



Proyecto de Fin de Máster en Sistemas Inteligentes

Curso 2009-2010

TravelCobber Técnicas para la integración de recomendaciones en Facebook

Autor:

Felipe Pablo Álvarez Arrieta

Director:

Pedro A. González Calero

Codirector:

Héctor Gómez Gauchía

Máster en Investigación en Informática
Facultad de Informática
Universidad Complutense de Madrid

Agradecimientos

Agradezco a D. Pedro González Calero y a D. Héctor Gómez Gauchía por haber aceptado dirigir mi Tesis de Fin de Master, por su disposición y disponibilidad para todas mis dudas y por la paciencia que han tenido conmigo a lo largo de este año.

Ha sido un placer volver a encontrarme con Héctor a quién ya tuve de profesor en la carrera de Ingeniería en Informática. He aprendido muchas cosas de él a lo largo de este año y he apreciado especialmente las interesantes charlas mantenidas en las reuniones del proyecto. Ha sido un estímulo conocer a Pedro, solo lamento no haberle tenido antes como profesor por lo mucho que hubiese podido aprender de él.

Quiero agradecer a todos los amigos que me han ayudado a probar la aplicación, sin importar lo ocupados que estuviesen en sus respectivos trabajos, especialmente a Paul Holthues, Francesco Ponticelli, Fernando García Gómez y Eva Shanibova porque sus comentarios y sugerencias me han permitido mejorar la aplicación.

Y por último y muy especialmente a mi madre Doña Carmen Mercedes Arrieta Cornejo por todo en general, y en particular no solo por comprarme mi primer ordenador (un ZX Spectrum 64K), sino por insistirme en que aprendiese a programarlo.

El abajo firmante, matriculado en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo de Fin de Máster: “TravelCobber. Técnicas para la integración de recomendaciones en Facebook”, realizado durante el año académico 2009-2010 dirigido por Pedro A. González Calero y con la colaboración externa de dirección de Héctor Gómez Gauchía en el Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA), y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Madrid, Septiembre 2010

Fdo. Felipe Pablo Álvarez Arrieta

Palabras clave para su búsqueda bibliográfica:

- Sistemas de recomendación
- Recomendaciones para grupos
- Razonamiento basado en casos
- Adaptación
- Personalización inteligente
- Modelado de usuario
- RIA
- Redes Sociales
- Facebook

- Recommender Systems
- Group recommender system
- Case-based reasoning
- Adaptation
- Personalización inteligente
- User modeling
- Rich Internet Applications
- Social Networks
- Facebook

Resumen

Las aplicaciones web comerciales afrontan dos problemas: el poco tiempo y las pocas páginas visitadas por cada usuario, y la baja conversión (visitas que acaban comprando). Para evitarlo se utilizan sistemas de recomendación que ofrecen rápidamente al usuario productos o servicios que le pueden interesar basándose en sus preferencias explícitas o implícitas. Sin embargo estas recomendaciones presentan algunas limitaciones. Por una parte tratan a todos los usuarios de la misma manera, por lo que es difícil que todos queden igualmente satisfechos. Además presentan los inconvenientes del arranque en frío de usuario, arranque en frío del producto, valoraciones inconsistentes, y cambios en las preferencias del usuario. Los sistemas de recomendaciones grupales tienen un cuello de botella: obtener el perfil de los usuarios (requiere el esfuerzo y el tiempo de los usuarios) y ofrecer métodos de negociación para que los usuarios lleguen al acuerdo final. Por otra parte, las redes sociales con un dominio donde los recomendadores se están rápidamente extendiendo, pero no se integran fácilmente en ellas. Para paliar estas limitaciones proponemos un modelo que integra el sistema de recomendación en una Red Social, teniendo en cuenta además el temperamento y el estado de ánimo de los usuarios para adaptarse mejor a ellos. Y mejorando la experiencia del usuario usando tecnologías RIA para facilitar la interacción con el sistema.

Abstract

Commercial Web applications face two problems: the short time and the few pages visited by each user, and the low conversion rates (visitors who purchase). To avoid this issue, Recommendations Systems offer the user quickly products or services that might interest him, according to his explicit or implicit preferences.

However these Recommendation Systems present some limitations. They don't consider the personality, temper and mood of the users. They treat all users the same way, no matter what personality, temper or mood they have. So it is difficult to get all the users equally satisfied. Besides they have the cold start user and cold start products drawbacks, inconsistent ratings and they do not capture the changes in user preferences. Group Recommender Systems also have two bottlenecks: get the profile of the Group members (it requires effort and time of the user), and provide mechanisms to help users to reach the final agreement.

To address these problems we propose a model that integrates the recommendation system in a Social Network (Facebook), taking into account the temper of the user to better suit them. And improving the user experience using RIA technologies

Índice de contenido

Resumen.....	5
Abstract.....	5
1 Introducción.....	9
2 Estado del Arte	13
2.1 Sistemas de Recomendación individuales.....	13
2.1.1 Clasificación de sistemas de Recomendación	13
2.1.2 Métricas usadas en Sistemas de Recomendación	16
2.2 Sistemas de Recomendación para grupos de usuarios.....	17
2.2.1 Fases del proceso de recomendación	20
2.2.1.1 Presentación al grupo preferencias individuales.....	21
2.2.1.2 Agregación preferencias y modelos individuales.....	23
2.2.1.3 Métodos de Agregación.....	26
2.2.1.4 Aceptabilidad individual de la recomendación.....	27
2.2.1.5 Negociación grupal final	28
2.3 Recomendación en Redes Sociales.....	30
Redes Sociales.....	30
Caso de estudio: Facebook	30
Recomendaciones para grupos en Facebook.....	31
2.4 Aplicaciones en Facebook	34
2.4.1 Juegos.....	34
2.4.2 Aplicaciones afectivas.....	37
2.4.3 Otras aplicaciones.....	43
2.4.4 Conclusiones aplicaciones Facebook.....	46
2.5 Rich Internet Applications (RIAs)	46
3 Análisis del Modelo	48
3.1 Arquitectura	49
3.2 Casos de Uso	54
4 Diseño e implementación.....	76
4.1 Patrones de diseño.....	76
4.2 Diseño de las Interfaces Gráficas	79
4.3 Personalización Inteligente	80
4.4 Implementación.....	84

5	Resultados obtenidos.....	89
6	Conclusiones.....	99
7	Documentación del sistema.....	102
8	Referencias.....	106

Índice de Tablas

Tabla 1.	Interacción Sistema de recomendación-usuario.....	15
Tabla 2.	Clasificación Sistemas de recomendación individual.....	16
Tabla 3.	Ventajas e inconvenientes de los sistemas de recomendación.....	16
Tabla 4.	Métricas usadas en los sistemas de recomendación.....	17
Tabla 5.	Sistemas de recomendación para grupos.....	18
Tabla 6.	Ventajas e inconvenientes Sistemas de recomendación para grupos.....	19
Tabla 7.	Análisis Sistemas de recomendación para grupos.....	20
Tabla 8.	Diferencias en el proceso de recomendación para grupos.....	20
Tabla 9.	Adquisición de las preferencias individuales.....	22
Tabla 10.	Agregación de las preferencias individuales.....	26
Tabla 11.	Negociación grupal final.....	28
Tabla 12.	Aplicaciones en Facebook.....	38
Tabla 13.	Filtros en las búsquedas en las aplicaciones afectivas en Facebook.....	38
Tabla 14.	Formas de contacto en las aplicaciones afectivas en Facebook.....	38
Tabla 15.	Ventajas e inconvenientes aplicaciones afectivas en Facebook.....	39

Índice de Ilustraciones

Ilustración 1. Gráfica y estadísticas de acceso a una aplicación web comercial.....	9
Ilustración 2. Ciclo de razonamiento basado en casos.....	14
Ilustración 3. Especificación preferencias individuales en el sistema <i>MusicFX</i>	21
Ilustración 4. Especificación preferencias individuales en el sistema <i>Travel Decision Forum</i>	23
Ilustración 5. Agregación valoraciones individuales en el sistema <i>POLYLENS</i> ...	24
Ilustración 6. Propuesta Sistema de Recomendación Grupal Integrado.....	29
Ilustración 7. Propuesta Sistema de Recomendación Grupal Integrado en una Red Social.....	33
Ilustración 8. Configuración del avatar en <i>PetSociety</i>	34
Ilustración 9. Hogar del avatar en <i>PetSociety</i>	35
Ilustración 10. Configuración del avatar en <i>Sacred Seasons</i>	35
Ilustración 11. Perfil de usuario en <i>Flirtable</i>	36
Ilustración 12. Pantalla principal de <i>Meet New People</i>	40
Ilustración 13. Perfil de usuario en <i>Meet New People</i>	40
Ilustración 14. Perfil de usuario en <i>True Match</i>	41
Ilustración 15. Configuración información usuario en <i>True Match</i>	41
Ilustración 16. Encuesta de personalidad en <i>True Match</i>	42
Ilustración 17. Usuarios con personalidad similar en <i>True Match</i>	42
Ilustración 18. Resultado test de personalidad en la aplicación <i>Personality</i>	43
Ilustración 19. Iteración RIA en el modelo de proceso de desarrollo Web.....	45
Ilustración 20. Arquitectura de la aplicación.....	49
Ilustración 21. Diagrama E/R Base de Datos.....	53
Ilustración 22. Diagrama de interacción Modelo-Vista-Controlador-Pasivo.....	77
Ilustración 23. Clases del subsistema.....	77
Ilustración 24. <i>eTraveler</i>	89
Ilustración 25. <i>TravelCobber</i> 1.0.....	89
Ilustración 26. Distribución de los usuarios por sexos.....	93
Ilustración 27. Distribución de los usuarios por temperamentos.....	93
Ilustración 28. Número de visitas en función de la edad.....	94
Ilustración 29. Opiniones respecto a la interfaz gráfica.....	95
Ilustración 30. Opiniones respecto a las opciones de la aplicación.....	95
Ilustración 31. Opiniones respecto a la rapidez de la aplicación.....	96
Ilustración 32. Opiniones respecto al resultado del test de temperamento.....	96
Ilustración 33. Opiniones respecto a la búsqueda por temperamento.....	97
Ilustración 34. Opiniones respecto a la aplicación.....	97
Ilustración 35. Satisfacción en función de la edad.....	98
Ilustración 36. Página de entrada a la aplicación <i>TravelCobber</i>	102
Ilustración 37. Inicio de sesión en <i>Facebook</i>	103
Ilustración 38. Página principal de la aplicación <i>TravelCobber</i>	103
Ilustración 39. Resultado de una búsqueda en la aplicación <i>TravelCobber</i>	104

1 Introducción

Una recomendación es una sugerencia que una persona ofrece a otra, sobre la conveniencia de realizar una acción, como por ejemplo adquirir un bien. Dicha sugerencia ayuda a que la persona decida que es lo que va a realizar. Las recomendaciones se realizan prácticamente en todas las actividades de nuestra vida. Podemos recibir recomendaciones sobre viajes, películas, libros, música, estudios, formas de actuar,...etc. Generalmente una persona necesita una recomendación, cuando no esta del todo segura sobre la acción que va a realizar.

En el campo de las aplicaciones Web, los sistemas de recomendación han cobrado una gran importancia en los últimos años, sobre todo en las aplicaciones Web comerciales. Las empresas que tienen sus negocios en Internet, se han dado cuenta de que el porcentaje de personas que visitan su Web y la dejan sin haber realizado ninguna compra es muy alto. Además el número de páginas que un usuario suele navegar por una web es muy bajo. Para ilustrar este problema en la (Fig. 1) se muestra un ejemplo de las estadísticas de acceso de los usuarios a una aplicación web comercial durante un mes. En dicho ejemplo el usuario medio permanece poco tiempo (90 segundos) y navega por pocas páginas (6 páginas por visita) antes de abandonar el sitio web. Por ello se invierte gran cantidad de recursos en construir sistemas de recomendación que ofrezcan al cliente una serie de productos que encajen con sus gustos y necesidades, con el objeto de realizar la venta. Por una parte facilitan que el usuario encuentre el producto que puede necesitar al reducir el espacio de búsqueda. Por otro lado intentan que el usuario encuentre rápidamente algo que le guste antes de que abandone la aplicación web, aumentando de esta manera la probabilidad de la venta. Desde este punto de vista los sistemas de recomendación serían herramientas empleadas como técnicas de venta y marketing.



Fig. 1. Gráfica y estadísticas de acceso a una aplicación web comercial.

Los sistemas de recomendación son también muy importantes en aplicaciones web de empresas o entidades que si bien no venden ningún

producto, si están interesadas en tener un gran número de visitas (en el caso que reciban ingresos por publicidad), o en ofrecer un mejor servicio a sus usuarios como llave al éxito del sitio web.

Al recomendar al usuario un producto de calidad y que encaje con sus gustos o necesidades, es probable que la satisfacción final del cliente por el objeto comprado sea mayor. La recomendación ayuda a que el usuario pueda encontrar lo que estaba buscando más rápida y fácilmente al reducir el espacio de búsqueda.

Los sistemas de recomendación web combinan ideas de recuperación y filtrado de información, modelado del usuario, inteligencia artificial e interacción hombre-maquina [Bridge-2].

Un gran número de productos son usados de forma grupal por las personas. La recomendación de productos para grupos es más compleja (incluso en la vida diaria), y presenta una serie de desafíos adicionales.

Los sistemas de recomendación actuales presentan una serie de limitaciones.

El *arranque en frío* [7] de los nuevos usuarios de los que no se dispone de la suficiente información para poder determinar que es lo que le puede gustar. El *arranque en frío del producto* sin las suficientes valoraciones para poder recomendarlo adecuadamente. Las *valoraciones inconsistentes* de los usuarios, y los frecuentes *cambios en las preferencias* de los usuarios con el paso del tiempo.

Los sistemas de recomendaciones para grupos al estar formados por agregaciones de los modelos individuales de los miembros del grupo, o agregaciones de las recomendaciones individuales para cada uno de los miembros del grupo, presentan las mismas limitaciones que los recomendadores individuales. Pero además presentan dos cuellos de botella adicionales. En la fase inicial cada uno de los miembros del grupo tiene que registrarse en la aplicación web que recomienda el producto y tiene que invertir tiempo y esfuerzo introduciendo sus datos personales, datos de su perfil y definir cuales son sus preferencias o los productos que le interesan. Además dichos sistemas deben proporcionar a los usuarios mecanismos de comunicación y negociación para que puedan ponerse de acuerdo en la aceptación final de una recomendación.

Para resolver el problema del arranque en frío del usuario en sistemas de recomendaciones para grupos como *MusicFx*[20] o *POLYLENS*[22], el usuario tiene que realizar una encuesta previa de preferencias sobre los contenidos recomendados. En el sistema de recomendación de sitios web para grupos *Let's Browse* [21] si existe una adquisición implícita de las preferencias individuales y grupales, pero al igual que en el sistema de recomendación de itinerarios turísticos para grupos *INTRIGUE*[23] no existe negociación de grupo y la decisión es tomada por un solo miembro.

En el sistema de recomendaciones para grupos *CATS* [16] si se tiene en cuenta la interacción entre los usuarios para llegar a la solución de grupo. Sin embargo su sistema tiene como inconveniente que el espacio de

interacción y negociación es un tablero táctil, por lo que se requiere la presencia física de todos los integrantes del grupo alrededor del mismo tablero. Limitando el número de personas que pueden usar simultáneamente la aplicación. En el sistema de recomendaciones de viajes para grupos *Travel Decision Forum*[4] también se tiene en cuenta la interacción entre los miembros del grupo para llegar a la solución final. Además los miembros del grupo pueden ver las preferencias de los demás, y las preferencias individuales tienen explicaciones textuales. Sin embargo presenta como inconveniente que los usuarios se deben registrar en la aplicación, introducir sus datos y especificar sus preferencias.

Por otra parte en este sistema al igual que en todos los sistemas anteriores se trata a todos los usuarios de la misma manera, sin tener en cuenta su personalidad, temperamento, ni estado de ánimo. Al tratar a todos los usuarios de la misma forma, es poco probable que todos queden igual de satisfechos con el sistema. En el estudio *Gómez-Gauchia* [1] se refleja que es importante capturar el estado de ánimo del usuario para ayudar a mejorar la interacción hombre-máquina y hacerla más intuitiva, suave y eficiente. Además detectar los cambios en el estado de ánimo del usuario puede servir para prevenir posibles bloqueos del usuario (*breakdowns*).

En el estudio de *Recio-García et al.*[8] se tiene en cuenta la personalidad en situaciones de conflicto de los distintos integrantes de un grupo. Sin embargo presenta el inconveniente de que los usuarios tienen que realizar una valoración previa de 50 películas y además tampoco resuelve el problema de la negociación grupal final.

Las redes sociales pueden ayudarnos a paliar estas limitaciones.

En los perfiles de los usuarios de las redes sociales podemos encontrar una gran cantidad de información que nos puede servir para solventar el problema del registro de los datos del usuario y de su arranque en frío.

En el perfil del usuario suele aparecer además de su nombre, su género, su edad, su localización, estado civil, incluso en algunos casos preferencias políticas, religiosas, y un listado de productos, actividades o lugares que les gustan.

La motivación social hace que los datos que aparecen en sus perfiles sean consistentes, y como suelen usar de forma frecuente la Red Social es probable que estos datos estén actualizados.

En vez de que un usuario tenga que registrar todos sus datos y preferencias en cada una de las aplicaciones web que utilizan un sistema de recomendación, sería preferible que el sistema de recomendación accediese a esos datos que el usuario ya ha definido en la Red Social, que son consistentes y se encuentran actualizados. Además en la Red Social los usuarios ofrecen mucha más información sobre sí mismos que la que están dispuestos a ofrecer en una aplicación web comercial.

Para intentar resolver las limitaciones de los sistemas de recomendación actuales, proponemos un modelo que integra el sistema de recomendación en una Red Social. Teniendo en cuenta además el temperamento y el estado de ánimo de los usuarios, para mejorar su interacción con el sistema y poder adaptarse mejor a las características del usuario para aumentar su

satisfacción con el sistema. En la implementación del modelo hemos utilizado técnicas RIA para mejorar la experiencia del usuario en la aplicación.

En la sección 2 exponemos el estado del arte de los sistemas de recomendación individuales, los sistemas de recomendación para grupos, y los sistemas de recomendación y aplicaciones en la Red Social Facebook. Hemos realizado una investigación previa sobre el estado del arte de estos sistemas porque como nuestro modelo se basa en ellos. Necesitábamos saber cuál es el estado actual de los mismos, sus ventajas e inconvenientes, para que nuestro modelo pudiera utilizar las ventajas y buenas ideas presentes en las soluciones actuales, pero intentando mejorar algunos de los inconvenientes que hemos detectados en los mismos.

En la sección 3 presentamos el modelo de integración del sistema de recomendación en la Red Social. En la sección 4 se encuentran los detalles sobre su implementación. Los resultados obtenidos al implementar el modelo propuesto se encuentran en la sección 5. Y para finalizar en la sección 6 enumeramos las conclusiones a las que hemos llegado al realizar esta investigación, así como las futuras extensiones que puede tener nuestro trabajo.

2 Estado del Arte

2.1 Sistemas de Recomendación individuales

Los sistemas de recomendación ofrecen al usuario una serie de elementos que pueden resultar de su interés, ayudándole en la toma de decisiones en el caso de volúmenes de información muy grandes.

2.1.1 Clasificación de sistemas de Recomendación

Clasificación en función de cómo se construyen las recomendaciones

a) Colaborativos

Para generar las recomendaciones se basan en la historia del usuario, principalmente en sus perfiles de valoración de productos. Las recomendaciones también se basan en los perfiles de compañeros de puntuación del usuario. Estos compañeros de puntuación son usuarios cuyas calificaciones de elementos se asemejan a las del usuario [*Bridge-2*]. Recomendara elementos que hayan obtenido altas puntuaciones de sus compañeros, aunque no figuren en el perfil del usuario.

Inconveniente:

- Previamente los elementos han de ser puntuados por los usuarios (arranque en frío del producto).
- Las valoraciones de los usuarios pueden ser inconsistentes [*Tintarev-7*].
- Se necesita conocer el perfil del usuario (Arranque en frío del usuario).

Ventajas:

- No se requiere descripción de los productos.
- Pueden captar detalles sutiles de los productos [7].
- Serendipity [7]: "Cuando una persona descubre accidentalmente algo que le gusta, mientras estaba buscando otra cosa que no tiene relación con lo encontrado"[*wikipedia*].

b) Basados en Casos

El razonamiento basado en casos (CBR) es una metodología para la resolución de problemas basada en la reutilización de conocimiento específico de experiencias previas y situaciones concretas del problema [*Aamodt-5*].

En un sistema basado en Casos se distinguen cuatro etapas [*Ontañon-6*]:

- *Recuperar*: Selección de casos relevantes para el problemas actual.
- *Reutilizar*: Adaptación de la solución de los casos al problema actual
- *Revisar*: Un experto examina la solución escogida, y determina si es correcta.
- *Recordar*: El sistema decide si incorpora la solución correcta del caso en la base de casos.

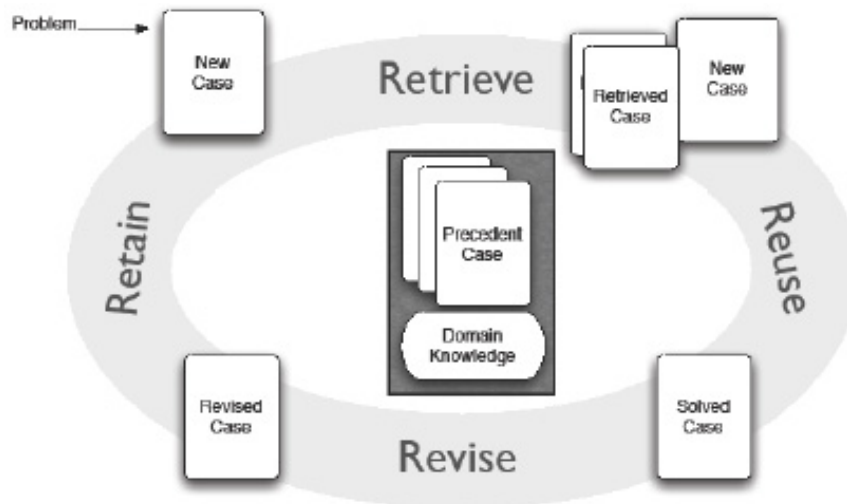


Fig. 2. Ciclo del razonamiento basado en casos.

El enfoque basado en casos es especialmente útil cuando el espacio de búsqueda es muy grande, y explorar todas las posibles soluciones del problema puede resultar demasiado costoso [6].

En los sistemas de recomendación basados en casos se utiliza solamente la primera etapa del ciclo, la recuperación de casos relevantes basada en similitud. Los casos son las descripciones de los elementos. El sistema genera una serie de recomendaciones para un usuario, recopilando los elementos que mejor se ajusten a la consulta del usuario, es decir los casos (descripciones de los productos) que tengan una mayor similitud con la consulta del usuario. Los casos son las descripciones de los productos, y las recomendaciones ofrecidas son los casos con mayor similitud a mí consulta.

Ventajas:

- No es necesario que los elementos hallan sido puntuados previamente por los usuarios para poder ser recomendados.
- Los elementos recomendados están ordenados por similitud, lo que sirve también para limitar el número de elementos, al no mostrar los elementos que tienen baja similitud con la consulta.

Inconvenientes:

- Se requiere descripción de los productos
- Arranque en frío del usuario [7], se necesita saber que aspectos son importantes para el usuario, lo cual requiere tiempo.
- Las preferencias del usuario pueden cambiar.
- Carencia de serendipity [7]. No recomiendan elementos diferentes a la consulta realizada por el usuario, estos elementos diferentes pueden ser productos que le hubiesen podido gustar al usuario.

Ejemplos de uso: Analog Devices

c) Híbridos

Combinan las técnicas colaborativas y basadas en casos para aumentar la exactitud de sus recomendaciones

Ejemplos de uso: PTVPlus

d) Bases de Datos

Basado en consultas a la base de datos de la aplicación web.
Inconveniente: No tiene aprendizaje.

e) Satisfacción de restricciones

Recomienda al usuario los elementos que satisfacen las restricciones establecidas por el usuario. [Vollrath 1998; Wilke 1998]
Inconveniente: Si el usuario introduce demasiadas restricciones, es probable que ningún elemento las cumpla. Si se introducen pocas restricciones, se ofrecerán como recomendaciones demasiados elementos.

Clasificación en función de su interacción con el usuario

En función de la forma en la que interaccionan con el usuario podemos considerar dos tipos de sistemas de recomendación:

a) *Reactivos*

El usuario realiza una consulta y el sistema de recomendación reacciona devolviendo las recomendaciones.

b) *Proactivos*

Se ofrecen las recomendaciones al usuario sin que este haya realizado una consulta previa.

Tipo	Interacción	
Reactivos	<ul style="list-style-type: none"> El usuario realiza una consulta y el sistema de recomendación reacciona devolviendo las recomendaciones. 	
Proactivos	<ul style="list-style-type: none"> Se ofrecen las recomendaciones al usuario sin consulta previa. 	

Tabla. 1. Clasificación sistemas de recomendación en función de su interacción con el usuario.

Clasificación en función de la estrategia de recomendación

- a) **Single-shot:** Ofrecen una serie de sugerencias por sesión.
- b) **Conversacional:** El usuario elabora sus necesidades, como parte de un dialogo de recomendación.

Tablas resumen clasificación sistemas de recomendación

Tipo	Solución	Recomendaciones
Colaborativos	<ul style="list-style-type: none"> Perfil del usuario (ratings). Perfiles compañeros de puntuación del usuario (usuarios cuyas calificaciones de elementos se asemejan a las del usuario [Bridge-2]) 	<ul style="list-style-type: none"> Recomendará elementos que hayan obtenido altas puntuaciones de sus compañeros, aunque no figuren en el perfil del usuario
Basados en Casos	<p>Útil: Espacio de búsqueda muy grande, explorar todas posibles soluciones del problema → costoso [Ontañón-6].</p> <p>CBR [5]</p> <ol style="list-style-type: none"> 1) Recuperar 2) Reutilizar 3) Revisar 4) Recordar 	<ul style="list-style-type: none"> Reutilización de conocimiento específico de experiencias previas y situaciones concretas del problemas [Aamodt-5] Recuperación de casos relevantes basada en similitud Elementos que mejor se ajusten a la consulta del usuario, es decir los casos (descripciones de los productos) que tengan una mayor similitud con la consulta del usuario.
Híbridos	Colaborativo + Basado en casos	<ul style="list-style-type: none"> Se beneficia de las ventajas de ambos enfoques.
Restricciones	Satisfacción de restricciones	<ul style="list-style-type: none"> Recomienda usuario elementos que satisfacen las restricciones establecidas por usuario. [Vollrath 1998; Wilke 1998] Inconveniente: demasiadas restricciones → ningún elemento las cumpla. pocas restricciones → demasiados elementos.
Bases Datos	Consultas a la BD	<ul style="list-style-type: none"> No tiene aprendizaje.

