Escuela Politécnica Superior de Jaén
Máster Universitario en Ingeniería Informática

Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

ESTUDIO DE INTERFACES GRÁFICAS PARA SISTEMAS DE RECOMENDACIÓN A GRUPOS

Alumno/a: **Rodríguez Molina, Juan Carlos**

Tutor/a: Prof. D. Luis Martínez López
D. Jorge Castro Gallardo
Dpto.: Departamento de Informática

Marzo, 2017

Escuela Politécnica Superior (Jaén)
Departamento de Informática

D. Luis Martínez López y D. Jorge Castro Gallardo, tutores del Proyecto Fin de Máster titulado: Estudio de interfaces gráficas para Sistemas de Recomendación a Grupos, que presenta Juan Carlos Rodríguez Molina, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Marzo de 2017

El alumno: Los tutores:

Juan Carlos Rodríguez Molina

Luis Martínez López

Jorge Castro Gallardo

Índice

Índice de Tablas

Índice de Figuras

CAPÍTULO 1
INTRODUCCIÓN

1. INTRODUCCIÓN

En este primer capítulo se introduce al lector en el contexto en el cual se inicia la especificación e implementación de la aplicación y especificar los objetivos que se pretenden alcanzar con su desarrollo, así como su motivación. Por último se especificará la estructura del proyecto para situar al lector y permitirle una mejor comprensión de este escrito.

1.1. Contexto

Antes de la llegada de Internet un consumidor de cualquier tipo de producto tenía un acceso limitado a la información relacionada tanto con el producto en sí como con otras posibles opciones. La publicidad se convertía, prácticamente, en la única forma de dar a conocer un producto, y el problema del usuario era como conseguir una información veraz. En el caso de productos culturales como el cine o la música, la radio o las revistas especializadas actuaban como únicos difusores de lo nuevo. Ahora la situación se ha invertido totalmente [12].

De la escasez de información se ha pasado a la sobrecarga, a tener acceso a una cantidad inagotable de creaciones culturales en tiendas online o en redes de intercambio. Ahora el problema ha tornado en como separar lo que queremos de lo que no queremos encontrar.

Para intentar ofrecer a cada persona lo que realmente le interesa, están comenzando a jugar un papel importante los sistemas de recomendación. La idea que subyace tras ellos es encontrar usuarios con gustos similares a los de otro determinado y recomendarle cosas que desconoce pero que gustan a aquellos con los que se tiene similitud [5].

Grandes plataformas online de la actualidad (como Amazon, Youtube, Spotify, Netflix, etc.), ofrecen al usuario, mediante estos sistemas basados en la similitud entre usuarios, una navegación más personalizada y ágil.

Los actuales algoritmos de recomendación han avanzado a una etapa en la que ofrecen una precisión muy alta en la predicción de elementos que gustarán a un

usuario o a todos los integrantes de un grupo, ya que el ser humano es un ser social e intenta realizar el mayor número de tareas en grupo. Esta es la motivación principal de la recomendación a grupos, en la que se fundamenta este proyecto y que se abordará en el siguiente capítulo.

Sin embargo, las recomendaciones resultantes no siempre son bien aceptadas por los usuarios. A veces los usuarios prefieren no compartir sus datos con el sistema debido a problemas de privacidad, o no confían en las recomendaciones del sistema, especialmente si no entienden cómo ha sido el proceso para generarlas [24].

Para ganarse la confianza del usuario, es importante explicarles el significado de las recomendaciones proporcionadas. Sin embargo una explicación aumentaría la carga cognitiva del usuario y por lo tanto no sería fácil de aceptar por el usuario. Una visualización del mecanismo o de los resultados puede ser útil, ya que "una imagen vale más que mil palabras" y puede ahorrar tiempo y esfuerzos de los usuarios al mostrar una representación intuitiva y comprensible de la recomendación, muy necesaria en las recomendaciones a grupos por su dificultad, y permitir controlar dichos mecanismos de visualización [37].

Es por ello que en este proyecto crearemos una aplicación de escritorio, desarrollada mediante el lenguaje de programación Java, y que, mediante el acceso a un servidor que genera recomendaciones de películas a grupos, nos permitirá obtener dicha información y representarla gráficamente utilizando distintas metodologías de visualización.

Una vez proporcionada la visualización de la información, el usuario tendrá control sobre dicha visualización, pudiendo ajustar diversos parámetros, obtener información adicional, o ver, por ejemplo, cómo influye la presencia de un miembro en una recomendación grupal.

1.2. Propósito

Este TFM propone el diseño y desarrollo de un prototipo de aplicación de escritorio que permita visualizar las recomendaciones a grupo obtenidas de un servicio web de forma gráfica y visual, mediante distintas técnicas, para que se adecúe de la mejor forma posible a cada tipo de producto y grupo.

1.3. Objetivos

El objetivo de este Trabajo de Fin de Máster es, principalmente, poder poner en práctica los conocimientos adquiridos durante el transcurso del Máster y del Grado en Ingeniería Informática, mediante la implementación de una aplicación de visualización de recomendaciones a grupos.

Con el fin de alcanzar este propósito general, se fueron marcando objetivos a lo largo de su desarrollo. Éstos objetivos se exponen a continuación:

- Revisión bibliográfica de Sistemas de Recomendación, Sistemas de Recomendación a grupo (SRG), Interfaces Gráficos y Visualización.
- Análisis, diseño e implementación mediante el framework Java de una aplicación de escritorio que permita visualizar gráficamente recomendaciones obtenidas mediante un SRG al que se accederá a través de un servicio web.
    - Desarrollar un componente que nos permita acceder y consumir de los distintos servicios web que necesitamos.
    - Desarrollar distintos tipos de visualización para las recomendaciones: Gráficas de líneas, Modelos jerárquicos y representación mediante escalado multidimensional.
- Desarrollar una memoria con la documentación del trabajo realizado.

1.4. Estructura del proyecto

A continuación, se realizará una breve descripción de los capítulos en los que está estructurado este proyecto, así como los contenidos expuestos en los mismos.

Capítulo 2. Sistemas de Recomendación.

En el segundo capítulo se describe qué son los sistemas de recomendación, dando una visión general de los mismos, para posteriormente profundizar en su clasificación, ventajas e inconvenientes de cada tipo, las tendencias actuales, y terminar prestando especial énfasis en los sistemas de recomendación a grupos, con los que se trabaja en este proyecto.

Capítulo 3. Visualización.

En este tercer capítulo se describe brevemente la importancia de visualizar las recomendaciones a grupo de manera gráfica, viendo cómo se relacionan entre ellas, y entre los distintos componentes del grupo.

Además se hace una revisión de las principales técnicas de visualización que existen en la actualidad.

Capítulo 4. Ingeniería del software.

Este capítulo recoge el proceso completo de ingeniería del software para el desarrollo del proyecto. Dado que estamos ante un proyecto de Ingeniera del Software, realizaremos todas las fases necesarias para el desarrollo de software (análisis de requerimientos, análisis y diseño del sistema e implementación, haciendo especial hincapié en el diseño de la interfaz de la aplicación Java) y las pruebas de usabilidad llevadas a cabo.

Además, en el apartado de implementación se recogerán las tecnologías empleadas, así como los lenguajes de programación y el tipo de arquitectura elegidos para el desarrollo del proyecto.

Capítulo 4. Conclusiones.

En este último capítulo es en el que se exponen las conclusiones generales derivadas de la realización de este proyecto.

ANEXOS I y II.

La sección final de esta memoria contiene dos anexos dedicados al Manual para la instalación de la aplicación Java, y al Manual de Usuario para su uso.

1.5. Planificación temporal

A continuación se añade una planificación temporal del proyecto. Esta planificación temporal es orientativa ya que, se están definiendo fechas de comienzo y de fin, sin a priori conocer la carga de tiempo disponible para el desarrollo.

Como base para la planificación del proyecto, se procede a crear una Estructura de Desglose del Trabajo (EDT), que de manera exhaustiva, jerárquica y descendente divida el proyecto en paquetes de trabajo formados por las tareas y entregables necesarios para lograr los objetivos del proyecto. Se incluyen en el plan todas las tareas necesarias para organizar el flujo de trabajo y de esta manera controlar su avance. Esta división clara y fácil de entender, estará planificada en el tiempo e identificada con el recurso humano asignado y responsable de cada una de las tareas. Pero en este proyecto, al realizarlo solo una persona, carece de sentido.

La EDT estará representada como un Organigrama, con un nivel de detalle descendente (desde la meta a las tareas específicas) en cuatro niveles:

Primer nivel - Meta: Creación del Proyecto "Estudio de interfaces gráficas para Sistemas de Recomendación a Grupos".

Segundo nivel - Objetivos: Los objetivos vendrán representados por las fases del ciclo de vida de un proyecto software, en nuestro caso Análisis, Diseño, Implementación y Periodo de Evaluación y despliegue.

Tercer nivel - Actividades: Las distintas fases, a realizar de manera secuencial, dentro de cada objetivo.

Cuarto nivel - Tareas: Cada una de las acciones que hay que realizar para llevar una actividad a cabo con éxito.

A continuación se puede ver, en la figura 1.1, la EDT de nuestro proyecto:

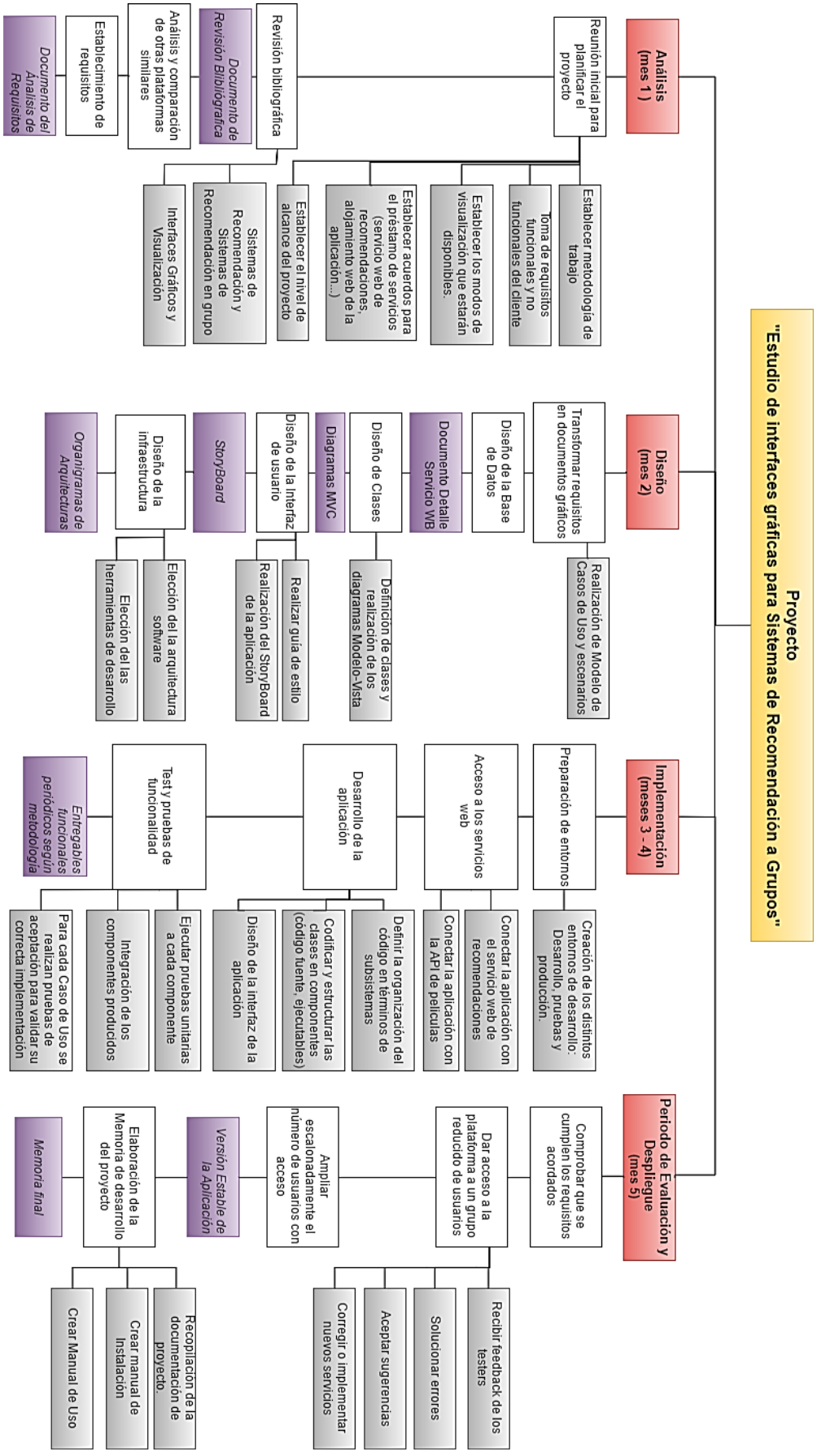

Figura 1.1 Diagrama de EDT.

Además se crea también un Diagrama de GANTT para ver la duración estimada en días reales de los objetivos más relevantes del proyecto. Dicha estimación es la siguiente:

Objetivo	Duración	F. Inicio	F. Fin	Predecesora
Análisis	29 días	03/10/2016	31/10/2016	-
Diseño	30 días	01/11/2016	30/11/2016	Análisis
Implementación	62 días	01/12/2016	31/01/2017	Diseño
Evaluación y Despliegue	30 días	01/02/2017	02/03/2017	Implementación
TOTAL	**151 días**			

Tabla 1.1 Estimación de la duración del proyecto.

A continuación se puede ver, en la figura 1.2, el Diagrama de GANTT de la planificación de las actividades.

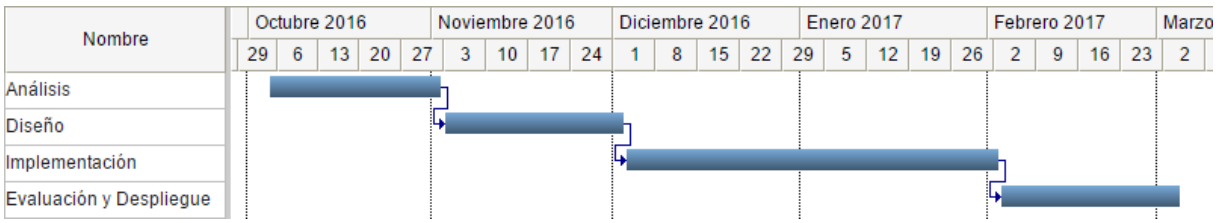

Figura 1.2 Diagrama de GANTT.

CAPÍTULO 2
SISTEMAS DE RECOMENDACIÓN

2. SISTEMAS DE RECOMENDACIÓN

En este capítulo se presenta una introducción a los sistemas de recomendación (SR), abarcando su clasificación y haciendo una revisión de las tendencias actuales en los SR. Finalmente concluiremos este capítulo conociendo en profundidad una de estas tendencias, los sistemas de recomendación a grupos.

2.1. Introducción

Como ya hemos visto en capítulos anteriores, Internet es una de las herramientas más importantes y versátiles creadas por el ser humano, pero es justo esa versatilidad el punto más débil que tiene Internet, ya que la mayoría de los usuarios se sienten impotentes ante la gran cantidad de información que hay en la red. A este fenómeno se le conoce como “sobrecarga de información” [21].

Esta cantidad de información a veces satura al usuario, bien porque no conoce todas las alternativas que se le ofrecen, o bien porque no tiene la experiencia suficiente para filtrar la información irrelevante. Por ello, es frecuente que busque la opinión de terceros acerca de un producto o servicio como respaldo para la toma de decisiones antes de decidir que alternativa es la que más le conviene.

Sería mucho más cómodo para los usuarios que un sistema informático monitorizase su actividad en la red y que descubriera lo que realmente les gusta y les consiguiese aquella información que fuese verdaderamente interesante para el usuario.

Dentro de este contexto, surgen los sistemas de recomendación que se presentan como la solución que cumple tanto las expectativas de las empresas (que buscan dar un mejor servicio a sus clientes, y suplir de algún modo el trato que se les da en un negocio físico, y que no es posible dar a través de una tienda virtual) como las de los usuarios, permitiendo a las primeras recomendar productos a los clientes en función de sus preferencias, y ayudar a los segundos a entender mejor la información que se les ofrece. Es por ello que los sistemas de recomendación tienen un papel cada vez más importante en el panorama actual, siendo su uso en aplicaciones web muy habitual desde hace algunos años [16].

Con la finalidad de reflejar los intereses de los usuarios y realizar recomendaciones, los sistemas de recomendación recopilan la información de los usuarios a través del proceso de retroalimentación. Este proceso es la pieza clave para el buen funcionamiento de un sistema de recomendación, porque sin la información recuperada por este, sería imposible conocer el interés de los usuarios y por esto, el sistema tampoco podría recomendarles contenidos interesantes.

La información necesaria para estos sistemas se pueden obtener de forma explícita, es decir que los usuarios expresan de forma voluntaria y directa que contenidos le gustan, normalmente a través de las valoraciones, o de manera implícita donde los objetos son evaluados sin la intervención directa de los usuarios, o sea, que la evaluación se realiza sin que el usuario lo perciba, a través de las acciones que usuario realiza durante la interacción con el sistema [30].

Actualmente existen varios tipos de SR, entre los que destacan los Sistemas de Recomendación Colaborativos, basados en contenido, basados en conocimiento e híbridos. Más adelante se verán las características de estos sistemas.

Después de haber introducido su objetivo, podemos llegar a definir los sistemas de recomendación como [31]:

«Conjunto de técnicas de recuperación de información que intentan descubrir el interés de los usuarios por determinados objetos, con la finalidad de ofrecerles un conjunto de objetos afines, relacionados a su perfil, en los que podría estar interesado».

Y por último, a modo de conclusión de esta sección, vamos a exponer varias razones por las cuales los proveedores de servicios, o incluso usuarios comunes, pueden querer aprovechar estas tecnologías [28]:

- *Incrementar el número de ítems vendidos*: Alcanzar este objetivo es posible porque los ítems recomendados encajan con las necesidades del usuario.

- *Vender ítems más diversos*: Permite al usuario seleccionar ítems que posiblemente hubieran sido difíciles de encontrar sin su recomendación.

- *Incrementar la fidelidad del usuario*: Cuando el sistema orienta correctamente en las recomendaciones a un usuario, este será fiel al sitio web.

- *Incrementar la satisfacción del usuario*: Mejora la experiencia del usuario con el sitio o con la aplicación.

- *Mejor entendimiento de lo que el usuario busca*: Permite describir las preferencias del usuario, tanto las recogidas explícitamente como las predichas por el sistema.

2.2. Elementos de un SR y esquema de funcionamiento

Aunque existen muchos tipos de sistemas de recomendación distintos, en función de la información y técnicas que utilizan a la hora de realizar las recomendaciones podemos decir que, en general, existen tres elementos comunes que están presentes en todos ellos [19]:

- **Base de datos de Items:** Ítem es el término general usado para denotar qué es lo que un sistema recomienda a un usuario. Un SR normalmente se centra en un tipo específico de ítem (por ejemplo películas, canciones o libros). Cada ítem contará con una serie de características propias que pueden ser utilizadas durante el proceso de recomendación, y cuya complejidad dependerá en muchos casos del ítem en sí. La calidad de los datos almacenados en nuestra base de datos jugará un papel crucial a la hora de realizar recomendaciones con mayor o menor calidad.
- **Perfiles de Usuario**: Representan a personas reales. Recordemos que el objetivo de un sistema de recomendación es el de ofrecer ítems que resulten interesantes a cada usuario en concreto. Un usuario va "dándole forma" a su perfil personal a medida que utiliza el sistema. El perfil refleja los gustos/preferencias del usuario, fundamentales a la hora de discriminar objetos durante la recomendación.
- **Valoraciones**: Se tienen registradas las valoraciones que realizan los usuarios sobre los ítems (cada interacción entre un usuario y el sistema de recomendación). La forma en la que los usuarios valoran los ítems

puede ser implícita (el usuario es monitorizado y se evalúan los ítems en función de su comportamiento) o explicita (el usuario decide que ítems valorar).

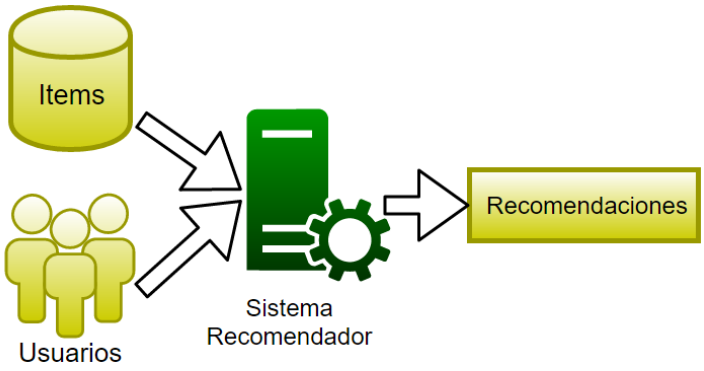

Figura 2.1 Esquema simplificado de un SR.

De forma adicional y dependiendo del sistema de recomendación, el sistema puede tener registrada más información sobre los usuarios y los ítems para llevar a cabo el proceso de recomendación.

Dado un usuario en concreto para el cual se desea realizar una recomendación, llamado formalmente *usuario activo*, y utilizando la información almacenada acerca de usuarios, ítems, y valoraciones, se trata de deducir qué ítems del conjunto serían valorados positivamente por el usuario. Tales ítems con mejor valoración serán los recomendados al usuario activo [10].

2.3. Clasificación

Una vez revisadas las bases de los Sistemas de Recomendación, es el momento de revisar los diferentes tipos existentes de los mismos. Los sistemas de recomendación pueden ser clasificados de acuerdo al tipo de información que utilizan para realizar las recomendaciones.

Tradicionalmente existen varios paradigmas de filtrado de información utilizados para la generación de recomendaciones, estos se clasifican en: Basados en contenido, que tratan de recomendar productos similares a los que le ha gustado

a un usuario determinado en el pasado, Filtrado colaborativo, que identifica a los usuarios cuyos gustos son similares a los de un usuario determinado y recomienda a este usuario los contenidos que les gusten a los demás usuarios, y el enfoque Híbrido, que es una combinación entre el basado en contenido y filtrado colaborativo [1]. Pero primero vamos a distinguir entre las recomendaciones no personalizadas y las recomendaciones personalizadas.

- **No Personalizados**

Se utilizan para solventar el problema de la ausencia de valoraciones cuando accede un nuevo usuario, lo que dificulta descubrir sus gustos y en consecuencia recomendar ítems en base a sus gustos. A este problema se le llama “Arranque en frío” (Cold Start), y se podría solucionar, por ejemplo, obteniendo valoraciones implícitas del usuario en las redes sociales.

Las ventajas de este tipo de SR es la facilidad de implementar para que los ítems más populares o mejores valorados sean mostrados a los usuarios. Como contra punto la recomendación en estos sistemas es la misma para todos los usuarios.

- **Personalizados**

Este tipo de recomendación si tiene en cuenta los gustos y opiniones de los usuarios, registrados de manera explícita en el SR, para realizar una recomendación única para cada usuario. A continuación vamos a ver los diferentes tipos de SR de recomendación personalizada:

2.3.1. Sistemas de recomendación basados en Contenido

Conocidos también con el nombre de sistemas de recomendación no colaborativos, tratan de recomendar items similares a los que les han gustado a un usuario determinado en el pasado. Estos items han sido previamente valorados por el usuario. Para saber que un ítem es parecido a otro se buscan “palabras clave” del ítem que calificó el usuario. Esto puede presentar un problema ya que si siempre se recomiendan ítems similares a los vistos anteriormente se puede llegar a lo que se

conoce como sobre-especialización. Un ejemplo de esto, es que si en un sistema de recomendación de películas un usuario solo ha valorado películas de temática o género acción, va a ser difícil que el sistema le recomiende películas de comedia o infantiles. Una de las soluciones posibles es añadir aleatoriedad a las recomendaciones del sistema. Los sistemas de recomendación basados en contenidos presentan las siguientes ventajas e inconvenientes [17]:

Ventajas:

- Recomendación por contenido y no por opiniones subjetivas de otros usuarios.
- El sistema puede generar explicaciones sobre la recomendación que hizo en base al historial del usuario.
- No hay Dispersión: El modelado de la información está presente en las características del documento y no necesitan proveerlas otros usuarios.

Inconvenientes:

- Necesita un modelo detallado de preferencias del usuario, que es complejo de construir y mantener.
- Sobre especialización: El usuario está limitado a que le recomienden ítems similares a los que recomendó.
- Subjetividad de los Contenidos: Dificultad en dominios con contenido difícil de analizar, (audio, gráficos, imágenes, vídeo).
- Problema del Usuario Nuevo: El usuario tiene que puntuar un número suficiente de ítems para que el sistema pueda realmente entender sus preferencias.
- Representación del Perfil del ítem: Para cada ítem se extraen ciertas características sobre las cuales se evalúa la similitud.
- Efecto Portafolio: Se da en dominios como recomendación de noticias, ya que es posible descartar noticias que pueden ser muy similares a previas, pero que al mismo tiempo presentan hechos nuevos e importantes.

- Problema Estabilidad vs Plasticidad: Es difícil para el sistema aprender a adaptarse a los cambios en el perfil del usuario hasta no haber recolectado un número suficiente de valoraciones actualizadas.

2.3.2. Sistemas de recomendación basados en Filtrado Colaborativo

Conocidos también como sistemas de recomendación colaborativos, tienen como objetivo conocer las preferencias del usuario y hacer recomendaciones sobre la base de datos de los usuarios y la comunidad. El sistema recomienda ítems de otros usuarios con "gustos" similares a los suyos. Por tanto, el sistema de recomendación calcula la similitud entre usuarios y crea los llamados "vecinos cercanos", es decir, usuarios que tienen valoraciones similares en los mismos ítems.

Para calcular la similitud los métodos más comunes son el coeficiente de correlación de Pearson, en el cual se mide la dependencia entre 2 variables, y el método de similitud es el coseno. El coeficiente de correlación de Pearson no es indicado para usuarios con pocas valoraciones ya que tiende a dar un alto nivel de similitud.

Los sistemas de recomendación basados en filtrado colaborativo presentan las siguientes ventajas e inconvenientes [17]:

Ventajas:

- No necesita modelo detallado de preferencias; basta con un vector de valoración de objetos.
- Permite recomendar contenidos difíciles de analizar.
- Permite recomendar ítems basados en las preferencias del usuario.
- Permite realizar recomendaciones válidas pero no esperadas.
- Puede aplicarse a cualquier tipo de ítem o producto: documentos, música, películas, libros, etc.
- Permite introducir novedad respecto a la experiencia previa del usuario.

Inconvenientes:

- Requiere mucho espacio de almacenamiento y tiempo de proceso para determinar usuarios parecidos.
- El coste computacional es elevado.
- Es imprescindible conocer la valoración de algunos objetos para que el proceso pueda funcionar.
- Problema de Cold-Start: Problema del Usuario Nuevo y de Ítem Nuevo.
- Problema de Dispersión: Si el número de usuarios es pequeño en relación al volumen de información en el sistema, se corre el riesgo de que el cubrimiento de las valoraciones se vuelva muy disperso, disminuyendo la colección de ítems recomendables.
- Problema de Escalabilidad: A medida que la cantidad de usuarios y de ítems crece, también crece la cantidad de cómputos de vecinos más cercanos para la determinación de usuarios similares, y como los cálculos se hacen en tiempo real, el sistema puede colapsar.
- Problema de la Oveja Gris: Existen usuarios donde sus perfiles caen entre clases existentes de usuarios, haciendo difícil determinar para ellos una recomendación adecuada.

Como podemos ver el filtrado colaborativo sufre de problemas de escalabilidad cuando la base de datos de usuarios crece. Para solucionar esto hay otra aproximación basada en similitud entre ítems, en lugar de entre usuarios, como hemos visto anteriormente.

En el filtrado colaborativo basado en ítem, la predicción se calcula teniendo en cuenta las valoraciones de los ítems. Esta técnica es similar a la recomendación basada en contenido, pero en lugar de calcular la similitud a partir de las características de los ítems, se calcula en base a los patrones de los usuarios.

