarrollador tendrá que afrontar nuevas dificultades a las que nosotros no nos hemos enfrentado. Sin embargo, así como Facebook va creciendo con mucha rapidez, también lo va a hacer el número de sus desarrolladores, con lo cual también podrán disponer de muchas más orientaciones y ayudas de las que hasta ahora hemos podido encontrar nosotros.

CAPÍTULO 21

Bibliografía.

Las referencias bibliográficas así como los sitios web y documentos electrónicos consultados a lo largo del desarrollo del proyecto son los siguientes:

-Monta un servidor web local con Xampp. Disponible en:

<<http://www.punto geek.com/2010/11/10/monta-un-servidor-web-local-con-xampp/>> [Consulta: 24 - 11 - 2010]

-Tutoriales generados por la comunidad. Disponible en:

<<http://forum.developers.facebook.net/viewtopic.php?id=79787>>
[Consulta: 20 - 3 - 2011]

-Graph API. Disponible en:

<<http://developers.facebook.com/docs/reference/api/>> [Consulta: 5 - 6 - 2011]

-Getting Started: Useful Links. Disponible en:

<<http://forum.developers.facebook.net/viewtopic.php?id=9163>>
[Consulta: 15 - 4 - 2011]

-¿Cómo obtener valores desde un “radio” button?. Disponible en: <

<http://www.forosdelweb.com/f18/como-obtener-valores-desde-radio-button-724265/>> [Consulta: Consulta: 2 - 5 - 2011]

-Formularios HTML. Disponible en:

<<http://es.kioskea.net/contents/html/htmlform.php3>> [Consulta: 20 - 5 - 2011]

- JA sample Facebook application with CodeIgniter. Disponible en:
<<http://junal.wordpress.com/2008/01/20/a-sample-facebook-application-with-codeigniter/>> [Consulta: 24 -5-2011]

- Stephanie Krieger. 12 sugerencias para crear mejores documentos. Disponible en:
<<http://www.microsoft.com/latam/office/evolucion/trabajo/getworkdone/DOCUMENTS.MSPX>> [Consulta: 26 -5-2011]

- [TLK] Graph API beginner's tutorial. Disponible en:
<<http://forum.developers.facebook.net/viewtopic.php?id=86504>> [Consulta: 29 - 5 - 2011]

- El conjunto de aplicaciones ideal para desarrollo con php. Disponible en:
<http://ja-jp.facebook.com/note.php?note_id=153179168035777> [Consulta: 29 - 11 - 2010]

- Funciones básicas en php. Disponible en:
<<http://www.trucoswebmasters.com/a295-43.html>> [Consulta: 20 - 2 - 2011]

- Atomic operations. Disponible en:
<<http://www.jvmultimedia.com/portal/node/16>> [Consulta: 10 - 4 - 2011]

- Conectar PHP con MySQL. Disponible en:
<<http://blog.unijimpe.net/conectar-php-con-mysql/>> [Consulta: 20 - 2 - 2011]

- Manual de PHP. Disponible en:
<http://www.forosdelweb.com/wiki/Manual_de_PHP> [Consulta: 15 - 3 - 2011]

- Funciones de MySQL. Disponible en: <
<http://www.php.net/manual/es/ref.mysql.php>> [Consulta: 15 - 3 - 2011]

- Step by step app development from a non-expert perspective. Disponible en:
< <http://forum.developers.facebook.net/viewtopic.php?id=37120>> [Consulta: 17 - 2 - 2011]

- PHP Tutorial. Disponible en: <<http://www.w3schools.com/php/default.asp>> [Consulta: 29 - 4 - 2011]

- Crea tu propia aplicación de Facebook en pocos minutos. Disponible en:
<<http://www.elwebmaster.com/general/crea-tu-propia-aplicacion-de-facebook-en-pocos-minutos>> [Consulta: 9 - 12 - 2010]

- Como hacer aplicaciones en Facebook. Disponible en:
<<http://gfaraj.wordpress.com/2007/09/11/como-hacer-aplicaciones-en-facebook/>> [Consulta: 9 - 12 - 2010]
- Gabriel Cuesta Arza. Cómo crear una aplicación en Facebook con PHP. Disponible en: <<http://www.faceblog.es/tutoriales/como-crear-una-aplicacion-facebook-con-php/>> [Consulta: 9 - 12 - 2010]
- Iñaki Huerta. Crear una aplicación con Facebook Graph Api, en modo FBML. Disponible en: <<http://blog.ikhuerta.com/crear-una-aplicacion-con-facebook-graph-api-en-modo-fbml/>> [Consulta: 23 - 5 - 2011]
- Crear aplicación "Hello World" en Facebook con PHP. Disponible en: <<http://www.cristalab.com/tutoriales/crear-aplicacion-hello-world-en-facebook-con-php-c73492l/>> [Consulta: 9 - 12 - 2010]
- Primeros pasos para generar aplicaciones de Facebook. Disponible en: <<http://www.maestrosdelweb.com/editorial/primeros-pasos-para-generar-aplicaciones-de-facebook/>> [Consulta: 23 - 2 - 2011]
- Step by Step Guide to Creating a First Facebook App. Disponible en: <<http://aleatory.clientsideweb.net/2009/07/05/step-by-step-guide-to-creating-a-first-facebook-app/>> [Consulta: 4 - 2 - 2010]
- E.A. Vander Veer. (2010). *Facebook*. Anaya Multimedia.
- E.A. Vander Veer. (2009). *Exprime Facebook*. Anaya Multimedia.
- Moore, M. H., Estrich, S., McGillis, D., y Spelman, W. (1984). *Delincuentes peligrosos: el blanco escurridizo de la justicia*. Cambridge: Harvard University Press.

CAPÍTULO 21

Glosario.

API: proviene de las siglas en inglés de *Application Programming Interface*, y se traduce como interfaz de programación de aplicaciones. Es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser usado por otro software como una capa de abstracción.

Aplicación web: conjunto de páginas que interactúan unas con otras y con diversos recursos en un servidor web, incluidas bases de datos. Esta interacción permite implementar características en su sitio como catálogos de productos virtuales y administradores de noticias y contenidos. Adicionalmente podrá realizar consultas a bases de datos, registrar e ingresar información, solicitudes, pedidos y múltiples tipos de información en línea en tiempo real.

Base de datos: colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Es decir, un sistema de archivos electrónico.

Dominio: es una red de identificación asociada a un grupo de dispositivos o equipos conectados a la red Internet. El propósito principal de los nombres de dominio en Internet es traducir las direcciones IP de cada nodo a términos memorizables y fáciles de encontrar.

Facebook: sitio web de redes sociales creado por Mark Zuckerberg. Actualmente es la mayor red social, con más de 600 millones de usuarios en todo el mundo.

Hosting: servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

Lenguaje de programación interpretado: es un lenguaje de programación que está diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados.

Php: acrónimo recursivo que significa PHP *Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas webs dinámicas.

Recomendador: usuario de la aplicación, almacenado en la base de datos de la misma, del que se extraerá información para realizar el cálculo de la recomendación.

Red social: estructura social compuesta por grupos de personas, que se encuentran conectadas por uno o varios tipos de relaciones, como amistad, familia, intereses comunes o conocimientos entre otras cosas.

SDK: proviene de las siglas en inglés *Software Development Kit*, y es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto. Es en realidad una API creada para permitir el uso de cierto lenguaje de programación.

Servidor web: es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

Usuario actual de la aplicación: persona que en ese momento está usando la aplicación, y que será aquella que quiere que se le recomiende alguna de las opciones de la aplicación