Tabla. 2. Clasificación sistemas de recomendación en función de la técnica empleada para construir las recomendaciones.

Tipo	Ventajas	Inconvenientes
Colaborativos (Amazon)	<ul style="list-style-type: none"> ✓ No se requiere descripción de los productos. ✓ Capturan detalles sutiles de productos [Tintarev-7] 	<ul style="list-style-type: none"> ✓ Se necesita un alto n° de valoraciones previas para buenas predicciones. ✓ Arranque frío producto: Previamente los productos han de ser puntuados por los usuarios ✓ Arranque frío usuario: Necesita conocer el perfil del usuario . ✓ Valoraciones inconsistentes de usuarios [7]. ✓ Poca diversidad [18] ✓ "Oveja gris" : Perfil gris del usuario.
Basados en Casos (LIBRA - libros, Syskill y Webert - webs)	<ul style="list-style-type: none"> ✓ No requiere valoración previa productos. ✓ Ordenación por similitud → limita el n° de recomendaciones ofrecidas. 	<ul style="list-style-type: none"> ✓ Arranque frío descripciones: Necesita la descripción de los productos ✓ Arranque frío usuario: Necesita conocer el perfil del usuario . ✓ Las preferencias del usuario pueden cambiar [Tintarev-7]. ✓ Carencia de serendipity, poca diversidad [18]
Híbridos (Google, yahoo)	<ul style="list-style-type: none"> ✓ Ventajas de ambos enfoques. 	<ul style="list-style-type: none"> ✓ Problemas de ambos enfoques. ✓ Establecer la proporción correcta entre colaboración/casos.

Tabla. 3. Ventajas e inconvenientes de los sistemas de recomendación.

2.1.2 Métricas usadas en Sistemas de Recomendación

Existen dos tipos de métricas [Cantador-18], las que se usan para medir la precisión de las recomendaciones (error absoluto medio, error cuadrático medio, precisión, recall, F-measure, receiver operating characteristic curve), y otras métricas que se utilizan para medir la capacidad de generar recomendaciones útiles y de calidad para todos los productos (Cobertura, Novedad, Serendipity, Tasa de aprendizaje, Confianza). Sin embargo a nivel académico no se introducen métricas para determinar si el sistema de

recomendación incrementa el número y tiempo de visitas a un producto y/o la tasa de conversión del mismo.

Tipo	Métricas
<p>PRECISIÓN</p> <p>Miden la precisión de las recomendaciones (y el soporte al usuario para obtener un producto de calidad)</p>	<ul style="list-style-type: none"> ❖ Error absoluto medio, Error cuadrático medio ❖ Precisión: % que un producto recomendado sea verdaderamente relevante ❖ Recall: % de que un producto relevante sea recomendado como relevante. ❖ F-measure: armonía entre precisión y recall. ❖ Receiver Operating Characteristic curve: compromiso entre presentar un alto número de items relevantes y unos pocos no relevantes <p>(Aplicables a producto que los usuarios eligen valorar, porque le gustan)</p>
<p>OTRAS</p> <p>(Miden la capacidad de generar recomendaciones útiles y de calidad para todos los productos)</p>	<ul style="list-style-type: none"> ❖ Cobertura: nº de productos que pueden ser recomendados ❖ Novedad y serendipity ❖ Tasa de aprendizaje: Como de rápido un sistema genera buenas recomendaciones. ❖ Confianza: Seguridad del sistema en la precisión de sus recomendaciones.

Tabla. 4. Tipos de métricas usados en los sistemas de recomendación.

2.2 Sistemas de Recomendación para grupos de usuarios

Las recomendaciones para grupos se aplican en sitios web donde los usuarios pueden especificar sus preferencias.

Gran parte de los productos que se recomiendan (películas, restaurantes, viajes) son usados de manera grupal en vez de individualmente, por lo que se han desarrollado sistemas de recomendación que ajusten sus sugerencias a las preferencias de ese grupo de usuarios. Se deben agregar los modelos o predicciones de cada usuario individual para llegar a una recomendación grupal.

Existen diferentes alternativas de sistemas que ofrecen recomendaciones para grupos.

Autores	Propuesta
<p>McCarthy et al.[20]</p>	<p><i>MusicFx:</i> Determina la música de ambiente que suena en un gimnasio, basándose en las preferencias de los usuarios que en ese momento se encuentran en el gimnasio. El sistema selecciona uno de los 91 canales musicales disponibles. También introducen algunas canciones al azar para introducir algo de variedad en su selección.</p>
<p>Lieberman et al.[21]</p>	<p><i>Let´s Browse:</i> Recomienda páginas web a grupos de personas que están navegando por la red juntas. Realiza una estimación implícita de los intereses de cada individuo analizando sus páginas web.</p>

O' Connor et al. [22]	<i>POLYLENS</i> : Recomienda películas a grupos de personas. http://www.movieLens.umn.edu
Ardissono et al.[23]	<i>INTRIGUE</i> : Recomienda tours a un guía de un grupo heterogéneo de turistas, que pueden contener subgrupos homogéneos como por ejemplo niños. Para realizar la recomendación grupal tiene en cuenta los intereses de los distintos subgrupos que lo componen.
Jameson [4]; Jameson et al. [14]	<i>Travel Decisión Forum</i> : Ayuda a que un grupo de personas lleguen a un acuerdo sobre las características de las vacaciones que van a realizar juntos.
Smyth et al. [24]	<i>I-SPY</i> : Es un motor de búsquedas web que personaliza sus resultados para una comunidad de personas con los mismos gustos, basándose en el modelo de las preferencias de búsqueda de dicha comunidad. El sistema en sí no es el encargado de proporcionar los resultados de las búsquedas. Reordena los resultados de las búsquedas ofrecidas por otro motor de búsqueda, en función de las preferencias de búsqueda de la comunidad (las consultas y selecciones hechas por sus miembros). Es un ejemplo de especificación y agregación de las preferencias de un grupo.
Recio-García et al.[8]	Recomendación de películas: Agregación de las preferencias individuales para llegar a la recomendación grupal. Las recomendaciones individuales de cada usuario son ponderadas en función de la personalidad de cada individuo a la hora de realizar la agregación grupal.
McCarthy et al.[16]	<i>CATS</i> : Sistema que permite que un grupo de personas lleguen a un acuerdo hacer del lugar y características del sitio donde van a ir a practicar esquí.

Tabla 5. Sistemas de recomendación para grupos.

Sistema	Ventajas	Inconvenientes
MusicFx McCarthy et al.[20]	- Serendipity: Los canales musicales cambian al azar a partir de los géneros musicales más valorados por los usuarios actuales.	- No es un sistema web. - Encuesta previa preferencia 91 canales música (Requiere Tiempo). - No se ven ni se pueden copiar las preferencias de otro usuario. (No especificación colaborativa de preferencias individuales).

<p>Let´s Browse Lieberman et al.[21]</p>	<p>- Adquisición implícita de las preferencias individuales y grupales.</p>	<p>- No existe negociación grupal final. - La decisión final es tomada por un solo miembro del grupo.</p>
<p>POLYLENS O´Connor et al. [22]</p>	<p>- Test de valoración de películas previo, para extraer los gustos individuales.</p>	<p>- No especificación colaborativa de preferencias individuales</p>
<p>INTRIGUE Ardissono et al.[23]</p>	<p>- Permite favorecer subgrupos de personas (discapacitados, niños)</p>	<p>- No hay negociación grupal final. - La decisión final es tomada por el guía turístico que usa la aplicación</p>
<p>Travel Decisión Forum Jameson [4]; Jameson et al. [14]</p>	<p>- Los miembros del grupo pueden ver las preferencias individuales de los demás. (ahorro tiempo y facilita acuerdo común) - Las preferencias individuales tienen explicaciones textuales - Tiene en cuenta la interacción entre usuarios para llegar a la solución grupal</p>	<p>- Los miembros del grupo deben registrar en la aplicación, introducir sus datos y especificar sus preferencias individuales. - No tiene en cuenta personalidad ni temperamento de los usuarios.</p>
<p>I-SPY Smyth et al. [24]</p>	<p>- Especificación y agregación de las preferencias de un grupo de personas.</p>	<p>- No hay negociación grupal final. - Se trata a todos los usuarios por igual</p>
<p>Recio-García et al.[8]</p>	<p>- Tiene en cuenta la personalidad de los usuarios a la hora de generar las recomendaciones grupales. Es decir, no se trata a todos los usuarios de la misma manera.</p>	<p>- No hay negociación grupal final. - Los usuarios tienen que realizar una valoración previa de 50 películas.</p>
<p>CATS McCarthy et al.[16]</p>	<p>- Tiene en cuenta la interacción entre usuarios para llegar a la solución de grupo.</p>	<p>- Requiere la presencia física de los usuarios alrededor del tablero táctil. - El tablero táctil limita el número de personas que pueden usar simultáneamente la aplicación. - No tiene en cuenta personalidad ni temperamento de los usuarios.</p>

Tabla 6. Ventajas e inconvenientes de los Sistemas de recomendación para grupos.

Sistema	Agregación Recomendaciones individuales	Agregación Modelos individuales	Negociación Grupal	Personalidad	Temperamento
MusicFx	X	0	0	0	0
Let´s Browse	0	X	0	0	0
POLYLENS	X	X	0	0	0
INTRIGUE	0	X	0	0	0
Travel Decisión Forum	0	X	X	0	0
I-SPY			0	0	0
Recio-García et al.[8]	X	0	0	X	0
CATS	X	0	X	0	0

Tabla 7. Análisis Sistemas de recomendación para grupos.

2.2.1 Fases del proceso de recomendación

En el estudio de [Jameson-4] se establecen cuatro particularidades de los sistemas de recomendación grupales que deben ser tenidos en cuenta en el diseño del sistema:

Fases del proceso de recomendación	Diferencias en la Recomendación grupal	Aspectos a desarrollar
Fase 1 Especificación de las preferencias [Usuarios]	Los miembros del grupo deben poder ver las preferencias de los demás miembros	Presentación al grupo de las preferencias individuales
Fase 2 Generación de la recomendación [Sistema]	Agregación de las preferencias de los usuarios	Como agregar las preferencias individuales
Fase 3 Presentación de la recomendación [Sistema]	La recomendación debe poder ser aceptable para cada individuo del grupo	Como presentar/tratar las restricciones de cada individuo

Fase 4 Aceptación recomendación [Usuarios]	Negociación grupal de la decisión.	Dar soporte a la negociación final en el grupo
---	---------------------------------------	--

Tabla. 8. Diferencias en el proceso de recomendación para grupos.

2.2.1.1 Presentación al grupo preferencias individuales

Formas de obtener las preferencias de cada individuo:

- **Explícitamente:** Las preferencias de cada individuo pueden ser introducidas explícitamente por cada uno de los individuos.
- **Implícitamente:** El sistema extrae las preferencias del individuo basándose en la información que dispone o puede obtener de él.

Por ejemplo, el sistema *Let's Browse* [Lieberman et al.- 21] extrae implícitamente las palabras en las que puede estar interesado cada usuario del grupo para realizar un búsqueda, analizando las palabras presentes en la página inicial de su sitio web. Posteriormente, en las búsquedas que realiza el grupo también analiza las palabras buscadas.

Sin embargo en otros sistemas como por ejemplo *MusicFx* [McCarthy et al. - 20] los usuarios tienen que especificar explícitamente en una encuesta su grado de satisfacción con cada uno de los 91 canales disponibles.

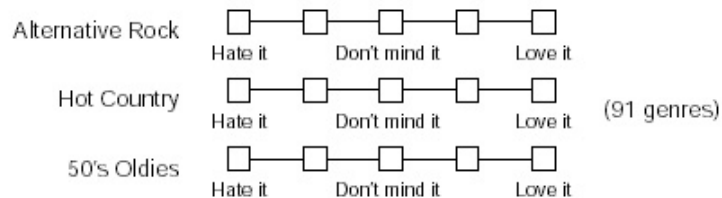


Fig. 3. Especificación explícita de las preferencias individuales en el sistema *MusicFX*.

Sistema	Adquisición Preferencias Individuales
MusicFx	- Cada individuo responde explícitamente una encuesta como le gusta cada uno de los 91 canales musicales disponibles. - No puede ver las preferencias de los demás.
Let's Browse	- Adquisición implícita de las preferencias individuales y grupales.
POLYLENS	- No se les pregunta explícitamente su valoración de cada película. - Valoran de 1 a 5 una serie de películas para extraer de forma implícita sus preferencias.

Travel Decisión Forum	- Los usuarios cumplimentan explícitamente sus preferencias a cerca de los atributos de cada uno de los destinos vacacionales.
I-SPY	- Combinación de selección explícita e implícita. - Cada usuario indica explícitamente su interés en el resultado de una búsqueda seleccionándolo explícitamente en la lista de resultados. - El sistema interpreta las selecciones de resultados de los usuarios, para establecer la relevancia de los resultados respecto a los términos buscados.
Recio-García et al.[8]	- Los usuarios rellenan en unas encuestas sus datos personales y preferencias individuales a cerca de películas. - Valoración de 50 películas heterogéneas.
CATS	- Valoración de los ítems en el tablero táctil.

Tabla. 9. Adquisición de las preferencias individuales.

Es recomendable que los miembros del grupo puedan ver las preferencias de cada miembro del grupo. También es conveniente que las preferencias tengan explicaciones para que los otros miembros del grupo las puedan entender mejor.

Ventajas:

- **Ahorro de esfuerzo:**
Introducir las preferencias individuales requiere tiempo, por lo que si un usuario tiene unas preferencias iguales a otro en cierto aspecto podría copiar esas preferencias y ahorrarse el tener que introducirlas
- **Aprender de los otros usuarios:**
Las preferencias de un usuario basadas en una experiencia o conocimiento previo pueden servir de guía u orientación a los otros miembros.
- **Dependencia de comportamientos:** Las preferencias de algún usuario pueden depender de las de otro usuario.
- **Acuerdo común:** Ver las preferencias de los demás facilita bajar las restricciones personales para llegar a un acuerdo común.

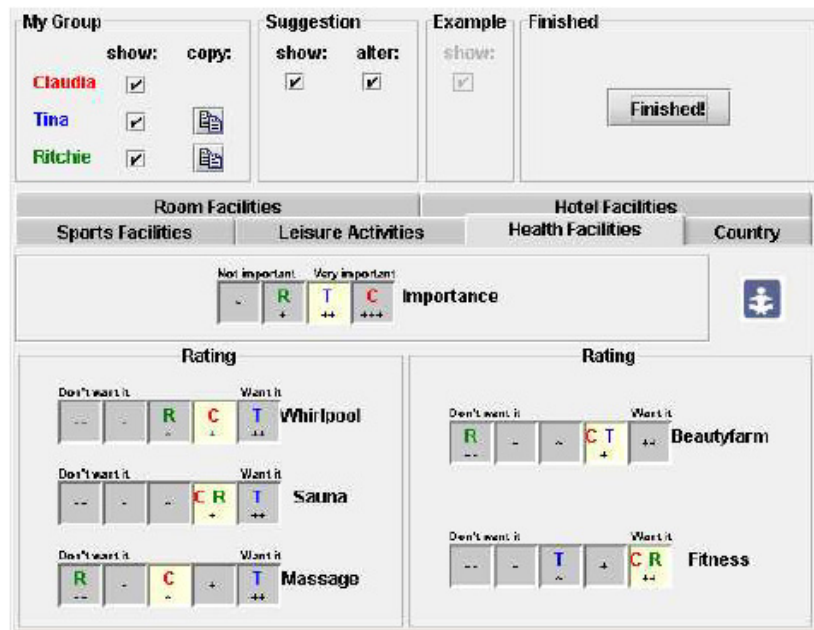


Fig. 4. Especificación explícita de las preferencias individuales en el sistema *TRAVEL DECISION FORUM*. La persona que cumplimenta sus preferencias individuales puede ver y copiar las preferencias del resto de miembros de su grupo.

En el sistema *TRAVEL DECISION FORUM* al rellenar las preferencias individuales se pueden ver y copiar las preferencias del resto de miembros del grupo. En el sistema *I-SPY* hay también cierto grado de especificación colaborativa de las preferencias individuales en la medida que sus usuarios pueden ver como las selecciones que hacen los otros miembros afectan a los resultados finales. Pudiendo de esta forma copiar la selección hecha por algún otro miembro del grupo.

En el estudio de [Jameson et al.-14] se recomienda que los sistemas de recomendación para grupos dispongan de una especificación colaborativa de las preferencias individuales.

2.2.1.2 Agregación de las preferencias y modelos individuales

El objetivo de la agregación de las preferencias individuales es obtener un modelo de las preferencias del grupo que permita llegar a un acuerdo grupal.

Existen dos alternativas para realizar la agregación de las preferencias individuales:

- **Agregación de las recomendaciones individuales:**
Las recomendaciones para el grupo se contruyen agregando las recomendaciones individuales para cada uno de sus individuos.
- **Agregación de los modelos individuales:**
Se construye un modelo del grupo a partir de los modelos individuales. Se ofrecen recomendaciones para ese modelo de grupo.

En algunos estudios se realiza una agregación de las preferencias individuales para obtener un modelado del grupo [McCarthy et al-16], en otros se realiza una agregación de las recomendaciones individuales para obtener las recomendaciones grupales [Recio-García et al-8].

Agregación de las preferencias individuales

El sistema debe poder determinar la valoración que cada usuario haría de los objetos recomendables.

Para cada individuo:

- Construir las valoraciones que cada individuo haría de cada objeto.
- Realizar un agregado de las valoraciones de individuales para cada objeto.

Se recomiendan los objetos que con mayor valoración total.

En el estudio de [Recio-García et al-8] las recomendaciones individuales de cada usuario son ponderadas en función de la personalidad de cada individuo a la hora de realizar la agregación grupal. Primero realizan un test TKI (*Thomas-Kilman Conflict Mode Instrument*) a cada individuo para determinar si su personalidad en situaciones de conflicto es:

- **Competitiva** (busca satisfacer sus objetivos).
- **Colaborativa** (tienen un objetivo de grupo).
- **Ausente** (evita tomar decisiones).
- **Acomodada** (lo que digan lo demás).
- **Comprometida** (moderadamente competitivo y colaborativo).

Las recomendaciones de las personas competitivas o colaborativas tendrán más peso a la hora de formar la recomendación grupal que la de las personas con personalidades ausentes o acomodadas.

En *POLYLENS* [O'Connor et al.-22] se obtiene la valoración del grupo para cada película, realizando un agregado de las valoraciones de cada uno de los individuos para esa película. Para ello aplican la estrategia "Least Misery" (ver Sección 2.2.1.3).

TITLE	GENRE	REVIEWS	GROUP	YOUR	casley@cs.uma.edu	casley@quasar
Pixote (1981)	Drama		★★★★★	★★★★★	★★★★★	★★★★★
Wrong Trousers, The (1993)	Animation, Comedy		★★★★★	★★★★★	★★★★★	★★★★★
After Life (1998)	Drama		★★★★☆	★★★★☆	★★★★☆	★★★★★
King of Masks, The (Bian Lian) (1996)	Drama		★★★★☆	★★★★★	★★★★☆	★★★★★

Fig. 5. Agregación de las valoraciones individuales en *POLYLENS*. Las tres columnas de la derecha son las valoraciones de los miembros del grupo para cada película. La columna GROUP muestra la valoración de grupo formada agregando las valoraciones individuales.

En el estudio [McCarthy et al.-17] se concluye que la agregación de las preferencias individuales para formar las preferencias de grupo ofrece mejores resultados que utilizar solamente los sistemas de recomendación para usuarios individuales. Y que las recomendaciones grupales generadas de esta manera es probable que satisfagan tanto a todos los miembros del grupo como a cada uno de sus miembros de forma individual.

Agregación de los modelos individuales

En vez de agregar las valoraciones individuales, se construye un modelo del grupo a partir de los modelos individuales. Y se ofrecen recomendaciones para ese modelo de grupo.

Etapas:

1. Construcción del modelo M del grupo.
2. Para cada objeto C_i generan la valoración V_i del grupo basandose en el modelo M del grupo.
3. Se recomiendan los objetos con mayor valoración V_i .

Ventajas:

1. Los miembros del grupo pueden examinar y negociar la formación del modelo de grupo.
2. Privacidad. Se evita tener que almacenar y mantener perfiles de usuarios individuales y los problemas legales que ello implica.

Sistema	Agregación Preferencias
MusicFx	<ul style="list-style-type: none"> - Se construye un modelo del grupo de personas que están en el gimnasio en cada momento. El modelo se actualiza con la llegada o salida de cada persona. - Cada determinar la preferencia grupal para cada género, realiza el promedio de las raíces de las valoraciones que para cada género han realizado los usuarios actuales. - Elección al azar entre los géneros más populares.
Let´s Browse	Combinación lineal modelos individuales.
POLYLENS	<ul style="list-style-type: none"> - Agregación de las valoraciones individuales. - Método: Least Misery
INTRIGUE	<ul style="list-style-type: none"> - No construye modelos ni valoraciones individuales. - Se construyen modelos de preferencias de subgrupos. - El modelo de grupo se forma por la media ponderada de los modelos de los subgrupos. <p>Por ejemplo las personas discapacitadas tienen un mayor peso para favorecer sus valoraciones.</p>
Travel Decisión Forum	Los usuarios son los que construyen el modelo de preferencias de grupo al rellenar los formularios.
I-SPY	<ul style="list-style-type: none"> - No existen modelos individuales. - El modelo de grupo es el conjunto formado por las consultas realizadas por el grupo y las páginas de resultados seleccionadas de esas consultas. - A la hora de generar resultados para una nueva consulta, se estima su relevancia para el grupo basándose en la frecuencia con la que la página fue seleccionada por miembros del grupo para esa consulta o similares.

Recio-García et al.[8]	- Agregación de las recomendaciones individuales para obtener las recomendaciones grupales. - El peso de las recomendaciones de cada individuo son ponderadas por la personalidad del individuo en situaciones de conflicto (test TKI)
CATS	Agregación de las preferencias individuales para obtener un Modelado del grupo.

Tabla 10. Agregación de las preferencias individuales.

2.2.1.3 Métodos de Agregación

La elección del método de agregación usado se debe basar en los objetivos que queremos alcanzar o satisfacer con la agregación.

- R_i**: Valoración del objeto i
- C_i**: Objeto candidato i
- M_j**: usuario j
- R_{ij}**: valoración que el usuario j hace del objeto i

Maximizar la satisfacción Media

Este método maximiza la satisfacción media promedio de los miembros del grupo con la recomendación propuesta.

$$R_i = 1/n * \sum_{j=1}^n r_{ij}$$

Minimizar la disconformidad

Su objetivo es minimizar la disconformidad de los miembros con las recomendaciones propuestas. De esta forma se evita que algún miembro del grupo se muestre insatisfecho con una recomendación que si bien puede ser la que tenga mayor valoración grupal a él no le guste. Pudiendo llegar a vetar la recomendación propuesta. (Ej: *PolyLens*)

$$R_i = \min_j r_{ij}$$

Asegurar cierto grado de equilibrio

Este método persigue asegurar cierto grado de equilibrio o justicia en las recomendaciones que se generan. Es preferible una solución que satisfice a todos los miembros del grupo de manera similar, a una que satisfice mucho a unos, pero no tanto a los demás.

Para ello a la valoración media de cada objeto se le resta su desviación media respecto a las valoraciones individuales multiplicado por un peso (w) que refleja como de equilibrada queremos que sea la solución.

$$R_i = 1/n * \sum_{j=1}^n r_{ij} - w * \text{desviación}(r_{ij})$$

w : peso que pondera la importancia del equilibrio en la solución

Evitar la manipulación

Se debe evitar que los miembros del grupo puedan manipular el proceso de agregación para que las recomendaciones propuestas les satisfagan. Por ejemplo si un miembro del grupo puede ver las preferencias de los demás, puede modificar las suyas originales para que el modelo de grupo se asemeje a su gusto en vez de a la realidad.

1. Evitar ver las preferencias individuales del resto de miembros del grupo
2. Utilizar cuando sea posible preferencias implícitas de los usuarios.
3. Eliminar las valoraciones de los miembros del grupo que no ofrezcan fiabilidad en sus valoraciones. De esta forma se puede aumentar la precisión de las recomendaciones.

Asegurar la comprensión de las recomendaciones

Es importante explicar al grupo porque se les recomienda determinado objeto. Para que de esta forma lleguen a comprender el porque de la recomendación.

Tratar de forma diferente grupos diferentes

En la agregación se deben tratar de forma diferente determinados subgrupos de personas para los que queramos que la solución no sea desagradable. Para ello las preferencias de los subgrupos de ponderan con unos pesos que reflejan la importancia del subgrupo. Y se realiza un promedio de las preferencias ponderadas con dichos pesos. De esta forma podemos dar preferencia a los intereses de determinados subgrupos de personas como pueden ser discapacitados, niños o alguna persona a la que se quiera agradar por encima de las demás.

2.2.1.4 Visualizar aceptabilidad individual de la recomendación

La recomendación debe satisfacer las restricciones de cada individuo del grupo para que pueda ser aceptada, por lo que se debe analizar la aceptabilidad de la recomendación grupal para cada individuo, subgrupos y grupo.

En el sistema *TRAVEL DECISION FORUM* los usuarios pueden ver como de aceptable es la recomendación para los demás miembros del grupo (lo que es útil en el caso que una persona este especialmente interesada en que otra quede contenta con la decisión final), en el sistema *CATS* cada usuario puede criticar la recomendación grupal para obtener una recomendación

similar que sea aceptable por él, por ejemplo si la recomendación es un hotel de tres estrellas, puede criticar que desea un hotel con categoría superior, como el usuario no descartó la recomendación, el sistema ofrece al grupo una recomendación de un hotel similar al anterior pero de mayor categoría.

2.2.1.5 Negociación grupal final

Se debe proporcionar un marco de trabajo donde los individuos se puedan comunicar, debatir y negociar que recomendación eligen.

En el sistema *TRAVEL DECISION FORUM* cada usuario dispone de un personaje animado que puede dialogar con los personajes animados de los demás miembros del grupo para llegar a la decisión final. En el sistema *CATS* el espacio de trabajo grupal es un tablero táctil, las personas se colocan alrededor del tablero y van valorando los productos (en este caso estaciones de esquí), cuando una estación es valorada positivamente por alguien el icono que la representa en un mapa en el tablero aumenta de tamaño, y disminuye de tamaño al ser valorada negativamente por alguien, de esta forma los usuarios valoran los productos y llegan a una recomendación final de forma colaborativa.