A continuación podemos ver una comparativa de cómo funciona el filtrado colaborativo basado en usuario y en ítem:

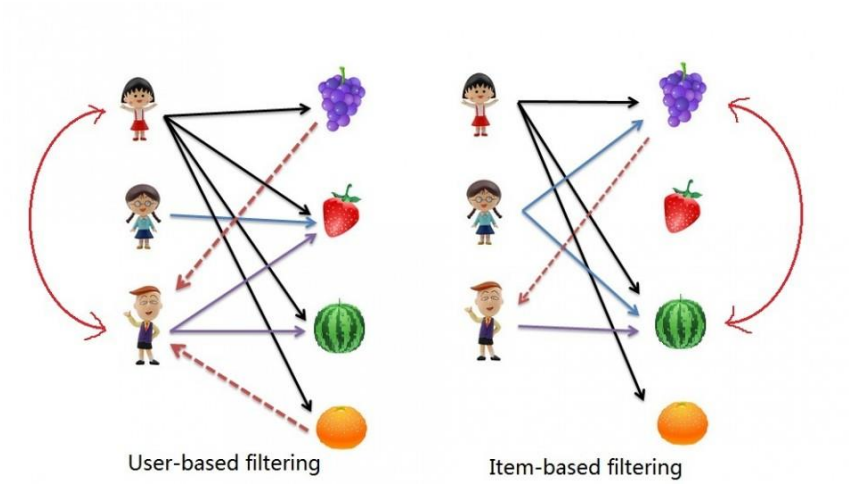

Figura 2.2 Comparativa ente filtrado basado en usuario y filtrado basado en item.

El basado en usuario consiste en recomendar al usuario u los ítems bien puntuados por usuarios v “similares” a u. En cambio, el basado en ítem consiste en recomendar al usuario u los ítems “similares” a los mejor puntuados por u.

2.3.3. Sistemas de recomendación Híbridos

Como hemos visto, dado que los sistemas colaborativos y los basados en contenido tienen sus ventajas e inconvenientes, y un carácter complementario, ha surgido un nuevo grupo de sistemas que trata de aunar las ventajas de ambos para efectuar mejores recomendaciones.

Lo que se busca es sobrellevar los inconvenientes de ambos sistemas para obtener mejores recomendaciones. Para crear un sistema híbrido colaborativo basado en contenido, los perfiles de usuario se mantienen según el análisis de los contenidos de los ítems, y directamente se comparan esos perfiles para determinar la similitud entre usuarios para una recomendación colaborativa [4].

2.3.4. Otros tipos de Sistemas de recomendación

Además de los antes mencionados, se proponen otros tipos de sistemas de recomendación [4], aunque de una u otra forma están relacionados con los tipos de sistemas de recomendación anteriores:

Recomendaciones demográficas

Clasifican a los usuarios de acuerdo a su perfil y hacen las recomendaciones basándose en clases demográficas. Las recomendaciones demográficas son similares a las recomendaciones basadas en el contenido con la excepción de que las similitudes están calculadas a partir de la utilización de información demográfica en lugar de valoraciones de los ítems.

Recomendaciones basadas en el conocimiento

Las sugerencias de los ítems se basan en inferencias sobre las necesidades de los usuarios y sus preferencias. Para ello se utiliza conocimiento en donde se tiene información sobre cómo un ítem específico responde a una necesidad en particular del usuario y, por lo tanto, la razón sobre la relación entre la necesidad y una posible recomendación.

Recomendaciones basadas en la utilidad

Son sistemas de recomendación que crean una función de utilidad para cada ítem la cual interviene directamente en el proceso de recomendación. La ventaja de este método es que permite evaluar elementos no atribuibles al producto o ítem en sí. Aspectos como la fiabilidad de un proveedor o la disponibilidad de un ítem estarían representados en la función de utilidad.

2.4. Tendencias

Además de los tipos de sistemas de recomendación expuestos en el punto anterior, actualmente están surgiendo nuevos métodos que permiten incorporar otro tipo de información al proceso de recomendación de los sistemas anteriores.

En la actualidad la investigación en el campo de los SR se está centrando en la mejora de las recomendaciones mediante el uso de información del usuario. La manera de obtener esta información es de lo más variada, pudiendo provenir de patrones de navegación, análisis de redes sociales, dispositivos móviles, etcétera.

Esto nos puede ayudar a mejorar las recomendación basándose, por ejemplo, en el lugar en el que está el usuario para no mostrarle todas las posibilidades, si no las más cercanas, o sabiendo con quien está para tener en cuenta los gustos de sus acompañantes.

A continuación se verán dos técnicas para mejorar la recomendación basándose en el contexto del usuario y realizando una recomendación a grupos.

2.4.1. Sistemas de recomendación sensibles al contexto

La función de estos SR es recomendar los ítems adecuados a los gustos e intereses del usuario, pero estos intereses pueden variar según el contexto en el que se encuentre. El contexto es cualquier información que pueda ser utilizada para caracterizar la situación de las entidades (persona, lugar u objeto) que se consideran relevantes para la interacción entre un usuario y una aplicación, incluyendo al propio usuario y a la aplicación.

Mientras que una cantidad sustancial de la investigación se ha realizado en el área de sistemas de recomendación, la gran mayoría de los enfoques existentes se centran en recomendar artículos a los usuarios o usuarios a los artículos y no toman en consideración cualquier tipo de contexto adicional.

Es por esta razón que los sistemas de recomendación que son sensibles al contexto se ocupan de modelar y predecir los gustos del usuario y preferencias mediante la incorporación de información contextual [29]. Esta información contextual puede ser de diferentes tipos, como el tiempo, la ubicación, compañero, el propósito de una compra, etc.

El proceso de recomendación que es sensible al contexto puede tomar una de las tres formas siguientes:

Pre-filtrado: En esta técnica se utiliza el contexto del usuario para reducir el conjunto de ítems para calcular la recomendación, esto permite usar técnicas de recomendación que no son eficientes con grandes conjuntos de ítems.

Un ejemplo de un pre-filtrado de datos contextual de un sistema de recomendación sería: dada una persona que desea ver una película en un domingo, entonces solo los datos de tipo domingo o fin de semana son utilizados para recomendar películas.

Post-filtrado: la información contextual es inicialmente ignorada y una vez calculadas las recomendaciones del usuario según sus preferencias se aplica un filtrado de estas recomendaciones según su contexto.

Modelado contextual: En este paradigma de recomendación, la información contextual se usa directamente en la técnica de modelado como parte de la estimación de valoración, es decir, en lugar de almacenar los datos de la forma "tradicional" (<usuario, ítem, puntuación>), esta técnica, construye un modelo en el que también se tiene en cuenta el contexto de la valoración del ítem quedando de la siguiente forma <usuario, ítem, contexto, puntuación>. Añadiendo así el contexto al SR en lugar de como simples funciones de filtrado o posterior al SR.

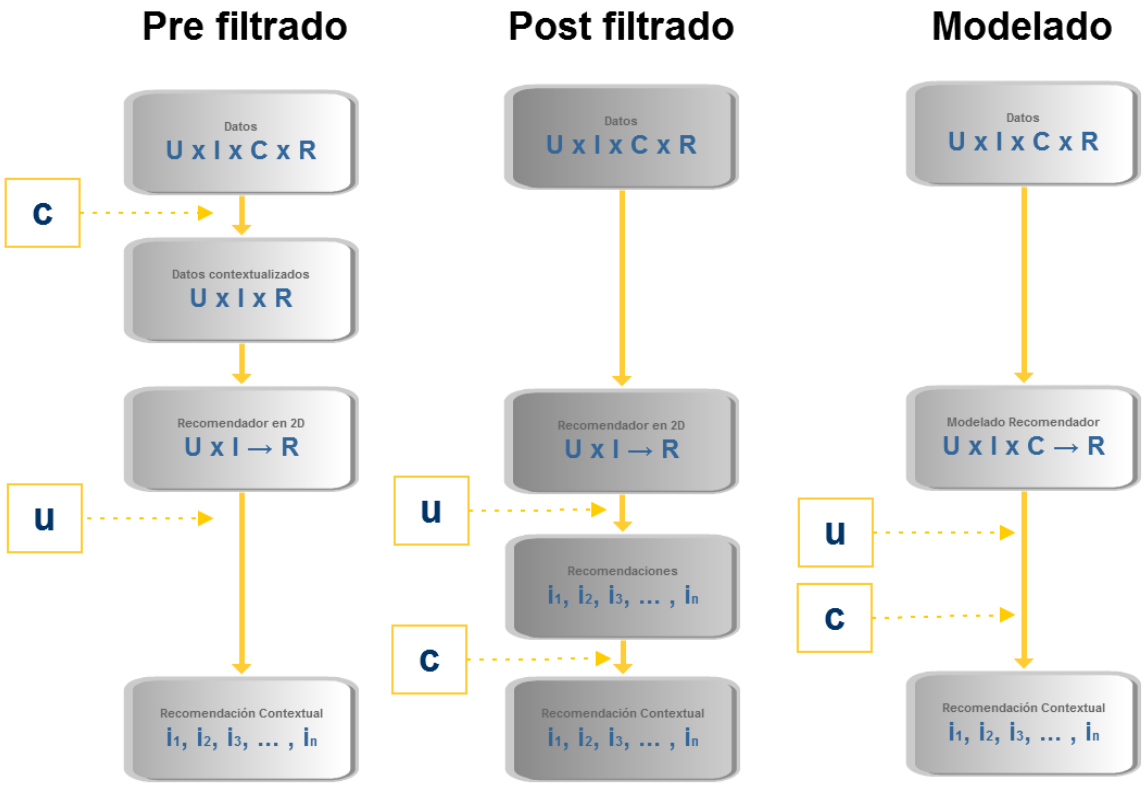

Figura 2.3 Paradigmas para la incorporación de contexto en sistemas de recomendación.

2.4.2. Sistemas de recomendación a grupo

Los sistemas de recomendación se han centrado tradicionalmente en hacer recomendaciones de elementos a usuarios individuales, pero en los últimos años especialmente ha aparecido una fuerte actividad de investigación en sistemas automatizados de recomendación personal, especialmente favorecida por el aumento asombroso de la actividad comercial en Internet.

Ha sido recientemente cuando se ha empezado a trabajar en el desarrollo de técnicas que permitan proponer recomendaciones a grupos de usuarios simultáneamente. Este tipo de sistemas plantea problemas específicos de las técnicas de recomendación, como son la necesidad de adquirir las preferencias del grupo, ayudar al grupo en el proceso de toma de decisiones de cuál es la mejor opción y explicar al grupo las razones de una recomendación.

En el siguiente punto abordaremos en profundidad este tipo de sistema de recomendación.

2.5. Sistemas de Recomendación a grupo

Debido a que este proyecto utiliza un sistema de recomendación a grupos, a continuación vamos a hacer una revisión un poco más en detalle de este tipo de SR.

Además, si nos centramos en el tipo de elementos que utilizaremos y visualizaremos en este proyecto, como son las películas, debemos considerar el hecho de que los usuarios frecuentemente ven la televisión o van al cine en grupo. Aunque el caso más común es ver películas con otros miembros del hogar (generalmente la familia), podemos encontrar una gran cantidad de escenarios en los que este hecho se convierte en una actividad social, ya que, no es raro en absoluto, especialmente entre los jóvenes, reunirse para ver una película.

Un factor importante a tener en cuenta cuando se trabaja con grupos es su caracterización respecto a la similitud de sus miembros. De esta manera, si queremos generar recomendaciones de películas para un grupo, lo primero que hay

que hacer es clasificarlos de acuerdo a los intereses y características de sus miembros, en grupos homogéneos o heterogéneos. En términos generales, una familia será un grupo heterogéneo, ya que los intereses particulares del padre en general serán diferentes a los de la madre, y los de ambos no similares a los de los niños, y éstos disímiles entre sí dependiendo de las diferencias de edad [34].

Por otro lado, podemos imaginar que los intereses de un grupo de amigos serán similares. Es decir, los intereses no serán idénticos, pero similares. Los perfiles de usuario de cada miembro serán similares. Es mucho más fácil recomendar para este grupo, ya que se puede sustituir al grupo por un usuario virtual y seguir teniendo una buena recomendación.

La mayoría de los sistemas de recomendación a grupos utilizan métodos similares a los métodos de adquisición de información que se aplican en los sistemas de recomendación para individuos. Básicamente se pueden dividir en métodos de adquisición de preferencias sin especificación explícita, en los que no se requiere que sus usuarios especifiquen explícitamente sus preferencias, ya que el sistema puede funcionar con información adquirida implícitamente sobre los usuarios, y por otro lado métodos de especificación de preferencias explícita, en los que sí se requiere una especificación explícita de las preferencias de los usuarios [25].

Una vez que el sistema ha adquirido el conocimiento necesario sobre los usuarios, puede adaptar la especificación de preferencias a los requerimientos de la recomendación para el grupo. Ejemplos de esta adaptación serían los sistemas que se centran en las preferencias negativas o los sistemas que comparten información sobre las preferencias especificadas.

Los primeros sólo tienen sentido si el procedimiento que se está utilizando para generar las recomendaciones es diseñado principalmente para evitar que el producto seleccionado sea especialmente contrario a los gustos de cualquier miembro del grupo.

En cuanto a los otros sistemas, se puede dar el caso que en un SRG interese que cada miembro pueda conocer las preferencias de los otros miembros del grupo,

por ejemplo, para aprender de otros miembros del grupo, para ahorrar tiempo a la hora de especificar sus propias preferencias, para poder anticipar actitudes y comportamientos de los otros componentes del grupo, o para asimilar los motivos de los demás componentes y así, llegar a un consenso más fácilmente.

Fácilmente podemos ver cómo de este enfoque surge un problema de manipulación si, por ejemplo, una persona que no quiere que salga un producto en particular lo califica como “muy deficiente” y asegurarse así que nunca salga recomendado.

Por otro lado, dependiendo del tamaño y la homogeneidad del grupo, puede ser difícil encontrar una recomendación que sea adecuada para cada miembro del grupo de manera individual. En la mayoría de los casos, el SR debe escoger aquella opción que satisfaga al mayor número de usuarios del grupo, de acuerdo con sus preferencias individuales. Por tanto, se necesita algún tipo de método de agregación para combinar la información sobre las preferencias individuales de los usuarios de forma que el sistema pueda obtener la recomendación idónea para el grupo en sí [22].

Existen tres aproximaciones básicas para resolver este problema:

- Mezclar las recomendaciones que se harían por separado a cada uno de los miembros del grupo. Es un método simple de agregación donde se unen las soluciones del SR para cada componente del grupo en una única lista.
- Agregar las valoraciones para cada usuario. Para cada producto candidato y para cada miembro del grupo, el sistema predice cómo ese componente evaluaría dicho producto, y devuelve una colección de candidatos que tengan las valoraciones previstas más altas.
- Construir un modelo de las preferencias del grupo. En este enfoque el sistema usa la información sobre las preferencias individuales de los componentes del grupo para construir un modelo de preferencias para el grupo en sí.

Elegido un enfoque general, hay que decidir qué procedimiento computacional se va a usar para la agregación. Hay varios objetivos que deben tenerse en cuenta a la hora de elegir uno, como la satisfacción total, la comprensibilidad, el grado de igualdad, esto variará dependiendo de la situación dada:

- **Maximizar la satisfacción media**. En esta función de agregación se calcula una media de la satisfacción predicha para cada miembro del grupo y se usa como base de la selección de productos candidatos. La función a utilizar sería:

$$R_i = average(r_{ij}) = 1/n * \sum_{k=1}^{n} r_{ij}$$

Ecuación 2.1.

Donde r_{ij} es la valoración de cada usuario u_i, para un producto p_j. R_i es el rating final que obtiene el producto p_j para el grupo.

- **Minimizar la miseria.** En esta función de agregación se elige la mínima de las valoraciones individuales. Aunque la satisfacción media sea alta, si una solución deja a un componente del grupo especialmente disconforme, se puede considerar como una situación indeseada. Es posible tener en cuenta este factor, por ejemplo, considerando que el rating más bajo debe ser siempre superior a un cierto umbral.

	A	B	C	D	E	F	G	H	I	J
Peter	10	4	3	6	10	9	6	8	10	8
Jane	1	9	8	9	7	9	6	9	3	8
Mary	10	5	2	7	9	8	5	6	7	6
Group Rating	1	4	2	6	7	8	5	6	3	6

Figura 2.4 Ejemplo de Minimización de la miseria.

- **Minimizar la penalización**. En esta función se considera una solución que satisfaga a cada persona de igual modo, asegurando un grado de justicia. Esta situación es en general preferible a una solución que

satisfaga a un cierto número de componentes a expensas de otros. La función a utilizar sería:

$$R_i = average(r_{ij}) - \omega * desviacionEstandar(r_{ij})$$

Ecuación 2.2.

Donde r_{ij} es la valoración estimada para cada usuario u_i para un producto p_j, ω es el peso que refleja la importancia del grado de justicia y R_i es el rating final que obtiene el producto p_j para el grupo.

CAPÍTULO 3
VISUALIZACIÓN

3. VISUALIZACIÓN

3.1. Introducción a la visualizacion en SR

La visualización de datos es uno de los mecanismos de los cuales se dispone para presentar gran cantidad de información de forma razonable a los usuarios finales, sin que estos se vean superados por una avalancha de datos. Una visualización de datos es un primer paso para el análisis y proyección de los datos disponibles, utilizando los recursos del sistema visual humano como procesador para detectar patrones, tendencias y/o anomalías. En este sentido, una visualización permite, de forma eficiente, medir y comparar datos, entre otras operaciones. Incluso un simple resumen de un conjunto de datos puede ser mejor transmitido y comprendido mediante una visualización que mediante el uso de texto y tablas numéricas [36].

Por lo tanto, acompañar (o incluso sustituir) los datos originales por una representación gráfica de estos puede ser muy efectivo para explicar el porqué de los mismos.

En concreto para el caso que nos ocupa, uno de los problemas más importantes que sufren los sistemas de recomendación es la dificultad que suele entrañar para el usuario comprender unas informaciones que generalmente adoptan forma de tablas, cuadros o listas, y su consiguiente falta de explicaciones. Lo que es lo mismo, los usuarios carecen de información clara, por lo que los resultados pueden no llegar a ser convincentes. Diseñar y emplear técnicas de visualización para los sistemas de recomendación puede abordar eficazmente este problema [9].

Gran parte de la investigación en el área de Sistemas de Recomendación se centra en el estudio de técnicas de recomendación, estas técnicas son dependientes del dominio de aplicación, de la efectividad de la técnica, así como de las métricas para poder evaluarlas. Sin embargo, el estudio de las técnicas de visualización en sistemas de recomendación ha tomado relevancia ya que han demostrado que mejoran la experiencia del usuario [38].

A saber:

- Ayudan a los usuarios a justificar y comprender el razonamiento detrás de una recomendación, para que puedan decidir el grado de confianza en ella.
- Aumentar la sensación de participación de los usuarios.
- Instruye a los usuarios en el proceso de recomendación.
- Aumenta la aceptación por parte de los usuarios de las recomendaciones.

Además se cree que la interacción puede ayudar aún más en el proceso de recomendación, ya que:

- Permite a los usuarios actualizar dinámicamente su perfil de preferencias durante una sesión de recomendación.
- Permite a los usuarios proporcionar calificaciones directamente sobre las entidades utilizadas para producir recomendaciones.
- Apoya la exploración de escenarios de "qué pasaría" basados en diferentes configuraciones de perfiles.

Se han utilizado muchas técnicas de visualización para proporcionar una comprensión más instintiva del sistema y revelar las relaciones que existen entre los datos. Los sistemas que muestran resultados de manera gráfica son conocidos por proporcionar más confianza, y hacer más fieles en su uso a sus propietarios y/o a sus usuarios. A continuación se presentan diversas técnicas de visualización utilizadas a lo largo del tiempo:

3.2. Técnicas de visualizacion

Incluso los gráficos más simples, como los de líneas, de barra, o gráficas de dispersión pueden mejorar la comprensión del usuario [11].

Es por ello que una de las primeras apariciones de un tipo de visualización para sistemas de recomendación, y utilizada por Foxtrot [20] (un sistema de recomendación de trabajos de investigación en línea a estudiantes de la Universidad de Southampton para el año académico), estaba basada en un enfoque de visualización en gráfico de líneas que mostraba la evolución del perfil de un usuario (en cuanto a intereses) a lo largo del tiempo. Esto ayudaba a los usuarios a comprender el porqué de los artículos relacionados que se les recomendaban.

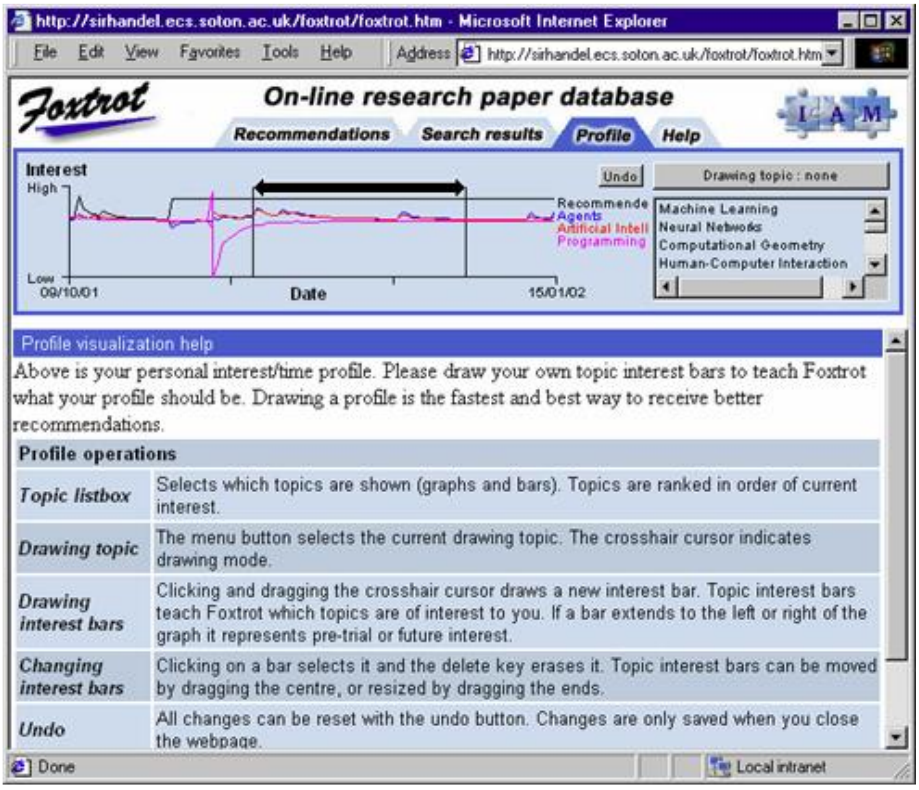

Figura 3.1 Sistema Foxtrot.

Otro de los tipos de visualización simples existentes es TasteWeights [3], un sistema híbrido de recomendación de música con una interfaz interactiva que genera predicciones de artículos de varios recursos web sociales y semánticos, como Wikipedia, Facebook, Y Twitter. El sistema emplea técnicas híbridas de la literatura tradicional de sistemas de recomendación, además de la interfaz, que sirve para explicar el proceso de recomendación y obtener preferencias del usuario final, lo que permite a los usuarios entender y controlar aspectos del proceso de recomendación que, de otro modo, pasarían desapercibidos.

Este sistema incita a los usuarios a ajustar sus gustos a través de controles deslizantes interactivos y otros componentes de la interfaz de usuario. Mientras un usuario arrastra un deslizador, los pesos de los elementos conectados cambian en tiempo real. Por ejemplo, en la figura 3.2, a medida que el usuario arrastra el control deslizante de "Pink Floyd" a la derecha, el valor de "Grupos de Música de Rock Inglés" aumenta simultáneamente y también los valores de "Beatles", "Rolling Stones", "Radiohead" y "Oasis".

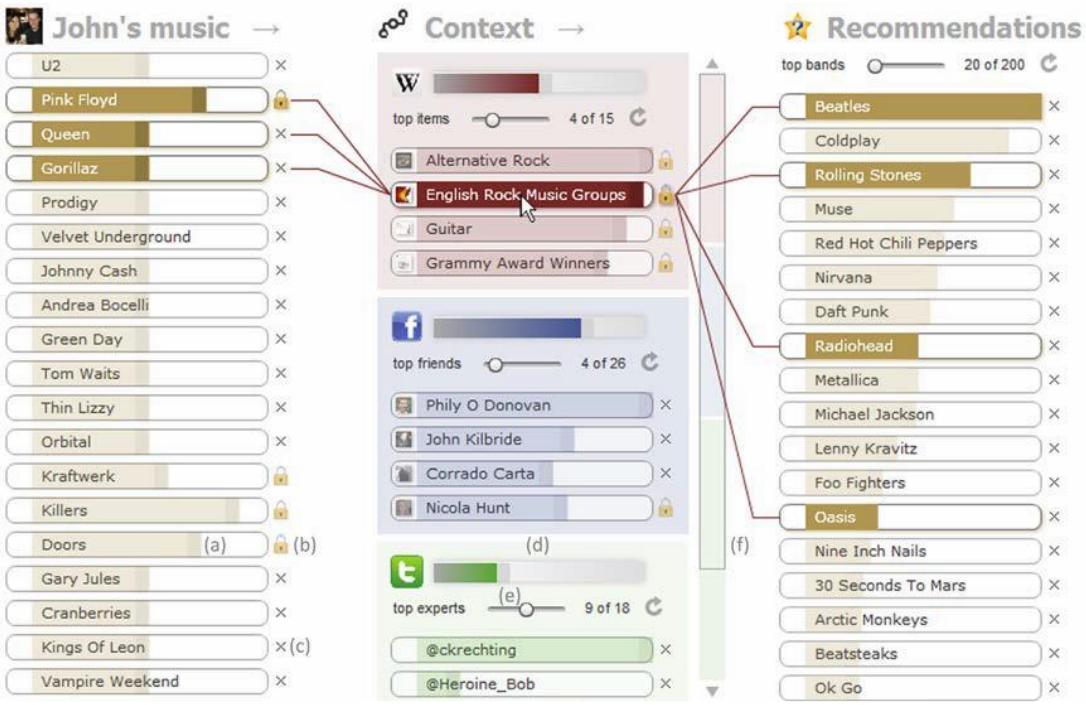

Figura 3.2 Sistema TasteWeights.

Además de la visualización anterior para representar relaciones de comparación entre pares, la visualización mediante grafos es adecuada cuando existen relaciones heredadas. Un ejemplo típico de este tipo de relación es un árbol de productos como se muestra en la figura 3.3. Una gráfica G se puede representar como = (N, E), donde N representa un conjunto de nodos y E representa un conjunto de aristas. Las aristas conectan dos nodos en N, y se dirigen de un nodo a otro, si y sólo si, hay una relación heredada entre dos nodos. Por lo tanto, los gráficos de jerarquía se adaptan bien a los RS porque los items heredan las preferencias de los usuarios.

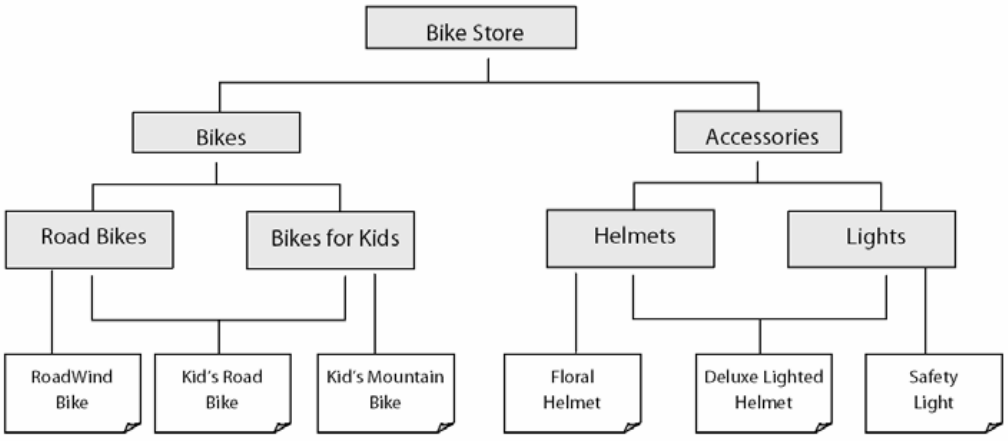

Figura 3.3 Ejemplo de árbol de productos.

Hay muchos diseños diferentes para representar un grafo, pero el diseño Node-Link [18] es el más simple.

Este grafo está formado por un conjunto de objetos llamados vértices o nodos, cuyas posiciones son calculadas, y están unidos por enlaces llamados aristas o arcos, que permiten representar relaciones o enlaces binarios entre elementos de un conjunto, dibujados como una curva. Los diseños creados por algoritmos de árbol y algoritmos de resorte pertenecen a esta categoría de grafo [2].