Sistema	Negociación grupal final
MusicFx	<ul style="list-style-type: none"> - No existe negociación grupal final. - El sistema va cambiando de canción conforme a las preferencias del grupo de personas presentes.
Let´s Browse	<ul style="list-style-type: none"> - No existe negociación grupal final. - La decisión final es tomada por un solo miembro del grupo.
POLYLENS	<ul style="list-style-type: none"> - No existe negociación grupal final explícita. - Se asume que los miembros del grupo llegaran a un acuerdo fuera de la aplicación (conversando entre ellos o por teléfono).
INTRIGUE	<ul style="list-style-type: none"> - No existe negociación grupal final. - La decisión final es tomada por el guía turístico que usa la aplicación.
Travel Decisión Forum	<ul style="list-style-type: none"> - Negociación grupal. - Diálogos entre los avatares que representan los miembros del grupo para llegar a un acuerdo final.
Recio-García et al.[8]	No existe negociación grupal final.

CATS	<ul style="list-style-type: none"> - Negociación grupal. - El tablero táctil proporciona un espacio de trabajo en grupo en el que los miembros del grupo pueden valorar y discutir las recomendaciones grupales
-------------	---

Tabla 11. Negociación grupal final.

Solo los sistemas *CATS* y *Travel Decision Forum* tienen en cuenta la interacción entre los usuarios para llegar a la solución grupal al ofrecer una negociación grupal final. Esto es importante porque generalmente las preferencias iniciales de cada individuo cambian conforme se usa la aplicación de recomendación y se interactúa con el resto de miembros del grupo.

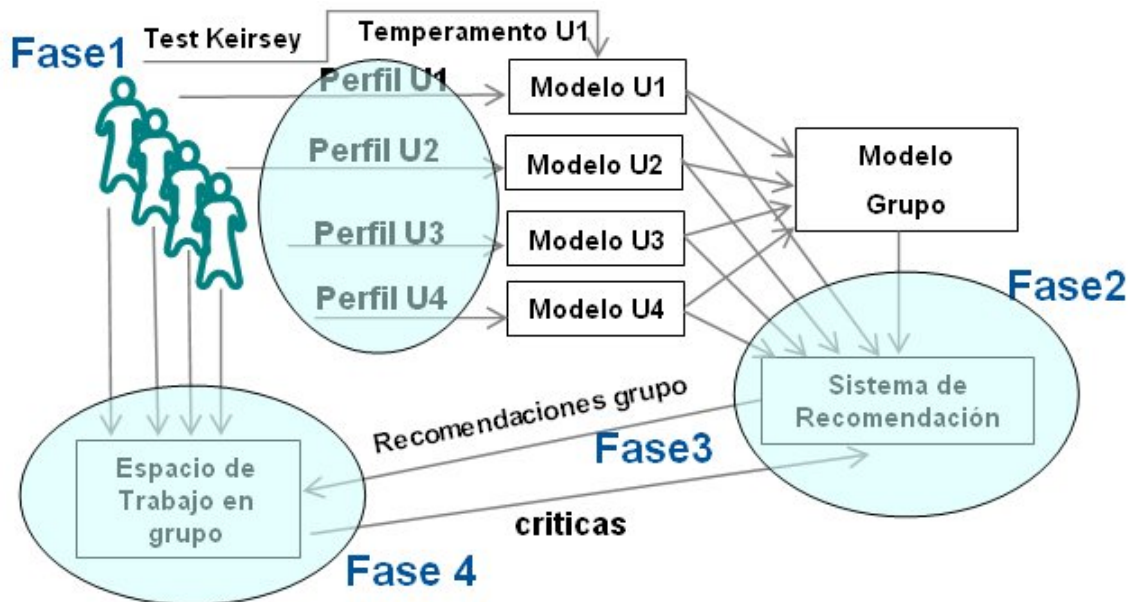


Fig. 6. Propuesta de Sistema de Recomendación Grupal Integrado.

Al integrar las propuestas de los estudios anteriores (Fig. 7) obtenemos un sistema en el que se realiza un modelado de los usuarios a nivel individual, se realiza un modelado del grupo, y se realiza una mejora de la interacción individual y grupal. A diferencia del estudio [8] en el que se realiza el test TKI para determinar la personalidad del usuario en casos de conflicto, realizamos del *Test de Keirsey* para determinar el temperamento del usuario. E incluimos dicho temperamento en el modelado del usuario para mejorar la interacción del sistemas con el usuario, adaptando la interfaz gráfica en función de su temperamento y incluyendo también el temperamento como una de las características a considerar por nuestro sistema de recomendación a lo hora de proporcionarle nuevas recomendaciones.

Sin embargo el sistema resultante presenta dos inconvenientes, en primer lugar es necesario que en la primera fase los miembros del grupo introduzcan en el sistema sus perfiles y preferencias, lo que requiere su

tiempo y disponibilidad, por otro en la cuarta fase hay que desarrollar un entorno de trabajo para que dichos usuarios se comuniquen entre ellos, lo que supone tiempo y coste de desarrollo y de la misma forma que en la fase 1 los usuarios deben de disponer de tiempo para entrar a dicho espacio para negociar con los demás.

De manera general es poco frecuente que todos los miembros de un grupo entren en un sistema para introducir sus preferencias, valorar productos y luego negociar, además esto habría que repetirlo para cada sistema de recomendación grupal de cada aplicación web comercial. Por ejemplo, todos los miembros del grupo tendrían que invertir tiempo indicando sus preferencias y perfiles en un sistema de recomendación de viajes grupales, y luego tendría que volver a realizar lo mismo desde el principio para un sistema que recomiende restaurantes o películas para grupos. Es decir, para cada actividad grupal se requiere la introducción de los perfiles y preferencias de cada uno de sus miembros.

Las redes sociales nos pueden ayudar a solventar estos dos problemas identificados en las fases primera y cuarta del proceso para recomendaciones grupales. Abordar dichos problemas desde las redes sociales supondría tanto un ahorro de tiempo para los usuarios, como un ahorro de tiempo en desarrollo el desarrollo de las fases 1 y 4 en los sistemas de recomendación.

2.3 Recomendación en Redes Sociales

Redes Sociales

En los últimos años han surgido un gran número de aplicaciones web que ofrecen sus redes sociales para que las personas se comuniquen con sus amigos o compañeros de trabajo. Los sitios más populares son *Facebook* y *MySpace*, en el caso de España los más populares son *Facebook* y *Tuenti*. En el caso de Rusia la red social más popular es *Vkontakte*. En dichos sitios los usuarios pueden añadir amigos, y compartir con ellos mensajes, emails, fotografías y realizar actividades con sus amigos como hablar en un chat o mandarse mensajes en un foro o jugar con ellos usando determinadas aplicaciones [Morris et al - 13].

El aumento del número de personas que utilizan las redes sociales ha sido espectacular, en 1997 solo en Estados Unidos existían 57 millones de usuarios. Un año después en 2008 existían ya 690 millones de usuarios en todo el mundo. Actualmente la red social Facebook cuenta con 500 millones de usuarios.

Caso de estudio: Facebook

Facebook surgió en Estados Unidos como una red social dirigida al ámbito universitario, actualmente es la principal red social (a nivel de aplicación web, a nivel general la mayor red social es la formada por los millones de

mensajes SMS que envían las personas con sus teléfonos móviles) y ha alcanzado un ámbito mundial.

Su crecimiento ha sido increíble, en 2007 tenía 9 millones de usuarios, en 2008 200 millones, y actualmente cuenta con 500 millones de usuarios registrados.

Además lo que resulta realmente importante es que sus usuarios se han acostumbrado a utilizar la red social de forma habitual y casi diaria. El 50% de sus usuarios entran en *Facebook* de forma diaria. Además según *Facebook*, un usuario medio tiene 130 amigos, es miembro de 12 grupos, pasa 55 minutos al día en Facebook, y al mes: recibe 8 solicitudes de amistad, declara que le gusta un contenido 9 veces, escribe unos 25 comentarios sobre contenidos de *Facebook*, se declara fan de 2 páginas de productos al mes, y recibe 3 invitaciones de eventos. *Facebook* cuenta actualmente con 700.000 páginas de negocio.

El número, frecuencia de visitas y el tiempo de permanencia de los usuarios en las redes sociales es muchísimo mayor que el de las aplicaciones web comerciales.

La motivación social hace que los usuarios de *Facebook* muestren un gran número de información en sus perfiles, edad, estudios, trabajo, situación sentimental, preferencias sexuales y políticas, a cerca de sus gustos: libros, películas, canciones, países preferidos. Forman parte de grupos con intereses y gustos comunes en determinadas actividades (estudios, trabajo, viajes, política,...). Con la información que un usuario de *Facebook* suele tener en su perfil, se puede realizar una inicialización del modelado del usuario, evitando el problema del arranque en frío del usuario (como expusimos en la sección sobre los sistemas de recomendación individuales), además las declaraciones de las personas que se han hecho fans de un producto o se han unido a un grupo con interés en el producto o la actividad, pueden solventar el problema del arranque en frío del producto.

Recomendaciones para grupos en Facebook

Actualmente son pocas las iniciativas que afrontan el problema de recomendar un producto o actividad a un grupo de personas en *Facebook*. En los grupos actuales se envían invitaciones para acudir a eventos de manera general a todos los miembros, donde solo se puede especificar si se acudirá o no al evento. Una persona recibe una invitación a un grupo de interés porque otra persona miembro de ese grupo le envía una invitación. No se valora el perfil del usuario, localización, gustos, pertenencia a otros grupos de interés, o la existencia en un grupo de personas con perfiles similares a los de un usuario para enviarle invitaciones o recomendaciones.

Existen dos estudios interesantes. *Collabio* [Bernstein-11] es una aplicación de *Facebook* donde un grupo de personas etiquetan a una persona con palabras que las definen, dichos etiquetados se agregan a una nube de palabras, al principio ocultando las letras de la palabra con imágenes (●●●●), el resto de personas pueden ganar puntos adivinando dichas etiquetas (cuando se adivina la etiqueta se muestra en la nube) o proporcionando nuevas etiquetas (que definen a la persona) que se añaden

a la nube de etiquetas (el tamaño de la palabra es más grande cuantas más personas piensan que la palabra etiqueta correctamente a la persona, y los puntos obtenidos por las personas que descubren las etiquetas es mayor cuanto más grande es la palabra). En el estudio las personas encuestadas afirman que las etiquetas de las nubes les definen correctamente, y que existen etiquetas que si bien ellos creen que no los definen, coinciden en que son apreciaciones que los demás suelen tener sobre ellos. La aplicación mantiene marcadores de puntuación a nivel local y global, y el estudio muestra que la motivación social de los participantes evita los problemas de las inconsistencias, falsedades o malos usos en las valoraciones de los usuarios.

Resulta atractiva la forma de agregación de etiquetas, ideas u opiniones de un grupo de personas en forma de nube, y se podría usar para mostrar una agregación de recomendaciones. Dar puntos por valorar recomendaciones, o por ser flexible para llegar a un acuerdo final también puede ser interesante.

En el estudio *Group Recommendation System for Facebook* [Baatarjav- 12] se realizan recomendaciones de grupos a personas en función del perfil de la persona. Es decir en función del perfil individual de una persona se le recomienda un grupo, teniendo en cuenta el perfil del grupo recomendado. Dicho perfil de grupo se obtiene a partir de los perfiles de las personas que son miembros de dicho grupo, para ello se realiza un clustering jerárquico de dichos perfiles donde se eliminan las características de los usuarios que no son relevantes para la mayoría del grupo. Posteriormente para recomendar un grupo a un individuo utilizan un árbol de decisión para a partir de las características del perfil del individuo recomendarle un grupo. Realizaron el estudio en la red de *Facebook* de la Universidad del Norte de Texas de Estados Unidos, a partir de los datos de 1580 usuarios y 17 grupos de intereses comunes existentes en dicha red. En sus conclusiones muestran que el sistema tiene una precisión en sus recomendaciones del 64%, y del 73% (aumento del 9%) después de eliminar el ruido de las características de usuarios no relevantes en un grupo.

Resulta de utilidad ver como es posible realizar un modelado de un grupo a partir de los perfiles de sus usuarios, sin embargo se puede mejorar la forma en que se realiza el modelado del grupo con taxonomías [Gómez-Gauchia-1] y/o con los métodos de agregación usados en el estudio [Recio-García et al-8] minimización de la penalización, satisfacción media, least misery, teniendo además en cuenta la personalidad de los miembros del grupo en situaciones de conflicto para poder determinar que recomendaciones grupales se adecuarían mejor a la interacción de un grupo de individuos, así como el temperamento de cada uno de los integrantes del grupo.



Fig. 7. Propuesta de Sistema de Recomendación Grupal Integrado en la red Social.

En el modelo de integración del sistema de Recomendación Grupal Integrado que expusimos en la sección 2.2, en la primera Fase 1: especificación de las preferencias de los usuarios, dichas preferencias se extraerían de las ya existentes en los perfiles de los usuarios en *Facebook*, teniendo en cuenta además el temperamento de los usuarios. La Fase 2: agregación de las preferencias y modelado del grupo, se harían dentro de la aplicación de recomendación. En la Fase 3: la aplicación presenta las recomendaciones al grupo y permite que los miembros del grupo interaccionen puntuando positivamente o negativamente las recomendaciones, interactuando con la aplicación. En la Fase 4: además de usar el espacio común de la aplicación para llegar a un acuerdo final, los usuarios pueden usar los mecanismos de comunicación sincronicos (chat) y asincronicos (posts) de *Facebook* para comunicarse entre ellos con el objeto de llegar a un acuerdo.

En resumen la interacción social necesaria en un sistema de recomendación grupal, se traslada de la aplicación web comercial, a un medio donde dicha interacción social entre los miembros de un grupo se realiza de forma habitual y mejor, la red social. Este enfoque aporta beneficios tanto a los usuarios (que se ahorran el tener que perder tiempo introduciendo información de sus preferencias en cada negocio web), para el sistema de recomendación (ahorro del tiempo y esfuerzo de desarrollo necesarios para que el sistema gestione las preferencias de los usuarios individuales, y proporcione formas de comunicación grupales para las decisiones de grupo), para los negocios (las recomendaciones pueden llegar a un número de personas muchísimo mayor), y para la red social al ofrecer un nuevo y mejor servicio social a sus usuarios.

2.4 Aplicaciones en Facebook

2.4.1 Juegos

FarmVille

En este juego el usuario es el dueño de una granja virtual. Su misión consiste plantar y recolectar vegetales y cuidar también los animales de la granja

Es uno de los juegos más extendidos y populares de *Facebook*. En Febrero del 2010 contaba con 400 millones de usuarios, 80 millones de usuarios mensuales, 54 millones de usuarios semanales y 27 millones de usuarios diarios.

Es un juego bastante sencillo, con unos gráficos 2D normales y un conjunto de actividades para realizar bastante limitado. Las claves de su éxito pueden ser su sencillez, la motivación social de los jugadores, y un sistema de recompensas que hace que el usuario se vaya sintiendo mejor a medida que avanza en el juego, puede cultivar más cosas, tener más dinero, más animales, más casas, ..etc.

PetSociety

Es un juego en el que tenemos que cuidar a nuestra mascota virtual. Hay que darla de comer, bañarla y jugar con ella de forma habitual. Podemos visitar las mascotas de nuestros amigos en el juego. Se puede también decorar la casa de nuestra mascota y recorrer la Villa del juego y por ejemplo entrar en sus tiendas para comprar comida para nuestra mascota u objetos para decorar su casa.



Fig. 8. Configuración del avatar en la aplicación PetSociety.

Su éxito se debe a la interacción que se puede realizar con las mascotas de otras personas (podemos ir a sus casas y por ejemplo jugar con ellas). Y también a un sistema de recompensa en forma de monedas que permite comprar más y mejores cosas para nuestra casa y para nuestra mascota.

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

En Julio del 2009 contaba con 30 millones de usuarios. 19 millones de usuarios mensuales, 10 millones de usuarios semanales y 4 millones de usuarios diarios.



Fig. 9. Imagen de un avatar mascota en su casa en la aplicación PetSociety.

Sacred Seasons

Es un juego de rol multijugador masivo en línea.

Esta ambientado en las mitologías de América del Norte. Un espíritu maligno se ha apoderado del reino, y ha llenado sus tierras de monstruos.

Es un juego basado en el desarrollo de la historia y en la realización de misiones.

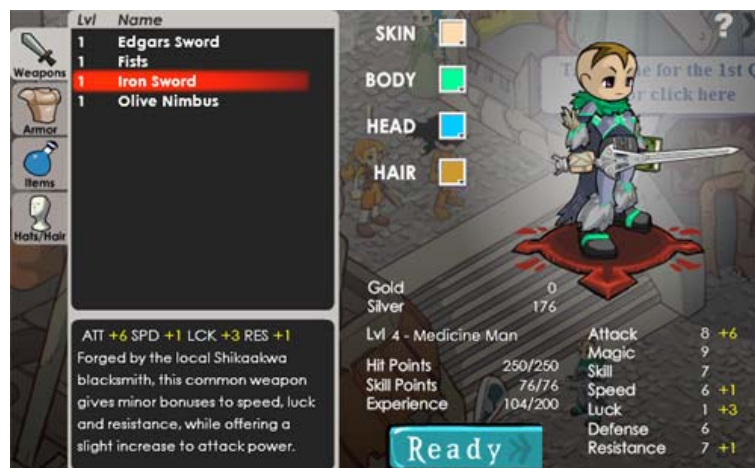


Fig. 10. Configuración del avatar en Sacred Seasons.

Según se avanza en el juego aparecen más misiones y desafíos. Cada región del juego tiene una apariencia y personalidad únicas. Las batallas son en vivo, y los jugadores que están online se pueden unir con sus amigos para pelear juntos contra los monstruos en tiempo real.

Ofrece la posibilidad de conocer gente nueva, y formar equipo con ella si se desea.

Vampiros

La aplicación Vampiros, es una aplicación bastante sencilla, pero que tuvo mucho éxito en los comienzos de *Facebook*.



Fig. 11. Imagen juego *Vampiros*.

El usuario es un vampiro, puede pelear con otros vampiros y también convertir a sus amigos en vampiros si estos aceptan su invitación (*Mordisco*) al juego.

Se puede pelear con otros vampiros, hombres lobo o zombies del juego. Para ello se selecciona el adversario, y se pulsa el botón 'pelear' (*fight*). A continuación aparece el resultado de la batalla, mostrando quién venció la pelea y cuantos puntos de juego gana. Los dos contendientes reciben en su muro de *Facebook* una notificación sobre el resultado de la batalla.

Para convertir amigos del *Facebook* en vampiros se les envía una invitación al juego. Si la aceptan son mordidos por nuestro vampiros, y se convierten en vampiros que forman parte de nuestro clan. Los vampiros que hemos convertido al juego son nuestros siervos y ganaremos puntos por cada batalla que ganen.

Se pueden ganar puntos en el juego ganando batallas o convirtiendo personas en vampiros. Según se van ganando puntos se van superando niveles en el juego. Con los puntos ganados en el juego se pueden adquirir armas y objetos que podemos utilizar en las batallas. Cuanto mayor sea el nivel de nuestro adversario más difícil será que ganemos la batalla.

No existen gráficos en movimiento ni en 2D ni 3D. Los gráficos del juego se limitan a una imagen que representa el nivel de nuestro avatar en el juego. Además esta imagen es la misma para todos jugadores. La primera versión del juego tenía una apariencia HTML muy sencilla similar a la de la imagen superior. En la última versión se ha mejorado la apariencia web del juego añadiendo mejoras de estilo y colorido, pero la mecánica del juego sigue

siendo la misma. Las batallas se limitan a elegir un adversario, pulsar un botón y ver en la siguiente pantalla si hemos ganado o perdido.

El juego es muy sencillo y con bastantes limitaciones. Su éxito estuvo motivado por la interacción social de sus jugadores en la red social *Facebook*, y por ser un pionero en este tipo de juegos en la red social.

- **Interacción social:**
Nuestros amigos nos invitan a participar. Y jugamos porque nos resulta divertido pelear contra nuestros amigos. Cuando entramos en *Facebook* vemos las notificaciones de las personas que han atacado a nuestro vampiro, y esto sirve como aliciente para que les devolvamos el ataque.
- **Juego pionero en *Facebook*:**
Fue uno de los primeros juego que aparecieron en la red Social. Permitted descubrir que además de publicar mensajes y fotografías en *Facebook*, se podía participar en juegos con el resto de amigos de la Red Social. Al ser de los primeros juego gran parte de la gente jugo al juego y lo recomendó a sus amigos.

El sistema de puntuación y niveles mantiene el interés del usuario en el juego.

Sin embargo con el tiempo el juego se vuelve aburrido y la gente termina abandonándolo.

Recientemente se ha lanzado la versión 3D del juego (*Ciudad de Vampiros*). Es un mundo virtual en 3D, en el que nuestro avatar puede ser un vampiro, un hombre lobo o un zombie. Podemos recorrer este mundo virtual, sus casas y parques. Participar en batallas y comunicarnos con el resto de jugadores online usando el chat.

Hombres Lobo

La aplicación Hombres Lobo es completamente similar a la de Vampiros. Cambia que en vez de un vampiro, nuestro avatar es un hombre lobo, pero tanto las batallas con otros hombre lobo, como la conversión de amigos del *Facebook* en hombre lobo es idéntica.

2.4.2 Aplicaciones afectivas

Son aplicaciones que permiten que sus usuarios encuentren a otras personas con fines sentimentales.

Los usuarios pueden introducir unos filtros para realizar la búsqueda de las personas en las que pueden estar interesadas. Como pueden ser el sexo, el rango de edad ó la localización de las personas. Por ejemplo, se puede buscar mujeres de entre 24 y 30 años que residan en mi ciudad.

Algunas de estas aplicaciones, como por ejemplo *Flirtable* y *TrueMatch* para hacer más atractiva la búsqueda tienen un sistema de puntuación y niveles.

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

Ofrecen puntos por ver los perfiles del resto de los usuarios. Según las personas van ganando más puntos van aumentando de nivel.

De las aplicaciones analizadas solo *TrueMatch* realiza un breve cuestionario para determinar la personalidad de sus usuarios. Y ofrecer posteriormente como resultados de las búsquedas, personas que tengan una personalidad similar al usuario que realiza la búsqueda. *Flirtable* ofrece un enlace a otra aplicación *Personality* que realiza encuestas de personalidad, pero no esta integrado dentro de la aplicación ni en los resultados de sus búsquedas.

Aplicación	Tecnología	Importancia
FarmVille	Flash	Gran Número jugadores
PetSociety	Flash	Gran Número jugadores
Sacred Seasons	Flash	Historia del juego
Vampiros	HTML + PHP	Sencillez
Hombre Lobo	HTML + PHP	Sencillez
Flirtable	HTML + PHP	Búsqueda de personas
MeetNewPeople	HTML + PHP	Búsqueda de personas
TrueMatch	HTML + PHP	Personalidad

Tabla. 12. Tabla aplicaciones en *Facebook*.

Juego	Sexo	Edad	Localización	Personalidad	Temperamento
Flirtable	X	X	X	0	0
MeetNewPeople	X	X	X	0	0
TrueMatch	X	X	X	X	0

Tabla. 13. Filtros en las búsquedas que ofrecen las aplicaciones afectivas en *Facebook*.

Juego	Chat	Mensajes	Regalos (imágenes)	Ver personas conectadas Facebook	Ver perfil	Chat Facebook	Mensajes Facebook	Agregar amigos Facebook
Flirtable	0	X	X	0	X	0	0	0
Meet New People	0	X	X	0	0	0	0	0
TrueMatch	0	X	X	0	0	0	X	X

Tabla. 14. Formas de contacto que ofrecen las aplicaciones afectivas en *Facebook*.

Juego	Ventajas	Inconvenientes
Flirtable	Nuestros amigos nos pueden introducir a contactos suyos en los que podemos estar interesados.	No tiene en cuenta la personalidad de las personas
MeetNewPeople	<ul style="list-style-type: none"> - Lista de mensajes predefinidos para enviar a los contactos. - Puntuaciones y niveles 	<ul style="list-style-type: none"> - Filtrado básico: Solo edad y localización. - No tiene en cuenta la personalidad de las personas. - No ofrece puntos ni niveles.
TrueMatch	<ul style="list-style-type: none"> - Filtra por Personalidad. - Búsqueda de personas con personalidad similar. - Muestra similitud de personalidad con otras personas. 	<ul style="list-style-type: none"> - No tiene en cuenta el temperamento de los personas. - Una vez que has mostrado el interés por una persona pasas a ser un interesado misterioso, la otra persona tiene que confirmar que esta interesada en ti entre las fotografías de un grupo de personas para poder entrar con contacto contigo.

Tabla. 15. Tabla ventajas e inconvenientes aplicaciones afectivas en *Facebook*.

Flirtable

Permite realizar búsquedas de personas filtrando por sexo, rango de edad, y localización. Para determinar la personalidad de los usuarios ofrece un link a la aplicación *Personality*. Permite que un amigo nuestro en el juego, nos introduzca a amigos suyos en *Facebook* en los que podemos estar interesados.

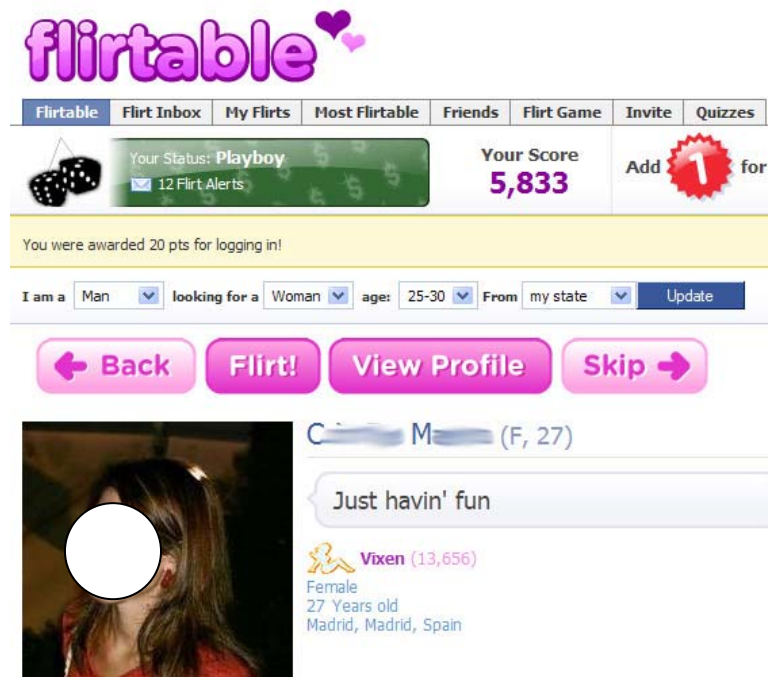


Fig 12. Perfil de usuario en Flirtable.

Meet New People

Permite realizar búsquedas de personas filtrando por sexo, rango de edad, localización. Ofrece una lista desplegable para elegir un mensaje de texto a enviar a otra persona. Por ejemplo: "Hi, I am interested".



Fig. 13. Pantalla principal de Meet New People.



Fig. 14. Perfil de usuario en *Meet New People*.

True Match

Como en los dos juegos anteriores permite buscar personas por sexo, edad y localización. Pero además permite encontrar personas con una personalidad compatible con la nuestra.

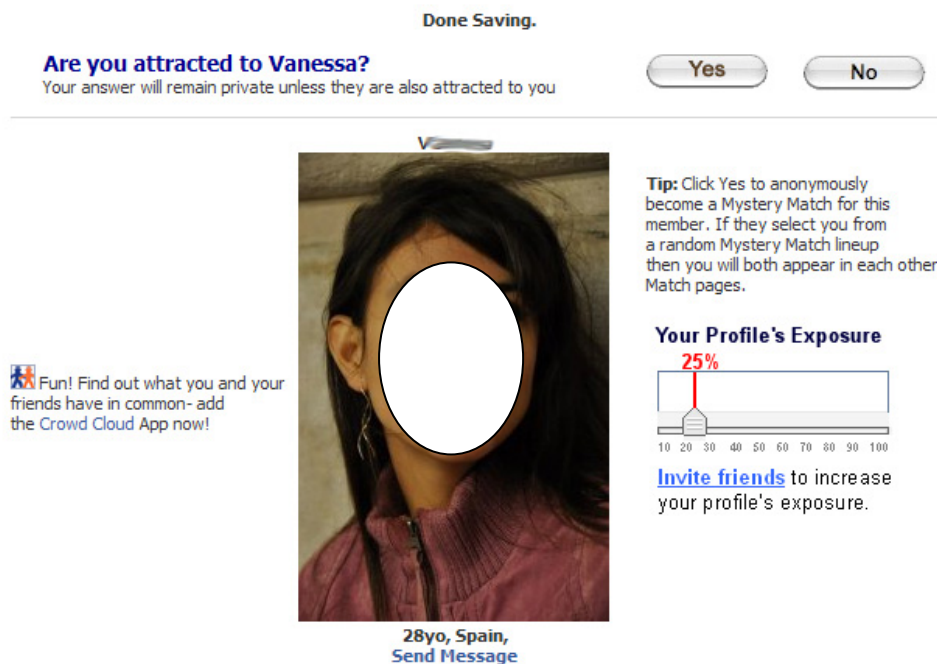


Fig. 15. Perfil de usuario en *True Match*.

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

En el juego se muestran las imágenes de usuarios. Si seleccionamos que estamos interesados en ella pasamos a ser un "Match Misterioso" para ella. Si la otra persona nos selecciona entre una serie de *matches* de relleno, pasaríamos a ser un match para ella y podríamos entrar en contacto a través del *Facebook*.

En el apartado de configuración podemos establecer nuestro perfil y nuestras preferencias de búsqueda.

The image shows two columns of configuration options for a user profile. The left column is titled 'Basic Information' and includes fields for birthday (with three dropdown menus), sex (Male), race (Hispanic), and country (Spain). Below these are checkboxes for 'Allow members to message me' (checked) and 'Looking For' (with checkboxes for Friendship, Dating, A Relationship, Networking, and Whatever I can get, all checked). The right column is titled 'Saved Search Preference' and includes 'Interested In' (Male and Female, with Female checked), 'Age Range' (31-35), 'Race Preference' (Any), 'Country' (radio buttons for US, CA, and OTHER, with OTHER selected), 'City' (Madrid), and another 'Country' dropdown (Spain).

Fig. 16. Configuración de la información del usuario en *True Match*.

También dispone de un pequeño test para establecer nuestra personalidad.

The image shows a 'Personality Quiz' with the instruction: 'Select a button below that is closest to how you would best describe your personality.' There are five rows of radio buttons, each with two options:

- Aggressive (radio 3 selected) vs. Avoid Conflict
- Expressive/Outgoing (radio 2 selected) vs. Quiet/Homebody
- Factual/Realist (radio 3 selected) vs. Idealist/Creative
- Logical (radio 2 selected) vs. Emotional
- Planner/Disciplined (radio 2 selected) vs. Spontaneous/Impulsive

Fig. 17. Encuesta de personalidad aplicación *True Match*.



Fig. 18. Enlaces a usuarios de la aplicación con personalidades similares a las del usuario (*Personality Matches*) en la aplicación *TrueMatch*.

Podemos ver una listado con los resultados de las personas que tienen una personalidad similar a la nuestra. Y podemos mandar un mensaje de texto a cada una de ellas a través de *Facebook* para entrar en contacto.

2.4.3 Otras aplicaciones

Emote

Esta sencilla aplicación permite seleccionar el emoticono (smiley) que representa nuestro estado de ánimo. El emoticono elegido se muestra en la página de *Facebook* del usuario. Permitiendo que nuestros amigos puedan tener una idea de nuestro estado de ánimo actual, si estamos alegres, tristes, llorando, enfadados,...etc.

La originalidad reside en que en vez de que el usuario tenga que introducir en un test o en una casilla su estado de ánimo, selecciona el icono que lo representa. El icono también sirve para romper las reticencias de ciertas personas, a las que no les gusta compartir este tipo de informaciones por considerarlas demasiado personales. La apariencia inocente y divertida de los emoticonos ayuda a romper estas barreras.

Sin embargo esta información sobre el estado de ánimo del usuario no se aprovecha para realizar ningún tipo de modelado sobre el estado de ánimo del mismo en esta aplicación ni en *Facebook*.

Pensamos que puede resultar interesante en el desarrollo de nuestra aplicación de recomendaciones para grupos, permitir que el usuario pueda reflejar su estado de ánimo actual mediante un emoticono. Para realizar un

modelado del estado de ánimo del usuario que nos permita mejorar la interacción de nuestra aplicación con los usuarios. En el estudio [Gómez-Gauchia-1] se refleja que es importante capturar el estado de ánimo del usuario para ayudar a mejorar la interacción hombre-máquina y hacerla más intuitiva, suave y eficiente. Además detectar los cambios en el estado de ánimo del usuario puede servir para prevenir posibles bloqueos del usuario (*breakdowns*).

También se pueden usar los emoticonos para que los usuarios que están participando en una decisión de grupos reflejen de forma amigable lo contentos o descontentos que están con las recomendaciones propuestas.

Los emoticonos pueden servir también para mejorar la interacción no solo del usuario con la aplicación, sino también la interacción entre los usuarios. El usuario puede ver el estado de ánimo del resto de compañeros de su grupo y adecuar sus acciones teniendo en cuenta el ánimo de los demás, evitando conflictos y favoreciendo alcanzar acuerdos finales.

TripAdvisor

Esta aplicación permite señalar en un pequeño mapa del mundo las ciudades en las que hemos estado. El mapa se muestra en nuestra página de *Facebook*, y sirve para que nuestros amigos puedan ver rápidamente los viajes que hemos hecho. Para marcar los viajes en el mapa solamente tenemos que seleccionar primero el país y luego la ciudad en unas listas desplegables. A continuación la ciudad que hemos seleccionado aparecerá marcada con un pequeño círculo azul en el mapa. También podemos indicar entre todos los viajes realizados cuales son los tres viajes que más nos han gustado. Esta valoración sirve para orientar a nuestros amigos. Y la aplicación la utiliza además para mostrarnos cuales son los tres destinos de viaje más apreciados por nuestros amigos de *Facebook* que han usado esta aplicación y han reflejado sus preferencias de viaje.

¿Qué personaje de Perdidos eres?

Esta aplicación permite saber a que personaje de la serie de televisión *Perdidos (Losts)* nos parecemos.

En *Facebook* existen un gran número de aplicaciones similares.

Se realizar un breve cuestionario previo de 10-15 preguntas, tras las cuales se nos muestra que personaje de determinada serie de televisión, película, personaje histórico, animal o color somos. Y se nos explica las virtudes del personaje al que nos parecemos. Dicho resultado se muestra en nuestra página de *Facebook*, de forma que todos nuestros amigos pueden ver a quién nos parecemos. Se muestra una imagen del personaje y la descripción del mismo.

Estas aplicaciones son muy usadas porque resulta bastante divertido y curioso realizar el tests y descubrir a que personaje te pareces. Que todos tus amigos puedan ver a quién te pareces fomenta la interacción social.

El inconveniente es que al igual que el caso de los emoticones, no se utiliza esta información para realizar un modelado de la personalidad del usuario.

Para nuestra aplicación nos resulta de interés realizar un cuestionario previo que nos permita modelar el temperamento del usuario. Y parece original poder reflejar el resultado con una imagen que refleje dicha personalidad o temperamento. Por ejemplo una imagen de una persona con aspecto formal para representar que el usuario tiene un temperamento en el que predomina el componente racional [Gómez-Gauchia-1]. Para fomentar la diversión y la interacción entre los usuarios mostraremos la imagen de un famoso que tenga el mismo temperamento que el usuario.

Personalidad

Esta aplicación ofrece una gran variedad de tests de personalidad. Los test suelen ser bastante divertidos y amenos de realizar. Como por ejemplo: *¿Qué sabor de helado eres? ¿O a que personaje de una serie de dibujos animados te pareces?* El test suele constar de unas 10 a 15 preguntas bastante sencillas de cumplimentar. El orden de las preguntas del test es aleatorio para evitar la monotonía en el caso que se vuelva a realizar. Se ofrece una imagen y una breve y concisa descripción de la personalidad resultante.

La originalidad de esta aplicación es que permite buscar y contactar a personas que han realizado el mismo test y que han obtenido resultados similares a los nuestros. También podemos ver el grado de similitud de personalidad que tenemos con el resto de usuarios de la aplicación.

El inconveniente que presentan es que no hacen ninguna mención ni al temperamento ni al estado de ánimo de los usuarios.



Fig. 19 Imagen en la que se ve el resultado del test y los perfiles de usuarios que han realizado el tests con el mismo resultado en la aplicación *Personality*.

El sistema también muestra el tanto por ciento de similitud de personalidad que se tienen con otras personas.

2.4.4 Conclusiones aplicaciones Facebook

Aspectos Importantes:

- Sistema de puntuación y niveles de juego. Para mantener el interés del usuario.
- Notificar las acciones del juego en el muro de *Facebook* para fomentar la interacción social.
- Poder invitar a los amigos al juego a través de *Facebook*.
- Interacción de los usuarios a través del chat, posts, o mensajes privados.
- En su mayoría son aplicaciones sencillas. HTML + un lenguaje dinámico (PHP, jsp, asp) + MySQL (datos).
- Fáciles de usar por todo el mundo.
- En ninguna de las aplicaciones analizadas se considera el temperamento del usuario.

2.5 Rich Internet Applications (RIAs)

El término RIA fue introducido por Macromedia en Marzo del 2002. Desde entonces se han producido grandes avances en las herramientas y en los lenguajes usados para crear RIAs. El principal motor del aumento de la importancia de RIAs ha sido el crecimiento de tecnologías como Flash y Flex, junto con el uso que compañías como Google han hecho de AJAX [27]. Las Rich Internet Application son aplicaciones web que ofrecen unas interfaces de usuario más usables, ricas e interactivas, similares a las interfaces de usuarios de las aplicaciones de escritorio. Las RIAs mejoran la usabilidad de la aplicación y la experiencia del usuario, y proporcionan una nueva arquitectura cliente-servidor que reduce significativamente el tráfico de red usando de forma más inteligente unos request asíncronos que envían solo pequeños bloques de datos[45].

Sin embargo, el desarrollo de RIAs es complejo, en la mayoría de los casos se realizan implementaciones ad hoc que suelen ser costosas en tiempo y generación de errores.

Ventajas:

- Interfaz de usuario más rica, usable e interactiva que proporciona una experiencia de usuario más atrayente sin tener que recargar páginas enteras. Ejecución en múltiples dispositivos.
- Actualización de partes de la página frente a recarga de toda la página.
- Feedback en tiempo real y validación de datos del usuario, disparados por eventos del usuario [27].
- Look and feel de las aplicaciones de escritorio.
- Incorporar multimedia en las aplicaciones, incluyendo audio y video [27].
- Ofrece una serie de capacidades, como por ejemplo el chat, en tiempo real y colaboraciones que son muy difíciles o imposibles de realizar en aplicaciones web tradicionales [27].

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

- Mayor procesamiento en la parte del cliente, ahorrando tanto tiempo de espera al usuario, como recursos del Servidor.
- Guía a los usuarios a la hora de rellenar los formularios de la web, la validación en la parte del cliente de formularios se produce de forma inmediata.
- Reducen el tiempo de espera del usuario.
- Aumento de la motivación del usuario.
- Acceso a múltiples servicios web y repositorios de contenidos.
- Trabajo en grupo y mayor colaboración y comunicación con el resto de usuarios de una forma eficiente.

Inconvenientes:

- Complejidad de diseño e implementación [28], [29]
- Alto coste de mantenimiento [28]
- Es frecuente un desarrollo ad hoc de las RIAs [29]

Conclusiones RIA

- ✓ **Web 1.0 → RIA** : Modelo RUP + extensión RIA (Una iteración más del Modelo de Proceso).
- ✓ **Diseño: Patrones** y **Codificación: Librerías (Dojo)** → Ahorro Tiempo y Costes de desarrollo
- ✓ RIA → para aumentar la **usabilidad e interactividad** aplicaciones web
- ✓ RIA → **Mayor naturalidad de las acciones**: arrastrar vs. click botón derecho ratón.
- ✓ web+RIA → **impacto positivo motivación** usuario.
- ✓ RIA **no garantiza lograr el objetivo de la aplicación web**, pero **contribuye a lograrlo** al aumentar la motivación del usuario.

3 Análisis del Modelo

Al estudiar el estado del arte de los sistemas de recomendación hemos encontrado las siguientes limitaciones:

Limitaciones:

- **Arranque en frío del usuario:**
En el caso de los nuevos usuarios no se dispone de la suficiente información para poder determinar que es lo que le puede gustar.
- **Arranque en frío del producto:**
El producto no tiene las suficientes valoraciones para poder recomendarlo adecuadamente.
- **Valoraciones inconsistentes:**
Las valoraciones que los usuarios hacen sobre los productos pueden ser inconsistentes.
- **Cambios en las preferencias:**
Es frecuente que las preferencias de los usuarios cambien con el paso del tiempo.
- **No se considera el temperamento del usuario:**
- **No se toma en cuenta el estado de animo del usuario:**
- **Volver a registrar datos personales y preferencias:**
El usuario tiene que registrarse e introducir sus datos y preferencias sobre los productos que le gustan en cada uno de los sistemas de recomendación.
- **Mecanismos de comunicación y negociación:**
En los sistemas de recomendación para grupos se debe proporcionar a los usuarios mecanismos de comunicación y negociación adecuados para que puedan ponerse de acuerdo en la aceptación final de una recomendación.

Objetivos del modelo

Para resolver estas limitaciones nos planteamos que nuestro modelo debía satisfacer los siguientes objetivos:

1. Resolver las limitaciones detectadas en los sistemas de recomendación.
2. Tratamiento personalizado del usuario, adaptando la aplicación al temperamento y estado de ánimo del usuario.
3. Facilitar la interacción del usuario con el sistema.

Características del modelo

Para poder cumplir estos objetivos determinamos que nuestro modelo debía tener las siguientes características.

1. Modelado del usuario capturando automáticamente la información del usuario disponible en la Red Social (*Facebook*).
2. Aplicación de la teoría de los temperamentos al modelo:
 - Dentro del modelo de usuario: Perfil extendido del usuario.

- En la personalización de la interfaz gráfica.
 - En la búsqueda de usuarios (amigos) y en la recomendación de contenidos (amigos, destinos de viaje).
3. Aplicación de técnicas RIA para simplificar la interacción del usuario con el sistema y mejorar su experiencia usando la aplicación

Nos hemos centrado en aplicar estas características en los sistemas de recomendación y ello nos ha llevado a tener que integrar varias técnicas de diseño e implementación de sistemas web.

3.1 Arquitectura

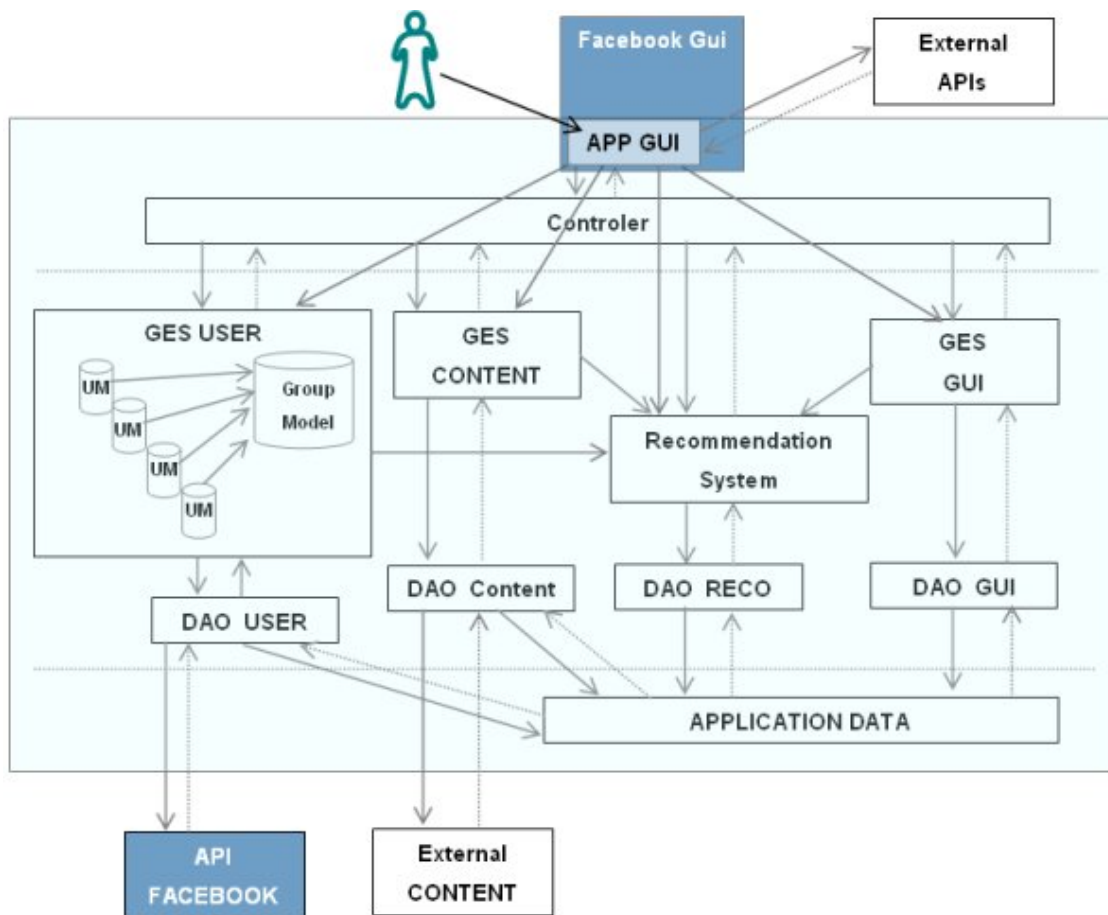


Fig. 20. Arquitectura de la aplicación.

Actores.

Existen tres tipos de actores en la aplicación en función de sus permisos de acceso.

- Administrador: Gestión de usuarios y grupos de usuarios (Modulo Ges User).
- Editor: Gestión de los Contenidos (destinos de viaje) (Modulo Ges Content). Gestión de la asignación de viajes en función del temperamento (Modulo GesContent). Gestión de las variaciones de la Gui en función del temperamento (Modulo GesGui).
- Usuario (Lector): Es el usuario normal de la aplicación, puede ver y modificar su perfil, buscar amigos, grupos y destinos de su interés.

CAPA DE PRESENTACION.

Gui.

La Capa de presentación del modelo esta formada por el Modulo GUI que contiene todas las interfaces gráficas.

El usuario de la aplicación interactúa solamente con las interfaces gráficas de la capa de presentación. Cuando un usuario accede a una interfaz gráfica y por ejemplo hace clic en "Mi Perfil", se lanza el evento "ver mi perfil" al controlador de la aplicación. La interfaz gráfica de nuestra aplicación esta contenida dentro de la interfaz gráfica de la Red Social. Esto permite que el usuario a la vez que utiliza nuestra aplicación tenga acceso a los métodos de comunicación sincronos y asincronos de la Red Social.

Controlador.

El controlador es el encargado de manejar los eventos lanzados desde la capa de presentación, actúa sobre el modelo y se encarga de lanzar los eventos de actualización a la capa de presentación.

CAPA DE LÓGICA.

La capa de lógica del modelo esta formada por los módulos GesUser, GesContent, GesGUI y el Sistema de Recomendación.

La capa de lógica contiene además la clase que contiene la información de configuración de la aplicación (en la que están definidos los parámetros necesarios para conectarse al Modelo de Datos y a la API de Facebook).

Modulo GesUser.

Es el Modulo de Gestión de los Usuarios y Grupos.

Contiene la lógica necesaria para crear, modificar, eliminar y obtener el modelo de los usuarios y crear, modificar, eliminar y obtener el modelo de los grupos.

Cuando un usuario accede a la aplicación se crea el modelo del usuario, accediendo a los datos de su perfil en la Red Social.

El modelo contiene las características del usuario, y proporciona métodos para acceder y modificar dichas características.

El modelo de un grupo se forma como un agregado de los modelos individuales de sus miembros.

Modulo GesContent.

Es el Modulo encargado de la gestión de los contenidos de la aplicación (destinos de viaje en el caso concreto de nuestra aplicación).

Contiene la lógica necesaria para dar de alta, modificar, eliminar, obtener contenido por Identificador, obtener contenido por nombre, mostrar contenido y devolver una lista de contenidos en función de unos parámetros de búsqueda.

Cuando un usuario solicita ver un contenido, se crea el modelo del contenido. El modelo solicita los datos del contenido al DAOContent

indicándole el identificador del contenido. El DAOContent devuelve la información encontrada en el Modelo de Datos con ese identificador o un código de error en caso contrario.

Cuando un usuario modifica un contenido, se modifica el modelo del contenido. Y el modelo solicita al DAOContent que actualice la información de dicho contenido en el Modelo de Datos.

A la hora de crear un contenido se pueden crear contenidos propios de la aplicación o acceder a un proveedor de contenidos externos. En el caso de contenidos externos, se guarda en el Modelo de Datos la información necesaria para que el sistema de recomendación pueda generar recomendaciones del contenido.

Modulo GesGui.

Es el Modulo encargado de la gestión de configuración de las interfaces gráficas.

Contiene la clase Gui que modela las características configurables de una interfaz gráfica.

Cuando un usuario accede a la aplicación se crea un modelo de la gui actual.

Si en el sistema de recomendación hay información (casos) suficientes para recomendar una gui al usuario, se realizan los cambios en el modelo de la gui que actualiza la interfaz gráfica.

Si el usuario realiza una configuración manual de la gui, se actualiza el modelo y se registra como un caso base en el sistema de recomendación.

El modelo de la Gui para ese usuario se almacena en el Modelo de Datos mediante el DAOGui. Y cuando un usuario vuelve a la aplicación se crea un modelo Gui que mediante un DAOGui obtiene la información de la configuración de la Gui para ese usuario en el Modelo de Datos.

Modulo Sistema Recomendación.

Este modulo contiene el sistema de recomendación de la aplicación.

Ofrece recomendaciones a los usuarios y grupos de usuarios.

Es un sistema de recomendación basado en casos. Se presenta a los usuarios y al grupo los casos más relevantes por similitud. En caso de que no exista ningún caso relevante, se ofrece una recomendación basada en la valoración de los contenidos.

Las recomendaciones para grupos se forman como agregados de las recomendaciones individuales para cada uno de los miembros del grupo.

En el caso particular de esta aplicación recomienda amigos y destinos de viajes a los usuarios y grupos de usuarios.

Cuando un usuario realiza una búsqueda de amigos se registra en el sistema de recomendación como un caso, en el que se guarda las características del usuario que realiza la búsqueda y los criterios de búsqueda. Cuando un usuario solicita ver el perfil de otro usuario también se registra como un caso en el sistema de recomendación.

De manera similar se registran como casos en el sistema de recomendación cuando un usuario realiza la búsqueda de un destino y cuando solicita ver el perfil de un destino.

Cuando un usuario realiza un cambio en la configuración de la interfaz gráfica, también se registra como un caso. En dicho caso se almacena las características del usuario y las características finales de la interfaz gráfica tras el cambio.

A la hora de generar una recomendación al usuario de amigos, destinos de viaje o variación de la interfaz gráfica, el sistema de recomendación busca los casos que tengan una mayor similitud con las características del usuario.

CAPA DE INTEGRACIÓN.

La capa de integración esta formada por los Modulos DAOUser, DAOContent, DAOReco, DAOGUI.

Contienen la lógica necesaria para registrar, modificar y eliminar la información contenida en el Modelo de Datos y devolver la información a la Capa de Lógica.

Es decir en vez de que por ejemplo el Modulo de Gestión de Usuarios acceda directamente a la base de datos ejecutando un sentencia SQL para recuperar los datos de un usuario con identificador = x, solicita el DAOUser que le de la información del usuario con identificador = x, y será el DAOUser el encargado de acceder a la base de datos, ejecutar la sentencia SQL, recuperar la información y devolver los datos a la clase del Modulo de Usuarios que se lo solicito.

Modulo DAOUser

Se encarga de realizar las altas, modificaciones, eliminaciones de usuarios, obtener usuario por identificador, obtener usuario por nombre y devolver una lista de usuarios en función de unos parámetros en el Modelo de Datos.

En el caso particular de las altas de usuarios accede a la API de Facebook para obtener los datos del usuario, mandárselos a la capa de lógica y guardarlos en el modelo de datos de la aplicación.

También se encarga de realizar las siguientes operaciones en el Modelo de Datos: altas, modificaciones, eliminaciones de grupos, obtener grupos por identificador, obtener grupos por nombre y devolver una lista de grupos que cumplan determinados parámetros de búsqueda.

Modulo DAOContent

Se encarga de realizar las altas, modificaciones, eliminaciones de contenidos, obtener destinos por identificador, obtener destino por nombre y devolver una lista de destinos en función de unos parámetros en el Modelo de Datos.

Modulo DAOGui

Se encarga de realizar las altas, modificaciones, eliminaciones de los registros de las configuraciones de la gui en el Modelo de Datos. Y de obtener la configuración de la gui de un usuario.

Casos de Uso

Casos de Uso de Usuarios.