El árbol clásico [27] se ha convertido en uno de los métodos básicos para describir los datos con relaciones heredadas. A veces es el único enfoque de estudio debido a su simplicidad y popularidad. Los árboles clásicos son sencillos y proporcionan representaciones 2D claras, pero se han propuesto muchas variaciones en el diseño clásico, como el radial para mejorar la eficiencia espacial y generar gráficos compactos. El uso de springs embedders [8], es bien conocido como grafo dirigido y también es estudiado debido a su simplicidad.

Los grafos que utilizan los spring embedders, que se puede traducir como ‘insertadores de muelles', tienen como objetivo principal el de realizar grafos atractivos siguiendo una serie de principios estéticos, como por ejemplo, utilizar todo el espacio asignado para distribuir la información, forzar la posición de los nodos o reducir el número de enlaces cruzados, entre otros. Los spring embedders comienzan asignando coordenadas a los nodos de modo que el grafo final sea estéticamente agradable al ojo humano. La asignación se realiza asemejando el funcionamiento de un muelle al que se asocian dos tipos de fuerzas: las fuerzas de atracción aplicadas a los nodos conectados y las fuerzas repulsivas aplicadas a los nodos desconectados, de acuerdo con la distancia y las propiedades del espacio de conexión.

El gráfico cambia iterativamente hasta que se vuelve estable. Un ejemplo simple de un diseño con spring embedders es un gráfico de red social que mueve dos nodos más cerca, o más lejos, dependiendo de la proximidad de su relación. El gráfico final se genera después de varias iteraciones para ajustar las posiciones relativas de cada nodo.

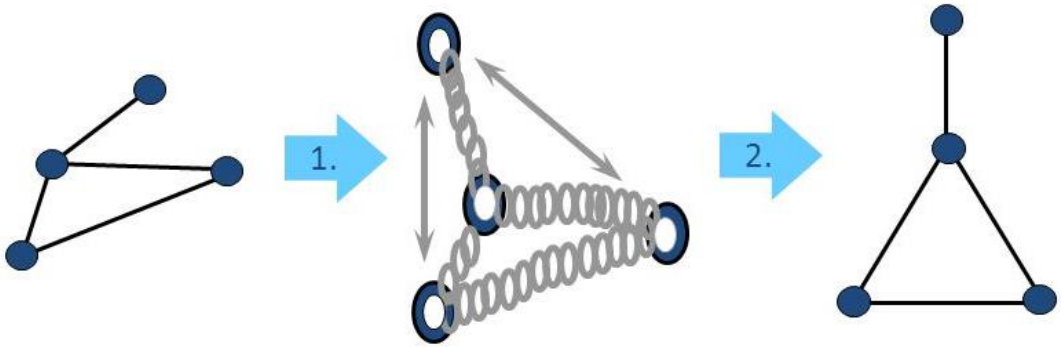

Figura 3.4 Ejemplo de transformación mediante spring embbeders.

La diferencia más importante entre los diseños de árboles y muelles es que las aristas en los diseños de árboles no forman un círculo, lo que los hace inapropiados para sistemas de recomendación basados en vecindario. Las recomendaciones para un usuario activo en sistemas de recomendación basados en vecindario se determinan agregando las opiniones de vecinos similares. En este caso, si se recomienda un elemento porque es preferido por dos vecinos, las aristas entre los nodos que representan al usuario activo, a los dos vecinos y a la recomendación forman un círculo. El diseño con spring embedders posiciona los nodos según una función de coste y, por lo tanto, genera un gráfico desorganizado para sistemas de recomendación. Por ejemplo, un gráfico con spring embedders que representa a miembros, vecinos y elementos como nodos, tiende a mezclar y dispersar todos los nodos y a dificultar que un miembro específico encuentre los nodos de recomendación directamente relacionados a primera vista. Por lo tanto, el diseño con spring embedders puede explicar las relaciones entre los nodos, pero carece de organización en el sistema de recomendación.

También se pueden utilizar redes Bayesianas [15] para mostrar relaciones predictivas como se muestra en la figura 3.5. Los datos se representan como nodos en la red y todos los nodos tienen uno o más enlaces de dependencia a los otros nodos que describen el grado de correlación entre ellos. Cuando el grado es menor que un umbral definido por el usuario, el enlace no se muestra en la red.

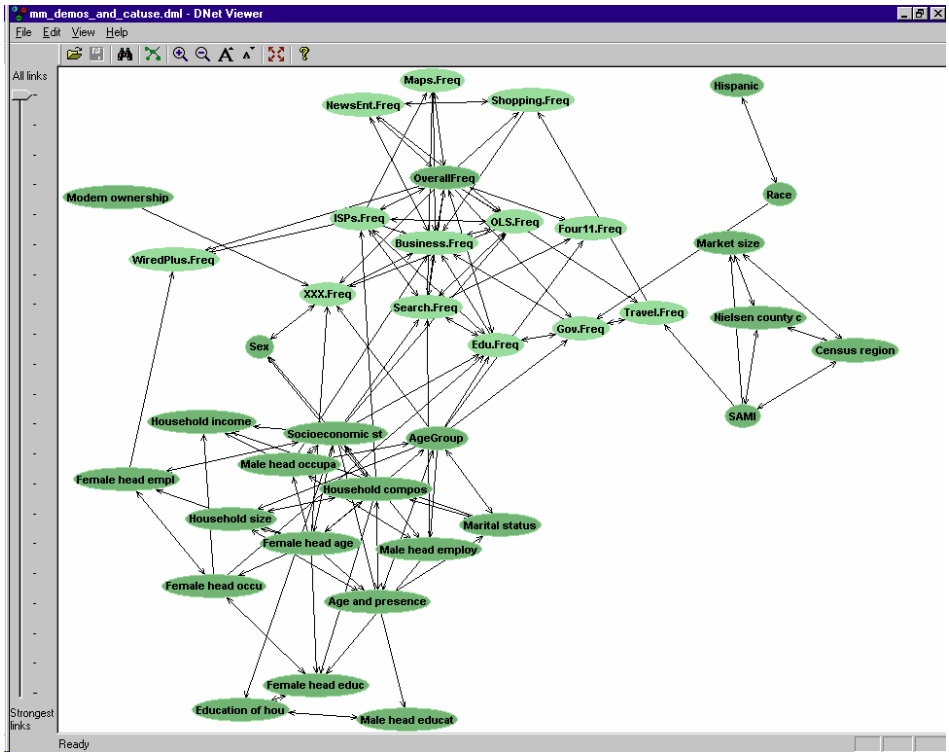

Figura 3.5 Visualización mediante red de Bayes.

Un sistema de recomendación basado en filtrado colaborativo, PeerChooser [23], utilizó una red para mostrar las relaciones entre las recomendaciones, el usuario activo, los vecinos y las recomendaciones. Variaciones de la red (como árboles) también se utilizan para explicar el sistema de recomendación.

PeerChooser es un sistema de recomendación colaborativo con una interfaz de explicación gráfica interactiva. Los usuarios reciben una explicación visual del proceso de filtrado y tienen la oportunidad de manipular su vecindario en diferentes niveles de granularidad para reescribir aspectos de sus requerimientos actuales. De esta manera, se supera el problema de la información de perfil redundante en los sistemas de filtrado colaborativo, además de proporcionar una interfaz de explicación.

La pantalla principal de PeerChooser incluye el gráfico que está centrado en el usuario activo, como se puede ver en la figura 3.6. Los vecinos del usuario activo son representados por los nodos conectados de tal manera que la longitud de la conexión entre el usuario activo y un vecino se basa en la similitud de sus clasificaciones (usando el Coeficiente de correlación de Pearson estándar).

Utiliza la tecnología OpenGL en una plataforma Java, haciéndola visualmente atractiva para el usuario final. Un usuario activo puede interactuar con el gráfico moviendo o eliminando iconos. Los nodos de género se colocan alrededor del gráfico y se puede hacer clic en ellos y moverlos libremente. Cuando esto ocurre, cada nodo vecino conectado es movido por una distancia relativa en la misma dirección.

Esto permite al usuario contrastar su opinión actual sobre un género en particular. Por ejemplo, si el icono de la comedia se mueve hacia el avatar del usuario activo (hacia el centro), entonces todos los usuarios a los que les gusta la comedia se acercarán y se volverán más similares al usuario activo de esta sesión. También permite mover los iconos de los vecinos de forma individual. Al pasar el ratón por encima de los elementos, los nodos se resaltan y las clasificaciones asociadas se muestran en el panel derecho.

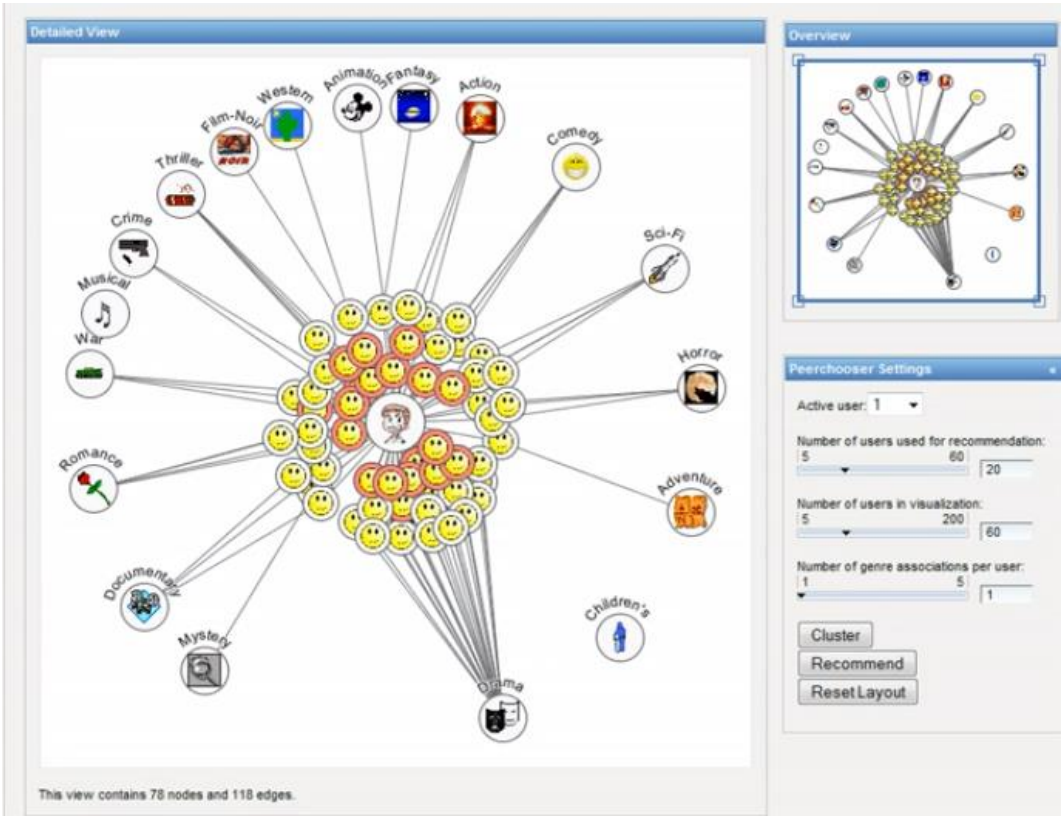

Figura 3.6 Sistema PeerChooser.

Los diagramas de Chord y Sankey, mostrados en la figura 3.7, también se han utilizado para representar relaciones cuantitativas entre nodos.

El diagrama de Chord [35] es un grafo dirigido ponderado que presenta un inconveniente: cuando se requieren muchos tipos de nodos, como usuarios, vecinos y recomendaciones pierden la legibilidad.

Los diagramas de Sankey [40] muestran claramente la información de jerarquía y son muy legibles en comparación con otros tipos de diseños de nodo-enlace, pero la técnica es impredecible y muy probable que produzca barras demasiado pequeñas.

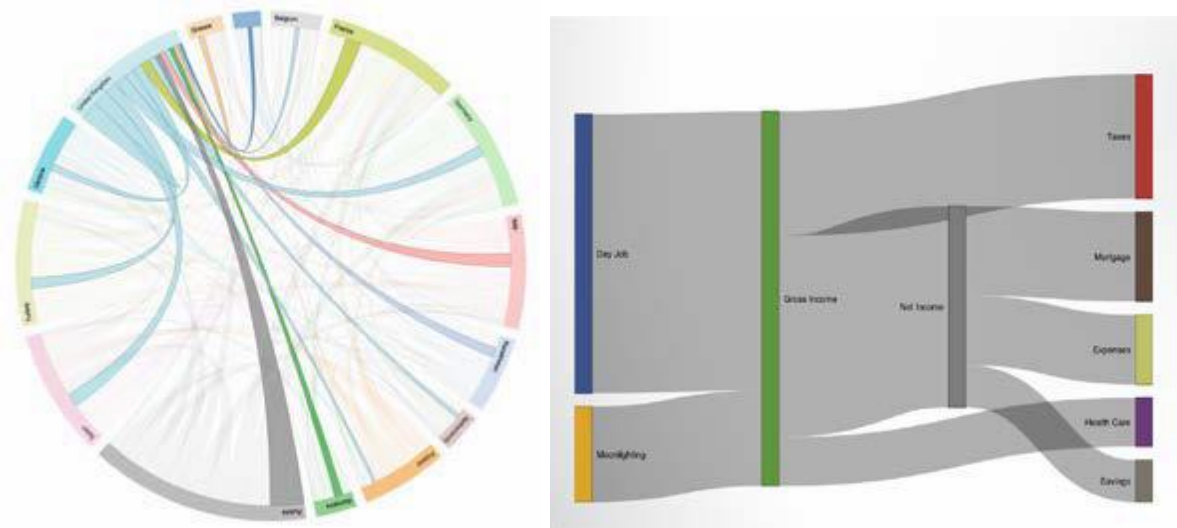

Figura 3.7 Diagramas de Chord y Sankey.

La visualización radial es muy útil para representar la distribución de atributos en datos jerárquicos. Sin embargo, también sufre de sus inconvenientes en términos de transición de vista, preservación del contexto, porciones finas, flexibilidad y soporte de datos de gran tamaño.

Para abordar estos problemas, se propuso FanLens [14], una variación del diseño en árbol junto con muchos otros diseños básicos, como gráficos de barras, líneas y dispersión, y que permite encontrar patrones de interacción y cambiar adaptativamente el diseño de los resultado mostrados a los usuarios, mejorando los enfoques existentes con nuevas características como la disposición incremental y la selección basada en la distorsión de ojo de pez. Este conjunto de herramientas

visuales, mostrado en la figura 3.8, también incluye especificación de jerarquía dinámica, mapeo dinámico de propiedades visuales, animación suave, etc.

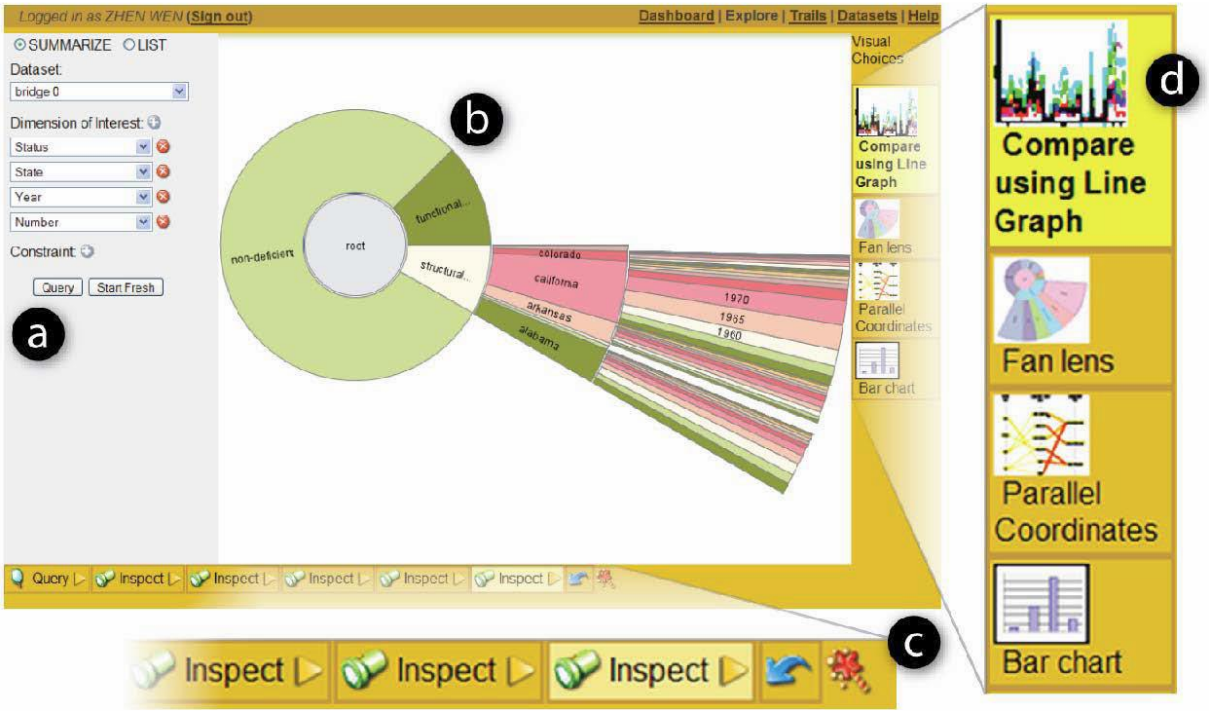

Figura 3.8 FanLens.

La transformación 2D como SOM [6] se utiliza para mapear y visualizar todos los servicios de acuerdo a sus relaciones y las relaciones entre los ítems, como se aprecia en la figura 3.9.

Los Self Organizing Maps (Mapas Auto-Organizados), o SOM, fueron presentados en 1982 por Teuvo Kohonen, profesor de la Academia de Finlandia, y proporcionan una forma de representar datos multidimensionales (vectores) en espacios de dimensión inferior, normalmente, en 2D. Este proceso de reducir la dimensionalidad de vectores es una técnica de compresión de datos conocida como Cuantización Vectorial. Además, la técnica de Kohonen crea una red que almacena información de forma que las relaciones topológicas del conjunto de entrenamiento se mantienen.

Uno de los aspectos más interesantes de los SOM es que aprenden a clasificar sin supervisión, lo que implica que no necesitamos un objetivo que aproximar, sino que genera la distribución a partir de la similitud entre los vectores.

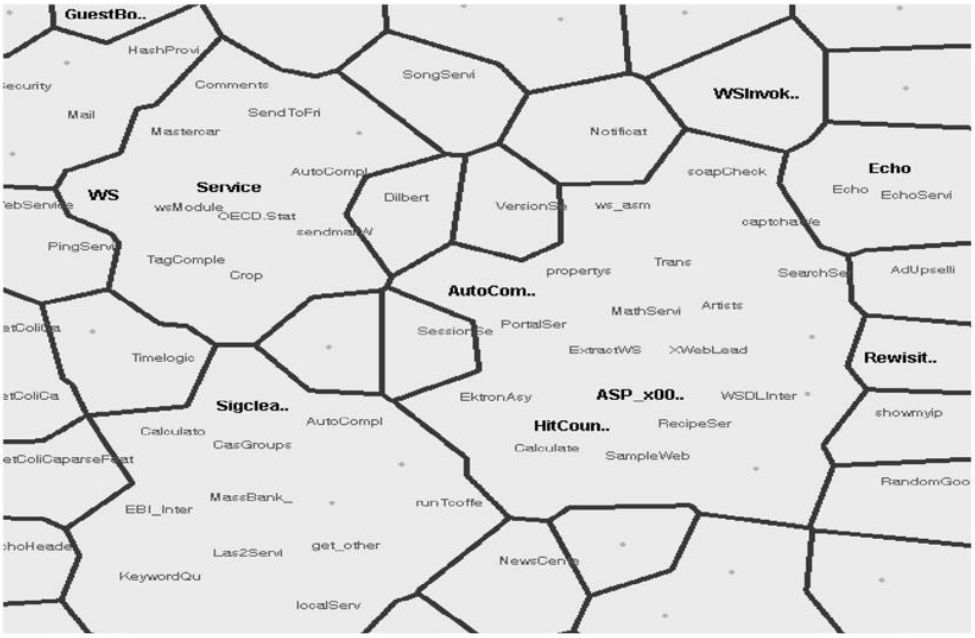

Figura 3.9 Visualización SOM.

Otras aplicaciones utilizan algoritmos de escalado multidimensional (MDS) para visualizar colecciones de datos relativamente grandes [28]. Existen una gran variedad de algoritmos MDS, como la escala clásica (por ejemplo, para visualizar listas de reproducción de música), MDM no métrica (por ejemplo, para visualizar una colección de imágenes), y SMACOF (por ejemplo, utilizado para visualizar catálogos de productos). Sin embargo, todos ellos se basan en el mapeo de una matriz de disimilitud (simétrica) en un espacio Euclídeo de baja dimensionalidad, de modo que las distancias entre pares de puntos representan las disimilitudes entre ellos.

La principal ventaja de MDS es que las distancias en el mapa realmente corresponden a la similitud entre los elementos. En segundo lugar, cuando se utiliza una medida de disimilitud flexible, MDS puede manejar valores faltantes y tipos de atributos mixtos. Las desventajas de MDS en comparación con SOM y Árboles son que puede haber mucho espacio vacío a lo largo del mapa y en cambio elementos muy similares pueden (parcialmente) superponerse entre sí en el mapa. Los espacios vacíos son una desventaja, ya que es necesario hacer más zoom en partes específicas del mapa para poder utilizar el mapa de una manera satisfactoria.

Hay que tener en cuenta que en la mayoría de las situaciones es imposible hacer un mapa de todas las disimilitudes perfectamente en dos dimensiones. Por lo

tanto, la solución es una permisividad en la que unas disimilitudes son mapeadas mejor que otras.

Por último, los gráficos de jerarquía llevan a la organización de componentes un paso más allá cuando cada nodo y cada arista tienen un indicador para especificar un tipo. Por ejemplo, en la figura 3.10, un gráfico de una película mostraría a los directores, productores y actores como tres tipos de roles. La película obtiene su guión del productor, su visualización del director, y su actuación de los actores. Se puede mostrar un gráfico de jerarquía usando diferentes niveles de nodos y tipos de líneas para representar tipos de nodos y de artistas. Usuarios, vecinos cercanos y recomendaciones se presentan como nodos y se asignan a diferentes niveles y se representan con diferentes colores. Desafortunadamente, los nodos en el mismo nivel están desordenados y carecen de legibilidad. Además, el uso de color y nivel a la vez para representar el tipo de nodo es redundante.

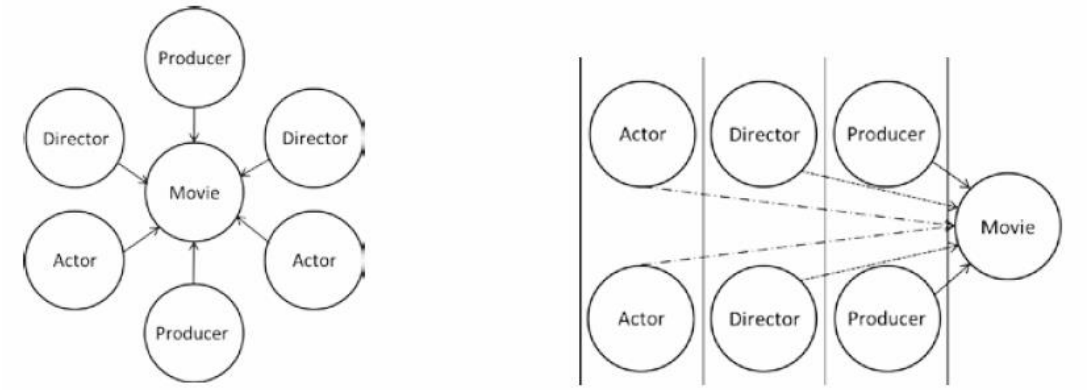

Figura 3.10 Relación de jerarquía en una película y su Gráfico de jerarquía equivalente.

CAPÍTULO 4
INGENIERÍA DEL SOFTWARE

4. INGENIERÍA DE SOFTWARE

Una vez definido, en el Capítulo 1, el propósito y los objetivos del proyecto, llega el momento de pasar a detallar el desarrollo del proyecto que se va a realizar.

En este capítulo se detalla el proceso de desarrollo de dicho software y las actividades de Ingeniería del Software que deben seguirse. No existe una definición única y estandarizada para la Ingeniería del Software pero, las dos que se presentan a continuación pueden resultar perfectamente válidas para este cometido [33]:

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

Las fases que establecen una buena aplicación de la Ingeniería del Software, y que se desglosan en este capítulo, son:

Especificación del catálogo de requisitos: Se debe identificar el tema principal que motiva el inicio del estudio y creación del nuevo software. A su vez, también se deben identificar los recursos de los que se dispone, es decir, conocer los recursos humanos y materiales que participan en el desarrollo de las actividades.

Análisis del sistema: Durante esta fase se plantean las necesidades que queremos satisfacer y se intenta explicar lo que debería hacer el software o producto final para satisfacer dichas necesidades.

Diseño del sistema: Es el proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o sistema con suficientes detalles como para permitir su reutilización física.

Implementación del sistema: Es la realización de una especificación técnica con un software u otro sistema de cómputo. En esta fase se generará el código correspondiente en el lenguaje elegido.

Pruebas y Validación: Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema.

4.1. Análisis del Sistema

En este punto se presenta la fase de análisis, parte inicial de todo proyecto software, donde se definen los requisitos pensados para el sistema. En el siguiente capítulo se tendrá en cuenta este catálogo de requisitos como base para el diseño de todos los aspectos de la aplicación. Es por ello por lo que la fase de análisis es de suma importancia para el devenir de todo producto software. Es en este punto donde se deben asentar las bases, a modo de cimientos, del proyecto, y, a partir de las cuales se construirá todo lo demás.

A continuación se desglosan la funcionalidad y las características del proyecto a modo de catálogo de requisitos, teniendo en cuenta tanto requisitos funcionales como no funcionales.

4.1.1. Catálogo de Requisitos

En nuestro caso, dado que no nos encontramos ante un proyecto comercial, sino ante un proyecto académico, el propósito es conocido desde el mismo momento de la concepción del mismo.

"Implementación en Java de componentes que permitan visualizar de distintas formas las recomendaciones de sistemas de recomendación en grupo."

El catálogo de requisitos es la especificación del comportamiento que se espera de cualquier proyecto software, el propósito último del proyecto, las propiedades que debe satisfacer y las restricciones a las que está sometido. Ésta es, sin duda, una etapa de vital importancia dentro del desarrollo de un proyecto software ya que, sin conocer el propósito del proyecto y todas las limitaciones de diversa índole a las que debe hacer frente, difícilmente se podrá realizar una aplicación software que cumpla dicho propósito.

Estudiando otras aplicaciones similares, se han predefinido una serie de requisitos que se consideran indispensables para el proyecto. A continuación, se muestra una enumeración y breve descripción de dichos requisitos establecidos para el diseño y desarrollo de la aplicación.

4.1.2. Requisitos Funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer. Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares.

Después del estudio de los posibles usos que los diferentes usuarios le van a dar al sistema, se han definido las siguientes funcionalidades que se consideran necesarias para que la utilización del sistema sea la apropiada.

RF 1. Acceso a la aplicación:

- Cualquier usuario que se instale la aplicación en su equipo tendrá la posibilidad de ejecutarla y acceder a ella.

RF 2. Establecer parámetros de configuración:

- El usuario podrá configurar distintos parámetros de la aplicación; podrá indicar las URL de los servicios de obtención de recomendaciones a grupo, así como indicar el número de recomendaciones que se mostrarán en cada uno de los modos de visualización.

RF 3. Generación de grupos:

- Funcionalidad para que el usuario pueda obtener las recomendaciones del grupo que el desee, indicando los usuarios que lo conforman.

RF 4. Obtención de recomendaciones:

- Cualquier usuario podrá obtener recomendaciones a grupos, para posteriormente visualizarlas, simplemente indicando la URL de los servicios desde la configuración de la aplicación, y los identificadores de los usuarios que lo conforman.

RF 5. Seleccionar el modo de visualización:

- Funcionalidad para que el usuario pueda seleccionar un modo de visualización de entre los disponibles.

RF 6. VISUALIZACION JERÁRQUCIA de GRUPO (VJG) – Ver de manera jerárquica las recomendaciones para el grupo:

- En el modo de visualización jerárquica, el usuario podrá ver las recomendaciones generales del grupo de manera jerárquica (a través de las distintas características de clasificación).

RF 7. VISUALIZACION JERÁRQUCIA INDIVIDUAL (VJI) – Ver de manera jerárquica las recomendaciones para un usuario del grupo:

- En el modo de visualización jerárquica, el usuario podrá ver las recomendaciones particulares de un solo usuario del grupo, de manera jerárquica (a través de las distintas características de clasificación).