	Caso de Uso	Objetivo
Aplicación	U1 – Registro	Que un usuario se registre en la aplicación
	U2 – Login	Que un usuario pueda entrar en la aplicación.
	U3 – Asignación de temperamento	Determinar el temperamento de un usuario de la aplicación.
	U4 – Configuración manual	Qué el usuario pueda configurar manualmente las opciones de personalización de la aplicación
	U5 – Logout	Qué un usuario pueda abandonar la aplicación
Usuarios	U6 – Ver mi Perfil	Qué un usuario pueda ver la información que se publica de su perfil en la aplicación
	U7 – Modificar mi Perfil	Qué un usuario pueda modificar los datos de su perfil
	U8 – Buscar amigos	Qué un usuario pueda buscar amigos en la aplicación
	U9 – Ver Perfil de otro usuarios	Qué un usuario pueda ver el perfil de otro usuario de la aplicación
Destinos de viaje	U10 – Buscar destinos de viaje	Qué un usuario pueda buscar destinos de viaje
	U11 – Ver destino de viaje en detalle	Que el usuario pueda ver los detalles de un destino de viaje
	U12 – Valorar destino de viaje	Que el usuario pueda valorar un destino de viaje
Grupos	U13 – Crear un grupo de amigos de viaje	Que un usuario pueda crear un grupo de amigos de viaje
	U14 – Buscar un grupo de amigos de viaje	Que un usuario pueda buscar grupos de amigos de viaje
	U15 – Unirse a un grupo de amigos de viaje	Que un usuario pueda unirse a un grupo de viaje ya existente
	U16 – Buscar destinos de viaje para mi grupo	Que un usuario pueda buscar destinos de viaje para el grupo
	U17 – Dejar un grupo de amigos de viaje	Que un usuario pueda abandonar el grupo de viaje

CASO DE USO Nº U1	
Nombre	Registro de usuario.
Objetivos:	Registrar un usuario en la aplicación.
Entrada:	Usuario nuevo en la aplicación. No esta registrado ni ha hecho login en la aplicación.
Salida:	Usuario registrado
Postcondiciones si éxito	Usuario registrado en el sistema.
Postcondiciones si fracaso:	Mensaje informativo sobre el problema acaecido.
Actor:	Usuario que se registra.

Secuencia de ejecución normal:

1. Solicitar los datos de registro al usuario en el formulario form_register (nombre de usuario, palabra de acceso y dirección de correo electrónico).
2. Validación en la parte cliente de los datos.
3. Recoger los datos del registro. Si el nombre de usuario ya existe en la Base de Datos volver al punto 1, indicando que el nombre de usuario ya existe en la aplicación.
4. Solicitar datos opcionales para realizar el registro (sexo, edad, localización,..etc.). Estos datos se recogen de forma implícita al conectarse a la aplicación a través de *Facebook*.
5. Crear el "modelo" del usuario (Acceso al Modelo de Datos).
6. Presentar la posibilidad de realizar el Test de Temperamento. Si realiza el test actualizar el Temperamento del usuario en el modelo.
7. Ejecutar ciclo CBR, para el usuario actual.
8. Preguntar si usuario satisfecho, si está satisfecho FIN. En otro caso Paso9
9. Automatización manual del usuario
10. Ejecutar ciclo CBR, para usuario actual y tipo personalización "Manual"

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son 7, 8, 9, 10.

CASO DE USO Nº U2	
Nombre	Entrar en la aplicación (Login de usuario).
Objetivos:	Usuario registrado pueda entrar en la aplicación.
Entrada:	Usuario registrado en la aplicación (login y password)
Precondiciones:	El usuario esta registrado en el sistema.
Salida:	Usuario logeado en la aplicación
Postcondiciones si éxito	Usuario logeado en el sistema.
Postcondiciones	Mensaje informativo sobre el problema acaecido.

si fracaso:	
Actor:	Usuario que hace login.

Secuencia de ejecución normal:

1. Solicitar los datos de login al usuario en el formulario form_login (login, palabra de acceso).
2. Validación en la parte cliente de los datos.
3. Recoger los datos del registro. Si el nombre de usuario no existe en la Base de Datos o no si la palabra de acceso no es la del nombre de usuario proporcionado, volver al punto 1. (Indicar que el nombre de usuario o la clave proporcionados son erróneos).
4. Crear el "modelo" del usuario (Acceso al Modelo de Datos).
5. Logear al usuario en la aplicación.
6. Actualizar el contador de visitas del usuario, incrementar los puntos del usuario por hacer login en la aplicación y actualizar el nivel del usuario en función de los puntos actuales.
7. Guardar en la sesión de la aplicación los datos del usuario.
8. Mostrar en la página de inicio de la aplicación un mensaje de bienvenida al usuario, mostrando su nombre y una descripción de su temperamento en caso de haber hecho el Test de temperamento.

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son 4,5,6.

CASO DE USO Nº U3	
Nombre	Establecer Temperamento Usuario.
Objetivos:	Registrar el temperamento del usuario.
Entrada:	Usuario que ha hecho login en la aplicación
Precondiciones:	El usuario esta registrado en el sistema y ha entrado en ella, haciendo login correctamente con su nombre de usuario y palabra clave.
Salida:	Temperamento del usuario.
Postcondiciones si éxito	Temperamento del usuario registrado en el sistema.
Postcondiciones si fracaso:	Mensaje informativo del problema que ha ocurrido.
Actor:	Usuario logeado.

Secuencia de ejecución normal:

1. Usuario pulsa el enlace "Test temperamento".
2. Mostrar al usuario el Test de Temperamento. Mostrar información explicativa sobre como rellenar el formulario.
3. Validar la correcta realización del test en la parte del cliente.
4. Procesar el formulario en la parte del cliente.
5. Guardar los datos sobre el temperamento del usuario en la Base de

datos.

6. Actualizar el temperamento del usuario en la Sesión.
7. Incrementar los puntos del usuario con la bonificación por realizar el test de temperamento.
8. Mostrar al usuario el resultado del test de temperamento. Incluyendo una breve descripción del temperamento resultante.
9. Actualizar las Gui en función de las variaciones establecidas para el tipo de temperamento del usuario.

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son 3,4,5.

CASO DE USO Nº U4	
Nombre	Configuración manual
Objetivos:	Configuración manual de la interfaz gráfica
Entrada:	Opción de configuración
Precondiciones:	La opción pertenece al conjunto de opciones de configuración
Salida:	Cambios aplicados en el sistema, registrado como un caso en el sistema CBR
Postcondiciones si éxito	Cambios aplicados en la interfaz gráfica
Postcondiciones si fracaso:	Mensaje informativo del problema que ha ocurrido.
Actor:	Usuario logeado.

Ejemplo: Configuración del menú de la aplicación.

	Opciones
Configuración menú de la aplicación	Horizontal Vertical Mapa

CASO DE USO Nº U5	
Nombre	Salir de la aplicación (Logout de usuario).
Objetivos:	Que un usuario registrado que ha hecho login en la aplicación pueda abandonarla.
Entrada:	Usuario que ha hecho login en la aplicación
Precondiciones:	El usuario esta registrado en el sistema y ha entrado en ella, haciendo login correctamente con su login y palabra clave.
Salida:	Usuario abandona la aplicación
Postcondiciones si éxito	Usuario abandona en el sistema
Postcondiciones si fracaso:	Mensaje informativo del problema que ha ocurrido.
Actor:	Usuario que hace logout.

Secuencia de ejecución normal:

1. Usuario pulsa botón de logout.
2. Presentar formulario opcional de satisfacción.
3. Actualizar la información del usuario contenida en variables de sesión en el Modelo de datos del sistema.
4. Redirigir al usuario a la pagina de su perfil en la Red Social.

CASO DE USO Nº U6	
Nombre	Ver mi Perfil
Objetivos:	Mostrar al usuario los datos de su perfil en la aplicación
Entrada:	Usuario que ha hecho login en la aplicación
Precondiciones:	El usuario esta registrado en el sistema y ha entrado en ella, haciendo login correctamente con su nombre de usuario y palabra clave.
Salida:	Datos del usuario en la aplicación
Postcondiciones si éxito	Ninguno.
Postcondiciones si fracaso:	Mensaje informativo del problema que ha ocurrido.
Actor:	Usuario que quiere ver su perfil.

Secuencia de ejecución normal:

1. Usuario pulsa el enlace "mi Perfil".
2. Recoger el identificador del usuario en la sesión.
3. Extraer del Modelo de Datos la información del usuario.
4. Modelar la información del usuario.
5. Guardar el modelo del usuario en la sesión.
6. Mostrar los datos al Usuario.

CASO DE USO Nº U7	
Nombre	Modificar mi Perfil
Objetivos:	Modificar los datos del perfil del usuario en la aplicación.
Entrada:	Cambios en los datos del usuario
Precondiciones:	El usuario esta registrado en el sistema y ha entrado en ella, haciendo login correctamente con su nombre de usuario y palabra clave.
Salida:	Mensaje informando al usuario sobre la correcta actualización de su perfil
Postcondiciones si éxito	Datos del usuario actualizados en el sistema.
Postcondiciones si fracaso:	Mensaje informativo del problema que ha ocurrido.
Actor:	Usuario que quiere ver su perfil.

Secuencia de ejecución normal:

1. Usuario pulsa el enlace "mi Perfil".
2. Recoger el identificador del usuario en la sesión.
3. Extraer del Modelo de Datos la información del usuario.
4. Modelar la información del usuario.
5. Guardar el modelo del usuario en la sesión.
6. Mostrar los datos al Usuario en un formulario de edición.
7. El usuario modifica sus datos.
8. Leer los datos del formulario y realizar la validación de los mismos en la parte del cliente.
9. Guardar los datos del usuario en el Modelo de Datos.
10. Actualizar los datos modificados del usuario en la sesión.
11. Mostrar mensaje al usuario indicando si sus datos se han actualizado correctamente.

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son 9,10.

CASO DE USO Nº U8	
Nombre	Buscar Amigos.
Objetivos:	Realizar una búsqueda entre los usuarios de la aplicación.
Entrada:	Criterios de búsqueda
Precondiciones:	El usuario esta registrado en el sistema y ha entrado en ella, haciendo login correctamente con su nombre de usuario y palabra clave.
Salida:	Datos de usuarios registrados en el sistema que se ajustan a los criterios de búsqueda introducidos por el usuario y mostrando antes los que tienen mayor valoración, y/o se ajustan mejor al tipo de usuario que realiza la búsqueda.
Postcondiciones si éxito	Se registra en el sistema de recomendación las características de los amigos que busca este tipo de usuario.
Postcondiciones si fracaso:	Mensaje informativo del problema que ha ocurrido.
Actor:	Usuario que ha hecho login en la aplicación

Secuencia de ejecución normal:

1. Mostrar al usuario un formulario en el que se pueda seleccionar los criterios de búsqueda de amigos: Sexo (Hombre, Mujer, Todos), Rango de edad (14-19, 20-29, 30-39, 40-49, 50-59, más de 59), localización (Todo el mundo, mi localización) y temperamento

(artesano, guardián, idealista, racional, todos). Dicho formulario se personalizara en función del modelado del usuario. En el formulario ya aparecerá seleccionado que se buscan personas del sexo opuesto al del usuario, de su mismo rango de edad, de su localización y de su mismo temperamento.

2. Leer los datos de la consulta.
3. Buscar los usuarios que se ajusten a los criterios de búsqueda en el Modelo de Datos.
4. Mostrar los usuarios que cumplen dichos criterios al usuario, dando preferencia a los que tienen mayor valoración, y/o se ajustan mejor al tipo del usuario que realiza la búsqueda.
5. El usuario podrá ver información más detallada de cada uno de ellos (Caso de Uso N°9) haciendo click sobre ellos. Si ningún usuario cumple los criterios de búsqueda mostrar un mensaje explicativo al usuario, indicándole que ningún usuario cumple los criterios de búsqueda. Indicar al usuario que intente realizar la búsqueda de nuevo con otros criterios de búsqueda.

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son 3.

CASO DE USO N° U9	
Nombre	Ver el Perfil de otro usuario.
Objetivos:	Ver los datos de otro usuario de la aplicación.
Entrada:	Identificador del usuario que se quiere consultar
Precondiciones:	El usuario esta registrado en el sistema y ha entrado en ella, haciendo login correctamente con su nombre de usuario y palabra clave. El usuario ha realizado previamente una búsqueda de usuarios, y pulsa el enlace de uno de ellos para ver su perfil.
Salida:	
Postcondiciones si éxito	Se registra como un caso en el sistema de recomendación que tipo de usuario se ha consultado en detalle. Incremento en los puntos del usuario que consulta. Incremento en la valoración de del usuario consultado.
Postcondiciones si fracaso:	Mensaje informativo del problema que ha ocurrido.
Actor:	Usuario logeado.

CASO DE USO N° U9

Secuencia de ejecución normal:

1. El usuario hace click en el enlace al perfil del usuario.
2. Leer el identificador del usuario a consultar.
3. Buscar los datos del usuario en el modelo de Datos. Si el identificador

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

del usuario no se encuentra en el Modelo de Datos, mostrar un mensaje explicativo al usuario. Si el usuario existe, incrementar la valoración del usuario en el Modelo de Datos.

4. Registrar como un caso en el sistema de recomendación el tipo de usuario consultado por este tipo de usuario.
5. Realizar el modelado del usuario.
6. Mostrar los datos del perfil del usuario consultado

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son 3.

CASO DE USO Nº U10	
Nombre	Buscar destinos de viaje.
Objetivos:	Realizar una búsqueda de los destinos de viaje que ofrece la aplicación.
Entrada:	
Precondiciones:	El usuario esta registrado en el sistema y ha entrado en ella, haciendo login correctamente con su nombre de usuario y palabra clave.
Salida:	Datos de los destinos de viaje registrados en el sistema que según a los criterios de búsqueda introducidos por el usuario. Se da prioridad a los destinos con mejor valoración y/o que mejor se ajustan al tipo de usuario
Postcondiciones si éxito	Se registra en el sistema que tipo de destino de viaje ha buscado este tipo de usuario.
Postcondiciones si fracaso:	Mensaje informativo del problema que ha ocurrido.
Actor:	Usuario logeado.

Secuencia de ejecución normal:

1. Mostrar al usuario un formulario en el que se pueda seleccionar los criterios de búsqueda de destinos de viaje (localización).
2. Leer los datos de la consulta.
3. Buscar los destinos que se ajusten a los criterios de búsqueda en el Modelo de Datos.
4. Mostrar los destinos que cumplen dichos criterios al usuario. El usuario podrá ver información más detallada de cada uno de ellos (Caso de Uso Nº11) haciendo click sobre ellos. Si ningún destino de viaje cumple los criterios de búsqueda mostrar un mensaje explicativo al usuario.

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son 3.

CASO DE USO Nº U11	
Nombre	Ver detalles de un destino de viaje.
Objetivos:	Ver los datos de un destino de viaje de la aplicación.
Entrada:	Identificador del destino
Precondiciones:	El usuario esta registrado en el sistema y ha entrado en ella, haciendo login correctamente con su nombre de usuario y palabra clave. El usuario ha realizado previamente una búsqueda de destinos de viaje, y pulsa el enlace de uno de ellos para obtener más información sobre el destino de viaje.
Salida:	Datos del destino de viaje seleccionado
Postcondiciones si éxito	Se registra en el sistema de recomendación que tipo de destino de viaje ha consultado en detalle este tipo de usuario. Y se aumenta la valoración del destino de viaje en el Modelo de Datos
Postcondiciones si fracaso:	Mensaje informativo del problema ha ocurrido.
Actor:	Usuario logeado.

Secuencia de ejecución normal:

1. El usuario hace click en el enlace del destino de viaje.
2. Leer el identificador del destino de viaje a consultar.
3. Buscar los datos del destino en el Modelo de Datos. Si el identificador del destino de viaje no se encuentra en el Modelo de Datos, mostrar un mensaje explicativo al usuario.
4. Leer los datos del destino de viaje.
5. Registrar como un caso en el sistema de recomendación el tipo de destino buscado por el tipo de usuario.
6. Incrementar la valoración del destino en el Modelo de Datos
5. Mostrar los datos del destino de viaje consultado.

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son 3.

CASO DE USO Nº U12	
Nombre	Valorar un destino de viaje.
Objetivos:	Realizar una valoración de un destino de viaje de la aplicación.
Entrada:	Identificador del destino de viaje
Precondiciones:	El usuario esta registrado en el sistema y ha entrado en ella, haciendo login correctamente con su nombre de usuario y palabra clave. El usuario ha realizado

	previamente una búsqueda de destinos de viaje. Dichos destinos aparecen en una lista. Cada uno tiene un campo de opciones para valorar (de 1 a 5) el destino y un botón " <i>Valorar</i> " para proceder a la valoración.
Salida:	Valoración del destino de viaje actualizada. Incremento en el nº de puntos del usuario.
Postcondiciones si éxito	Se registra en el sistema la valoración actualizada del destino de viaje, y la valoración que ha realizado el usuario. Incremento en los puntos del usuario.
Postcondiciones si fracaso:	Mensaje informativo del problema que ha ocurrido.
Actor:	Usuario logeado.

Secuencia de ejecución normal:

1. El usuario selecciona la valoración (de 1 a 5) que otorga al destino.
2. El usuario hace click en el botón "*Valorar*" de ese mismo destino.
3. Buscar el destino del viaje en el Modelo de Datos. Si el identificador del destino de viaje no se encuentra en el Modelo de Datos, mostrar un mensaje explicativo al usuario.
4. Actualizar la valoración del destino de viaje.
5. Incrementar los puntos del usuario, por haber realizado la valoración del destino.
6. Registrar el tipo de destino valorado por este tipo de usuario, y la valoración realizada.
5. Mostrar la valoración actualizada del destino de viaje.
6. Mostrar un mensaje indicando al usuario los puntos que ha ganado por valorar un destino. Mostrar también el número de puntos totales del usuario.

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es el 3.

Casos de Uso de Editores.

	Caso de Uso	Objetivo
Gestión de destinos de viaje	E1 – Añadir un destino	Registrar un destino en la aplicación
	E2 – Ver datos de los destinos	Ver los datos de los destinos registrados en la aplicación.
	E3 – Ver detalles de un destino	Ver todos los datos de un destino de viaje registrado en la aplicación.
	E4 – Modificar un destino	Modificar los datos de un destino registrado en la aplicación
	E5 – Eliminar un destino	Eliminar un destino de viaje de la aplicación
Gestión de personalización de la aplicación	E6 – Ver la asignación de los características de destinos de viaje en función del tipo de temperamento	Ver los datos de las asignaciones de características de viaje en función del temperamento
	E7 – Asignación de una característica de destino de viaje a un tipo de temperamento	Asignar características de destinos de viaje a un temperamento
	E8 – Ver la asignación de personalizaciones de la aplicación en función del temperamento	Ver los datos de las asignaciones de las características de la aplicación en función del temperamento
	E9 – Asignación de una personalización de la aplicación en función del temperamento	Asignar características de personalización de la aplicación a un temperamento

Gestión de destinos de viaje.

CASO DE USO Nº E1

Nombre: **Añadir un destino de viaje.**

Objetivos: Añadir un destino de viaje en la aplicación.

Entrada: Datos del nuevo destino de viaje en la aplicación.

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de editor.

Salida: Nuevo destino de viaje en la aplicación.

Postcondiciones si éxito: Nuevo destino de viaje registrado en el sistema.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de editor.

Secuencia de ejecución normal:

1. Solicitar los datos obligatorios del destino del viaje al editor en el formulario form_destino.
2. Validación en la parte cliente de los datos.
3. Recoger los datos del formulario. Si el nombre de destino ya existe en el Modelo de Datos volver al punto 1, indicando que el nombre de destino ya existe en la aplicación, para que proporcione uno diferente.
4. Solicitar datos opcionales de destinos.
5. Insertar los datos del destino en el Modelo de Datos de la aplicación.
6. Mostrar un mensaje indicativo del resultado de la operación.

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son 3, 5.

CASO DE USO Nº E2

Nombre: **Ver destinos de viaje.**

Objetivos: Ver los datos de los destinos de viaje de la aplicación.

Entrada: Ninguna.

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de editor.

Salida: Datos de los destinos de viaje registrados en la aplicación.

Postcondiciones si éxito: Ninguna.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de editor.

Secuencia de ejecución normal:

1. El editor pulsa el enlace para ver los datos de los destinos registrados en el sistema.
2. Buscar los datos de los destinos registrados en el Modelo de Datos.
3. Si en el sistema no hay destinos registrados se muestra un mensaje explicativo.
4. Mostrar los datos de los destinos en un listado. De cada destino se muestran de forma breve su identificador, nombre, localización, y valoración.
5. En la fila de cada destino aparecen tres enlaces: uno para ver información más detallada del destino (Caso de uso: E3), otro para modificar los datos de ese destino (Caso de uso: E4), y otro para eliminar el destino de la aplicación (Caso de uso: E5).

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es el 2.

CASO DE USO Nº E3

Nombre: **Ver detalles de un destino de viaje.**

Objetivos: Consultar todos los datos de un destino de viaje.

Entrada: Identificador del destino.

Precondiciones: El editor ha hecho login en la aplicación.

Salida: Datos del destino registrados en la aplicación.

Postcondiciones si éxito: Se muestran los datos del destino.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de editor.

Secuencia de ejecución normal:

1. Solicitar el identificador del destino a consultar.
2. Buscar el destino en la Base de Datos.
3. Si no existe ningún destino con ese identificador en el Modelo de Datos volver al punto 1, indicando que no existe ningún destino con el identificador proporcionado.
4. Leer los datos del destino de la Base de Datos.
5. Mostrar todos los datos del destino que existen en el Modelo de Datos. También se muestra un enlace para modificar los datos de ese destino (Caso de uso: E4), y para eliminarlo de la aplicación (Caso de uso: E5).

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores en el 2..

CASO DE USO Nº E3

Nombre: **Modificar un destino de viaje.**

Objetivos: Modificar los datos de un destino de viaje.

Entrada: Nuevos datos del destino.

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de editor. El destino de viaje ya existe en el sistema.

Salida: Actualización de los datos del destino de viaje.

Postcondiciones si éxito: Datos del destino de viaje actualizados.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de editor.

Secuencia de ejecución normal:

1. Solicitar el identificador del destino a modificar.
2. Buscar el destino en el Modelo de Datos.
3. Si no existe ningún destino con ese identificador en el Modelo de Datos volver al punto 1, indicando que no existe ningún destino con el identificador proporcionado.
4. Mostrar los datos del destino que existen en el Modelo de Datos.
5. El editor modifica los datos.
6. Leer los nuevos datos del destino.
7. Actualizar los datos del destino en la Base de Datos.
8. Mostrar un mensaje indicativo del resultado de la operación.

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. Los pasos más sensibles a la aparición de errores son el 2,7.

CASO DE USO Nº E4

Nombre: **Eliminar un destino de viaje.**

Objetivos: Eliminar un destino de viaje existente en la aplicación.

Entrada: Identificador del destino ha eliminar.

Precondiciones: El editor ha hecho login en la aplicación.

Salida: El destino eliminado ya no esta registrado en la aplicación.

Postcondiciones si éxito: Datos del destino eliminados en el sistema.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de editor.

Secuencia de ejecución normal:

1. Solicitar el identificador del destino a eliminar.
2. Buscar el destino en el Modelo de Datos.
3. Si no existe ningún destino con ese identificador en el Modelo de Datos volver al punto 1, indicando que no existe ningún destino con el identificador proporcionado.
4. Mostrar los datos del destino que existen en el Modelo de Datos.
5. Mostrar mensaje pidiendo la confirmación de la eliminación de dicho destino en la aplicación.
6. Eliminar los datos del destino en el Modelo de Datos.
7. Mostrar un mensaje indicativo del resultado de la operación.

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores en el 6.

Gestión de la personalización de la aplicación en función del temperamento.

CASO DE USO Nº E5

Nombre: **Ver la asignación de características de destinos de viaje en función del tipo de temperamento.**

Objetivos: Consultar las asignaciones destinos de viaje a temperamentos.

Entrada: Ninguna.

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de editor.

Salida: Datos de las asignaciones.

Postcondiciones si éxito: Ninguna.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de editor.

Secuencia de ejecución normal:

1. El editor pulsa el enlace para ver los datos de las asignaciones de destinos a temperamentos registrados en el sistema.
2. Buscar los datos de las asignaciones de destinos a temperamentos registrados en el Modelo de Datos.
3. Si en el sistema no hay asignaciones registradas se muestra un mensaje explicativo.
4. Mostrar los datos de las asignaciones en una ventana. Para cada característica de los destino de viaje se muestra su peso para cada uno de los temperamentos.
Ejemplo: Naturaleza: Artesanos: 0.8 Guardianes: 0.6 Idealistas: 0.7 Racionales: 0.5
5. En la fila de cada característica aparece un enlace para modificar la asignación (Caso de uso: E6).

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es el 2.

CASO DE USO Nº E6

Nombre: **Asignación de un características de destinos de viaje a un tipo de temperamento.**

Objetivos: Establecer el peso de una característica de los destinos, para cada uno de los temperamentos.

Entrada: Ninguna.

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de editor.

Salida: Datos de las asignación actualizados.

Postcondiciones si éxito: Ninguna.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de editor.

Secuencia de ejecución normal:

1. Solicitar el identificador de la característica.
2. Buscar los datos de la característica de destinos registrados en el Modelo de Datos.
3. Si en el sistema no existen características registradas se muestra un mensaje explicativo.
4. Mostrar el peso de la característica de los destino de viaje para cada uno de los temperamentos.
Ejemplo: Naturaleza: Artesanos: 0.8 Guardianes: 0.6 Idealistas: 0.7 Racionales: 0.5
5. El editor modifica los pesos y pulsa el botón "Modificar".
6. Los datos se actualizan en el Modelo de Datos.

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es el 2.

CASO DE USO Nº E7

Nombre: **Ver la asignación de personalizaciones de la aplicación en función del tipo de temperamento.**

Objetivos: Consultar las asignaciones opciones de personalización a temperamentos.

Entrada: Ninguna.

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de editor.

Salida: Datos de las asignaciones.

Postcondiciones si éxito: Ninguna.

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de editor.

Secuencia de ejecución normal:

1. El editor pulsa el enlace para ver los datos de las asignaciones de opciones de personalización a temperamentos registrados en el sistema.
2. Buscar los datos de las asignaciones de personalizaciones a temperamentos registrados en el Modelo de Datos.
3. Si en el sistema no hay asignaciones registradas se muestra un mensaje explicativo.
4. Mostrar los datos de las asignaciones en una ventana. Para cada tipo de personalización se muestra el peso de cada una de sus opciones para cada uno de los temperamentos.