RF 8. VJ vs Individual (VJvI) – Ver la aportación de un usuario del grupo frente a las recomendaciones generales del grupo:

- En el modo de visualización jerárquica, el usuario podrá ver la aportación que realiza un usuario, con sus recomendaciones, sobre las recomendaciones del grupo.

RF 9. VJ vs Grupo (VJvG) – Ver influencia de las recomendaciones generales del grupo sobre un miembro de él:

- En el modo de visualización jerárquica, el usuario podrá ver cómo influye el grupo en sus recomendaciones.

RF 10. VISUALIZACIÓN DE LINEAS a GRUPO (VLG) - Ver las recomendaciones para el grupo:

- En el modo de visualización en gráfica de líneas, el usuario podrá ver las recomendaciones generales del grupo, siendo el eje de la Y los valores de la recomendación, y el eje de la X los ítems recomendados.

RF 11. VISUALIZACIÓN DE LINEAS INDIVIDUAL (VLI) - Ver las recomendaciones para un usuario del grupo:

- En el modo de visualización en gráfica de líneas, el usuario podrá ver las recomendaciones particulares de un solo usuario del grupo, siendo el eje de la Y los valores de la recomendación, y el eje de la X los ítems recomendados.

RF 12. VL-C – Combinar recomendaciones:

- En el modo de visualización en gráfica de líneas, el usuario podrá ver, de manera combinada, las recomendaciones para el grupo y/o para el usuario y/o usuarios particulares que desee, Siendo cada una de estas una línea del gráfico.

RF 13. VISUALIZACIÓN ESCALADO MULTIDIMENSIONAL a GRUPO (MDSG) - Ver las recomendaciones para el grupo:

- En el modo de visualización de escalado multidimensional, el usuario podrá ver las recomendaciones generales del grupo, mostrándose estás en forma de coordenadas en 2 dimensiones.

RF 14. MDSI – Comparar las recomendaciones del grupo con las de un usuario:

- En el modo de visualización de escalado multidimensional, el usuario podrá comparar la visualización de las recomendaciones para el grupo, con las de un usuario en particular, mostrándose las dos a la vez.

RF 15. Ver la información de una película:

- Tanto en el modo de visualización jerárquico, como en el modo de visualización de escalado multidimensional, el usuario podrá ver la información de la película que seleccione.

Así quedan definidas de manera general las funcionalidades que la aplicación debe proporcionar a sus usuarios.

4.1.3. Requisitos No Funcionales

Los requerimientos no funcionales son aquellos que restringen los requerimientos funcionales. Son requisitos complementarios o atributos de calidad que proporcionan al usuario las funcionalidades requeridas de forma eficiente. Especifican propiedades del sistema o del producto en sí (plataforma, velocidad, rendimiento...), del diseño de la interfaz gráfica con el usuario, y de todas las restricciones impuestas por la organización, en lugar de su comportamiento (requisitos funcionales).

Seguridad

- Para poder utilizar la aplicación hace falta conocer las url de los servicios de recomendación.
- No hace falta autentificarse en el sistema.

Mantenibilidad y portabilidad

- Disponible para cualquier plataforma PC con sistema operativo Windows 64 bits, en cualquiera de sus versiones.
- Será necesaria una conexión a internet, ya sea por WI-FI o cable.

Interfaz y usabilidad

- La aplicación debe constar de una interfaz sencilla, atractiva e intuitiva. De tal forma que su uso no suponga un impedimento o esfuerzo al usuario a la hora hacer uso de la aplicación.
- La introducción de los datos de configuración debe estar estructurada procurando evitar errores.
- Las herramientas empleadas a la hora de desarrollar la aplicación, así como los algoritmos implementados, están lo suficientemente contrastados para garantizar que los resultados que darán a los usuarios finales van a ser útiles y de calidad.

Rendimiento

- Se esperan tiempos de respuesta no superiores a 5 segundos en el proceso de petición de las recomendaciones al servidor, recepción y carga.
- Los cálculos que se realicen en la aplicación no suponen demasiada carga para el dispositivo, por lo que el rendimiento será bastante óptimo.

Los requerimientos no funcionales que también se deben obtener y analizar para este proyecto son los referentes a las necesidades hardware y software de los equipos informáticos. Puesto que este proyecto consta de una arquitectura cliente servidor, habrá que especificar los requisitos mínimos de la maquina servidor, y de la máquina cliente:

Cliente

Por un lado, los requerimientos hardware del equipo informático del cliente son bastante simples, ya que únicamente necesitará un equipo con una tarjeta gráfica de gama baja-media (o simplemente una GPU integrada en la placa base del equipo), conexión a Internet, y requisitos de procesamiento, memoria RAM y almacenamiento no demasiado potentes.

Y por otro lado, los requerimientos software del cliente son: Un Sistema operativo Microsoft Windows en cualquiera de sus versiones (compatibles con 64

bits) y, para el correcto funcionamiento de la aplicación y de las librerías gráficas de los modos de visualización, será necesaria la instalación de la plataforma Java JRE en su versión 8, y en su modalidad de 64 bits.

Servidor

En el caso del servidor de Recomendaciones, **aunque este no entra en el ámbito ni alcance de este proyecto**, debería ser capaz de atender eficazmente las solicitudes de recomendaciones.

Dicho servidor está actualmente en funcionamiento y los únicos requisitos hardware mínimos para un funcionamiento normal serían:

- Procesador: Suficientemente potente para calcular las recomendaciones en un tiempo ajustado.
- Memoria RAM: debe ser la suficiente para permitir un uso fluido del sistema.
- Almacenamiento: Suficiente para almacenar los datos de los ítems que recomienda.
- Conexión y disponibilidad asegurada todo el tiempo.

Requisitos de la Interfaz de la Aplicación

La interfaz de la aplicación puede afectar a la experiencia del usuario final, por lo que nos tenemos que centrar en integrar una usabilidad mínima para que los usuarios trabajen de manera cómoda y eficiente.

Jackob Wielsen [39] definió la usabilidad como el atributo de calidad que mide lo fáciles que son de usar las interfaces de las aplicaciones. El grado de usabilidad de un sistema interactivo es, por su parte, una medida empírica y relativa de la usabilidad del mismo, es decir, de la sencillez de uso y facilidad de aprendizaje para interactuar con éste. Es empírica porque no se basa solamente en opiniones o sensaciones, denominados datos cualitativos, sino también en pruebas de usabilidad

realizadas para obtener datos cuantitativos (como por ejemplo, qué zonas observa el usuario, cuantos clics realiza para llevar a cabo una acción, etc.). Y a la vez es relativa porque el resultado depende de las metas y objetivos planteados o de una comparación con otros sistemas similares.

A pesar de ello, en un esfuerzo de homogeneizar las diferentes premisas y definiciones de los diferentes autores y entidades, los principios básicos en la usabilidad que deberemos utilizar se pueden resumir en:

- Facilidad de Aprendizaje: facilidad con la que nuevos usuarios desarrollan una interacción efectiva con el sistema o producto. Está relacionada con la predictibilidad, sintetización, familiaridad, la generalización de los conocimientos previos y la consistencia.
- Facilidad de Uso: facilidad con la que el usuario hace uso de la herramienta, con menos pasos o más naturales a su formación específica. Tiene que ver con la eficacia y eficiencia de la herramienta.
- Flexibilidad: relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.
- Robustez: es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. Está relacionada con la capacidad de observación del usuario, de recuperación de información y de ajuste de la tarea al usuario.

Los principales beneficios que se obtienen empleando los conceptos y técnicas de usabilidad a la hora de desarrollar una aplicación se pueden resumir en:

- Incremento en la productividad del usuario.
- Incremento en la satisfacción del usuario final.
- Reducción de los costes en desarrollo y mantenimiento de las aplicaciones.

- Disminución de los costes de capacitación y apoyo a los usuarios. Todos estos beneficios implican una reducción y optimización general de los costes de producción, así como un aumento en la productividad.

La usabilidad permite mayor rapidez en la realización de tareas y reduce posibles pérdidas de tiempo a los usuarios.

Requisitos de los Usuarios Finales

Los usuarios implicados en la utilización de la aplicación deben de poseer una serie de requisitos y capacidades para interactuar con el sistema.

- *Conocimientos relativos al problema*: el usuario en este aspecto tiene que tener diversos conocimientos relativos a las Recomendaciones de ítems a Grupos, para poder identificar y entender lo que se muestra en las distintas visualizaciones.
- *Habilidad con la tecnología*: Tiene que tener unos conocimientos básicos de utilización de aplicaciones de escritorio y del sistema operativo Windows, ya que en otro caso no podrá trabajar en el sistema.

4.1.4. Casos de uso

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto, estos describen todas las posibles interacciones que se pueden realizar entre un usuario del sistema y la aplicación con el fin de alcanzar su objetivo. De esta manera se presentan una serie de escenarios que indican la sucesión de respuestas ante las acciones iniciadas por los usuarios sobre el sistema.

Esta especificación de casos de uso facilita la comprensión del sistema y expresa la intención con la que el usuario interactuará con la aplicación, ampliando el conocimiento de dichas necesidades, y que junto con los requisitos anteriormente descritos, establece una firme base a la hora de comenzar con la fase de diseño.

Los elementos de los que consta cada diagrama de casos de uso son los siguientes:

Actor: se trata del rol tomado por la entidad que actúa de forma externa al sistema, interactuando con el mismo con el objetivo de satisfacer una necesidad mediante las funcionalidades que el sistema presenta.

Podemos encontrar los siguientes actores en el sistema:

- **Servidor de recomendaciones**: servidor de la Universidad de Jaén que consta de un motor de recomendación y una base de datos, al cual se conecta nuestra aplicación para hacer peticiones de información.
- **Servidor de información de películas**: servidor externo abierto que consta de una API y una base de datos, al cual se conecta nuestra aplicación para hacer peticiones de información.
- **Usuario**: usuario que interactúa con la aplicación y realiza las peticiones a los servidores externos.

Caso de uso: es la operación realizada tras un evento, siendo éste realizado por petición del actor u otra entidad externa, o por acción de otro caso de uso.

Relaciones: el vínculo entre actor y caso de uso. Puede ser de diversos tipos:

- **Relación de asociación**: relación básica entre casos de uso, o un actor y un caso de uso que representa la invocación de una operación.
- **Relación de inclusión:** interacción entre casos de uso, donde la invocación de una operación de uno depende del resultado del caso de uso incluido. Se marca con <<include>>.
- **Relación de extensión:** interacción entre casos de uso, y se entiende como una función opcional del sistema. Se marca con <<extend>>.

El primer caso de uso que debemos definir será el denominado diagrama frontera, o lo que es lo mismo, un diagrama que nos mostrará el funcionamiento en general del sistema:

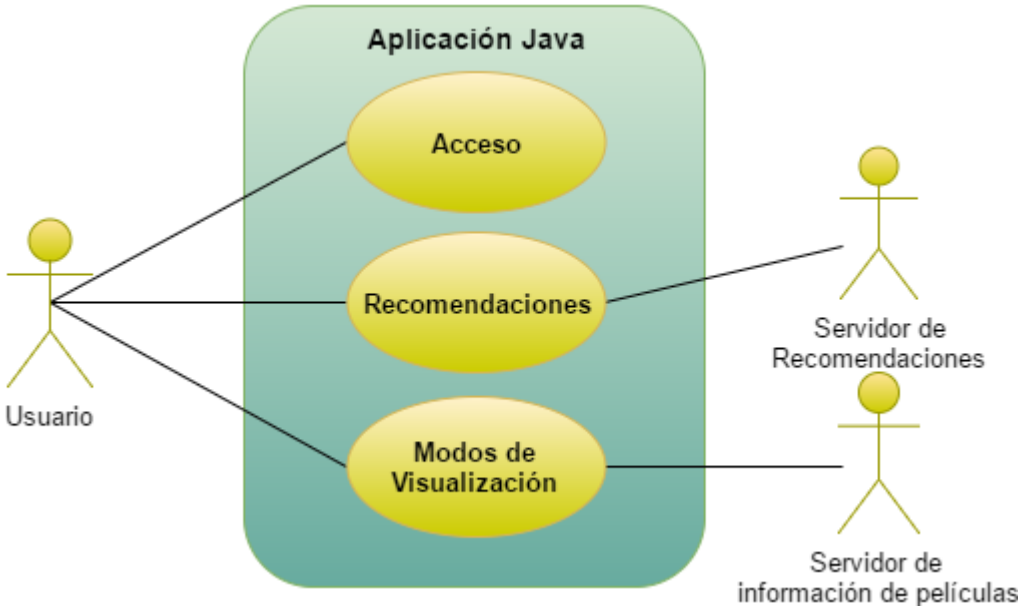

Figura 4.1 Diagrama frontera.

Este diagrama frontera está compuesto por 3 casos de uso principales que recogen todo lo expuesto en la especificación de requerimientos. Estos son:

- Acceso y uso de la aplicación [Requisito funcional 1].
- Recomendaciones [Requisitos funcionales 2, 3, 4].
- Modos de visualización [Requisitos funcionales 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 y 15].

Una vez definido esto, vamos a pasar a ver los diferentes Casos de Uso que han surgido de nuestro diagrama frontera.

Además se va a complementar cada diagrama de caso de uso mostrado con una descripción textual del mismo, consiguiendo así una explicación más detallada y formal de cada uno.

Se definen con el siguiente contenido:

- Nombre del caso de uso
- Breve descripción del caso de uso
- Actor: Actor o actores que interacciona con el caso de uso en cuestión.
- Condiciones de entrada: Condiciones que deben cumplirse de forma previa para realizar la operación.
- Operaciones básicas del escenario principal.
- Camino alternativo.
- Condiciones de salida: Estado del sistema después de la realización de la operación.

Estos son los diagramas de caso de uso considerados:

Caso de Uso #1. Acceso

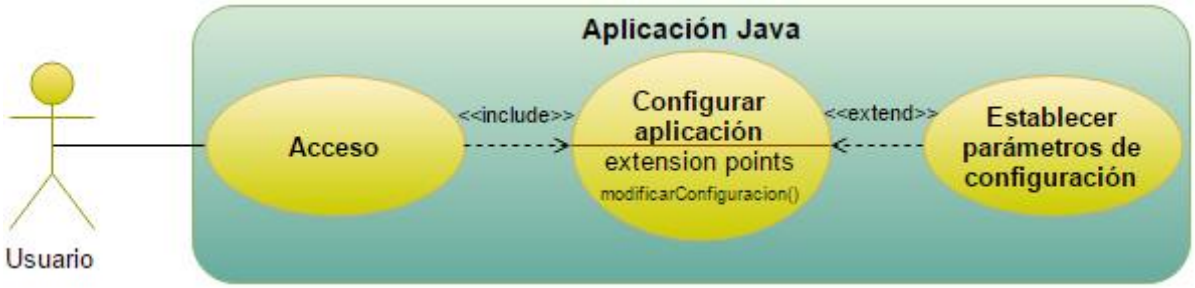

Figura 4.2 Caso de uso #1. Acceso.

Nombre	Acceso
Descripción	Permite entrar en el sistema y hacer uso de él
Actor	Usuario
Condiciones de entrada: La aplicación está instalada en el equipo.	
Operaciones básicas: 1. El usuario ejecuta la aplicación y pasa a primer plano. 2. El usuario debe de introducir una configuración inicial, como la url del servicio de obtención de recomendaciones.	
Alternativas: 1. b. La aplicación no se ha instalado correctamente, o falta alguna dependencia (como jre de Java).	
Condiciones de salida: El usuario a la aplicación y está listo para utilizarla.	

Tabla 4.1. Caso de uso #1. Acceso.

Caso de Uso #2. Recomendaciones.

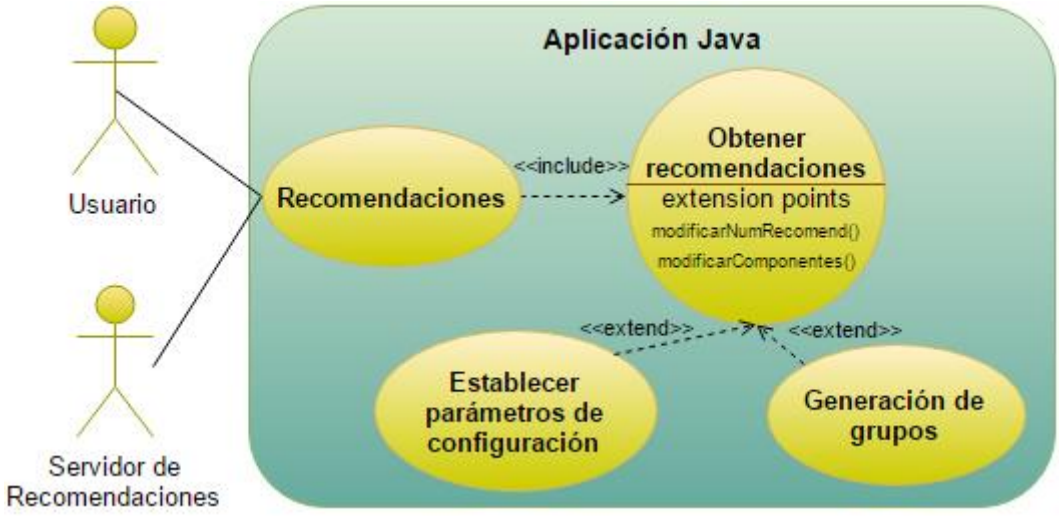

Figura 4.3 Caso de uso #2. Recomendaciones.

Nombre	Recomendaciones
Descripción	Permite al usuario obtener recomendaciones para el grupo elegido
Actor	Usuario, Servidor de Recomendaciones
Condiciones de entrada: Haber configurado correctamente las URL de los servicios.	
Operaciones básicas: 1. El usuario indica el número de recomendaciones que desea recibir. 2. El usuario indica los componentes de los que se compone el grupo. 3. El usuario pulsa buscar, y la aplicación solicita las recomendaciones.	
Alternativas: 1. b. El usuario no lo indica (valor por defecto). 2. b. Alguno de los usuarios indicados como miembros del grupo no existe. 3. b. No hay conexión con el servidor. 4. El usuario cancela el proceso de obtención de recomendaciones.	
Condiciones de salida: La aplicación recibe las recomendaciones para el grupo indicado.	

Tabla 4.2. Caso de uso #2. Recomendaciones.

Caso de Uso #3. Modos de Visualización.

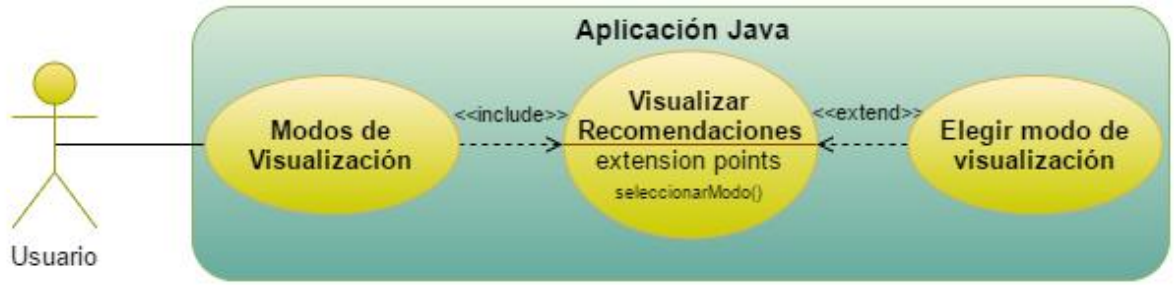

Figura 4.4 Caso de uso #3. Modos de visualización.

Nombre	Modos de Visualización
Descripción	Permite al usuario seleccionar un modo de visualización para mostrar las recomendaciones
Actor	Usuario
Condiciones de entrada: Se han recibido correctamente las recomendaciones.	
Operaciones básicas: 1. El usuario selecciona un modo de visualización de entre los disponibles, y pulsa en visualizar.	
Alternativas: 1. b. Si el usuario no lo cambia, se accede al primer método disponible.	
Condiciones de salida: El usuario accede al modo de visualización seleccionado.	

Tabla 4.3. Caso de uso #3. Modos de visualización.

Caso de Uso #4. Modo de Visualización Jerárquico.

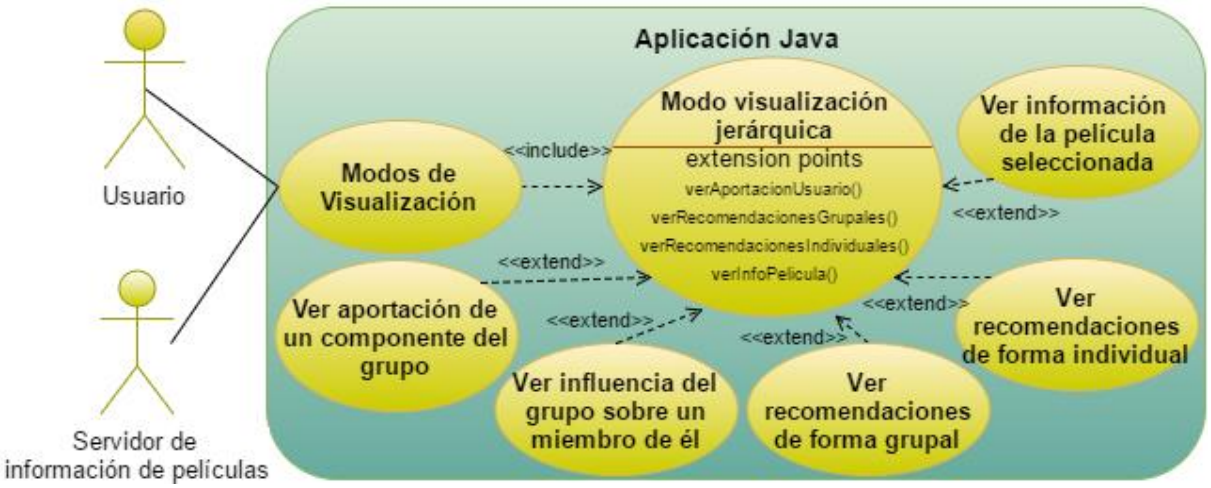

Figura 4.5 Caso de uso #3. Modo de visualización jerárquico.

Nombre	Modo de Visualización Jerárquico
Descripción	Permite al usuario visualización las recomendaciones de manera jerárquica
Actor	Usuario, Servidor de Información de películas.
Condiciones de entrada: Se ha seleccionado el modo de visualización jerárquico.	
Operaciones básicas: 1. El usuario selecciona visualizar las recomendaciones de manera grupal. 2. El usuario navega, jerárquicamente, por las distintas categorías de clasificación de las películas. 3. Estando en el último nivel, donde aparecen las películas, el usuario selecciona ver la información para esa película. 4. Se solicita al servidor de información de películas que nos envié los datos para esa película. 5. Se muestran los datos de la película por pantalla. 6. El usuario puede ver la aportación que realiza un miembro del grupo al grupo a cualquier nivel de la jerarquía.	
Alternativas: 1. b. El usuario selecciona visualizar las recomendaciones de manera individual, seleccionado un miembro del grupo. 4. b. No hay conexión con el servidor. 5. b. Se muestra un mensaje en pantalla si se ha producido cualquier tipo de error. b. El usuario puede ver la cómo influye un miembro del grupo al grupo a cualquier nivel de la jerarquía	
Condiciones de salida: El usuario visualiza las recomendaciones de manera jerárquica y puede interactuar con la visualización.	

Tabla 4.4. Caso de uso #3. Modo de visualización jerárquico.

Caso de Uso #5. Modo de Visualización de Gráfico de Líneas.

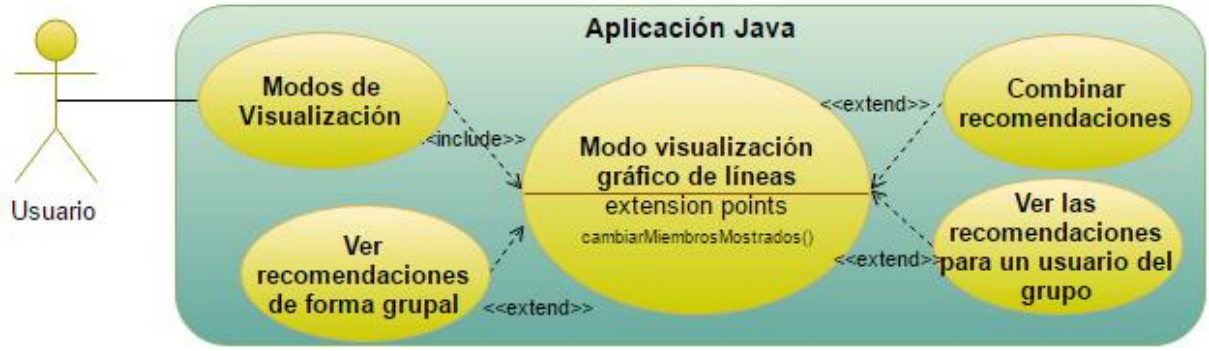

Figura 4.6 Caso de uso #5. Modo de visualización de gráfico de líneas.

Nombre	Modo de Visualización de Gráfico de Líneas
Descripción	Permite al usuario visualización las recomendaciones mediante un gráfico de líneas
Actor	Usuario
Condiciones de entrada: Se ha seleccionado el modo de visualización de gráfico de líneas.	
Operaciones básicas: 1. El usuario selecciona que individuos del grupo quiere ver en el gráfico de líneas, y, sí además, quiere ver la línea con las recomendaciones grupales, para realizar comparaciones. 2. La visualización se actualiza mostrando los miembros elegidos.	
Alternativas: -	
Condiciones de salida: El usuario visualiza las recomendaciones mediante un gráfico de líneas y puede interactuar con la visualización.	

Tabla 4.5. Caso de uso #5. Modos de visualización de gráfico de líneas.

Caso de Uso #6. Modo de Visualización Escalado multidimensional.

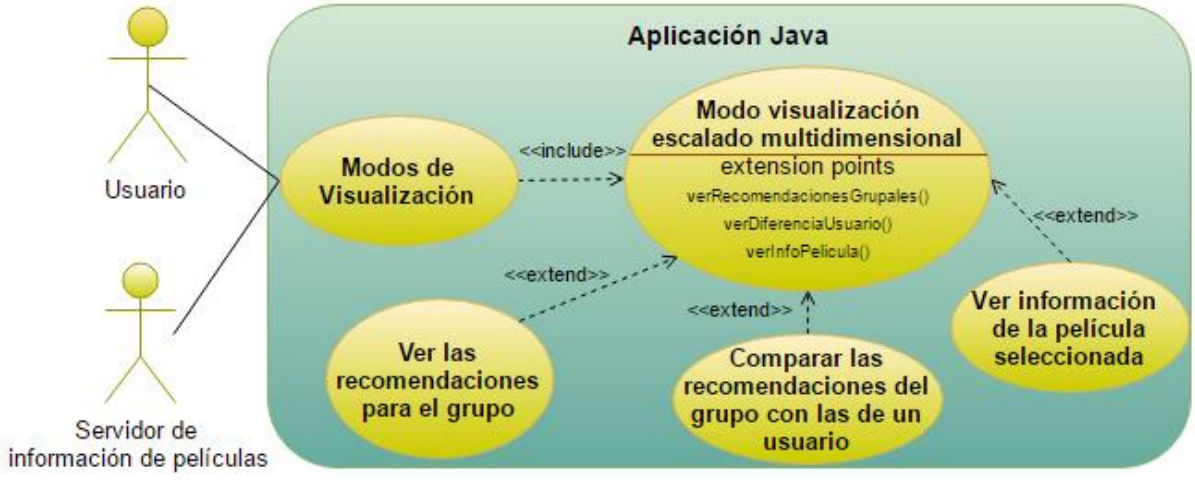

Figura 4.7 Caso de uso #6. Modo de visualización escalado multidimensional.

Nombre	Modo de Visualización Escalado multidimensional
Descripción	Permite al usuario visualización las recomendaciones por escalado multidimensional.
Actor	Usuario, Servidor de Información de películas.
Condiciones de entrada: Se ha seleccionado el modo de visualización por escalado multidimensional.	
Operaciones básicas: 1. El usuario selecciona visualizar las recomendaciones mediante escalado multidimensional. 2. El usuario visualiza las recomendaciones grupales representadas mediante coordenadas. 3. El usuario puede comparar estas recomendaciones grupales con las de un miembro individual del grupo, seleccionado dicho usuario. 4. El usuario visualiza las recomendaciones grupales junto con las de un miembro individual. 5. El usuario selecciona ver la información para una película. 6. Se solicita al servidor de información de películas que nos envié los datos para esa película. 7. Se muestran los datos de la película por pantalla.	
Alternativas: 3. b. Mientras el usuario no seleccione un miembro del grupo, se mostrarán las recomendaciones del primer miembro del grupo. 6. b. No hay conexión con el servidor. 7. b. Se muestra un mensaje en pantalla si se ha producido cualquier tipo de error.	
Condiciones de salida: El usuario visualiza las recomendaciones por escalado multidimensional y puede interactuar con la visualización.	