Ejemplo:

Personalización del Menú				
Opción	Artesanos	Guardianes	Idealistas	Racionales
Horizontal desplegable	0.5	0.7	0.6	0.9
Horizontal simple	0.4	0.6	0.5	0.2
Vertical desplegable	0.5	0.5	0.4	0.9
Vertical simple	0.4	0.4	0.5	0.2
Mapa Texto	0.5	0.4	0.6	0.8
Mapa Imágenes	0.9	0.3	0.8	0.2

5. Para cada característica aparece un enlace para modificar los pesos de las asignaciones (Caso de uso: E8).

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es el 2.

CASO DE USO Nº E8

Nombre: **Asignación de una personalización de la aplicación en función del tipo de temperamento.**

Objetivos: Establecer el peso las opciones de una personalización de la aplicación, para cada uno de los temperamentos.

Entrada: Ninguna.

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de editor.

Salida: Datos de las asignación actualizados.

Postcondiciones si éxito: Ninguna.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de editor.

Secuencia de ejecución normal:

1. Solicitar el identificador de la característica.
2. Buscar los datos de la característica de destinos registrados en el Modelo de Datos.
3. Si en el sistema no existen características registradas se muestra un mensaje explicativo.
4. Mostrar el peso de las opciones de personalización para cada uno de los temperamentos.

Ejemplo:

Personalización del Menú				
Opción	Artesanos	Guardianes	Idealistas	Racionales
Horizontal desplegable	0.5	0.7	0.6	0.9
Horizontal simple	0.4	0.6	0.5	0.2
Vertical desplegable	0.5	0.5	0.4	0.9
Vertical simple	0.4	0.4	0.5	0.2
Mapa Texto	0.5	0.4	0.6	0.8
Mapa Imágenes	0.7	0.3	0.8	0.2

5. El editor modifica los pesos y pulsa el botón "Modificar".

6. Los datos se actualizan en la Base de Datos.

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es el 2,5.

Casos de Uso de Administradores.

	Caso de Uso	Objetivo
Gestión de usuarios	A1 – Añadir un usuario	Registrar un usuario en la aplicación
	A2 – Ver datos de los usuarios	Ver los datos de los usuarios registrados en la aplicación.
	A3 – Ver detalles de un usuario	Ver todos los datos de un usuario registrado en la aplicación.
	A4 – Modificar un usuario	Modificar los datos de un usuario registrado en la aplicación
	A5 – Eliminar un usuario	Eliminar un usuario registrado de la aplicación
Gestión de grupos de usuarios	A6 – Añadir un grupo	Registrar un grupo en la aplicación
	A7 – Añadir usuarios a un grupo	Añadir un usuario a un grupo de la aplicación
	A8 – Ver datos de los grupo	Ver los datos de los grupos registrados en la aplicación.
	A9 – Ver detalles de un grupo	Ver todos los datos de un grupo registrado en la aplicación.
	A10 – Modificar un grupo	Modificar los datos de un grupo registrado en la aplicación
	A11 – Eliminar un grupo	Eliminar un grupo registrado de la aplicación
Aplicación	A12 – Consulta estadísticas y gráficos de la aplicación	Ver las estadísticas y gráficos referentes al uso de la aplicación

Gestión de usuarios de la aplicación.

CASO DE USO Nº A1

Nombre: **Añadir un usuario.**

Objetivos: Registrar un usuario en la aplicación.

Entrada: Datos del nuevo usuario en la aplicación.

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de administrador.

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

Salida: Nuevo usuario registrado en la aplicación.

Postcondiciones si éxito: Nuevo usuario registrado en el sistema.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de administrador.

Secuencia de ejecución normal:

1. Solicitar los datos de registro del usuario en el formulario form_register (nombre de usuario, palabra de acceso y dirección de correo electrónico).
2. Validación en la parte cliente de los datos.
3. Recoger los datos del registro. Si el nombre de usuario ya existe en la Base de Datos volver al punto 1, indicando que el nombre de usuario ya existe en la aplicación.
4. Solicitar datos opcionales para realizar el registro (sexo, edad, localización,..etc.). Estos datos se recogen de forma implícita al conectarse a la aplicación a través de *Facebook*.
5. Insertar los datos del usuario en la Base de Datos de la aplicación.
6. Mostrar un mensaje indicativo del resultado de la operación.

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es el 5.

CASO DE USO Nº A2

Nombre: **Ver datos de los usuarios.**

Objetivos: Ver los datos de los usuarios registrados en la aplicación.

Entrada: .

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de administrador.

Salida: Datos de los usuarios registrados en la aplicación.

Postcondiciones si éxito: Ninguna.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de administrador.

Secuencia de ejecución normal:

1. El administrador pulsa el enlace para ver los datos de los usuarios registrados en el sistema.
2. Buscar los datos de los usuarios registrados en el Modelo de Datos.
3. Si en el sistema no hay usuarios registrados se muestra un mensaje explicativo.

4. Mostrar los datos de los usuarios en un listado. De cada usuario se muestran de forma breve su identificador, nombre, fecha del último acceso a la aplicación, y el número de visitas a la aplicación.

5. En la fila de cada usuario aparecen tres enlaces: uno para ver información más detallada del usuario (Caso de uso: A3), otro para modificar los datos de ese usuario (Caso de uso: A4), y otro para eliminar al usuario de la aplicación (Caso de uso: A5).

Secuencia alternativa:

- Si se produce algún error en el proceso se sacará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es el 2.

CASO DE USO Nº A3

Nombre: **Ver detalles de un usuario.**

Objetivos: Ver todos los datos de un usuario registrado en la aplicación.

Entrada: Identificador del usuario a consultar.

Precondiciones: El actor ha hecho login en la aplicación y tiene permisos de administrador.

Salida: Datos del usuario registrado en la aplicación.

Postcondiciones si éxito: Se muestran los datos del usuario registrado en el sistema.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de administrador.

Secuencia de ejecución normal:

1. Solicitar el identificador del usuario a consultar.
2. Buscar el usuario en el Modelo de Datos.
3. Si no existe ningún usuario con ese identificador en el Modelo de Datos volver al punto 1, indicando que no existe ningún usuario con el identificador proporcionado.
4. Leer los datos del usuario de el Modelo de Datos.
5. Mostrar todos los datos del usuario que existen en el Modelo de Datos. También se muestra un enlace para modificar los datos de ese usuario (Caso de uso: A4), y para eliminarlo de la aplicación (Caso de uso: A5).

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es el 5.

CASO DE USO Nº A4

Nombre: **Modificar un usuario.**

Objetivos: Modificar los datos de un usuario registrado en la aplicación.

Entrada: Nuevos datos del usuario ha modificar.

Precondiciones: El usuario ha hecho login en la aplicación y tiene permisos de administrador.

Salida: Actualización de los datos del usuario registrado en la aplicación.

Postcondiciones si éxito: Datos de usuario registrado en el sistema actualizados.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de administrador.

Secuencia de ejecución normal:

1. Solicitar el identificador del usuario a modificar.
2. Buscar el usuario en la Base de Datos.
3. Si no existe ningún usuario con ese identificador en la Base de Datos volver al punto 1, indicando que no existe ningún usuario con el identificador proporcionado.
4. Mostrar los datos del usuario que existen en la Base de Datos.
5. Leer los nuevos datos del usuario.
6. Actualizar los datos del usuario en la Base de Datos.
7. Mostrar un mensaje indicativo del resultado de la operación.

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores en el 6.

CASO DE USO Nº A5

Nombre: **Eliminar un usuario.**

Objetivos: Eliminar un usuario registrado de la aplicación.

Entrada: Identificador del usuario ha eliminar.

Precondiciones: El actor ha hecho login en la aplicación y tiene permisos de administrador.

Salida: El usuario eliminado ya no esta registrado en la aplicación.

Postcondiciones si éxito: Datos de usuario eliminados en el sistema.

Postcondiciones si fracaso: Mensaje informativo del problema que ha ocurrido.

Actor: Usuario con permisos de administrador.

Secuencia de ejecución normal:

1. Solicitar el identificador del usuario a eliminar.
2. Buscar el usuario en el Modelo de Datos.
3. Si no existe ningún usuario con ese identificador en el Modelo de Datos volver al punto 1, indicando que no existe ningún usuario con el identificador proporcionado.
4. Mostrar los datos del usuario que existen en el Modelo de Datos.
5. Mostrar mensaje pidiendo la confirmación de la eliminación de dicho usuario en la aplicación.
6. Eliminar los datos del usuario en el Modelo de Datos.
7. Mostrar un mensaje indicativo del resultado de la operación.

Secuencia alternativa:

- Si se produce algún error en el proceso se mostrará un mensaje de error explicativo de lo que ha ocurrido. El paso más sensible a la aparición de errores es en el 6.

4 Diseño e implementación

4.1 Patrones de diseño

Modelo-Vista-Controlador.

Para la implementación la aplicación hemos decidido utilizar un Modelo-Vista-Controlador pasivo de tres capas.

El patrón/arquitectura Modelo Vista Controlador MVC divide una aplicación interactiva en tres componentes.

- El **modelo** contiene la funcionalidad básica y los datos.
- Las **vistas** muestran información al usuario.
- Los **controladores** median entre vistas y controladores.

Desde la capa de vista se generan eventos y se puede acceder al modelo.

La vista genera un evento, el controlador captura el evento lo trata y actúa en consecuencia sobre el modelo, el controlador actualiza la vista, y la vista accede al modelo.

El controlador es el encargado de manejar los eventos, actúa sobre el modelo y se encarga de lanzar los eventos de actualización de la capa de Vista.

Interacción en MVC. Modelo Pasivo.

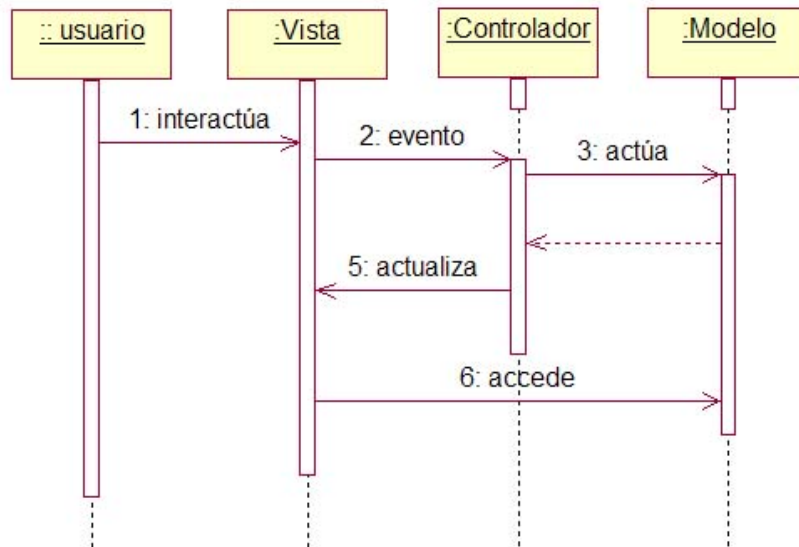


Fig. 22. Diagrama de interacción Modelo-Vista-Controlador Pasivo.

Modelo de tres capas

El modelo de tres capas considera una capa de presentación, otra de lógica, y otra de integración.

- La **capa de presentación** encapsula toda la lógica de presentación necesaria para dar servicio a los usuarios que acceden a la aplicación.
- La **capa de lógica** proporciona los servicios de la aplicación.
- La **capa de integración** es responsable de la comunicación con recursos y sistemas externos.

Clases del subsistema

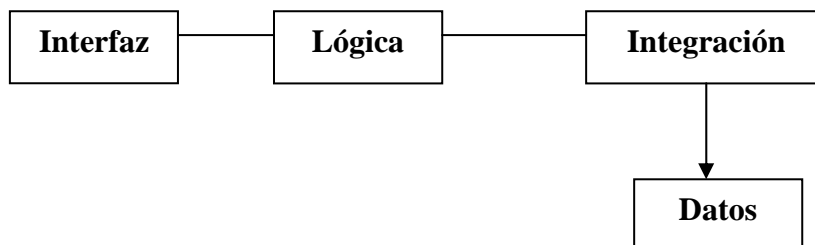


Fig. 23. Clases del subsistema.

Ventajas:

- Se puede modificar cualquier capa sin afectar a las demás.
- Facilita la división y organización o reparto del trabajo entre los miembros del equipo.
- Reduce la comunicación necesaria entre los integrantes del equipo.

Inconvenientes:

- Mayor complejidad arquitectónica.

Actualización asíncrona de partes

Con el surgimiento de la web 2.0 aparecen una serie de patrones [26], como la actualización asíncrona de partes de la página web, el patrón de experiencias de usuario mejorada, patrón de web sincronizada que suponen una gran mejora en la usabilidad y en la experiencia de usuario.

Este patrón se puede aplicar cuando para actualizar la información mostrada en una parte de una página web, es suficiente con recibir del servidor web la información actualizada de esa parte, en vez de volver a recibir del servidor la página completa [26].

Ahorrándose de esta forma recursos tanto en la parte del cliente como en la del servidor.

La aplicación Web envía una solicitud de carga (a petición del usuario, por un evento de actualización periódico, o por un cambio en el estado de algún elemento) de pequeñas porciones de la página, cuando llega el request (asíncrono) del Servidor con los nuevos datos, se actualiza solo esa porción de la página [45].

En la parte del cliente se requiere un mecanismo que envíe una petición al servidor, en el servidor un componente que escuche las peticiones y envíe los request. El navegador debe tener mecanismos que le permiten recibir el request asíncrono del servidor y actualizar la parte correspondiente de la página.

Ventajas:

- Hace que las páginas web sean mucho más interactivas y rápidas de descargar en la parte del cliente.
- Ahorro de recursos en la parte del servidor.

Inconvenientes:

- Demasiados componentes actualizables de forma asíncrona en una misma página pueden reducir su rendimiento.
- Personas con discapacidades físicas, que acceden a los contenidos a través de lectores de pantalla o táctiles, tienen dificultades para percibir la actualización de un contenido que ya se ha procesado por el lector. (En estos casos se debe permitir deshabilitar la actualización asíncrona).

Experiencia de usuario mejorada

Se utiliza para mejorar la experiencia del usuario (la interacción hombre-computadora) que utiliza una aplicación web, y la usabilidad de la aplicación en sí [26]. La capa de presentación de una aplicación web se puede adaptar a las características del usuario o al contexto en el que el usuario utiliza la aplicación. Para que dichas adaptaciones y mejoras de la experiencia de usuario sean más sencillas de implementar se recomienda usar el patrón Modelo-Vista-Controlador y el modelo de proceso RUP en el desarrollo de aplicaciones web. Los contenidos se deben ofrecer en el formato más adecuado al usuario. La aplicación web debe ser eficiente. Se debe optimizar la usabilidad de los procesos de usuario, para lo que se puede tener en cuenta la localización del usuario, si previamente se dispone de información suya, si ya es cliente de la aplicación, las capacidades técnicas (navegador, ancho de banda).

Se deben evitar ciertos errores en el diseño de las interfaces de usuario: formularios demasiado extensos, borrar todos los datos del formulario si el usuario se ha equivocado en algún dato, localizar la validación de los datos (formatos telefónicos o postales) del formulario estrictamente al país de origen del sitio web, campos de edición sin formato (es mejor mostrar un calendario para seleccionar la fecha que un campo de edición en blanco), solicitar información innecesaria, no mostrar la información necesaria para realizar un proceso, una pobre presentación gráfica o agrupación poco lógica de los elementos de la interfaz gráfica.

Según un estudio realizado por *Allurent*, al aplicar el patrón RUE en la aplicación web de la cadena de venta de libros Borders, la mejora de la experiencia de usuario produjo un aumento del 62% de la conversión (visitantes que acaban comprando), del 41% en el número de productos vistos, y un 11% en las recomendaciones. Por lo que se recomienda utilizar el patrón RUE para disminuir la alta tasa de abandonos que suelen tener los negocios web.

Con el surgimiento de la web 2.0 aparecen una serie de patrones [26], como la actualización asíncrona de partes de la página web, el patrón de experiencias de usuario mejorada, patrón de web sincronizada que suponen una gran mejora en la usabilidad y en la experiencia de usuario.

Las web tradicionales están siendo reemplazadas por las Rich Internet Applications (RIA). Que son aplicaciones que proporcionan al usuario unas interfaces más ricas e interactivas similares a las aplicaciones de escritorio. Las RIAs ofrecen un gran número de ventajas a las aplicaciones web.

4.2 Diseño de las Interfaces Gráficas

De manera general para el desarrollo de cada interfaz he partido de un boceto, un simple borrador dibujado en papel de cómo debería ser dicha interfaz.

En la interfaz gráfica principal de la aplicación distinguimos los siguientes apartados:

Esquema inicial:

A) CABECERA
B) MENU: Enlaces a los distintos servicios de la aplicación
C) Sección Central: Contenidos
D) Pie : Información de contacto

El estilo de la aplicación está definido con hojas de estilo CSS. La validación y envío de los formularios de la aplicación se realiza utilizando JavaScript [46].

La página inicial es la única página que se carga completamente en la aplicación web.

Cuando el usuario hace click en un enlace para ir a otra sección por ejemplo a "Mi perfil", utilizando AJAX [45] se realiza una petición asíncrona de un fragmento de página que contiene la información del perfil del usuario. Y dicha información aparece en la Sección Central de la página principal

La sección central de la aplicación es en la se muestran los fragmentos de HTML solicitados utilizando AJAX para todas las opciones de la aplicación.

Debajo del "pie" de la aplicación se han situado dos componentes *Facebook*. Uno que permite ver que amigos nuestros están usando la aplicación. Y otro que permite que los usuarios compartan en la Red Social que el sitio les gusta, pulsando el botón "I like it", y que escriban un *post* que se muestra también en su perfil de *Facebook*.

Como la página principal solo se carga una vez, y el resto de opciones solo se refrescan en la sección central de la página principal, estos componentes *Facebook* solo se cargan una vez, evitando de esta forma retrasar la carga del resto de funciones de la aplicación.

4.3 Personalización Inteligente

Las **variaciones software** son todas las acciones posibles que podemos realizar para personalizar cierto dominio [Gómez-Gauchia-1]. Por ejemplo realizar determinadas acciones para cambiar el color de fondo de la aplicación en función del estado de ánimo del usuario. Podríamos establecer un fondo de color verde para intentar calmar al usuario en función de su estado de ánimo, o porque puede ser un color de preferencia para personas con un temperamento racional. Y establecer un color más vivo e intenso si queremos mantener la atención y el interés del usuario, por ejemplo en el

caso que detectemos que su estado de animo refleje que se esta aburriendo.

Las opciones que tenemos para personalizar una aplicación en *Facebook* depende de la tecnología que utilicemos para implementarla.

Vamos a desarrollar una aplicación web utilizando tecnologías HTML + CSS + PHP + AJAX. Esta combinación de tecnologías nos ofrece grandes posibilidades de personalización.

A grandes rasgos:

- **Apariencia:** Usando hojas de estilos CSS podemos personalizar la apariencia de todos los elementos HTML (párrafos, enlaces, listas, textos, tablas, secciones) usados en la aplicación web. Su tamaño, color, grosos, tipos de letra, tamaño de las letras, ..etc. También se puede cambiar el posicionamiento de las distintas partes o secciones de la web (maquetación CSS con div). Por ejemplo si tenemos 3 secciones pueden estar las 3 en una fila, o las 3 en una misma columna.
- **Contenido:** PHP y Ajax nos permiten personalizar los elementos de la GUI que queremos mostrar y su disposición. También se puede cambiar dinámicamente el contenido HTML que tiene cada sección. Lo que permite adaptar la Gui y sus contenidos a las características del usuario. Por ejemplo en la página de inicio podemos dar más prioridad a recomendar hoteles o Bed&Breakfast en función del modelo del usuario. O mostrarle información especifica de interés solo para el usuario.
- **Idioma:** La aplicación puede estar disponible para distintos idiomas. Se puede programar de forma que para adaptarla a un nuevo idioma baste simplemente con añadir un solo archivo de texto que contenga todas las palabras, frases y párrafos que se utilizan en la aplicación. De esta forma además se puede cambiar de forma rápida y automática el lenguaje que utiliza la aplicación.
- **Formalidad:** Dentro de un mismo idioma podemos graduar la formalidad o el tratamiento con el que nos dirigimos al usuario. Por ejemplo en italiano podemos saludar a las personas diciéndolas "*Ciao*", sin embargo en la cultura italiana saludar a las personas mayores de esta forma sería considerado una falta de educación. Deberíamos dirigirnos a ellos con un "*Salve*" como muestra de respeto.
- **Dificultad conceptual:** Además del idioma y la formalidad podemos graduar y adaptar la densidad o dificultad conceptual de los textos para adaptarlos a las características formativas, intelectuales, y cognitivas del usuario.

Variaciones GUI		
Hoja de estilo		
Opciones	Valores	
Rojo	estilo_rojo.css	
Amarillo	estilo_amarillo.css	
Verde	estilo_verde.css	
Azul	estilo_azul.css	
Cabecera de la página web		
Opciones	Valores	
Imagen sobria	-	
Imagen artística	-	
Imagen impactante	-	
Texto	Travel Cobber	
Menú de la aplicación web		
Opciones	Valores	
Barra horizontal	-	
Barra vertical	-	
Mapa de texto	-	
Mapa de imágenes	-	
Mostrar Valoraciones de elementos:		
Opciones	Valores	
Textual	Malo Regular Aceptable Bueno Excelente	
Numérico	1-5	
Barra horizontal		
Imágenes de estrellas	***	
Imágenes	Pulgar hacia arriba o hacia abajo	
Mostrar resultado tests:		
Opciones	Valores	
Textual	Tu temperamento predominante es X1 con un %, tienes bastante de X2 con un %, un poco de X3 %y muy poco de X4 %.	
Numérico	80% X1, 20% X2, 10% X3,10% X4	
Grafico 2D	Barras verticales	
Grafico 3D	Diagrama de quesitos	
Paginación de elementos		
Opciones	Elementos por página	Elementos por página
Pocos	1 a 4	item item item item

Medio	1 a 8	item item item item item item item item
Muchos	1 a 16	item item item item item item item item item item item item item item item item

Tipo de letra

Opciones	Ejemplo
Verdana	Ipsum lorem
Times New Roman	Ipsum lorem
Comic Sans	Ipsum lorem
Impact	Ipsum lorem

Efectos de Texto

Opciones	Valores
Parpadeo	Si No
Desplazamientos	Horizontal Vertical No
Sombreado	Grueso Fino No
Efectos	Si No

Enlaces

Opciones	Valores
Texto	Inicio Atrás Anterior Siguiente
Iconos	-

Botones

Opciones	Valores
Texto	Entrar Salir Ver Actualizar Borrar
Iconos	-

Fondo de la página web.

Colores las ventanas

Iconos de la aplicación

Opciones	Valores
Texto	-
Colores vivos y fuertes	-

Estáticos	-
Dinámicos	-

Personalizar mensajes según el tipo de usuario

Opciones	Valores
Inicio de sesión	Formal informal no
Fin de sesión	Formal informal no
Fechas importantes	Cumpleaños fiestas no

Mostrar cada tiempo aleatorio un cita, chiste, comentario, ayuda

Densidad gramatical

Nº de etapas de los procesos (1 etapa, varias etapas, muchas etapas)

4.4 Implementación

Existen dos opciones de desarrollo para construir una aplicación para Facebook [25]:

- a) Construir una aplicación web y conectarla a Facebook usando el API – Facebook Connect.
- b) Construir una aplicación usando su lenguaje de marcado FBML.

Hemos optado por construir una aplicación web y conectarla a Facebook usando el API Facebook Connect [42], porque en la actual hoja de ruta de *Facebook* se ha abandonado el desarrollo de aplicaciones Facebook con FBML. A partir del 2011 no se podrán construir más aplicaciones con FBML.

Básicamente nuestra aplicación se muestra dentro de *Facebook* mediante un *iframe*. Y utilizamos el API de *Facebook* para realizar el registro, login y acceder a los datos de nuestros usuarios en la Red Social.

Al utilizar el API de *Facebook* para realizar el login de los usuarios, no hemos tenido que implementar los casos de uso del registro, ni login de los usuarios porque se hacen a través de *Facebook*.

Hemos implementado la aplicación con tecnología PHP [50].

La capa de presentación esta formada por archivos PHP que escriben dinámicamente el contenido HTML. El estilo de las páginas esta definido usando hojas de estilo CSS. En validación de los formularios y en determinadas funciones de la aplicación se utiliza código *JavaScript* [46]. Utilizamos *AJAX* [45] para realizar el refrescado asíncrono de los contenidos de todas las opciones de la aplicación en la parte central de la página principal. Para mostrar los mapas con vista del satélite que muestran la localización tanto de usuarios como de destinos accedemos al API de *Google Maps* [47]. Para mostrar las imágenes de los usuarios y de los destinos accedemos al API *GRAPH* de *Facebook*.

La capa de lógica y la capa de integración esta formada por clases de objetos implementadas con PHP.

La capa de datos de la aplicación es una base de datos MySQL[49], el código de la base de datos se encuentra en el archivo travelcobber.sql

Del estudio sobre el estado del arte de la sección 2 hemos tomado algunas ideas que nos han parecido interesantes. Y las hemos aplicado en nuestra aplicación. Por ejemplo poder determinar el estado de ánimo del usuario con iconos que reflejen el estado de ánimo, o disponer de un sistema de recompensas de puntuación, niveles y marcadores para fomentar la motivación y participación de los usuarios.

En el directorio raíz de la aplicación los siguientes archivos y carpetas:

Capa de Presentación:

Config_site_en_US → Palabras utilizadas en la aplicación en inglés

Config_site_en_ES → Palabras utilizadas en la aplicación en español

Todas las palabras y textos utilizados en la aplicación están definidos en estos archivos, en su correspondiente idioma.

De esta manera para que la aplicación este disponible en otro idioma. Solo hay que añadir el nuevo archivo en el que se definen los textos en ese idioma. Esto supone un gran ahorro de tiempo, al no tener que crear una nueva versión de la aplicación e ir teniendo que localizar y cambiar todos los textos en cada una de las interfaces gráficas.

Index.php → Pagina de inicio de la aplicación

Controlador.php → Es el controlador que responde a los eventos de la interfaz gráfica (index.php) e interacciona con la Capa de Lógica

admin/ → interfaces gráficas de la zona privada de administración y gestión

destinos/ → interfaces gráficas de los destinos de viaje

games/ → interfaz gráfica del marcador de la aplicación (SCORE TOP-10)

img/ → imágenes utilizadas en la aplicación web

includes/ → archivos de inclusiones donde están definidas secciones de la página principal de la aplicación

jpgraph → Librería para realizar los gráficos con la información estadística.

js/ → Funciones javascript

js/quizz.js → Código javascript que valida y envía los formularios de la aplicación.

js/ajax.js → Código javascript con el que se realizan las peticiones AJAX.