Tabla 4.6. Caso de uso #6. Modo de visualización escalado multidimensional.

4.1.5. Escenarios

Los escenarios, que suelen basarse en la información reunida durante el análisis de los casos de uso, son muy variados. Algunos se centran en el nivel funcional, mientras que otros proporcionan información detallada sobre el nivel de tarea. El término “caso de uso” se utiliza para hacer referencia a muchas cosas distintas y algunos tipos de casos de uso son muy similares a los escenarios que se describen aquí. Por lo general, los escenarios de alto nivel se utilizan en la fase de análisis de productos nuevos y los escenarios de un nivel más detallado se utilizan en el rediseño de productos ya existentes.

Los escenarios son informes individuales y ficticios con datos sobre el flujo de trabajo. Un escenario es una descripción de un actor que utiliza un producto para conseguir un fin. Los escenarios suelen ser relatos que cuentan una historia en la que se describe una o más tareas desarrolladas en una situación ambiental concreta.

Con el desarrollo de escenarios se suelen identificar aspectos importantes que afectan a la utilización de un producto en el mundo real y que no se pueden identificar ni tenerse en cuenta de otro modo. Los escenarios son útiles a lo largo del proceso de diseño, especialmente al desarrollar descripciones de tareas para las pruebas de usabilidad.

Hay que recordar que las personas son muy distintas. No hay que dar por sentado que todos los usuarios, incluyendo los usuarios con discapacidad, utilizan el producto de la misma forma. Las personas utilizan técnicas de interacción, estrategias de adaptación y configuraciones de tecnologías de apoyo distintas. Cada persona tiene experiencias, expectativas y preferencias diferentes.

Se pueden hacer varios escenarios por cada caso de uso, pero uno para cada caso de uso debería ser suficiente para ver las cosas más claras.

- **Nombre del escenario:** Acceso a la aplicación
- **Descripción:** Juan quiere acceder u utilizar la aplicación.

- Flujo de eventos:

1. Juan instala la aplicación.
2. Juan abre la aplicación desde el ejecutable.
3. Pulsa el botón configuración.
4. Introduce la url 'http://serezade.ujaen.es:8047/delfos-web-ml-100k/' perteneciente a los servicios de obtención de recomendaciones.
5. Pulsa el botón de 'Guardar'
6. Si la url es válida, Juan está preparado para acceder a todas las funcionalidades de la aplicación.

- **Nombre del escenario:** Obtener recomendaciones
- **Descripción:** Pedro quiere obtener las recomendaciones para un grupo.

- Flujo de eventos:

1. Pedro pulsa el botón configuración.
2. Indica que quiere recibir '50' recomendaciones para el modo de visualización jerárquico.
3. Pulsa el botón de 'Guardar'
4. En el menú principal, Pedro indica que quiere recibir recomendaciones para el grupo formado por los usuarios "1, 3, 5, 10, 25"
5. Pulsa el botón de 'Comenzar'
6. Pedro recibe las recomendaciones para el grupo.

- **Nombre del escenario:** Seleccionar el modo de visualización.
- **Descripción:** Ana quiere seleccionar el modo de visualización en el cual verá las recomendaciones.

- Flujo de eventos:

1. Ana pulsa en el menú desplegable de modos de visualización en el menú principal de la aplicación.
2. Indica que quiere visualizar las recomendaciones que obtendrá mediante el modo de 'Escalado Multidimensional'.
3. Ana esta lista para recibir las recomendaciones y visualizarlas en el modo que ha seleccionado.

- **Nombre del escenario:** Modo de Visualización Jerárquico
- **Descripción:** Luis quiere visualizar las recomendaciones mediante el modo de visualización jerárquica.

- Flujo de eventos:

1. Luis selecciona el modo de visualización jerárquica.
2. Pulsa el botón de 'Comenzar'
3. Envía la solicitud de recomendaciones, las recibe y se crea el modelo de visualización.
4. En la visualización que se muestra, Luis selecciona ver las recomendaciones grupales.
5. Selecciona la categoría 'Drama'
6. Selecciona el año '1986'
7. Selecciona la película 'La vie en Rose'
8. Luis ve los datos de esa película.
9. Luis pulsa 'Ver aportación de un usuario' para ver cómo afectan las películas de 1986 de un miembro del grupo a la recomendación grupal.
10. En el menú que se despliega, Luis selecciona del usuario 5.
11. Se muestra la aportación del usuario 5.
12. Luis pulsa sobre 'Ocultar aportación'.
13. Ahora Luis pulsa en el centro del gráfico jerárquico para cerrarlo y volver al inicio del gráfico.
14. Luis selecciona ver las recomendaciones a nivel de un miembro del grupo.
15. Selecciona el usuario 10.
16. Selecciona la categoría 'Animación
17. Selecciona el año '1995'
18. Selecciona la película 'Toy Story'
18. Luis ve los datos de esa película.
19. Luis pulsa 'Ver influencia en el grupo' para ver cómo influyen las películas de 1995 del usuario 10 en la recomendación grupal.
20. Se muestra la influencia en el grupo.
21. Luis pulsa sobre 'Ocultar influencia'.

- **Nombre del escenario:** Modo de Visualización Gráfico de Líneas
- **Descripción:** María quiere visualizar las recomendaciones mediante el modo de visualización en gráfico de líneas.

- Flujo de eventos:

1. María selecciona el modo de visualización mediante gráfico de líneas.
2. Pulsa el botón de 'Comenzar'
3. Envía la solicitud de recomendaciones, las recibe y se crea el modelo de visualización.
4. En la visualización que se muestra, María marca las casillas de los usuarios 10 y 25 para combinar la línea de las recomendaciones grupales que aparece, con las líneas de estos usuarios.
5. María desmarca la casilla de la línea de las recomendaciones grupales.
6. Por último marca la casilla del usuario 1, para que aparezca su línea.

- **Nombre del escenario:** Modo de Visualización Escalado Multidimensional
- **Descripción:** Antonio quiere visualizar las recomendaciones mediante el modo de visualización escalado multidimensional.

- Flujo de eventos:

1. Antonio selecciona el modo de visualización por escalado multidimensional.
2. Pulsa el botón de 'Comenzar'
3. Envía la solicitud de recomendaciones, las recibe y se crea el modelo de visualización.
4. En la visualización que se muestra, Antonio ve a la derecha el gráfico que corresponde a las recomendaciones a nivel grupal.
5. Hace clic en el círculo de la película 'Casablanca'
6. Antonio ve los datos de esa película.
7. Antonio pulsa 'Ver diferencia con un usuario' para comparar el gráfico grupal con el gráfico de uno de los miembros del grupo.
8. En el menú que se despliega, Antonio selecciona del usuario 5.
9. Se muestra el gráfico de las recomendaciones para el usuario 5 a la izquierda de la pantalla.
10. Pulsa sobre Ver diferencia con un usuario' cambiar el gráfico del miembro del grupo que se muestra por otro miembro.
11. En el menú que se despliega, Antonio selecciona el usuario 10.
12. Se muestra el gráfico de las recomendaciones para el usuario 10 en el lugar donde se encontraba el gráfico para las recomendaciones del miembro del grupo mostrado anteriormente.

Con esto se han definido los escenarios necesarios para ver cómo funciona el sistema normalmente, completando así la primera fase de la ingeniería del software del proyecto, el Análisis del Sistema.

4.2. Diseño del Sistema

4.2.1. Visión global

La etapa de diseño consiste en traducir los requerimientos funcionales y no funcionales en una representación de software. El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería.

El objetivo del diseño es producir un modelo o representación de una entidad que se va a construir posteriormente.

Hay tres características que sirven como parámetros generales para la evaluación de un buen diseño. Estos parámetros son los siguientes:

1. El diseño debe implementar todos los requisitos explícitos obtenidos en la etapa de análisis.
2. El diseño debe ser una guía que puedan leer y entender los que construyen el código y los que prueban y mantienen el software.
3. El diseño debe proporcionar una idea completa de lo que es el software.

El diseño de todo el proyecto sería complejo llevarlo a cabo en una sola fase, por lo que lo mejor es modularizar la fase de diseño en varias fases en la que cada una definirá una parte de la aplicación.

- **Fase de estructura del sistema,** donde se describirán las **clases** que conforman el modelo del sistema.
- **Fase de estructura de los datos,** donde se detallará el uso que se realiza de los datos y como se gestionan.
- **Fase de definición de la interfaz de la aplicación**, donde se prototiparán las pantallas de nuestra aplicación, mediante **storyboards**.

4.2.2. Diagrama de Clases

Antes de mostrar el diagrama de clases, vamos a explicar cómo va a ser el funcionamiento o la lógica del sistema, para ver más claramente el paso de información o comunicación entre las distintas capas diseñadas. A la hora de diseñar

una aplicación con una interfaz gráfica de usuario es importante seguir el esquema de diseño Modelo-Vista-Controlador (MVC) [26], en el que se definen tres roles bien diferenciados:

- **Modelo**: el modelo es la representación de la información de un problema.
- **Vista**: una vista es una posible visualización de la información contenida en un modelo. Un modelo puede tener varias vistas definidas.
- **Controlador**: el controlador se encarga de coordinar la interacción entre las vistas y los modelos. Cada vez que un modelo cambia internamente, actualiza sus vistas.

El mensaje más importante que proporciona este esquema de diseño es, que hay que mantener al modelo lo más independiente posible de la vista y el controlador, de forma que se pueda cambiar uno u otros sin necesidad de realizar ninguna modificación en los demás componentes.

En nuestra aplicación tendremos una capa modelo de la información que representará todos los datos del sistema.

Analizamos cada uno de los paquetes de nuestro diagrama de clases:

- Paquete **models**: Implementa la lógica de la aplicación, es decir, almacena, organiza y contiene los métodos para manipular los datos.
- Paquete **controllers**: Es la capa controladora, llevará a cabo el control del sistema y permitirá al usuario realizar cualquier operación sobre éste. Utilizará el Modelo para obtener y guardar información, y será la encargada de operar con la información para mostrarla en la Vista.
- Paquete **views**: Implementa todo lo relacionado con la interfaz y la forma de hacer visualizar gráficamente la información al usuario.
- Paquete **utils**: Contiene recursos necesarios para el manejo de información.

A continuación vamos a ver con más detalle los paquetes de nuestra aplicación:

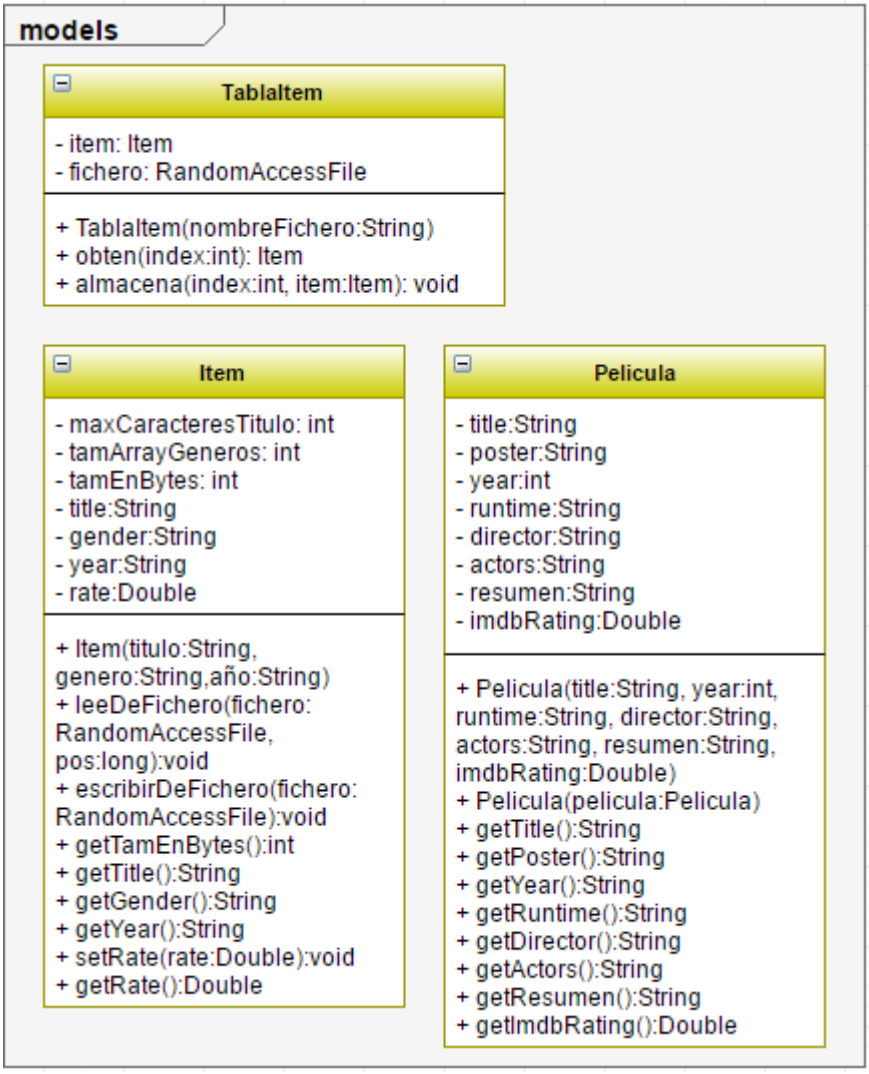

Figura 4.8 Paquetes models.

Analizamos ahora este primer diagrama correspondiente a los Modelos: Aquí tenemos todas las clases necesarias para la correcta gestión de los datos de la aplicación. Vamos a definirlas lo más sencillamente posible.

- **Item**: Esta clase es el modelo de datos para mapear y guardar y trabajar con las películas que recibimos del servidor de recomendaciones.
- **TablaItem**: Esta clase se utiliza para crear, leer y modificar el fichero indexado donde se guarda la información básica de las películas para agilizar los procesos.
- **Película**: Esta clase es el modelo de datos para mapear y trabajar con los datos informativos de las películas que recibimos del servidor de información de películas.

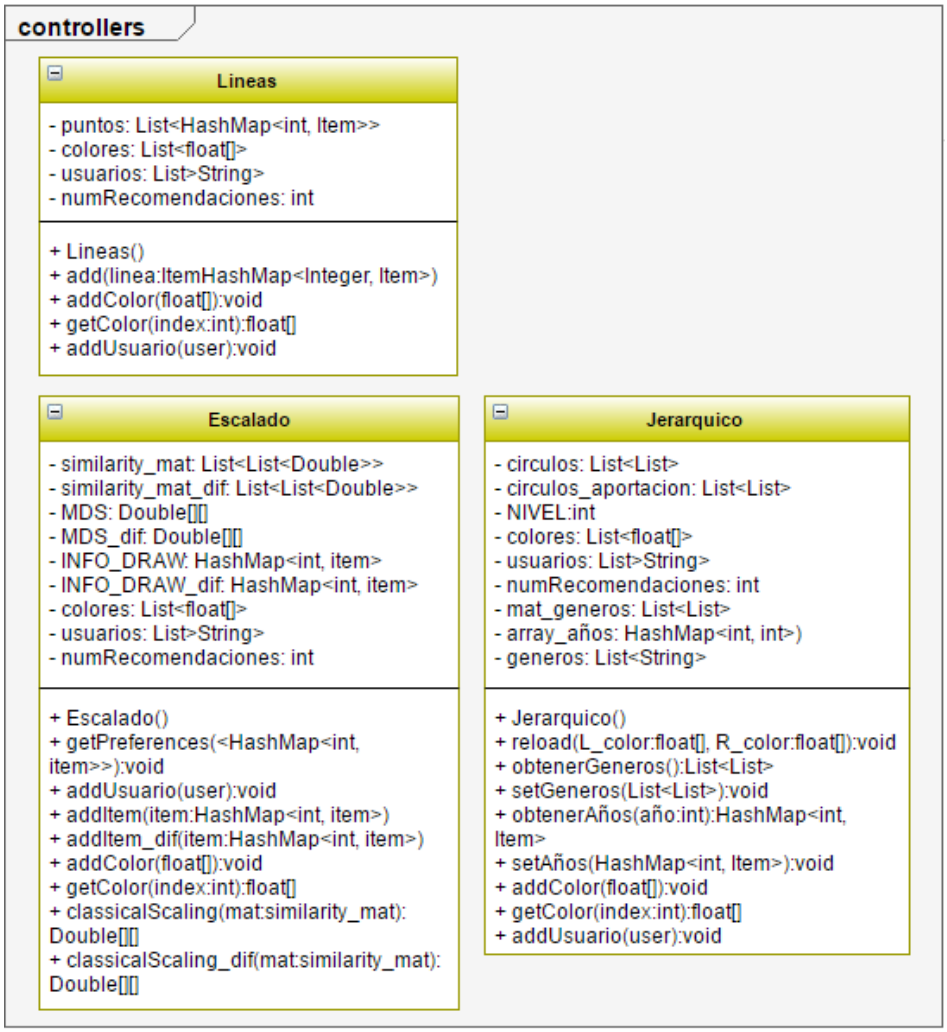

Figura 4.9 Paquetes controllers.

Analizamos ahora este diagrama correspondiente a los Controladores: Aquí tenemos todas las clases necesarias para el correcto tratamiento de los datos, procesamiento, y comunicación con la vista para mostrarlos. Vamos a definirlas lo más sencillamente posible.

- **Líneas**: Esta clase es el controlador para gestionar el modo de visualización en gráfico de líneas, y es el encargado de crear cada una de las líneas con los datos de las películas correspondientes.
- **Escalado**: Esta es la encargada, principalmente, de convertir los datos de las películas, que tienen multitud de dimensiones, en coordenadas 2D y poder visualizarlas como puntos en el espacio.
- **Jerárquico**: Esta clase es la encargada de clasificar las películas en cada uno de los niveles de la jerarquía, y de ir gestionando el despliegue del gráfico jerárquico.

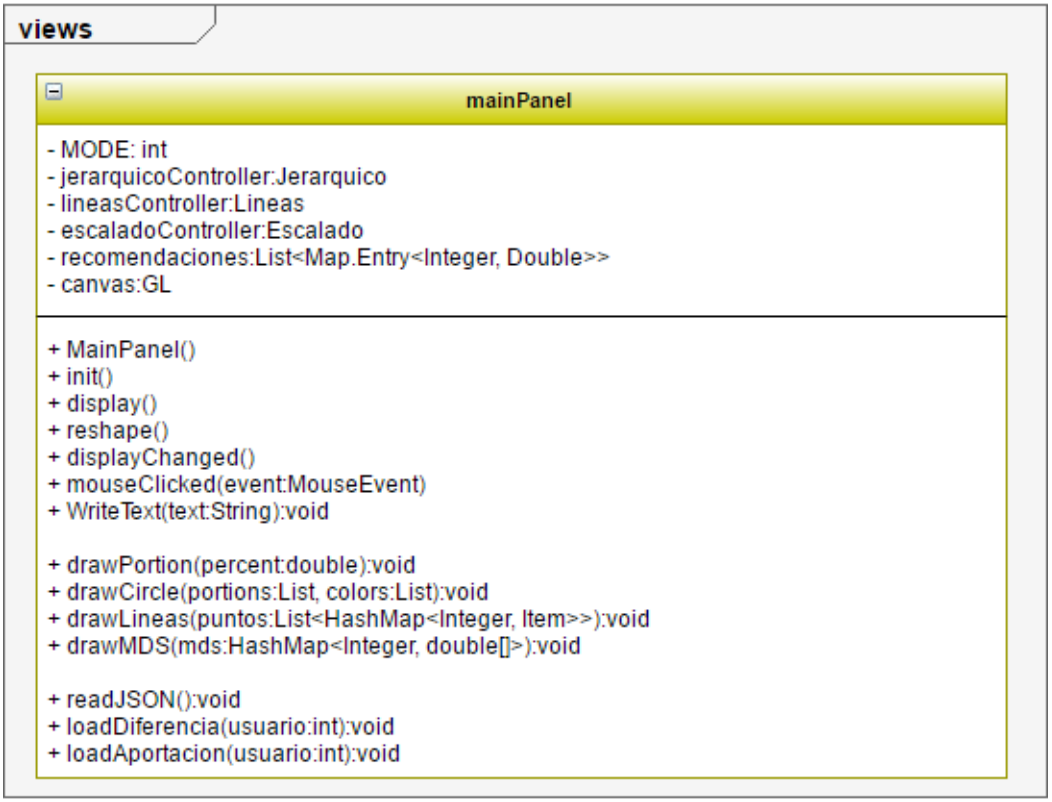

Figura 4.10 Paquetes views.

Analizamos ahora este diagrama correspondiente a la Vista: Aquí tenemos la única clase necesaria para visualizar las recomendaciones en cualquiera de los tres tipos de visualización disponibles. Vamos a definirla lo más sencillamente posible.

MainPanel: Esta clase es la vista de nuestra aplicación, y está compuesta de los siguientes métodos:

- Métodos propios de OpenGL: Son necesarios para mostrar gráficos en pantalla.
- drawPortion y drawCircle: Encargados de visualizar las recomendaciones en modo jerárquico. Primero se muestra el círculo del nivel correspondiente a la jerarquía, para después subdividir el círculo en porciones que corresponden a cada uno de los tipos de clasificación.
- drawLineas: Encargada de representar las recomendaciones en forma de línea.
- drawMDS: Encargada de representar las películas en puntos 2D.
- loadDiferencia y loadAportacion: Encargados de mostrar información comparativa para varios modos de visualización.

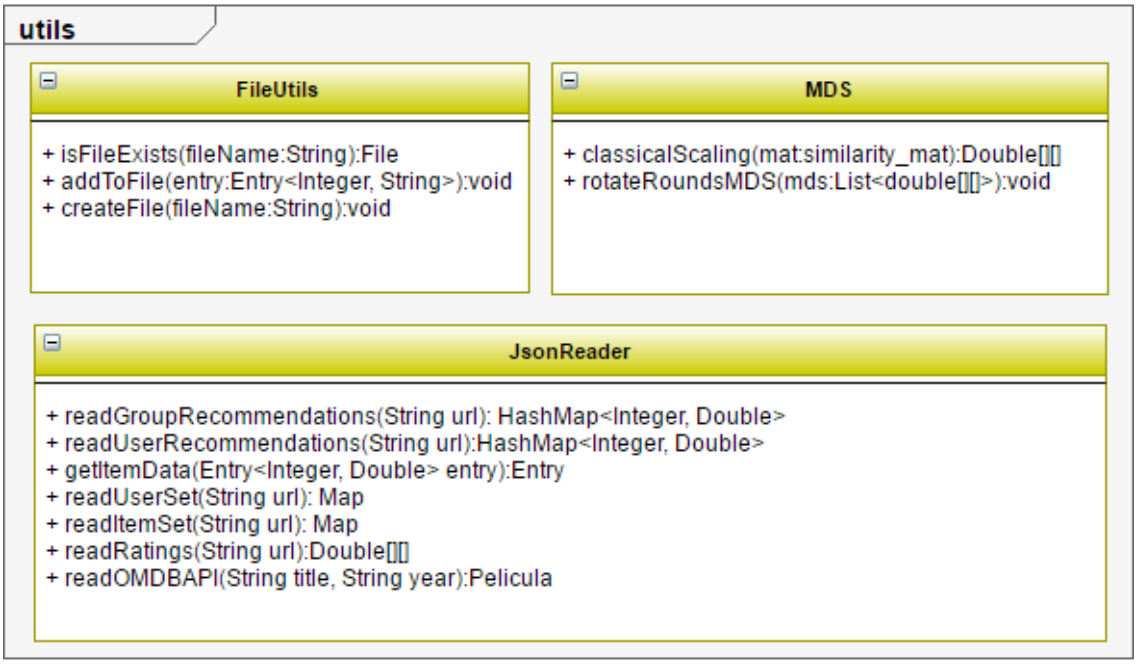

Figura 4.11 Paquetes utils.

Analizamos, por último, este diagrama correspondiente al paquete Utils: Aquí tenemos todas las clases adicionales que necesitamos para realizar tareas que den soporte a las funcionalidades principales de la aplicación. Vamos a definirlas lo más sencillamente posible.

- **FileUtils**: Esta clase es la responsable de realizar tareas sobre el fichero de almacenamiento que utilizamos en este proyecto. Es la encargada de crear el fichero si no existe y de mantenerlo.
- **MDS**: Esta es la encargada, principalmente, de convertir los datos de las películas, convertidos en una matriz de similitudes entre ellas, y que tienen multitud de dimensiones, en coordenadas 2D mediante el escalado multidimensional para poder visualizarlas como puntos en el espacio. También es la encargada de rotar los gráficos resultantes para que no pierdan la coherencia al realizar cambios.
- **JSONReader**: Esta clase es la encargada de comunicarse con los servicios web, tanto el de obtención de recomendaciones a grupo, como el que solicita información para una película concreta. Se encarga de recibir los resultados de los servicios, en formato JSON, des-serializarlos, y crear los objetos necesarios para que los controladores trabajen con esos tipos de datos.

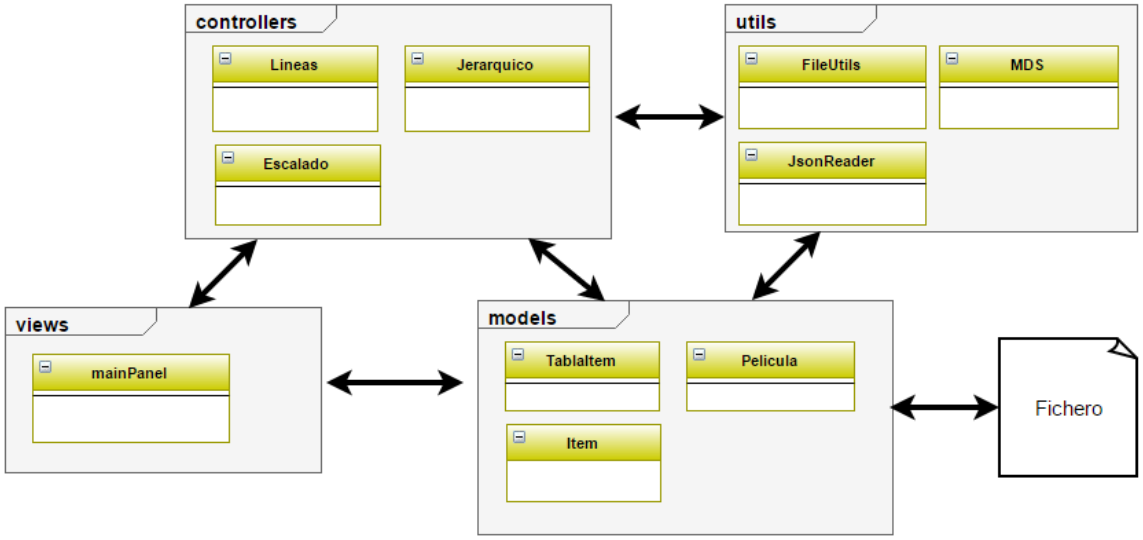

Figura 4.12 Relación de paquetes.

4.2.3. Gestión de los datos

Para trabajar y gestionar los datos de las recomendaciones a grupo que recibimos del servicio web, he optado en esta ocasión por almacenar parte de esa información que recibimos en un **fichero**, en vez de utilizar algún sistema gestor de base de datos.

He optado por esta opción ya que los datos que almacenaré de las respuestas son muy reducidos y simplemente se utilizarán para agilizar la carga de los modos de visualización. De este modo, no he creído oportuno adjuntar al proyecto ningún programa o librería externa de gestión de bases de datos, que aumentarían el peso y el rendimiento de la aplicación, considerando que la gestión que quería para los datos era muy básica, y no necesitaba la mayoría de las opciones y funcionalidades que los GBBDD ofrecen.

Un fichero es un conjunto de información relacionada, tratada como un todo y organizada de forma estructurada. Es una secuencia de dígitos binarios que organiza información relacionada con un mismo aspecto.

Los ficheros están formados por registros lógicos que contienen datos relativos a un mismo elemento u objeto (por ejemplo, en mi caso, los datos básicos de

clasificación de una película). A su vez, los registros están divididos en campos que contienen cada una de las informaciones elementales que forman un registro (por ejemplo, el año o los géneros).

Hemos de resaltar que los datos están almacenados de tal forma que se puedan añadir, suprimir, actualizar o consultar individualmente en cualquier momento.

Como los ficheros suelen ser muy voluminosos, solo se pueden llevar a la memoria principal partes de ellos para poder procesarlos. La cantidad de información que es transferida entre el soporte en el que se almacena el fichero, y la memoria principal del ordenador, en una sola operación de lectura/grabación, recibe el nombre de registro físico o bloque.

Normalmente en cada operación de lectura/grabación se transfieren varios registros del fichero, es decir un bloque suele contener varios registros lógicos. Al número de registros que entran en un bloque se le conoce con el nombre de factor de blocaje, y a esta operación de agrupar varios registros en un bloque se le llama bloqueo de registros.

Para la gestión de estos datos clasificatorios utilizaré un **Fichero de Acceso Aleatorio o Directo**. En este tipo de ficheros se puede acceder a un registro indicando la posición relativa del mismo dentro del archivo o, más comúnmente, a través de una clave que forma parte del registro como un campo más, y que permite identificar y localizar un registro de manera ágil y organizada. También, estos archivos deben almacenarse en dispositivos de memoria masiva de acceso directo, como son los discos magnéticos.

Cada uno de los registros se guarda en una posición física, que dependerá del espacio disponible en memoria masiva, de ahí que la distribución de los registros sea aleatoria dentro del soporte de almacenamiento. Para acceder a la posición física de un registro se utiliza una dirección o índice, no siendo necesario recorrer todo el fichero para encontrar un determinado registro.

A través de una transformación específica aplicada a la clave, se obtendrá la dirección física en la que se encuentra el registro. Según la forma de realizar esta transformación, existen diferentes modos de acceso:

En el acceso directo la clave coincide con la dirección, debiendo ser numérica y comprendida dentro del rango de valores de las direcciones. Es el método más rápido.

La medida básica de posicionamiento del puntero en el fichero es el byte, dependiendo del tipo de codificación de caracteres que empleemos (Unicode, ANSI) se utilizarán 1 o 2 bytes por carácter respectivamente. Teniendo esto en cuenta, el puntero avanzará uno en uno o de dos en dos bytes para poder leer o escribir cada carácter.

Otras características fundamentales de los ficheros de acceso directo o aleatorio son:

- Posicionamiento inmediato.
- Registros de longitud fija.
- Apertura del fichero en modo mixto, para lectura y escritura.
- Permiten múltiples usuarios utilizándolos.
- Los registros se borran colocando un cero en la posición que ocupan.
- Permiten la utilización de algoritmos de compactación de huecos.
- Los archivos se crean con un tamaño definido, es decir, con un máximo de registros establecido durante la creación.
- Esta organización sólo es posible en soportes direccionables.
- Se usan cuando el acceso a los datos de un registro se hace siempre empleando la misma clave y la velocidad de acceso a un registro es lo que más nos importa.
- Permiten la actualización de los registros en el mismo fichero, sin necesidad de copiar el fichero.
- Permiten realizar procesos de actualización en tiempo real.

Cada uno de los registros que tendrá nuestro fichero de películas seguirá la siguiente estructura:

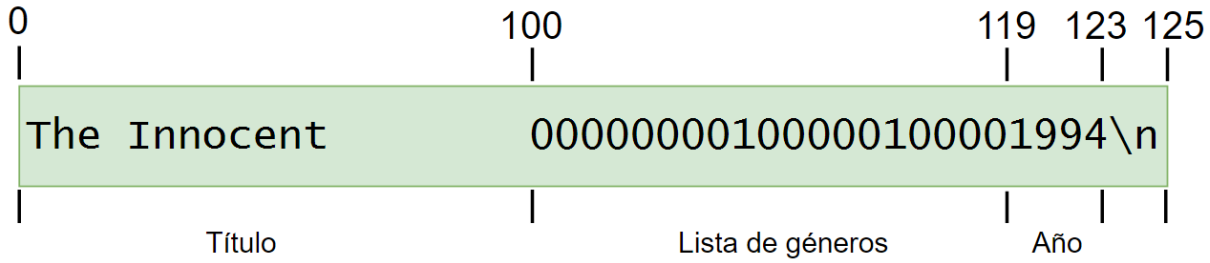

Figura 4.13 Registro de película.

- Los primeros 100 bytes corresponden al título de la película, que se completará de espacios en blanco si la longitud del título es menor que el tamaño del campo.
- Los siguientes 19 bytes corresponden a la lista de géneros. Cada byte x es un valor booleano que indica si la película es del género x o no. El orden de los géneros es siempre el mismo, y nos viene establecido por la respuesta del servicio.
- Los siguientes 4 bytes corresponden al año de la película.
- Los 2 últimos bytes corresponden a los símbolos CRLF: CR (retorno de carro) y LF (salto de línea).

Para almacenar esta cadena de bytes, Java dispone de la clase RandomAccessFile, para crear ficheros que contienen bytes y que permiten el acceso aleatorio, pudiendo usarse para leer, o leer y escribir a la vez.

Estos se enumeran con un índice que empieza en cero que está almacenado en el sistema y que se llama "puntero de lectura/escritura".

Las lecturas y escrituras se hacen a partir de él, en secuencia. Además, si escribimos pasado el final del fichero, su tamaño se amplía. Así, la estructura que tendrá nuestro fichero de películas será la siguiente:

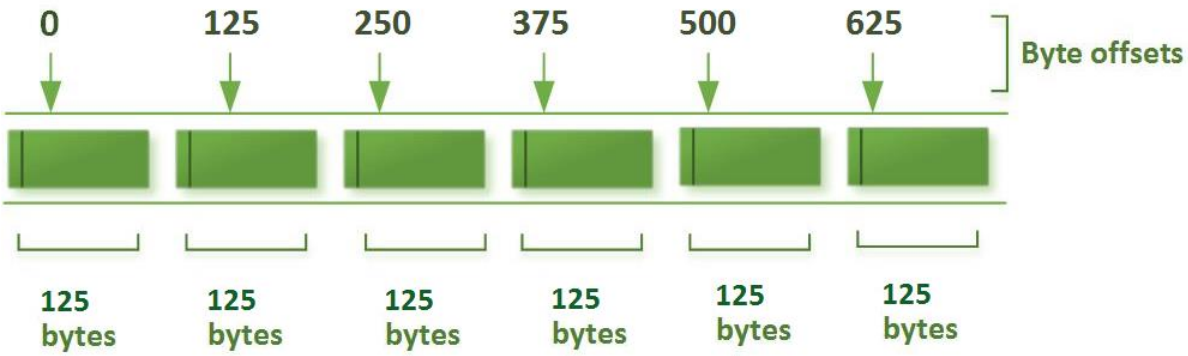

Figura 4.14 Estructura fichero películas.

Por último simplemente indicar, como ya he avanzado antes, que este fichero solamente se consultará para agilizar el tiempo de carga del modo de visualización correspondiente. Cuando solicitamos las recomendaciones para un grupo, el servidor nos devuelve únicamente los identificadores de las películas recomendadas, teniendo que volver a realizar otra petición para obtener su información. Con el uso del fichero, guardamos esa información para, en futuras consultas leerla de ahí y no tener que solicitarla al servicio de recomendaciones.

4.3. Diseño de la interfaz

Una vez realizado el diseño de la estructura de nuestra aplicación, llega el momento de diseñar la interfaz. Esta será la apariencia visual de la aplicación. Pero a la hora de diseñar una interfaz, hay que tener en cuenta la usabilidad.

La usabilidad se refiere a la facilidad con la que una persona puede usar o aprender a usar una herramienta fabricada por un ser humano. En interacción persona-ordenador se refiere al grado en el cual un usuario específico puede lograr un objetivo específico en un tiempo determinado.

Para obtener un bueno grado de usabilidad en nuestra interfaz, vamos a crear prototipos de las pantallas y StoryBoards para lograr un bueno diseño de interfaz.

4.3.1. Metáforas

Al igual que tradicionalmente usamos las metáforas para referirnos a conceptos abstractos en el uso del lenguaje, en informática usamos las metáforas para describir visualmente las tareas que la interfaz permite desarrollar.

Actualmente tienen un papel predominante en el diseño de interfaces debido a que permiten a los desarrolladores desarrollar programas que puedan ser usados por comunidades de usuarios muy diversos, al poder comunicar ideas, conceptos u objetos a través de su semejanza con otra cosa de forma familiar y accesible [32].

Las metáforas utilizadas en este proyecto utilizan unos iconos de diseño plano para no recargar demasiado la interfaz y son las siguientes:

Información

Esta metáfora aparece en la pantalla del menú principal, y permite desplegar una ventana con la información referente a la aplicación y a la autoría del proyecto.

Figura 4.15 Icono para la metáfora “Información”.

Configuración

Esta metáfora aparece en la pantalla del menú principal, y permite desplegar una ventana desde la cual el usuario podrá introducir diversos parámetros de configuración en la aplicación.

Figura 4.16 Icono para la metáfora "Configuración".

Volver

Esta metáfora aparece en los modos de visualización, y permite volver al menú principal de la aplicación en cualquier momento.

Figura 4.17 Icono para la metáfora "Volver".

Obteniendo Recomendaciones

Esta metáfora aparece una vez hemos pulsado en 'comenzar el proceso de visualización'. Se muestra mientras se solicitan las recomendaciones al servidor, se obtienen, y se crea el modelo de visualización.

Figura 4.18 Icono para la metáfora "Obteniendo Recoemndaciones".

Usuarios o Grupo

Esta metáfora aparece en el modo de visualización jerárquico. Sirve para obtener la visualización de las recomendaciones a nivel grupal si se pulsa sobre el símbolo de varias personas, o visualizar las recomendaciones de un solo usuario, si se pulsa sobre el símbolo de una sola persona.

Figura 4.19 Icono para la metáfora "Usuarios o grupos".

4.3.2. Prototipos de interfaz

En este apartado vamos a presentar los prototipos de las pantallas de navegación que tendrá nuestra aplicación. Estas pantallas son un esbozo de lo que será el sistema para el usuario final, por lo que no representan un diseño final.

Menú principal

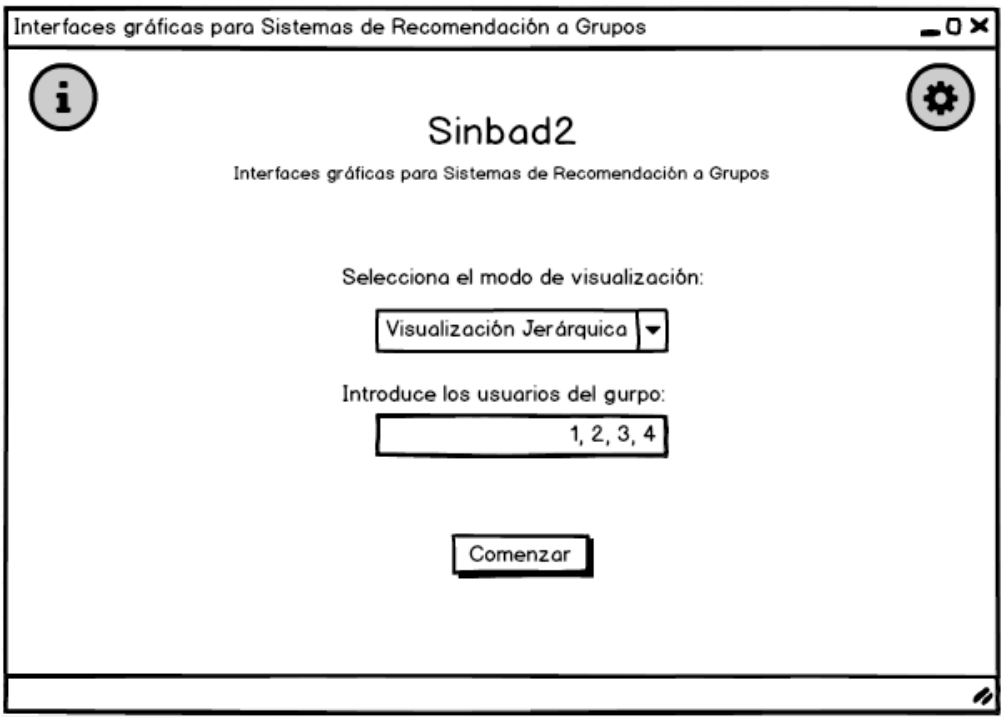

Figura 4.20 Pantalla Menú Principal.

En el menú principal el usuario debe seleccionar el modo de visualización que desea e indicar los usuarios que compondrán el grupo para el cual se obtendrán recomendaciones. De esta forma accederá al resto de la funcionalidad que ofrece la aplicación.

Configuración de la aplicación

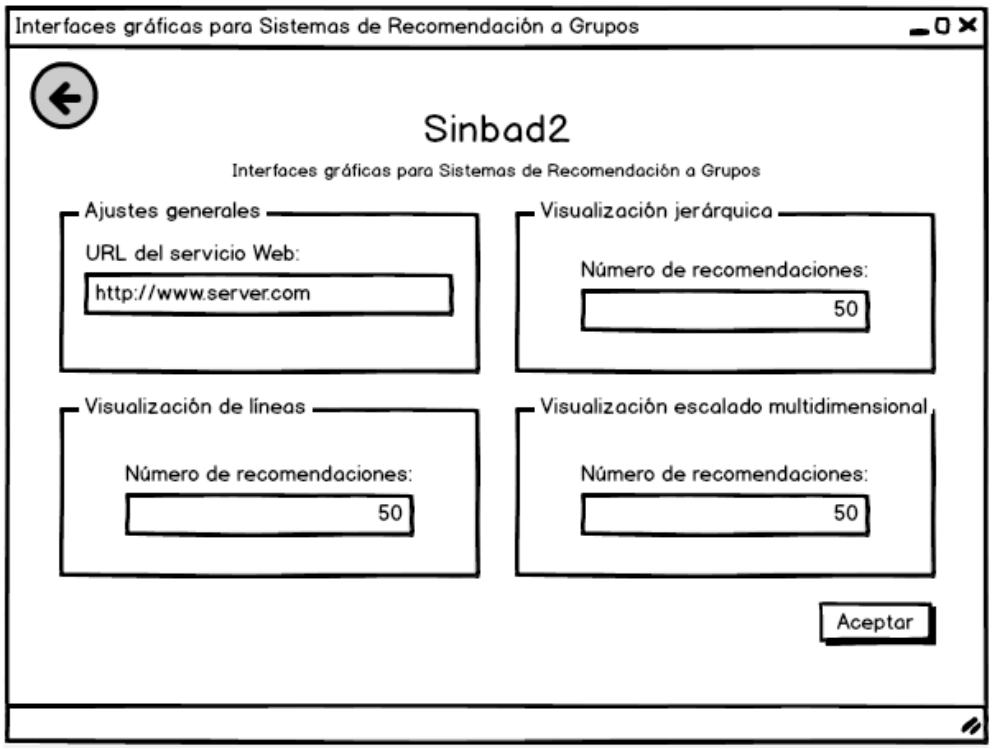

Figura 4.21 Pantalla Configuración.

Desde la pantalla de configuración de nuestra aplicación el usuario puede modificar diferentes parámetros.

El primero de ellos es la dirección o direcciones URL de los servicios web donde está alojado el Servidor de recomendaciones y mediante el cual obtendremos las recomendaciones a grupo.

El resto de parámetros configúrales son los que hacen referencia al número de recomendaciones que queremos obtener y que solicitaremos al servicio. El usuario podrá establecer un número de recomendaciones para cada uno de los modos de visualización disponibles. Si el usuario no realiza ninguna modificación, se mantendrán los valores por defecto prefijados en la instalación de la aplicación.

Petición y carga de las recomendaciones

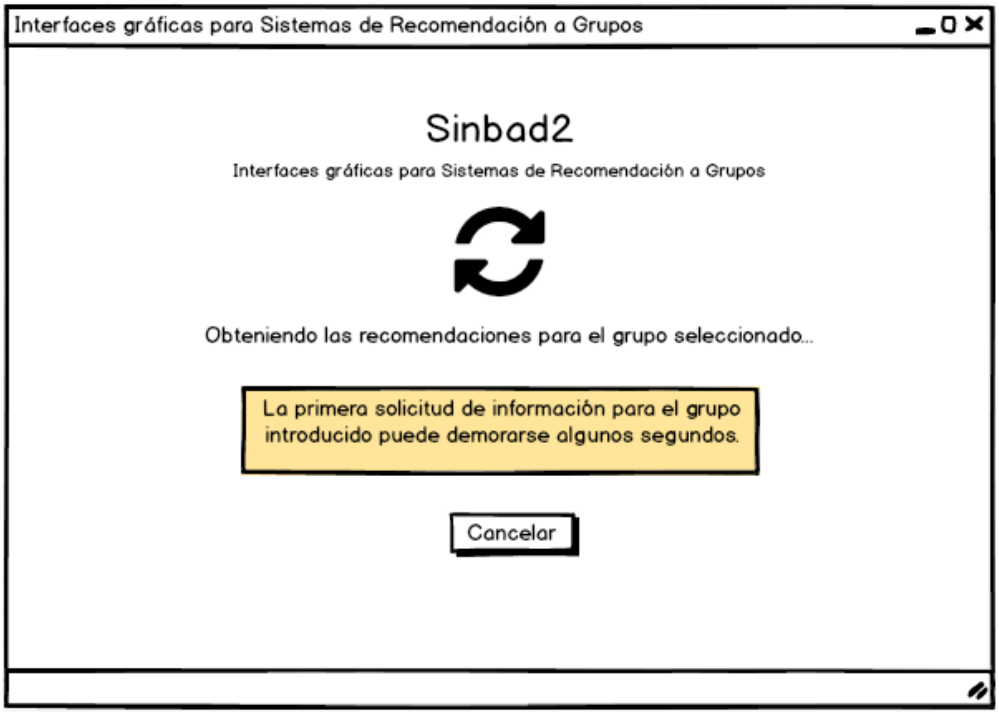

Figura 4.22 Pantalla Petición de recomendaciones.

Esta pantalla de carga aparecerá una vez seleccionado el modo de visualización y solicitado las recomendaciones. Mientras se muestra esta pantalla, se estarán recibiendo recomendaciones y creando el modelo de visualización seleccionado.

Modo de visualización Jerárquico

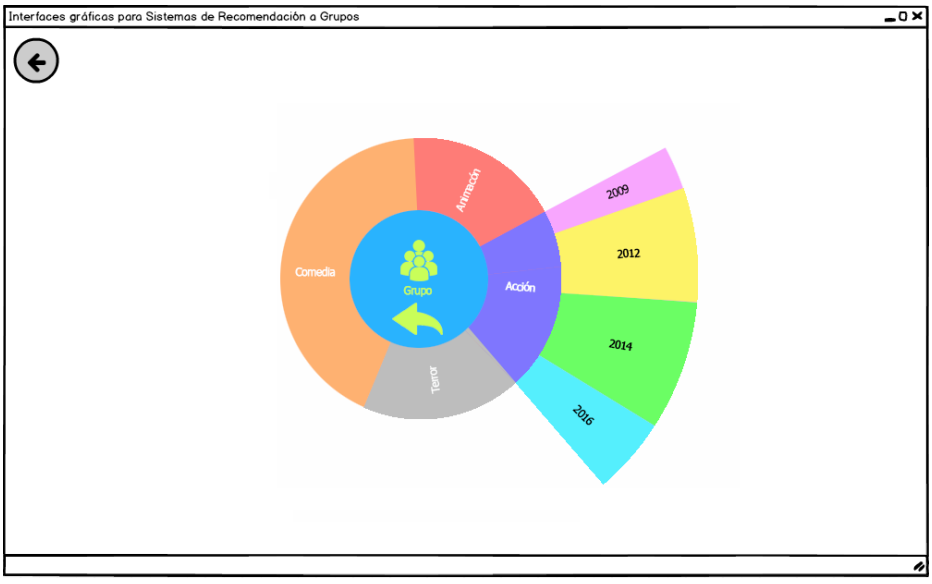

Figura 4.23 Pantalla Modo de visualización jerárquico.

En esta pantalla se mostrará el modo de visualización jerárquico. El usuario podrá interactuar con ella e ir viendo las recomendaciones recibidas ordenadas en categorías jerárquicas.

Modo de visualización Gráfico de Líneas

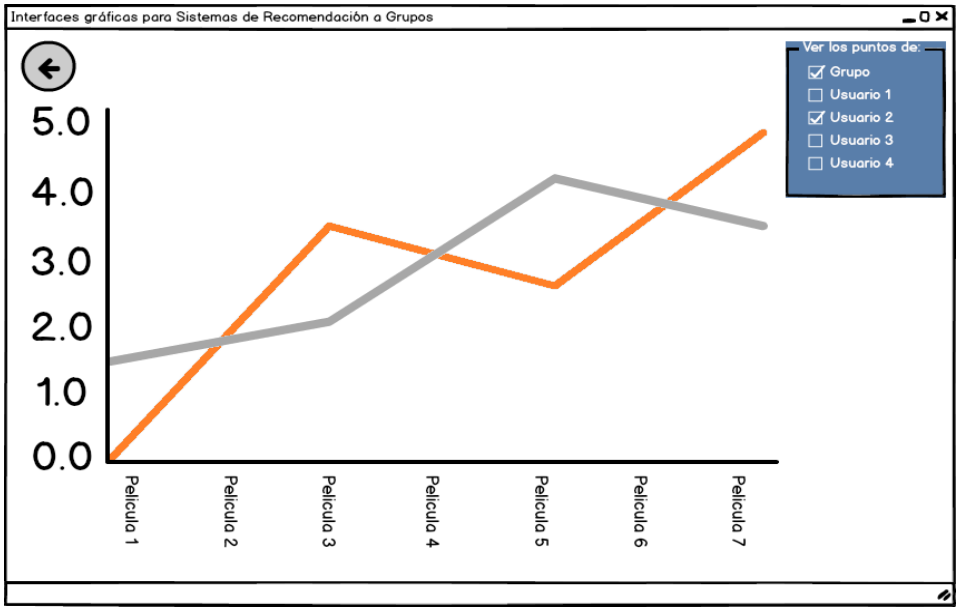

Figura 4.24 Pantalla Modo de visualización gráfico de líneas.

En esta pantalla se mostrará el modo de visualización gráfico de líneas. El usuario podrá interactuar con ella y seleccionar los datos que quiere o no ver.

Modo de visualización Escalado multidimensional

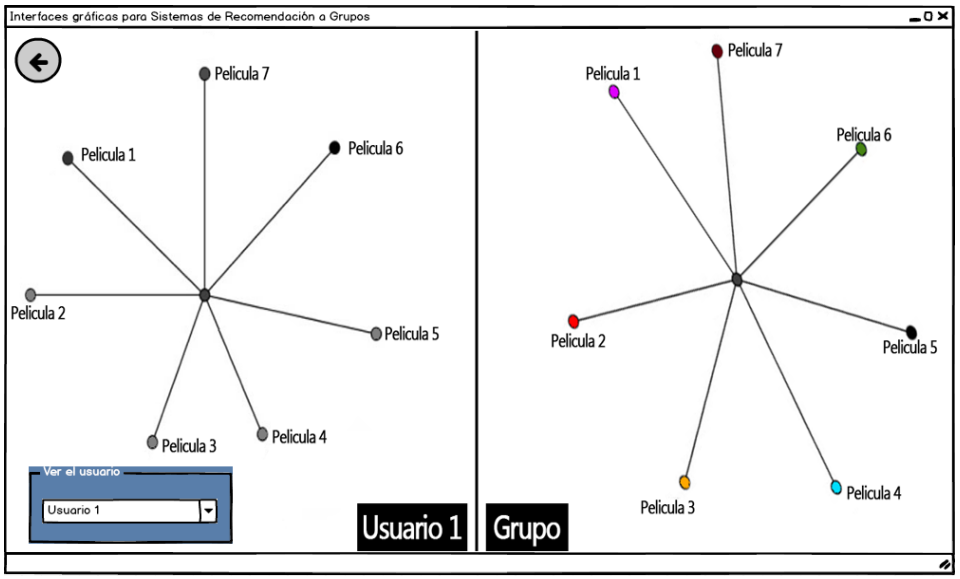

Figura 4.25 Pantalla Modo de visualización escalado multidimensional.

En esta pantalla se mostrará el modo de visualización por escalado multidimensional. El usuario podrá interactuar con ella y comparar las recomendaciones de los distintos usuarios del grupo.

4.3.3. StoryBoards

Para mostrar la interacción que se produciría al utilizar el usuario la aplicación, vamos a diseñar los “Storyboards”, que no son más que las herramientas que permiten mostrar, a modo de secuencia, las diferentes pantallas por las que el usuario va pasando al realizar cada una de las posibles acciones que la aplicación nos permite llevar a cabo [7].

Se disponen las capturas de pantalla de la interfaz unidas en una misma imagen y relacionadas mediante flechas, de tal manera que se indique así el camino que seguiría la interacción (el origen de cada flecha indica cuál es el elemento que desencadena la acción en la pantalla de origen, y el destino de la flecha, cuál sería el resultado de esa acción, es decir, la pantalla que se mostraría a continuación).

El storyboard servirá de prototipo para ser evaluado por el usuario o cliente que ha encargado la aplicación y de esa manera poder modificar así las correcciones en fases tempranas; con el fin de que las correcciones, en caso de haberlas, no sean costosas (en tiempo, trabajo y/o económicamente), ya que el coste de estas correcciones aumenta cuanto más tarde se hagan.

Los storyboards están bastante relacionados con los casos de uso y, se han intentado agrupar de una manera comprensible y que incluya cada una de las acciones que puede llevar a cabo un usuario en nuestra aplicación. Son los siguientes:

- Storyboard de Acceso y Uso (configuración).
- Storyboard de obtención de Recomendaciones.
- Storyboard de Selección del modo de visualización.
- StoryBoard de Modo de Visualización Jerárquico
- Storyboard de Modo de Visualización Gráfico de Líneas.
- StoryBoard de Modo de Visualización Escalado Multidimensional

Storyboard de Acceso y Uso (Configuración). [Caso de Uso #1]

Para acceder y usar la aplicación (véase figura 4.8):

0. El usuario abre la aplicación desde el ejecutable.
1. El usuario hace clic en el botón de configuración.
2. El usuario introduce la URL de los servicios de obtención de recomendaciones a grupo.
3. El usuario hace clic en Aceptar.

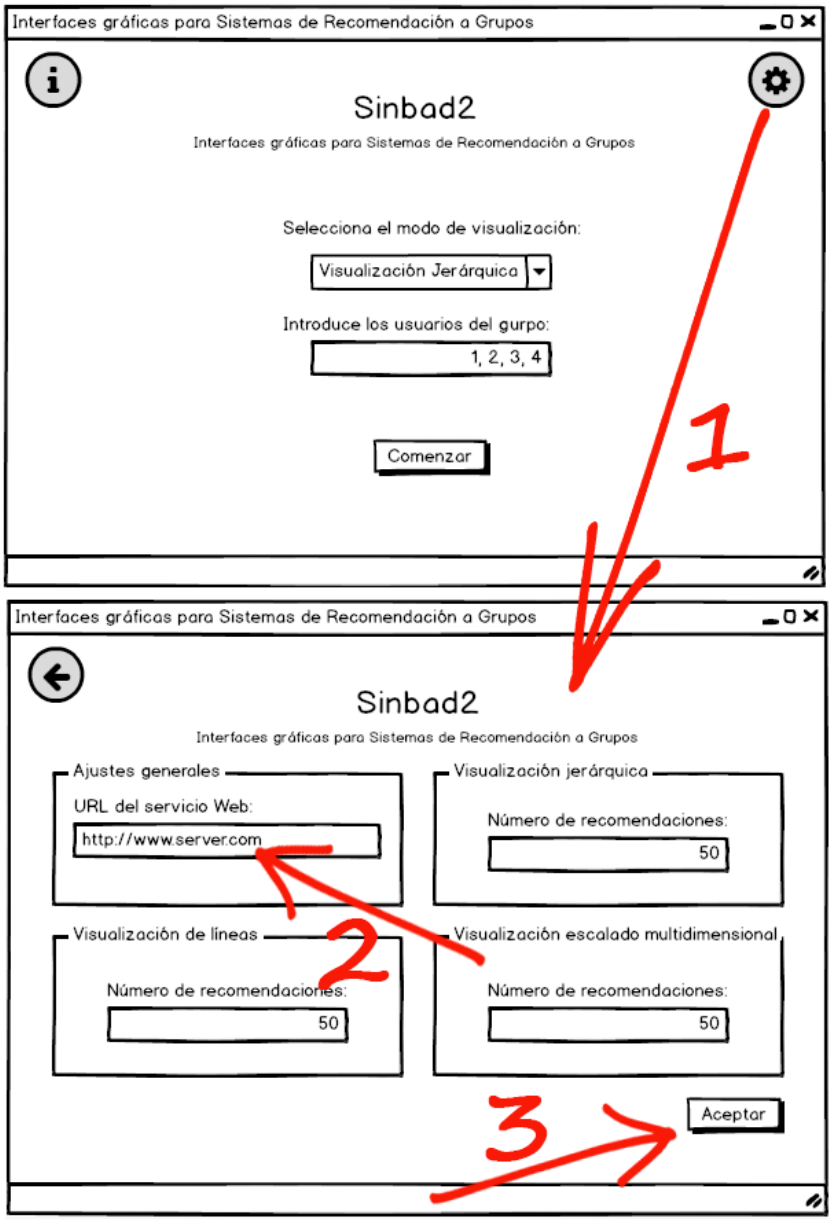

Figura 4.26 StoryBoard de Acceso y Uso.