Log/phperror.txt → Log de errores

quizzes → interfaces gráficas de los test de temperamento y satisfacción

theme/ → hojas de estilo CSS

users/ → interfaces gráficas de usuarios y grupos

Capa Lógica:

Src/AppConfig.php → Clase que contiene los parámetros de configuración de la aplicación.

Src/facebook.php → Clase que contiene el código necesario para conectarse al API de Factbook[42].

Src/logica → Contiene las clases de la Capa de lógica.

Src/logica/Usuarios/

Usuario.php → Clase Usuario.

Modela la información de un usuario de la aplicación
Contiene los atributos, accedentes y mutadores del Modelo del Usuario,
y los métodos altaUsuario(listaParametros),
modificaUsuario(listaParametros), eliminaUsuario(id),
obtenerUsuario(id), obtenerUsuario(nombre)

Usuarios.php → Clase Usuarios.

Es una clase que contiene un Array con usuarios de la aplicación.
Métodos: obtenerUsuarios(parametros de consulta)

Grupo.php → Clase Grupo.

Modela la información de un grupo de usuarios de la aplicación.
Contiene los atributos, accedentes y mutadores del Modelo del Grupo,
y los métodos altaGrupo(listaParametros),
modificaGrupo(listaParametros), eliminaGrupo(idGrupo),
ponUsuario(idUser), quitaUsuario(idUser)
obtenerGrupo(idGrupo), obtenerGrupo(name),

Grupos.php → Clase Grupos

Es una clase que contiene un Array con grupos de la aplicación.
Métodos: ponGrupo(idGrupo), quitaGrupo(idGrupo),
obtenerGrupos(parametros de consulta).

Src/logica/Destinos/

Destino.php → Clase Destino

Modela la información de un destino de viaje.
Contiene los atributos, accedentes y mutadores del Modelo del Destino,
y los métodos altaDestino(listaParametros),
modificaDestino(listaParametros), eliminaDestino(id),
obtenerDestino(id), obtenerDestino(nombre).

Destinos.php → Clase Destinos

Es una clase que contiene un Array con destinos de la aplicación.
Métodos:
ponDestino(idDestino), quitaDestino(idDestino),
obtenerDestinos(parametros de consulta).

Src/logica/Gui/

Gui.php → Clase Gui

Clase que modela los elementos configurables de una interfaz gráfica.
Se utiliza para manejar la información de la configuración de la
interfaz gráfica que realiza un usuario

Contiene los atributos, accedentes y mutadores del Modelo de la GUI, y
los métodos:

- altaGui(idUsuario, listaParametros): Guarda la configuración de la interfaz gráfica de un usuario.
- modificaGui(idUsuario, listaParametros): Modifica la configuración de la interfaz gráfica del usuario.
- eliminaGui(idUsuario): Eliminar el registro de la configuración de la interfaz gráfica del usuario.

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

- obtenerGui(idUsuario): Obtener la configuración de la interfaz gráfica de este usuario.

Src/logica/Recomendador

Recomendador.php → Clase Recomendador

Modela el sistema de Recomendación de la aplicación

Contiene los atributos, accedentes y mutadores del Modelo del sistema de recomendación.

Métodos:

- dameAmigos(): Devuelve un listado con los usuarios que tienen mayor valoración
- dameAmigos (usuario, temperamento): Devuelve un listado con las recomendaciones de amigos para este tipo de usuario y tipo de temperamento basándose en los casos de uso registrados en el sistema.
- dameContents (usuario, temperamento): Devuelve un listado con las recomendaciones de contenidos para este tipo de usuario y temperamento, basándose en los casos de uso registrados en el sistema.
- dameContents(): Devuelve un listado con los contenidos con mayor valoración.
- dameGUI(usuario, temperamento): Devuelve una recomendación de configuración de la interfaz gráfica para este tipo de usuario y temperamento, basándose en los casos de uso registrados en el sistema.
- ponCaso (User1, temperamento1, User2, temperamento2): Registra como caso de uso que el usuario1 con temperamento1 esta interesado en un tipo de usuario2 con temperamento2.
- ponCaso (User, temperamento, Content): Registra como un caso de uso el tipo de Contenido en el que esta interesado un usuario con este tipo de temperamento
- ponCaso (User, temperamento, Gui): Registra como un caso de uso la configuración de la interfaz grafica que hace este tipo de usuario con este tipo de temperamento.

Consideramos que un usuario esta interesado en otro cuando mira los detalles de su perfil, y consulta más información sobre él. Y que esta interesado en un contenido cuando mira los detalles del contenido, y consulta más información sobre el contenido.

Capa Integración:

Src/integración → Contiene las clases de la Capa de integración.

Son las clases que utiliza cada clase de la capa de Lógica para acceder al Modelo de Datos (la base de datos MySQL en este proyecto).

Src/integración/Usuarios

DAOUsuario.php → Clase DAOUsuario.

Cuenta con los siguientes métodos:

- altaUsuario(listaParametros): Registra los datos de un usuario en en el Modelo de Datos
- modificaUsuario(listaParametros): Modifica los datos de un usuario registrados en el Modelo de Datos

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

- eliminaUsuario(idUsuario): Registra en el Modelo de Datos que el uso pasa al estado inactivo.
- obtenerUsuario(idUsuario): Obtiene los datos del usuario que tiene ese número de identificación
- obtenerUsuario(nombre): Obtiene el usuario por su nombre
- obtenerUsuarios(consulta): Obtiene un listado de los usuarios que cumplen la restricciones de la consulta.

DAOGrupo.php → Clase DAOGrupo.

Src/integración/Destinos

DAODestino.php → Clase DAODestino.

Src/integración/Gui

DAOGui.php → Clase DAOGui.

Src/integración/DAOReco.php → Clase DAOReco.

5 Resultados obtenidos

eTraveler

Para analizar los técnicas y problemas de los sistemas de recomendación individuales, desarrollamos un sistema de recomendación en una aplicación web que simula una Agencia de Viajes online en la que se pueden reservar plazas de hotel, vuelos o trenes.

En la página de inicio de la aplicación web se ofrecen las recomendaciones al usuario. Al hacer clic en un producto se dirige directamente al usuario a la página de compra. Los usuarios registrados con permisos de edición pueden entrar en la zona privada de gestión de la web y gestionar las recomendaciones ofrecidas al usuario.

Tecnologías empleadas: Modelo Vista Controlador. Capa Lógica: JAVA. Capa Presentación: HTML y .jsp. Capa de Datos: MySQL. Controlador: STRUTS.

Con este experimento hemos estudiado mejor las características de los sistemas de recomendación. Y hemos determinado cuales son las características que deben tener los destinos de viaje o los productos para poder ser recomendados adecuadamente.

Esto nos ha servido para realizar un buen modelado del sistema de Recomendación y del sistema de Gestión de Contenidos.

The screenshot displays the eTraveler web application interface. At the top, there is a login section with fields for 'Usuario' and 'Clave', and buttons for 'Entrar' and 'Reseteo'. Below this is a main product list table with columns: Nombre, Plazas disponibles, Precio, Descripción, Tipo, Seleccionar, Unidades compra, and Acciones. The table lists products like 'Meninas', 'Ritz', 'Palasa', 'Madrid-Milan', and 'Milan-Madrid'. To the right, there are sections for 'Novedades' and 'Recomendaciones'. A green arrow points from the 'Recomendaciones' section to a 'Detalle del Producto' sidebar on the right. This sidebar shows details for a flight: 'Nombre: Madrid-Calabria', 'Unidades: 100', 'Precio Venta: 230.0 €', 'Descripción: Air Malta', 'Estado: activo', 'Origen: Madrid', 'Destino: Reggio', 'Salida: 17:00', 'Llegada: 19:48'. Below the sidebar is a 'Comprar' button. At the bottom, there is an 'Área de gestión' section with a navigation menu and a 'Listado de Recomendaciones' table with columns: Cód. Rec., Posición, Descripción, Tipo, Estado, Acciones Rec., Cód. Producto, Producto, Unidades, Precio Venta, Descripción Producto, Tipo Producto, Estado Producto, and Acciones Prod. The table lists recommendation items like 'Oferta Primavera', 'Oferta Primavera', and 'BBB barato (Calabria, Italia)'.

Fig. 24. eTraveler.

TravelCobber 1.0

Versión inicial de la aplicación *TravelCobber* disponible en *Facebook* (en producción).

La aplicación permite que sus usuarios puedan encontrar amigos con su mismo temperamento e intereses. Además pueden encontrar destinos de viaje que pueden ser de su interés.

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

Tecnologías empleadas: Modelo Vista Controlador. Capa Lógica: PHP. Capa Presentación: HTML y PHP. Capa de Datos: MySQL. Controlador: controlador.php (programado por nosotros).

Los usuarios registrados en *Facebook* pueden acceder a la aplicación con un solo click. Al acceder a la aplicación se realiza un modelado automático del usuario a partir de su información disponible en *Facebook*.

Al introducir destinos de viaje en la aplicación se realiza un modelado del destino de viaje en el que se incluyen la url de una página que ofrezca más información sobre el destino, el identificador de una página sobre el destino en *Facebook* y su localización. Al guardar esta información, en la capa de presentación podemos acceder a diversas APIs para proporcionar más información al usuario. Con el identificador de la página de *Facebook* utilizando el API GRAPH de *Facebook* podemos acceder entre otras cosas a las fotografías que han subido los miembros de esa página. Al disponer de la localización del destino (latitud y longitud), accedemos al API Google Maps[47] para obtener un mapa híbrido (vista del satélite más mapa de calle) con un marcador señalando la localización del destino de viaje.

El usuario puede:

- Conocer que tipo de temperamento tiene.
- Buscar amigos en *Facebook*, filtrando los resultados por sexo, edad, localización y temperamento.
- Modificar la información de su perfil en la aplicación.
- Ver los perfiles de otros usuarios.
- Acceder al perfil *Facebook* de otros usuarios.
- Buscar destinos de viaje. Ver información detallada del destino, como su localización en mapas y dirigirse a páginas en *Facebook* sobre esos destinos de viaje.
- Realizar la encuesta de satisfacción.



Fig. 25. Imagen de la version 1.0 de la aplicación TravelCobber.

El usuario con permisos de administración puede acceder a la zona privada de administración donde puede realizar las siguientes acciones:

- Gestionar usuarios.
- Ver los datos y gráficos con la información de la aplicación.

El usuario con permisos de edición puede gestionar los destinos de viaje.

Además se utiliza el API de Google Maps para mostrar la localización en el mundo de los usuarios de la aplicación en una imagen con vista satélite.

La aplicación aprende del uso que hacen de ella los usuarios. Inicialmente recomienda los usuarios y destinos de viaje que tienen una mayor valoración o se ajustan más a los intereses del usuario. Pero conforme se van registrando como casos bases en el sistema de recomendación, los tipos de usuarios y destinos que interesan a cada tipo de usuario. Se irán ofreciendo mejores recomendaciones basadas en los casos de uso de los usuarios anteriores.

Actualmente se esta depurando esta versión con el feedback con las opiniones y errores que están reportando los usuarios.

TravelCobber 1.1

Esta versión añade a la anterior las siguientes características.

- Configuración de la aplicación.
- Adaptación de la aplicación al temperamento.

Esta versión esta desarrollada pero no esta disponible a los usuarios de *Facebook*. Esta alojada en el servidor en el entorno de staging. Y no se subirá a producción hasta que se termine de depurar la versión 1.0 con el feedback recibido por los usuarios, y hasta que no termine la comprobación de la versión 2.0 en el entorno de staging.

TravelCobber 1.2

Esta versión añade a la anterior las siguientes características.

- Formación de grupos de usuarios

Esta versión no esta disponible todavía a los usuarios de *Facebook*.

Esta alojada en el entorno de desarrollo del servidor, y se están terminando todavía ciertos aspecto de su desarrollo relativos a la formación de grupos de usuarios.

Experimentos

La versión 1.0 de la aplicación ha probado online con las personas que han usado la aplicación y que estaban disponibles en el chat de *Facebook* para comprobar que podían realizar las funcionalidades y casos de uso de la aplicación.

Se ha comprobado como podían hacer login en la aplicación, como se recogían automáticamente sus datos de *Facebook* en el Modelo de Datos de nuestra aplicación. Y como podían realizar todos los casos de uso de la aplicación, realizar el test de temperamento, el test de satisfacción, consultar y actualizar su perfil, buscar amigos y ver los perfiles de otros usuarios en la aplicación.

Esta comprobación de la aplicación con usuarios y datos reales ha sido muy importante para corregir errores de la aplicación que no les permitían entrar en la aplicación, no se recogían algunos de sus datos de su perfil de *Facebook*, no podían realizar búsquedas dentro de nuestra aplicación o su perfil no se visualizaba para el resto de usuarios.

La aplicación se ha probado con usuarios reales localizados en diferentes países como España, Italia, Alemania, Israel, Rusia y Estados Unidos.

La aplicación se ha probado con los siguientes navegadores Web: Internet Explorer, FireFox, Google Chrome, Opera, Safari. Esto ha servido para corregido los bugs que aparecían en la aplicación al usarla con determinados navegadores, concretamente con Internet Explorer.

Conseguir que la aplicación funcionase (concretamente el código JavaScript[46]) y se visualizase correctamente en estos navegadores ha sido bastante complicado. Ha supuesto mucho tiempo y esfuerzo de desarrollo. Según aparece en los capítulos 1 y 2 del libro "*Beginning Google Web Toolkit*" [44] utilizar Google Web Toolkit podría ser una solución para estos problemas. Por lo que sería recomendable probar esta opción en futuros desarrollos.

Además algunos de los usuarios que han probado la aplicación nos han expresado sus opiniones (a través de mensajes en el *Facebook*), sobre aspectos de la aplicación que según ellos se podían hacer mejor. Y sus críticas referentes a las cosas de la aplicación que no les habían gustado o que no entendían a la hora de utilizar la aplicación.

Las opiniones y críticas de estos usuarios se han tomado en cuenta, y han sido muy valiosas para mejorar la aplicación.

Usuarios de la aplicación

Los usuarios de la aplicación con permisos de administración pueden acceder a una zona privada de administración.

En dicha zona entre otras funciones se encuentran los datos estadísticos de la aplicación.

Entre esos datos estadísticos existen unos gráficos que se actualizan automáticamente con la información de los usuarios.

Esta información sirve para que los gestores de la aplicación puedan determinar el perfil de sus usuarios (edad, sexo, temperamento, localización). Y ciertas informaciones como por ejemplo el número de veces

que los usuarios entran en la aplicación en función de su sexo, edad, temperamento o localización.

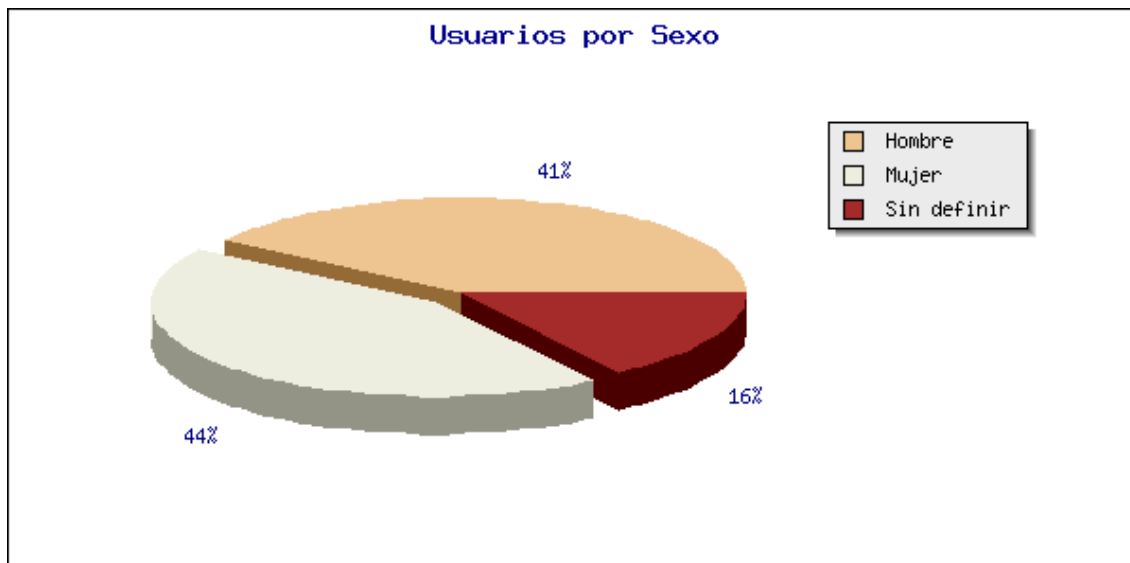


Fig. 26 Distribución de los usuarios de la aplicación *TravelCobber* por sexos..

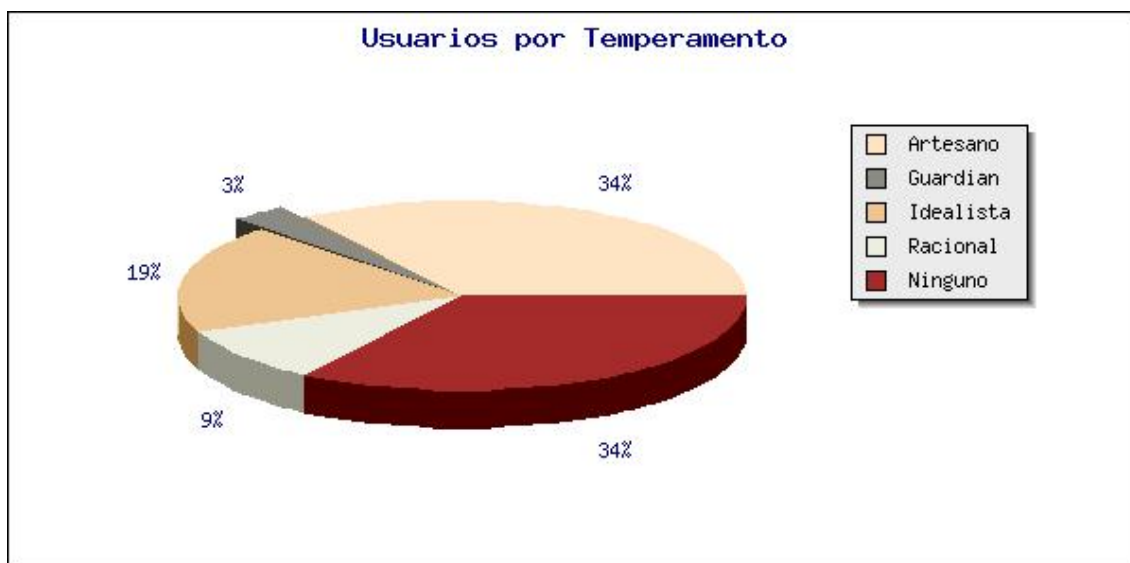


Fig. 27. Distribución de los usuarios de la aplicación *TravelCobber* por temperamentos.

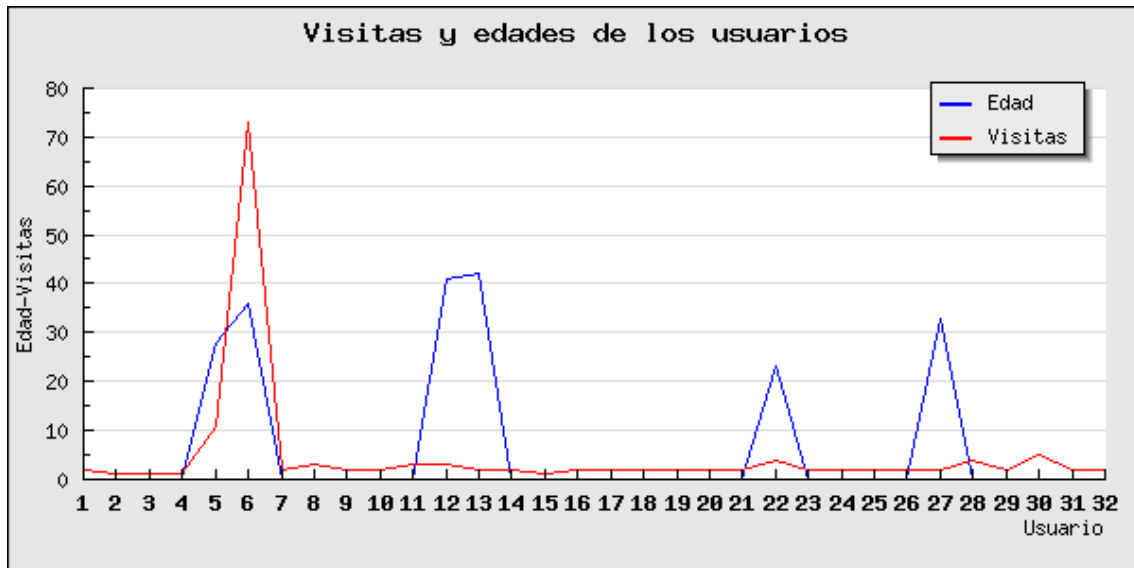


Fig. 28. Número de visitas de los usuarios de la aplicación TravelCobber en función de su edad.

En el gráfico de la figura 28 se muestra la distribución de las visitas de los usuarios a la aplicación *TravelCobber* y se relaciona con la edad de los usuarios. Esta información puede servir para que el administrador de la aplicación realice los cambios que estime oportunos en la configuración de aplicación, para intentar aumentar el número de visitas de ciertos rangos de edad.

Por otra parte en esa misma figura se observa que son pocos los usuarios para los que se ha podido determinar su edad a partir de sus datos de *Facebook*.

La aplicación determina la edad de los usuarios a partir de la fecha de nacimiento que figura en su perfil de *Facebook*. Sin embargo son pocos los usuarios que ponen esta información en su perfil. La mayoría en todo caso solo pone el día y el mes de su cumpleaños. Solo el 9% de los usuarios de la aplicación tenían esta información disponible en su perfil. Es decir para el 81% de los usuarios provenientes de *Facebook* no se podía determinar su edad.

A pesar que en nuestra aplicación los usuarios en la zona de su perfil pueden actualizar sus datos, son muy pocos los que pusieron su edad.

La conclusión es que los usuarios están más dispuestos a compartir cierta información sobre datos como el género (85% de las personas) y su temperamento (66% de las personas), que por ejemplo la edad. Solo el 19% de los usuarios están dispuestos a que otros usuarios conozcan su edad.

Encuestas

En la aplicación existe un cuestionario de satisfacción que el usuario puede cumplimentar con el objeto de obtener su satisfacción con la aplicación.

En la zona de administración también se puede acceder a los gráficos que analizan los resultados de las encuestas, y que se actualizan

automáticamente con los resultados de las encuestas registrados también automáticamente en el Modelo de datos.



Fig. 29. Gráfico con las opiniones de los usuarios respecto a la interfaz gráfica.

Podemos decir que la interfaz gráfica le gusta a un 66% de los usuarios, sin embargo hay un 33% a los que no les ha llegado a convencer la interfaz.



Fig. 30. Gráfico con las opiniones de los usuarios respecto a la opciones que se les presentan en la aplicación.

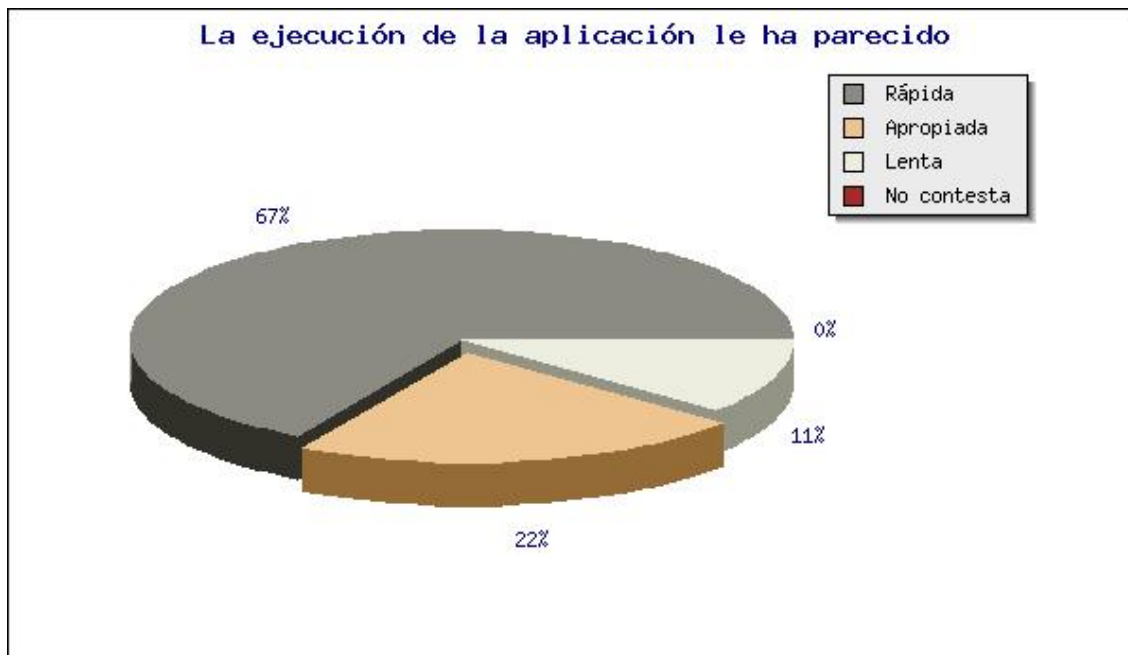


Fig. 31. Gráfico con las opiniones de los usuarios respecto a la rapidez de la ejecución de la aplicación.

A un alto porcentaje de la aplicación (89%) le parece bien la velocidad de ejecución de la aplicación.



Fig. 32. Gráfico con las opiniones de los usuarios respecto al resultado del test de temperamento.

Un 33% de los usuarios no están del todo conformes con el temperamento asignado. En algunos casos esto se debe a que algunas personas pensaron que solo tenían que responder a la única pregunta que se veía en la

pantalla. No hicieron el scroll hacia abajo para responder las otras 15 preguntas.

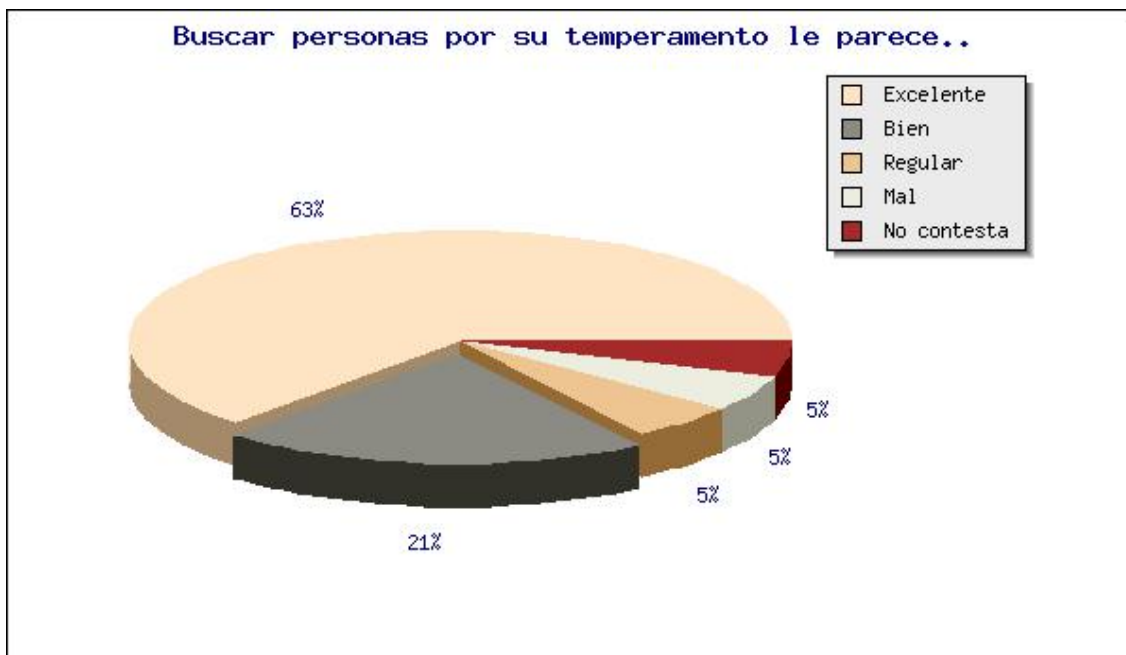


Fig. 33. Gráfico con las opiniones de los usuarios respecto a la búsqueda de usuarios por temperamento.

Un 84% de los usuarios consideran de utilidad poder buscar personas por su temperamento.

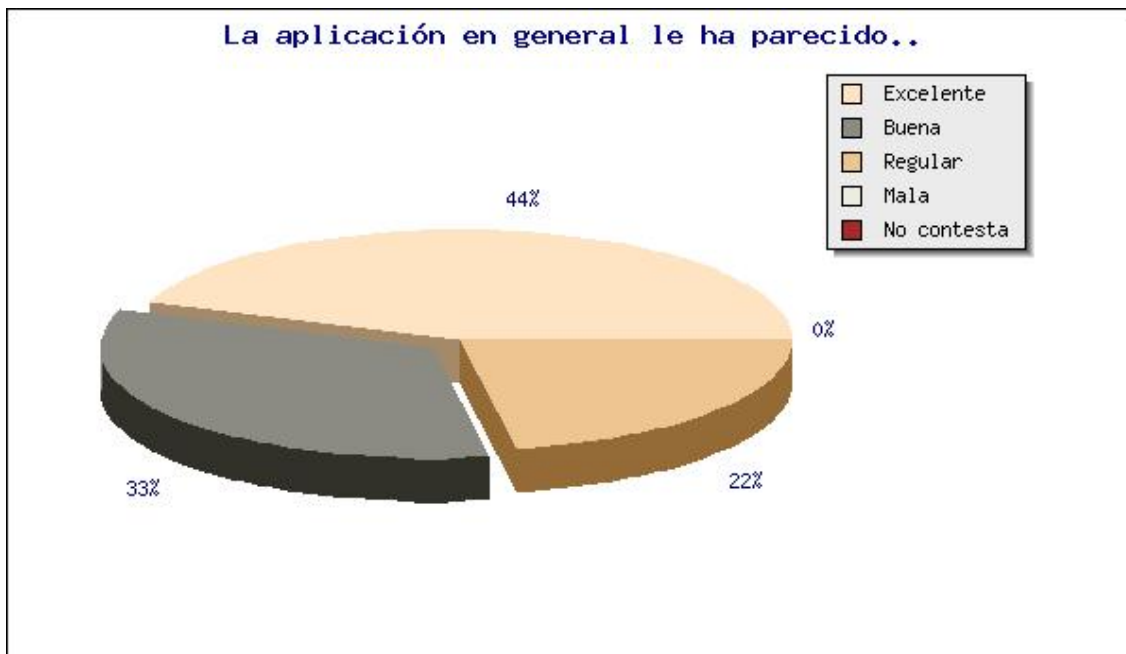


Fig. 34. Gráfico con las opiniones de los usuarios respecto a la aplicación en general.

Podemos decir que la aplicación ha gustado al 77% de los usuarios.

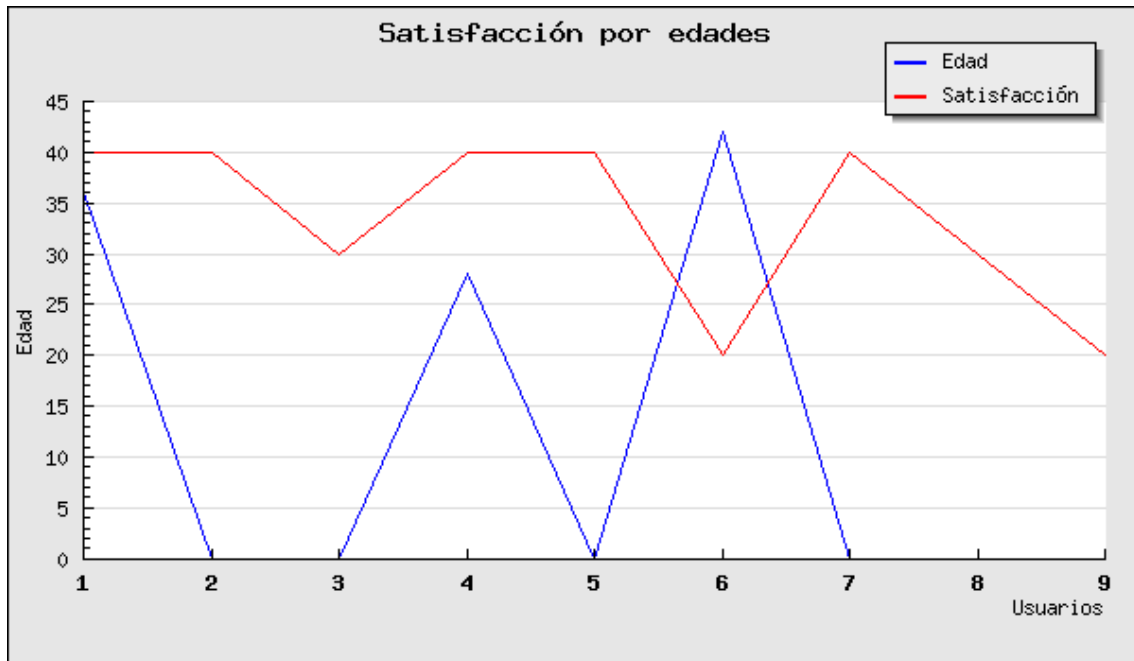


Fig. 35. Satisfacción de los usuarios de la aplicación TravelCobber en función de su edad.

En la zona de administración también se pueden ver gráficos que relacionan los resultados de la encuesta de satisfacción con determinadas características de los usuarios. Se puede conocer como de bien se esta satisfaciendo a los usuarios en función de su edad, genero, temperamento o localización. Para de esta forma poder tomar las medidas oportunas para aumentar la satisfacción de estos usuarios.

En general los resultados de las encuestas son muy favorables. Sin embargo hay que tener en cuenta que los usuarios que las han respondido son personas con las que mantenemos una relación de amistad. Quizás esto implique una valoración poco objetiva de la aplicación, que puede explicar unos resultados tan favorables.

De todas formas me parece muy positivo que la interfaz gráfica sea aceptable por los usuarios, que la aplicación se ejecute con rapidez. Y poder comprobar que determinar el temperamento de los usuarios es algo relevante para ellos. Y que en general los usuarios estén contentos con la aplicación.

6 Conclusiones

A nivel de sistemas de recomendación individuales, el sistema de recomendación se debe elegir en función de las características de la aplicación web. Se deben introducir métricas para estudiar el aumento del número de visitas y conversiones procedentes del recomendador y usar dichos datos para actualizar el posicionamiento de las recomendaciones [9]. El sistema debe poder gestionar las recomendaciones. Las recomendaciones deben tener explicaciones.

Un correcto modelado del usuario mejora tanto las recomendaciones como la adaptabilidad y la experiencia del usuario de la aplicación web.

Hay que tener en cuenta el contexto del usuario a la hora de generar la recomendaciones, por ejemplo si el contexto es un terminal móvil se deben ofrecer recomendaciones single-shot, y no recomendaciones conversacionales.

Es conveniente desarrollar un recomendador inteligente, entendiendo por inteligencia la consecución de objetivos [Russel-10]. En el estado inicial (arranque en frío) proporcionaría recomendaciones basadas en consultas al Modelo de Datos con la información extraída de la Red Social sobre los datos de usuario y sus preferencias. Según se adquiriera conocimiento de los usuarios y las métricas se ofrecerá la composición de recomendaciones colaborativas y basadas en casos que produzcan mejores resultados.

A nivel de recomendadores grupales, la integración de los estudios realizados en el campo de los recomendadores grupales en un único modelo, y la integración de dicho modelo en las redes sociales puede generar beneficios tanto a los usuarios (que se ahorran el tener que perder tiempo introduciendo información de sus preferencias en cada negocio web), para el sistema de recomendación (ahorro del tiempo y esfuerzo de desarrollo necesarios para que el sistema gestione las preferencias de los usuarios individuales, y proporcione formas de comunicación grupales para las decisiones de grupo), para los negocios (las recomendaciones pueden llegar a un número de personas muchísimo mayor), y para la red social al ofrecer un nuevo (mejores técnicas de agregación y modelado de grupos) y mejor servicio social a sus usuarios.

El modelo de integración de un sistema de recomendación en la Red Social mitiga las limitaciones por los problemas de arranque en frío del usuario existentes en los sistemas de recomendación. Cuando un nuevo usuario llega por primera vez a la aplicación se realiza un Modelado del usuario extrayendo automáticamente los datos registrados en su perfil de la Red Social.

También sirve para mitigar el problema del arranque en frío del producto (la necesidad de un gran número de valoraciones previas de los productos para realizar buenas recomendaciones en los sistemas de recomendación colaborativos) porque cuando los usuarios ingresan en la aplicación la información referentes a los productos y actividades que le gustan, no solo sirve para modelar las preferencias del usuario, sino que también sirve como valoración de la calidad del contenido. Si muchos usuarios han

registrado en su lista de gustos de la Red Social que les gusta un mismo contenido, podemos entender que a esos usuarios les gusta ese contenido, y que ese contenido tiene una buena valoración o calidad.

Soluciona también el problema del cambio en las preferencias del usuario, porque cuando el usuario entra en la aplicación se actualizan sus datos y preferencias con los registrados en la Red Social.

Por otra parte la motivación social también resuelve el inconveniente de las valoraciones inconsistentes de los usuarios.

Además se puede introducir diversidad en las recomendaciones presentando productos, que si bien no encuadran en las preferencias del usuario, les han gustado a amigos del usuario.

En cuanto a los sistemas de recomendación para grupos el Modelo de Integración del Sistema de Recomendación en la Red Social cuenta con las ventajas de los recomendadores aprendidas al realizar el estudio de su estado del arte. Como por ejemplo que los miembros del grupo puedan ver las preferencias individuales del resto del grupo. Pero evitando que los usuarios se tengan que registrar, e introducir sus datos y preferencias cada vez que quieran obtener una recomendación de una aplicación web.

El modelo cuenta además con la ventaja de poder personalizar la aplicación en función del temperamento y estado de animo de los usuarios, para mejorar su la experiencia del usuario y su satisfacción. Siento esto algo que no se había contemplado en los sistemas de recomendación actuales.

Sin embargo, tras los experimentos realizados hemos podido comprobar que una cosa es poder acceder al perfil de los usuarios de la Red Social. Y otra que los usuarios compartan sus datos en dicha Red Social. Hemos comprobado como hay datos que si están dispuestos a compartir, tales como genero o las cosas que les gustan, sin embargo hay otros que no quieren difundir como es el caso de la edad. La localización es un dato que tampoco es muy frecuente en los perfiles de los usuarios.

Por eso parece una buena idea realizar una encuesta con el pretexto de conocer cual es el temperamento de famoso que tiene el usuario. Porque de esta forma se le motiva a realizar la encuesta. Y podemos determinar un rasgo como el temperamento, que sirve para realizar un mejor modelado del usuario. Y posteriormente adaptar la aplicación a su temperamento para mejorar la experiencia del usuario. Sin hacer la encuesta no podríamos determinar su temperamento, pero además el usuario como en el caso de la edad tampoco estaría muy dispuesto en compartir este tipo de información.

Los resultados de las encuestas son muy positivos. La interfaz gráfica gusta a los usuarios. La aplicación se ejecuta con rapidez. Les gusta la aplicación. El temperamento les parece algo importante a la hora de buscar nuevos amigos.

Sin embargo, pensamos ampliar este estudio con un mayor número de usuarios, con los que además no mantengamos ningún tipo de relación de amistad. Para poder así determinar la validez de estos resultados. Y porque por otra parte nos resulta muy interesante poder ampliar el estudio para determinar que información comparten los usuarios en la Red Social.

Futuras Extensiones

- Intercambio de mensajes de texto entre los usuarios de la aplicación.
- Integración del Chat de Facebook en la aplicación.
- Integración de APIs de Google (imágenes, links, RSS) para ofrecer más información a los usuarios.

7 Documentación del sistema

Manual de usuario

Para acceder a la aplicación es necesario disponer de un navegador web y contar con conexión a Internet.

1º) Escribir la siguiente url en la barra del navegador:

<http://apps.facebook.com/travelcobber/>

Aparecerá la página de entrada a la aplicación.



Fig. 36 Página de entrada a la aplicación *TravelCobber*.

2º) Hacer click en el botón que pone "*Connect with Facebook*".

3º) En caso de no haber iniciado una sesión en *Facebook*, aparecerá una pantalla para que iniciemos una sesión en *Facebook*.



Fig. 37. Inicio de sesión en *Facebook*.

Para ello debemos el correo electrónico y password de nuestra cuenta de *Facebook*. En el caso de no disponer de una cuenta en *Facebook* hay pulsar el enlace "*Registrarse en Facebook*" crear una cuenta y posteriormente volver a la url de entrada a la aplicación (Paso 1).

4º) Si el correo electrónico y el password de acceso a *Facebook* son correctos entraremos en la página principal de *TravelCobber*.

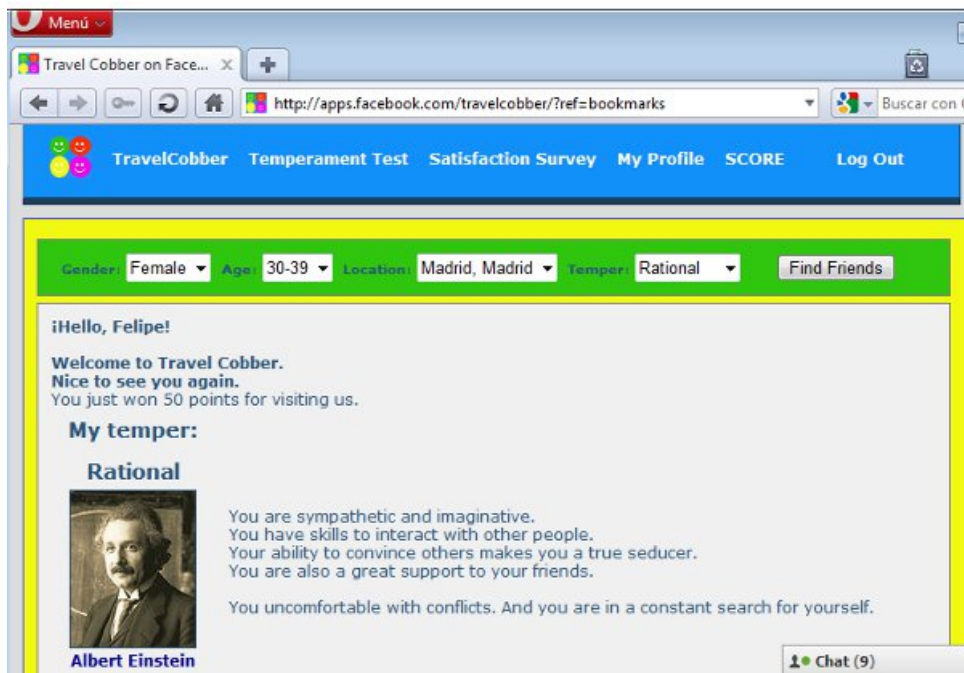


Fig. 38 Página principal de la aplicación *TravelCobber*.

5º) En la parte superior de la página principal hay un menú horizontal con siguientes opciones:

- **TravelCobber:** Para volver a la página de inicio
- **Destinos:** Permite acceder a destinos de viaje. Se muestra una imagen del destino, y su valoración. Al hacer "*click*" sobre el destino se muestran en la misma página más información sobre el destino, incluyendo un enlaces a la pagina sobre el destino en *Facebook*, o un mapa de *Google Maps*[47] mostrando la localización del destino

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

- **Test de Temperamento:** Para realizar el test de temperamento.
- **Encuesta de Satisfacción:** Para realizar el test de satisfacción
- **Mi Perfil:** Para ver y poder modificar nuestro perfil en la aplicación.
- **SCORE:** Permite ver un listado con los 10 usuarios que tienen mayor puntuación en la aplicación.
- **LogOut:** Para salir de la aplicación.

6º) **Buscar amigos:**

- Introducir los criterios de búsqueda género, rango de edad, localización y temperamento de los amigos que estamos buscando.
- Pulsar el botón "Buscar Amigos".

7º) **Ver el perfil de otros usuarios:**

Al realizar una búsqueda aparece como resultado las imágenes y los nombres de los usuarios que cumplen los criterios de búsqueda. Para poder ver sus perfiles tenemos que hacer "click" con el ratón en la imagen o el nombre del usuario.



Fig. 39 Resultado de una búsqueda en la aplicación *TravelCobber*.

Una vez que aparece el perfil del usuario, podemos ver sus datos, sus intereses y acceder a su perfil de Facebook haciendo "click" con el ratón sobre el icono que tiene forma de lupa.

Manual de sistema

Para poder ejecutar la aplicación es necesario tener un servidor web que escuche las peticiones http de los usuarios, en dicho servidor deberemos instalar además un intérprete de páginas PHP 5.0, y MySQLServer para alojar la base de datos mysql de la aplicación.

Master Investigación en Informática (U.C.M) – Tesis Fin de Master

En la carpeta en la que nuestro servidor busca las páginas web alojaremos la carpeta "travelcobber".

Accederemos a MysqlServer\bin para crear la base de datos de la aplicación a partir del fichero travelcobber.sql.

8 Referencias

1. Gómez-Gauchia, H., Díaz-Agudo, B, González-Calero, P.: COBBER: Ontology Based Model for Human Centered Computing. Universidad Complutense Madrid, Spain (2006)
2. Bridge, D., Göker, M., McGinty, L.: Case-based recommender systems. Cambridge University Press, United Kingdom (2006)
3. McGinty, L., Smyth, B.: On the Role of Diversity in Conversational Recommender Systems. University College Dublin, Ireland
4. Jameson, A.: More than the sum of its Members: Challenges for Group Recommender Systems. Germany
5. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Communications 7 (1), 39-59 (1994)
6. Ontañón, S., Mishra, K., Sugandh, N., Ram, A.: Learning from Demonstration and Case-Based Planning for Real-Time Strategy Games. Georgia Institute of Technology, Atlanta
7. Tintarev, N.: User-centered issues in Recommender Systems. Telefónica Research, Barcelona
8. Recio-Garcia, J. Jimenez-Diaz, G., Sanchez-Ruiz, A. Diaz-Agudo, B.: Personality Aware Recommendations to Groups. Universidad Complutense Madrid, Spain
9. Governor, J., Hinchcliffe, D., Nickull, D.: Web 2.0 Architectures. O'Reilly Media, Inc., 2009.
10. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall, 1995
11. Bernstein, M., Tan, D., Smith, G., Czerwinski, M., Horvitz, E: Collabio: A Game for Annotating People within Social Networks. MIT CSAIL, Microsoft Research, (2009)
12. Baatarjav, E., Phithakkitnukoon, S., Dantu, R.: Group Recommendation System for Facebook. University of North Texas, Denton, Texas, USA,(2008)
13. Morris, J., Millen, D.: Identity Management: Multiple Presentations of Self in Facebook. IBM Research, (2007)
14. Jameson, A., Smyth, B.: What a Difference a Group Makes: Web-Based Recommendations for Interrelated Users. DFKI German Research Center for Artificial Intelligence, University College Dublin.
15. Chen, Y.-L., Cheng, L.-C., Chuang, C.-N.: A group recommendation system with consideration of interactions among group members. National Central University, Taiwan (2008)
16. McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P.: CATS: A synchronous Approach to Collaborative Group Recommendation. University College Dublin (2006)
17. McCarthy, McGinty, L., Smyth, B.: Case-Based Group Recommendation Compromising for Success. University College Dublin (2007)
18. Cantador, I.: Exploring the conceptual space in hybrid recommender systems. Universidad Autónoma de Madrid (2008)

19. Cantador, I., Castells, P.: Extracting Multylayered Semantic Communities of Interest from Ontology-based and Hybrid Recommendations. Universidad Autónoma de Madrid (2008)
20. McCarthy, J.F., Anagnost, T.D.: MusicFX: An arbiter of group preferences for computer supported collaborative workouts. In: Proceedings of the 1998 Conference on Computer Supported Cooperative Work. (1998) 363 – 372
21. Lieberman, H., Van Dyke, N.W., Vivacqua, A.S.: Let ´ s Browse: A collaborative Web browsing agent. In Maybury, M.T., ed.: IUI99: International Conference of Intelligent User Interfaces. ACM, New York (1999) 65-68
22. O ´ Connor, M., Cosley, D., Konstan, J., Riedi, J.: PolyLens: A recommender system for groups of users. In: Proceedings of the European Conference on Computer-Supported Cooperative Work. (2001)
23. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: INTRIGUE: Personalized recommendation of tourist attraction for desktop and handset devices. Applied Artificial Intelligence 17 (2003) 687-714
24. Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., Boydell, O.: Exploiting query repetition and regularity in an adaptive community-based web search engine. User Modeling and User-Adapted Interaction: The Journal of Personalization Research 14 (2005) 383-423
25. http://wiki.developers.facebook.com/index.php/Creating_a_Platform_Application
26. Governor, J., Hinchcliffe D., Nickull D: Web 2.0 Architectures. Chapter 7 Specific Patterns of Web 2.0. O'Reilly Media, Inc., 2009.
27. Tretola, R.: Beginning Adobe Air. Building Applications for the Adobe Integrated Runtime. Wiley Publishing, Inc. 2008
28. Preciado, J., Linaje, M., Morales-Chaparro, R., Sanchez-Figueroa, F., Zhang, G., Kroi, C., Koch, N.: Designing Rich Internet Applications Combining UWE and RUX-Method. Universidad de Extremadura (Spain), Ludwig -Maximilians-Universitt Mnchen (Germany). ICWE 2008
29. Meli, S., Gmez, J., Prez, S., Daz, O.: A Model-Driven Development for GWT-Based Rich Internet Applications with OOH4RIA. Universidad de Alicante (Spain), Universidad del Pas Vasco (Spain). ICWE 2008
30. <http://www.w3.org/TR/2009/WD-wai-aria-practices-20091215/>
31. Koch, N., Pigeri, M., Zhang, G., Morozova: Patterns for the Model-Based Development of RIAs. Ludwig -Maximilians-Universitt Mnchen (Germany), Cirquent GmbH (Germany), LifeScience GmbH (Germany), ICWE 2009.
32. Hennicker, R., Koch, N.: Systematic Design of Web Applications with UML. Ludwig – Maximilians-Universitt Mnchen (Germany). 2001.
33. Preciado, J., Linaje, M., Morales-Chaparro, R., Sanchez-Figueroa, F., Zhang, G., Kroi, C., Koch, N.: Designing Rich Internet Applications Combining UWE and RUX-Method. Universidad de Extremadura (Spain), Ludwig -Maximilians-Universitt Mnchen (Germany). ICWE 2008

34. Meliá, S., Gómez, J., Pérez, S., Díaz, O.: A Model-Driven Development for GWT-Based Rich Internet Applications with OOH4RIA. Universidad de Alicante (Spain), Universidad del País Vasco (Spain). ICWE 2008
35. Tretola, R.: Beginning Adobe Air. Building Applications for the Adobe Integrated Runtime. Wiley Publishing, Inc. 2008
36. Deitel, P.J., Deitel, H.M.: AJAX, Rich Internet Applications, and Web Development for Programmers. Pearson Education 2008.
37. http://es.wikipedia.org/wiki/Dojo_toolkit
38. Rossi, G., Urbietta, M., Ginzburg, J., Distanto, D., Garrido, A.: Refactoring to Rich Internet Applications. A Model Driven Approach. Facultad de Informática UNLP (Argentina), Universidad de Buenos Aires (Argentina), University of Sannio (Italy)
39. Linaje, M., Preciado, J., Sanchez-Figueroa, F.: Engineering Rich Internet Applications User Interfaces over Legacy Web Models. Internet Computing, IEEE, Volume: 11, Issue: 6, pp.53-59, Nov-Dec. 2007.
40. Mesbah, A., Van Deursen, A.: Migrating Multi-page Web Applications to Single-page AJAX Interfaces. csmr, pp. 181-190, 11th European Conference on Software Maintenance and Reengineering (CSMR'07), 2007.
41. Álvarez, F.: Recomendaciones para grupos en Redes Sociales: Facebook. Universidad Complutense Madrid, Spain (2010)
42. Gerakines, N.: Facebook Application Development. Wiley Publishing Inc., Indianapolis, Indiana (2008)
43. Boronczyk, T., Psinas, M, E.: PHP y MySQL. Anaya Multimedia
44. Smeets, B., Boness, U., Bankras, R.: Beginning Google Web Toolkit. Apress (2008)
45. Zakas, N.,C., McPeak, J., Fawcett, J.: Professional Ajax. Wiley Publishing Inc., Indianapolis, Indiana (2006)
46. Goodman, D.: JavaScript Bible. 4th Edition. Hungry Minds, Inc., Indianapolis, Indiana (2001)
47. Purvis, M., Sambells, J., Turner, C.: Beginning Google Maps Applications with PHP and Ajax. Apress (2006)
49. Ullman, L.: MySQL. Pearson Educación S.A. (2003)
50. Boronczyk, T., Naramore, E., Gerner, J., Scouarnec, Y.,L., Stolz, J., Glass, M.,K.: Desarrollo Web con PHP 6, Apache y MySQL. Anaya Multimedia (2010)