StoryBoard de obtención de Recomendaciones. [Caso de Uso #2]

Para poder configurar y obtener las recomendaciones a grupo (véase figura 4.9):

1. El usuario hace clic en el botón de configuración.
2. El usuario configura el número de recomendaciones que quiere obtener para cada uno de los modos de visualización disponibles. (tienen un valor por defecto).
3. El usuario hace clic en Aceptar.
4. El usuario introduce los números (identificadores) de los usuarios que forman parte del grupo para el que se generarán recomendaciones.
5. El usuario hace clic en Comenzar.

5. b. Si alguno de los usuarios no existe, o se produce algún error que impide obtener las recomendaciones, se nos muestra un mensaje de error.

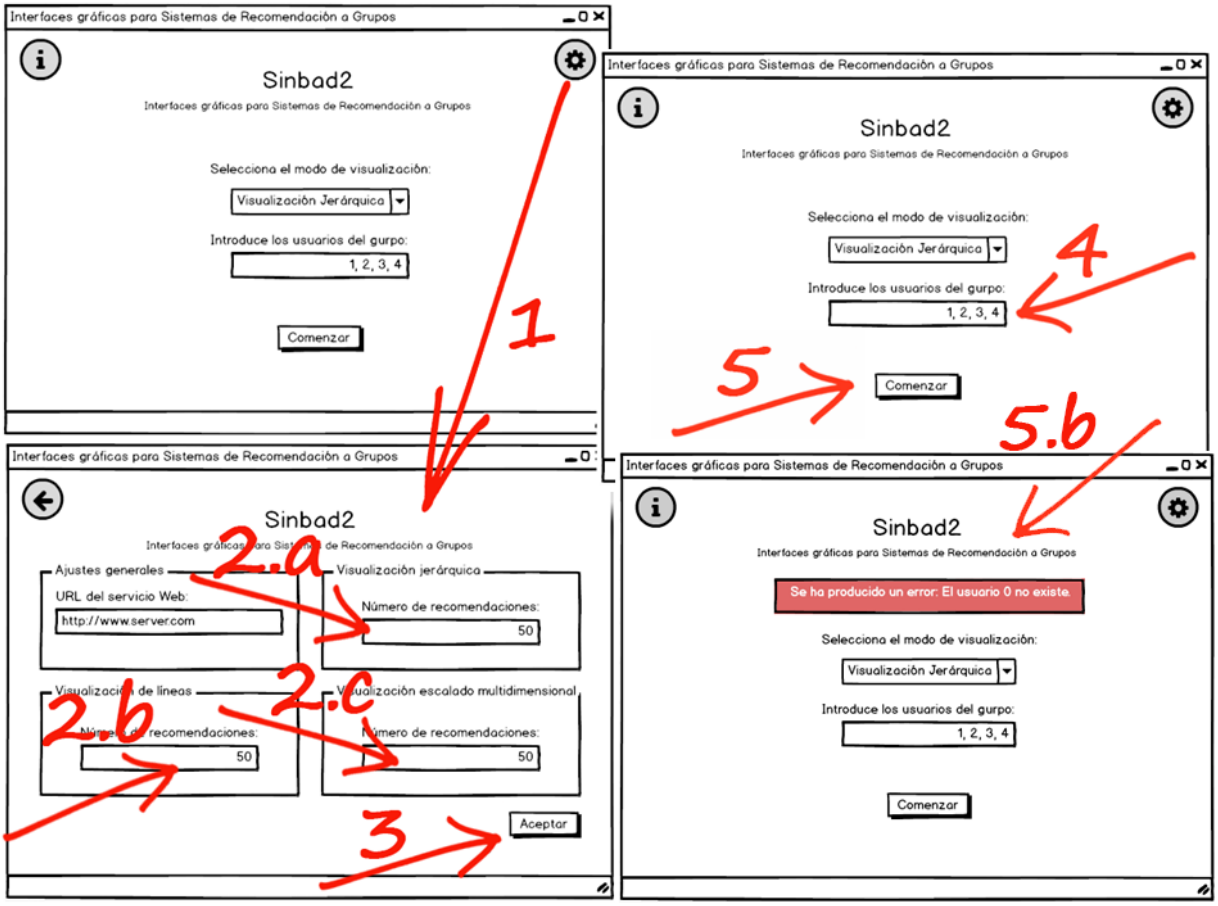

Figura 4.27 StoryBoard de obtención de recomendaciones.

Storyboard de Selección del modo de visualización. [Caso de Uso #3]

Para seleccionar el modo de visualización con el que queremos representar las recomendaciones que obtendremos (véase figura 4.10):

1. El usuario hace clic en el desplegable de selección del modo de visualización.
2. El usuario hace clic en uno de los modos de visualización disponibles que aparecen.
3. El usuario hace clic en comenzar.

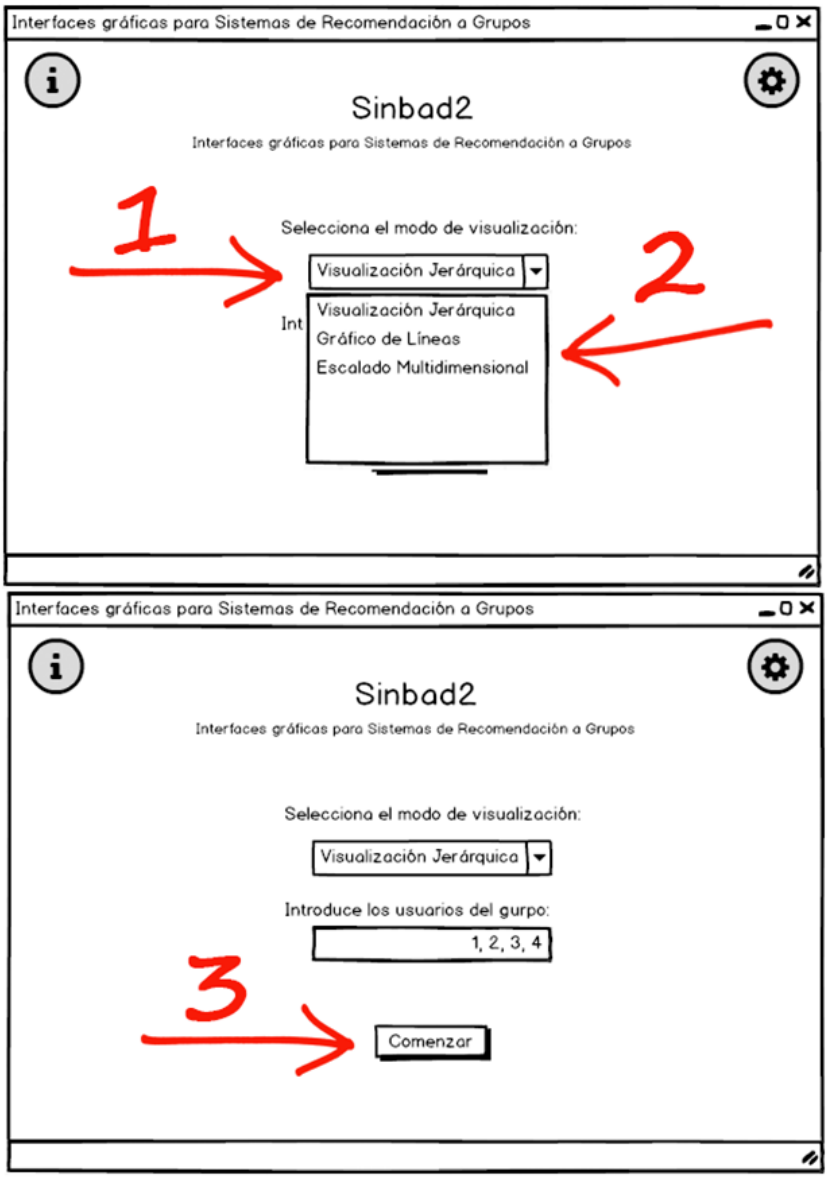

Figura 4.28 StoryBoard de selección de método de visualización.

StoryBoard de Modo de Visualización Jerárquico. [Caso de Uso #4]

Para visualizar las recomendaciones del grupo mediante el modo de visualización jerárquico (véase figura 4.11):

1. El usuario selecciona el modo de visualización jerárquico.
2. El usuario hace clic en comenzar.
3. Se muestra un panel de espera mientras se obtienes las recomendaciones y se crea el modelo de visualización.
4. a. Se muestra el modo de visualización jerárquico.

4. b. Si El usuario cancela el proceso, vuelve al menú principal.

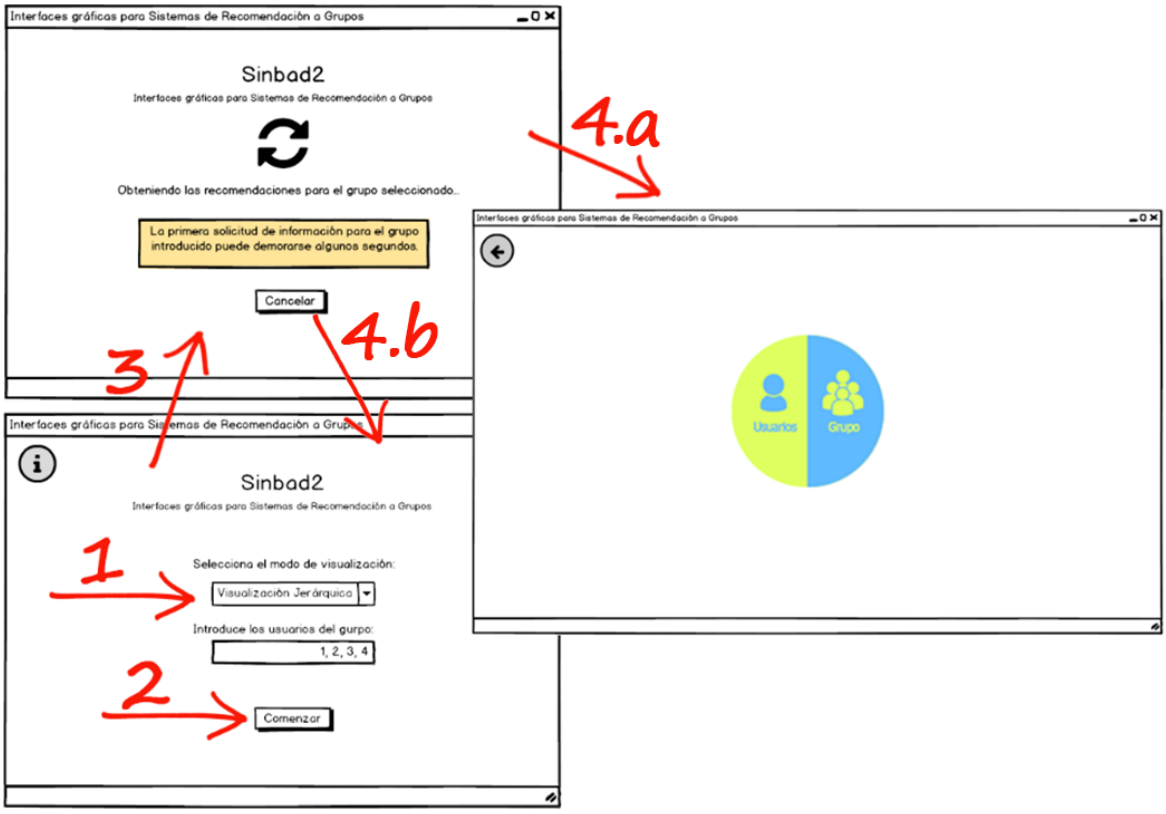

Figura 4.29 StoryBoard de Modo de visualización jerárquico.

Para navegar por el modo de visualización jerárquico (véase figura 4.12):

1. a. El usuario hace clic en “Grupo” y accede a la visualización para las recomendaciones grupales.

1. b. El usuario hace clic en “Usuario” y accede a la visualización para las recomendaciones individuales de un miembro del grupo.

2. [Usuario] El usuario que miembro del grupo quiere ver.

3. 4. El usuario navega por los distintos niveles de clasificación, el género de la película, el año y, por último, las películas.

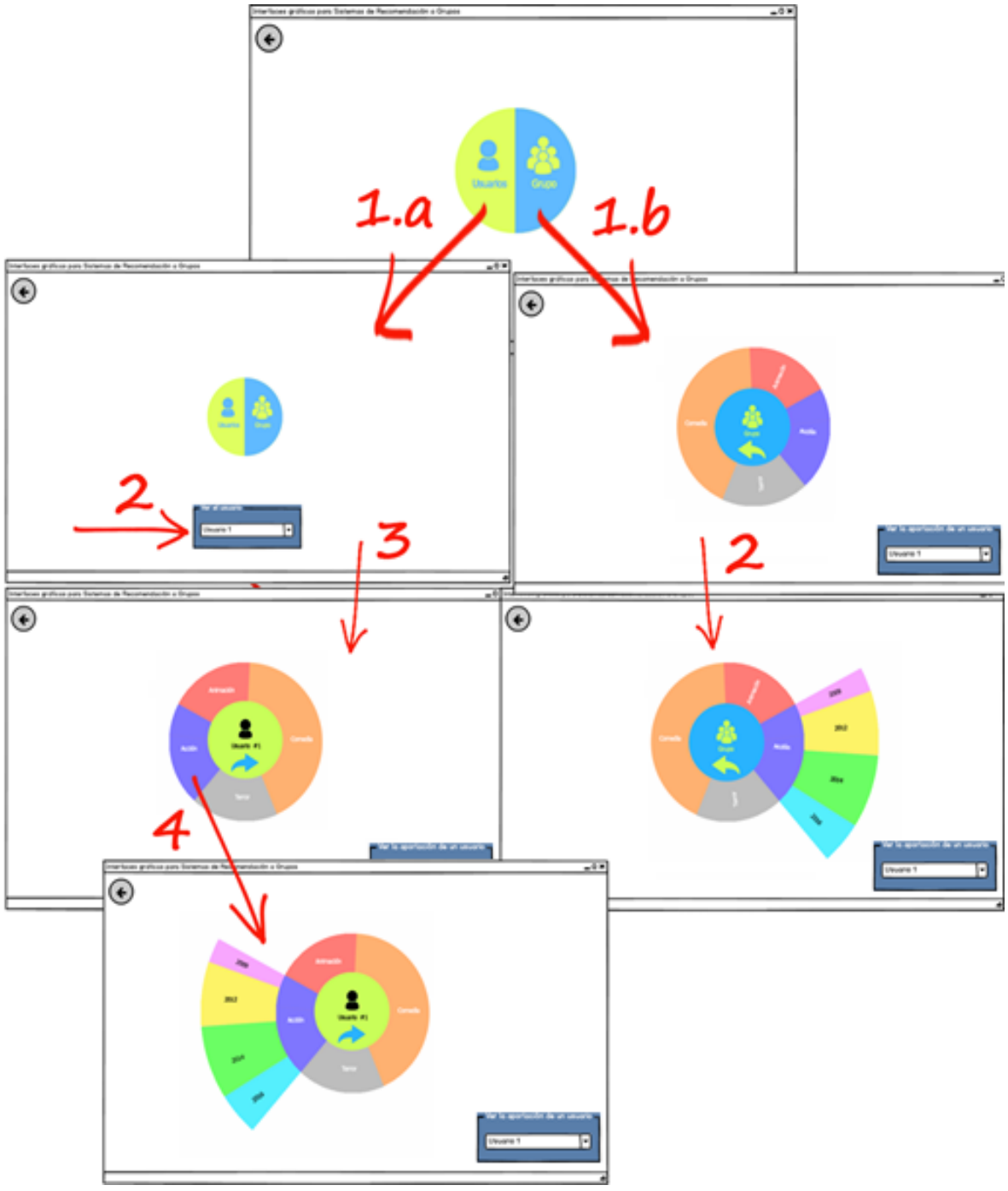

Figura 4.30 StoryBoard de Navegación jerárquica por categorias para usuario y grupo.

Para ver la información de una película desde el modo de visualización jerárquico (véase figura 4.13):

1. El usuario navega hasta el último nivel de clasificación, el nivel de las películas.
2. El usuario hace clic en una película de las disponibles.
3. Se muestra un panel con datos relevantes de la película (cartel, actores, director, etc.).

Figura 4.31 StoryBoard de ver información de una película en la visualización jerárquica.

Para ver la aportación de un usuario a la recomendación grupal, o la influencia del grupo en una recomendación individual desde modo de visualización jerárquico (véase figura 4.14):

1. El usuario navega hasta el nivel que quiere comparar.
2. El usuario selecciona el usuario con el que quiere comparar (o el grupo).
3. Se muestra la cobertura (aportación del usuario o grupo) sobre la información original.
4. Para ocultar la aportación, el usuario hace clic en “ocultar aportación”.

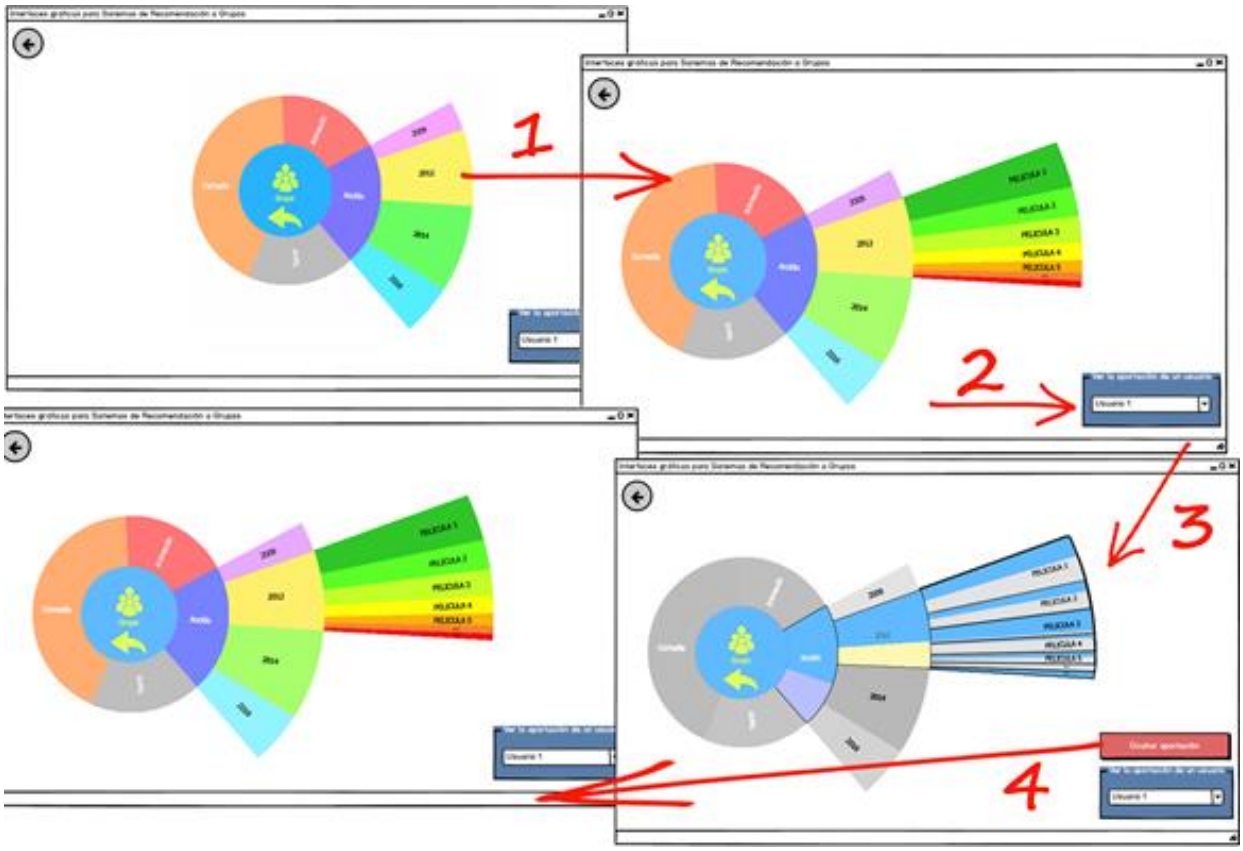

Figura 4.32 StoryBoard de ver la aportación de un miembro del grupo a la recomendación grupal, y viceversa.

StoryBoard de Modo de Visualización Gráfico de Líneas. [Caso de Uso #5]

Para visualizar las recomendaciones del grupo mediante el modo de visualización en gráfico de líneas (véase figura 4.15):

1. El usuario selecciona el modo de visualización ‘Gráfico de Líneas’.
2. El usuario hace clic en comenzar.
3. Se muestra un panel de espera mientras se obtienes las recomendaciones y se crea el modelo de visualización.
4. a. Se muestra el modo de visualización por gráfico de líneas.

4. b. Si El usuario cancela el proceso, vuelve al menú principal.
5. El usuario puede seleccionar las líneas de que miembro del grupo quiere que aparezcan o desaparezcan en la visualización (la línea del grupo también se puede añadir/ocultar).
6. El modo de visualización se actualiza automáticamente.

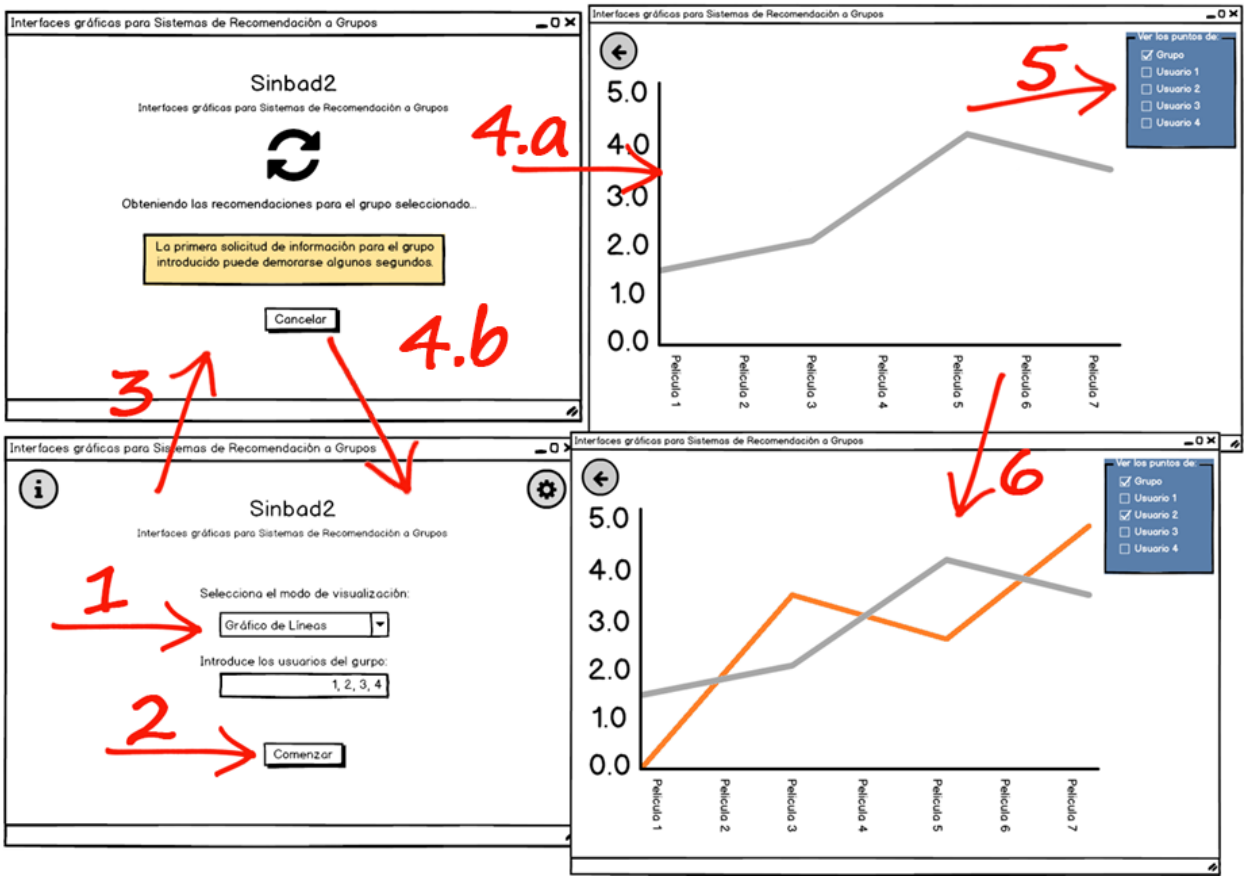

Figura 4.33 StoryBoard de Modo de visualización gráfico de líneas.

StoryBoard de Modo de Visualización Escalado Multidimensional. [Caso de Uso #6]

Para visualizar las recomendaciones del grupo mediante el modo de visualización de escalado multidimensional (véase figura 4.16):

1. El usuario selecciona el modo de visualización por escalado multidimensional.
2. El usuario hace clic en comenzar.
3. Se muestra un panel de espera mientras se obtienes las recomendaciones y se crea el modelo de visualización.
4. a. Se muestra el modo de visualización mediante escalado multidimensional.

4. b. Si El usuario cancela el proceso, vuelve al menú principal.

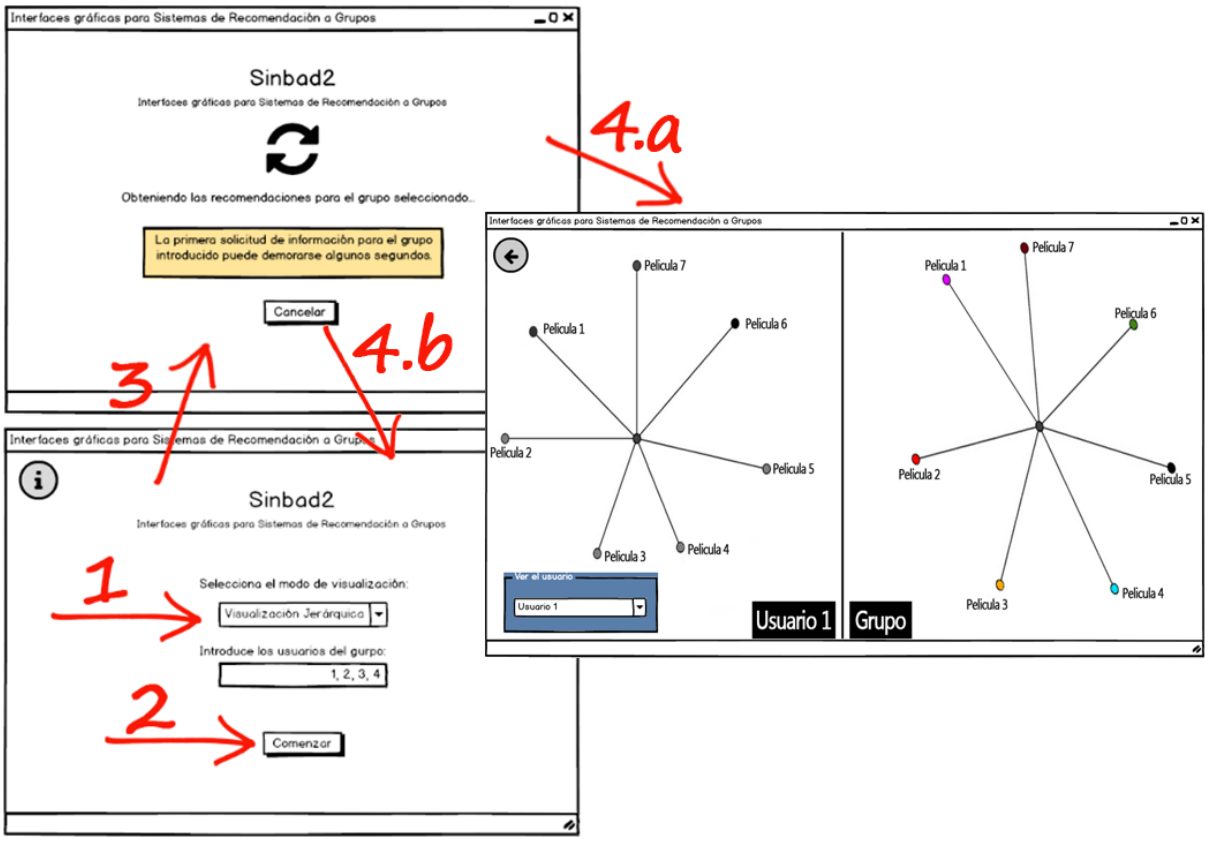

Figura 4.34 StoryBoard de Modo de visualización por escaldo multidimensional.

Para cambiar el usuario (sus recomendaciones) que se muestran junto a las grupales en el modo de visualización Escalado multidimensional (véase figura 4.17):

1. El usuario hace clic en el desplegable de la parte inferior y selecciona el miembro del grupo que quiere comparar con la representación de las recomendaciones grupales.
2. La visualización se actualiza con el miembro seleccionado.

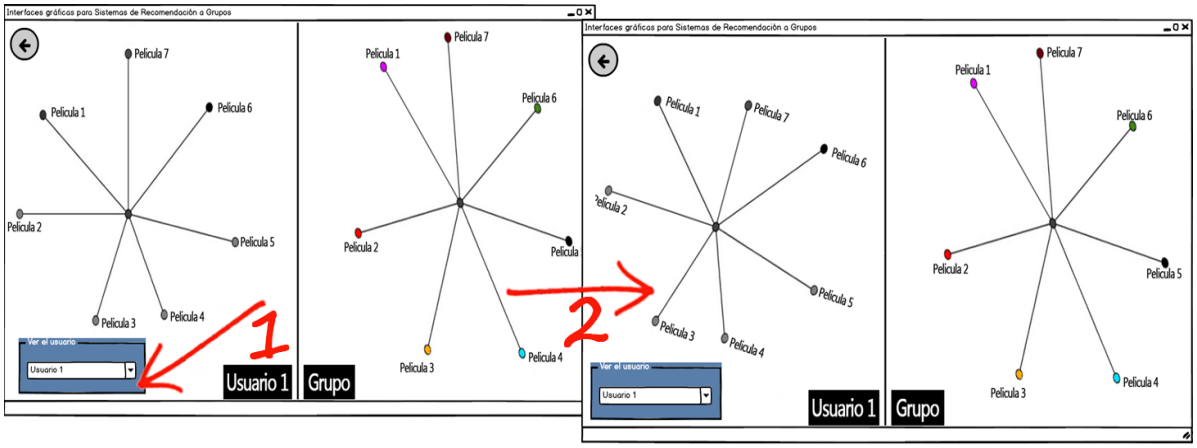

Figura 4.35 StoryBoard de Modo de visualización por escaldo multidimensional.

Para ver la información de una película desde el modo de visualización por escalado multidimensional (véase figura 4.18):

1. El usuario hace clic en uno de los puntos que representan a las películas.
2. Se muestra un panel con datos relevantes de la película.

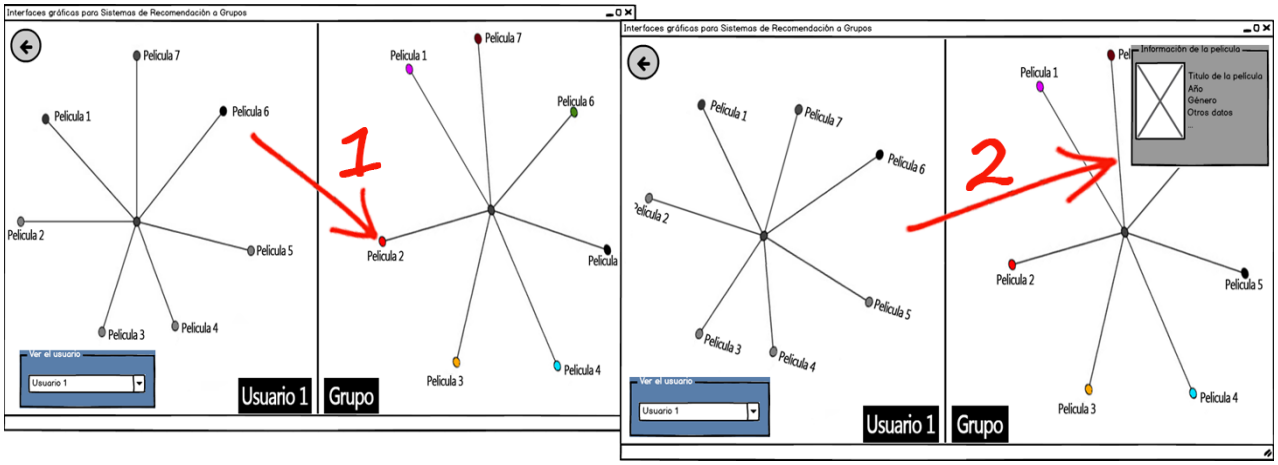

Figura 4.36 StoryBoard de ver información de una película en la visualización por escalado multidimensional.

4.4. Implementación

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consisten, esencialmente, en llevar a código fuente mediante el lenguaje de programación elegido, todo lo diseñado en la fase anterior, a través del uso de herramientas de desarrollo.

Además necesitaremos establecer la arquitectura y el conjunto de tecnologías que utilizaremos para realizar esta implementación.

La arquitectura es la estructura operacional de nuestro sistema, y, en este caso, utilizaremos una arquitectura cliente-servidor, que es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos, llamados servidores, y el demandante, llamado cliente.

En cuanto a las tecnologías, el lenguaje de programación elegido será Java, y se utilizará la librería gráfica OpenGL, en su versión para dicho lenguaje: JOGL. A continuación se describen detalladamente cada una de estas tecnologías.

4.4.1. Lenguajes de Programación

Al ser una aplicación de visualización gráfica, hemos optado por desarrollarla mediante el uso de la librería **OpenGL**, utilizada para escribir aplicaciones que produzcan gráficos 2D y 3D.

Por otra parte, gracias a las ventajas para desarrollar interfaces gráficas para aplicaciones de escritorio que nos ofrece Java con **Swing**, hemos decido utilizar este lenguaje de programación para codificar nuestra aplicación. Por ello, finalmente, hemos desarrollado el proyecto utilizando **JOGL**, que es una biblioteca que permite acceder a OpenGL mediante programación en Java.

Java es un lenguaje compilado e interpretado. Todo programa en Java ha de compilarse y el código que se genera bytecodes es interpretado por una máquina virtual. De este modo se consigue la independencia de la máquina, el código compilado se ejecuta en máquinas virtuales que si son dependientes de la plataforma. Java es un lenguaje orientado a objetos de propósito general.

Aunque Java comenzará a ser conocido como un lenguaje de programación de applets que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto. Su sintaxis es similar a la de C y C++ pero hasta ahí llega el parecido. Java no es una evolución ni de C++ ni un C++ mejorado.

Podemos decir que el motivo de ser tan conocido e innovador es que tiene una filosofía única: "escribe una vez y ejecuta en cualquier lugar" ("write once, run everywhere"). En otras palabras, lo programas una vez y lo ejecutas en cualquier sistema. Además, Java se creó con cinco objetivos principales:

1. Debería utilizar el paradigma de programación orientada a objetos.

2. Permitir ejecutar un mismo programa en múltiples sistemas operativos.

3. Incluir por defecto soporte para trabajar en red.

4. Ser diseñado para ejecutar código en sistemas remotos de manera fiable.

5. Ser fácil de utilizar y utilizar las mejores partes de otros lenguajes como C++.

Esa filosofía es la que hizo a Java un mito, pero además Java tiene algunas características que lo hacen muy útil: es de los lenguajes más fuertes, con mayor cantidad de librerías para hacer cualquier tipo de cosa y con una de las mayores comunidades.

Y entre sus principales características se encuentran:

Es simple: Basado en el lenguaje C++ pero donde se eliminan muchas de las características Orientadas a objetos que se utilizan esporádicamente y que creaban frecuentes problemas a los programadores. Esta eliminación de causas de error y problemas de mantenimiento facilita y reduce el coste del desarrollo de software.

Es orientado a objetos: Java da buen soporte a las técnicas de desarrollo OOP y en resumen a la reutilización de componentes de software.

Es distribuido: Java se ha diseñado para trabajar en ambiente de redes y contienen una gran biblioteca de clases para la utilización del protocolo TCP/IP, incluyendo HTTP y FTP. El código Java se puede manipular a través de recursos URL con la misma facilidad que C y C++ utilizan recursos locales (archivos).

Es interpretado: El compilador Java traduce cada fichero fuente de clases a código de bytes (Bytecode), que puede ser interpretado por todas las máquinas que den soporte a un visualizador de que funcione con Java. Este Bytecode no es específico de una máquina determinada, por lo que no se compila y enlaza como en el ciclo clásico, sino que se interpreta.

Es sólido: El código Java no se quiebra fácilmente ante errores de programación. Así el relaje que existe en la declaración y manejo de tipos en C y C++ se torna en restricciones en Java, donde no es posible la conversión forzada (cast) de enteros en punteros y no ofrece soporte a los punteros que permitan saltarse reglas de manejo de tipos. Así en Java no es posible escribir en áreas arbitrarias de memoria ni realizar operaciones que corrompan el código. En resumen se eliminan muchas de las posibilidades de "trucos" que ofrecían C y C++.

Es seguro: Como Java suele funcionar en ambientes de redes el tema de seguridad debe interesar en sobremanera. Las mismas características antes

descritas que evitan la corrupción de código evitan su manipulación. Actualmente se está trabajando en encriptar el código.

Tiene una arquitectura neutral: El compilador crea códigos de byte (Bytecode) que se envía al visualizador solicitado y se interpreta en la máquina que posee un intérprete de Java o dispone de un visualizador que funciona con Java.

Es portable: Al ser de arquitectura neutral es altamente portable, pero esta característica puede verse de otra manera: Los tipos estándares (int, float...) están igualmente implementados en todas las máquinas por lo que las operaciones aritméticas funcionaran igual en todas las máquinas.

Alto desempeño: al ser código interpretado, la ejecución no es tan rápida como el código compilado para una plataforma particular. El compilador Java suele ofrecer la posibilidad de compilar Bytecode en código máquina de determinadas plataformas, y según Sun este código resultar de una eficacia similar a compilaciones de C y C++.

Permite multihilos: Java puede aplicarse a la realización de aplicaciones en las que ocurra más de una cosa a la vez. Java, apoyándose en un sistema de gestión de eventos basado en el paradigma de condición y monitores C.A.R. permite apoyar la conducta en tiempo real e interactiva en programas

Es dinámico: al contrario que C++ que exige se compile de nuevo la aplicación al cambiar una clase madre Java utiliza un sistema de interfaces que permite aligerar esta dependencia. Como resultado, los programas Java pueden permitir nuevos métodos y variables en un objeto de biblioteca sin afectar a los objetos dependientes.

4.4.2. OpenGl y JOGL

OpenGL es una librería gráfica escrita originalmente en C que permite la manipulación de gráficos 3D a todos los niveles [13]. Esta librería se concibió para programar en máquinas nativas Silicon Graphics bajo el nombre de GL (Graphics Library). Posteriormente se consideró la posibilidad de extenderla a cualquier tipo de

plataforma y asegurar así su portabilidad y extensibilidad de uso con lo que se llegó al término Open Graphics Library, es decir, OpenGL.

La librería se ejecuta a la par con nuestro programa independientemente de la capacidad gráfica de la máquina que usamos. Esto significa que la ejecución se dará por software a no ser que contemos con hardware gráfico específico en nuestra máquina. Si contamos con tarjetas aceleradoras de vídeo, tecnología MMX, aceleradoras 3D, pipelines gráficos implementados en placa, etc. por supuesto gozaremos de una ejecución muchísimo más rápida en tiempo real.

Así esta librería puede usarse bajo todo tipo de sistemas operativos e incluso usando una gran variedad de lenguajes de programación. Podemos encontrar variantes de OpenGL para Windows, Unix, Linux, Iris, Solaris, Delphi, Java e incluso Visual Basic. No obstante, su uso más extenso suele ser el lenguaje C o C++.

JOGL (Java OpenGL) es una biblioteca que permite acceder a OpenGL mediante programación en Java. Actualmente está siendo desarrollado por el Game Technology Group de Sun Microsystems, y es la implementación de referencia para Java Bindigs for OpenGL. JOGL permite acceder a la mayoría de características disponibles para los programadores de C, con la excepción de las llamadas a ventanas realizadas en GLUT ya que Java contiene sus propios sistemas de ventanas (AWT y Swing), y algunas extensiones de OpenGL.

La API OpenGL, escrita en C, es llamada por JOGL gracias a la Java Native Interface (JNI). Por tanto, el sistema en el que se está programando debe tener soporte para OpenGL para que pueda funcionar JOGL correctamente.

JOGL se diferencia de otras bibliotecas Java para OpenGL en que simplemente expone las funciones de la OpenGL, basadas en un lenguaje procedural (lenguaje C), por medio de métodos contenidos en unas pocas clases, en lugar de intentar realizar un mapeo completo del código OpenGL para transformarlo y adaptarlo al paradigma de orientación a objetos. De hecho, la mayoría del código de JOGL está en realidad autogenerado a partir de las cabeceras de las bibliotecas C de OpenGL, mediante una herramienta llamada Gluegen, que fue programada específicamente para dicho propósito.

Esta decisión en el diseño tiene sus ventajas y sus desventajas. La naturaleza procedural y de máquina de estados de OpenGL es inconsistente con la forma habitual de programar en Java, lo cual puede dejar perplejos a muchos programadores. Sin embargo, la conversión directa realizada de las funciones OpenGL a métodos Java, permite la conversión del código de aplicaciones C ya existentes mucho más simple. La fina capa de abstracción proporcionada por JOGL hace que la ejecución sea muy eficiente, aunque resulta mucho más difícil de programar que otras bibliotecas de mucho más alto nivel como Java3D. Dado que la mayoría del código está autogenerado, los cambios que se produzcan en OpenGL son rápidamente adaptados a JOGL.

En 2007, JOGL proporciona acceso completo a la especificación 2.0 de OpenGL.

Por último, explicaremos cómo está conformada la estructura de un programa escrito en Java utilizando librerías de JOGL. Como trabajaremos con gráficos de OpenGL será necesario implementar la interfaz conocida con el nombre de GLEventListener además de extender de la clase JFrame o JPanel debido a que debemos montar los gráficos en un canvas de la interfaz de usuario.

La interfaz GLEventListener declara eventos los cuales son utilizados por el código cliente para manipular el renderizado de OpenGL a través de GLAutodrawable, (algo parecido a cuando se implementa la interfaz ActionListener para escuchar los eventos de los botones los cuales son manipulados por el código cliente a través de ActionEvent).

La interfaz GLEventListener utiliza cuatro métodos principales los cuales será necesario declarar dentro de la estructura de nuestro programa para que éste funcione:

1 - init(GLAutodrawable drawable)

Este método es llamado por drawable inmediatamente después de que el contexto de OpenGL (procedimiento que realizará un objeto GLCanvas) es inicializado. Puede ser utilizado para la inicialización de los gráficos de OpenGL que

GLCanvas utilizará tales como el color de fondo, color de los objetos que se dibujarán, luces que se manejarán, etc.

2 - reshape(GLAutodrawable drawable, int x, int y, int width, int height)

Este método es llamado por drawable durante el primer redibujado (repaint) y después de que el componente se redimensiona (resize). El cliente puede actualizar el punto de vista (viewport) de los gráficos apropiadamente.

3 - display(GLAutodrawable drawable)

Este método es llamado por drawable para iniciar el renderizado de OpenGL a petición del cliente. Dentro de este método se incluirán los gráficos que GLCanvas dibujará y será llamado cada vez que se le solicite, o bien, cuando todos los GLEventListeners hayan sido notificados de que ocurrió algún evento.

4 - displayChanged(GLAutodrawable, boolean modeChanged, boolean deviceChanged)

Este método es llamado por drawable cuando ocurre algún cambio en el visualizador (pantalla) asociado con GLAutoDrawable. Los dos parámetros de tipo boolean indican el tipo de cambio que ha ocurrido.

4.4.3. Servicio de recomendaciones

En los puntos anteriores hemos visto la estructura que tendrán los datos para ser almacenados en nuestra aplicación, pero nos quedan aún por resolver de qué manera accederá dicha aplicación a esos datos. Tal como se describió en capítulos anteriores, será a través de una Servicio Web, Aportan la interoperabilidad necesaria entre varias aplicaciones independientemente de sus propiedades o de las plataformas sobre las que estén instaladas. De esta manera, los cambios a lo largo del tiempo en uno no deben repercutir en el otro. A esta flexibilidad se le dará cada vez más importancia, debido a que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada vez más común.

Esta independencia entre datos se puede ver como una API, que es un mecanismo de comunicación entre componentes software, proporcionando un conjunto de funciones para acceder a los servicios obviando la complejidad de un sistema o aplicación.

La justificación del uso de esta tecnología reside en varios aspectos, como por ejemplo que se basan en estándares y protocolos basados en texto, que permiten acceder a su contenido y entender su funcionamiento de una manera más fácil, o por otro lado que permiten que servicios y aplicaciones ubicados en diferentes lugares geográficos puedan ser combinados entre ellos para proveer servicios integrados fácilmente.

En la siguiente tabla se definen las llamadas al servicio, de la siguiente manera:

- URI: Identificador Recursos Uniforme, el nombre de la función.
- Método: Método HTTP utilizado para la llamada al servicio.
- Parámetros: Datos adicionales necesarios para la llamada.
- Definición: Breve descripción del funcionamiento de la llamada.

URI	Método	Parámetros	Definición
/Recommendation/Group/Recommend/:users	GET	:users	Genera las recomendaciones para el grupo :users
/Recommendation/RecommendToIndividual/Recommend/:user	GET	:user	Devuelve las recomendaciones para el usuario :user

Tabla 4.7. Llamadas al servicio web de recomendaciones.

4.4.4. Herramientas de desarrollo

Para el desarrollo de la aplicación Java se ha empleado el entorno de desarrollo NetBeans en su versión 8.0.2.

Por otro lado se han utilizado una serie de librerías externas para aportarnos diversas funcionalidades:

- **JOGL**: Librería que nos permite acceder a los componentes de OpenGL mediante instrucciones en lenguaje Java.
- **Json Reader**: librería que nos permite des-serializar las respuestas JSON del servidor, para permitir leerlas más fácilmente.
- **MDSJ**: Librería que realiza las funciones de escalado multidimensional, generando puntos en 2D a partir de ítems multidimensionales.

Posteriormente, se dispondrán unos anexos para explicar cómo instalar las herramientas necesarias para el correcto funcionamiento de la aplicación.

4.4.5. Arquitectura del sistema

En este caso tenemos un tipo de arquitectura Cliente-Servidor. El rol de servidor lo llevarán a cabo tanto el servicio web de recomendaciones, como la API de información de peliculas. Por su parte, el rol de cliente/interfaz lo llevará a cabo nuestra aplicación de escritorio.

Vamos a intentar explicarlo un poco mediante la siguiente figura:

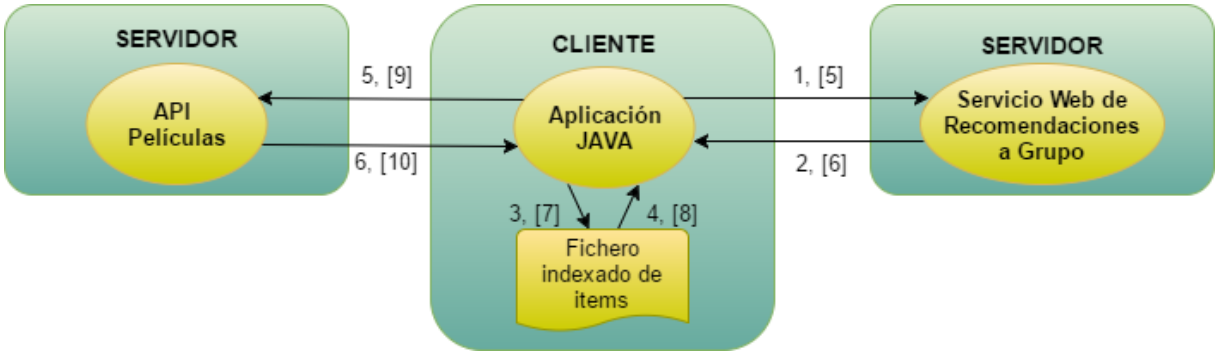

Figura 4.37 Arquitectura Cliente-Servidor de la aplicación.

1. La aplicación envía una petición, mediante URL, al servicio web de recomendaciones, con los datos de los usuarios que forman el grupo.
2. El servicio web recibe la petición, calcula las recomendaciones para el grupo indicado, y devuelve los identificadores de las películas recomendadas.
3. La aplicación consulta en un fichero indexado local, para cada película, si dispone de la información de clasificación para ella.
4. Si no dispone de la información deberá solicitarla al servicio web.
[5]. La aplicación envía una petición, mediante URL, al servicio web de recomendaciones, con el identificador de la película.
[6]. El servicio web recibe la petición y devuelve la información de clasificación de la película.
[7]. La aplicación almacena la información de manera indexada para futuras consultas.
[8]. La aplicación obtiene toda la información de clasificación para todas las películas recomendadas, y la visualiza según el modo de visualización elegido.
[9]. Una vez visualizada, al seleccionar una película en la visualización, la aplicación envía una petición, mediante URL, a una API de datos de películas para mostrar, entre otras cosas, el poster.
[10]. La aplicación recibe los datos de la película y los muestra por pantalla.

Este tipo de arquitectura nos proporciona un sistema centralizado en el que el cliente no tiene que disponer de una máquina con gran capacidad de almacenamiento para toda la base de datos de las películas, ni gran capacidad de procesamiento para el cálculo de las recomendaciones. Estas tareas residen y se ejecutan en el servidor, por lo que el usuario accede a los datos que le corresponden a través de la aplicación cliente Java, y simplemente necesita disponer de una maquina con una unidad de procesamiento gráfico básica para visualizar las recomendaciones recibidas.

4.5. Pruebas y Validación

Una vez que el sistema ha sido integrado, comienza esta etapa. Es donde es probado para verificar que el sistema es consistente con la definición de requisitos y la especificación funcional. Por otro lado, la verificación consiste en una serie de actividades que aseguran que el software implementa correctamente una función específica. Al finalizar esta etapa, el sistema ya puede ser instalado en ambiente de explotación.

Para la realización de estas pruebas, vamos a enumerar unos casos de test que se corresponderán con los requisitos funcionales definidos anteriormente.

4.7.1. Casos de Prueba

Los casos de prueba diseñados serán los siguientes:

TEST 1: [RF 1. Acceso a la aplicación]

Pre-condiciones:	La aplicación está instalada en el equipo.
Acción:	El usuario ejecuta la aplicación.
Checkpoint 1:	La aplicación pasa a primer plano

TEST 2: [RF 2. Establecer parámetros de configuración]

Pre-condiciones:	La aplicación se está ejecutando en el equipo.
Acción:	El usuario ajusta los parámetros de configuración.
Checkpoint 1:	Los parámetros de configuración se cambian.

TEST 3: [RF 3. Generación de grupos]

Pre-condiciones:	La aplicación se está ejecutando en el equipo.
Acción:	El usuario indica los identificadores de los usuarios que conforman el grupo, separados por coma.
Checkpoint 1: **Checkpoint 2:**	El formato del campo es valido El formato del campo no es valido

TEST 4: [RF 4. Obtención de recomendaciones]

Pre-condiciones:	Se han configurado correctamente las URL de los servicios. Se han indicado los usuarios que conforman el grupo.
Acción:	El usuario indica los componentes de los que se compone el grupo y lanza la petición.
Checkpoint 1: **Checkpoint 2:** **Checkpoint 3:** **Checkpoint 4:**	Se obtienen las recomendaciones. Mensaje error: No hay conexión con el servidor Mensaje error: Alguno de los usuarios indicados como miembros del grupo no existe El usuario cancela el proceso de obtención de recomendaciones

TEST 5: [RF 5. Seleccionar el modo de visualización]

Pre-condiciones:	La aplicación se está ejecutando en el equipo.
Acción:	El usuario indica el modo de visualización que desea, desde un menú desplegable.
Checkpoint 1:	El modo se selecciona correctamente.

TEST 6: [RF 6. VISUALIZACION JERÁRQUCIA de GRUPO (VJG) – Ver de manera jerárquica las recomendaciones para el grupo]

Pre-condiciones:	Se ha seleccionado el modo de visualización jerárquico. Se han obtenido las recomendaciones.
Acción:	El usuario selecciona y accede al modo de visualización jerárquico, y selecciona ver las recomendaciones grupales.
Checkpoint 1:	El usuario visualiza las recomendaciones grupales de manera jerárquica y puede interactuar con la visualización

TEST 7: [RF 7. VISUALIZACION JERÁRQUCIA INDIVIDUAL (VJI) – Ver de manera jerárquica las recomendaciones para un usuario del grupo]

Pre-condiciones:	Se ha seleccionado el modo de visualización jerárquico. Se han obtenido las recomendaciones.
Acción:	El usuario selecciona y accede al modo de visualización jerárquico, y selecciona ver las recomendaciones para un individuo concreto.
Checkpoint 1:	El usuario visualiza las recomendaciones de un individuo del grupo de manera jerárquica y puede interactuar con la visualización

TEST 8: [RF 8. VJ vs Individual (VJvI) – Ver la aportación de un usuario del grupo frente a las recomendaciones generales del grupo]

Pre-condiciones:	Se ha seleccionado el modo de visualización jerárquico. Se ha seleccionado ver la recomendación grupal.
Acción:	El usuario selecciona ver que aportación realiza uno de los miembros del grupo a la recomendación global.
Checkpoint 1:	El usuario visualiza la aportación del miembro seleccionado sobre la visualización general de las recomendaciones grupales.

TEST 9: [RF 9. VJ vs Grupo (VJvG) – Ver influencia de las recomendaciones generales del grupo sobre un miembro de él]

Pre-condiciones:	Se ha seleccionado el modo de visualización jerárquico. Se ha seleccionado ver la recomendación de un individuo concreto.
Acción:	El usuario selecciona ver que influencia tiene el individuo visualizado sobre la recomendación global.
Checkpoint 1:	El usuario visualiza la aportación del miembro seleccionado bajo la visualización general de las recomendaciones grupales.

TEST 10: [RF 10. VISUALIZACIÓN DE LINEAS a GRUPO (VLG) - Ver las recomendaciones para el grupo]

Pre-condiciones:	Se ha seleccionado el modo de visualización gráfico de líneas. Se han obtenido las recomendaciones.
Acción:	El usuario selecciona y accede al modo de visualización gráfico de líneas.
Checkpoint 1:	El usuario visualiza las recomendaciones del grupo en forma de línea en un gráfico de líneas y puede interactuar con la visualización

TEST 11: [RF 11. VISUALIZACIÓN DE LINEAS INDIVIDUAL (VLI) - Ver las recomendaciones para un usuario del grupo]

Pre-condiciones:	Se ha seleccionado el modo de visualización gráfico de líneas.
Acción:	El usuario selecciona ver la línea de un individuo del grupo.
Checkpoint 1:	El usuario visualiza las recomendaciones del individuo en forma de línea en un gráfico de líneas y puede interactuar con la visualización.

TEST 12: [RF 12. VL-C – Combinar recomendaciones]

Pre-condiciones:	Se ha seleccionado el modo de visualización gráfico de líneas.
Acción:	El usuario selecciona ver la líneas de uno o varios individuos del grupo, junto con la del grupo (o no) deseleccionando otras.
Checkpoint 1:	El usuario visualiza las líneas de las recomendaciones de las entidades marcadas, pudiendo realizar comparaciones.

TEST 13: [RF 13. VISUALIZACIÓN ESCALADO MULTIDIMENSIONAL a GRUPO (MDSG) - Ver las recomendaciones para el grupo]

Pre-condiciones:	Se ha seleccionado el modo de visualización de escalado multidimensional. Se han obtenido las recomendaciones.
Acción:	El usuario selecciona y accede al modo de visualización de escalado multidimensional.
Checkpoint 1:	El usuario visualiza las recomendaciones grupales a la derecha de la pantalla y puede interactuar con la visualización

TEST 14: [RF 14. MDSI – Comparar las recomendaciones del grupo con las de un usuario]

Pre-condiciones:	Se ha seleccionado el modo de visualización de escalado multidimensional. Se han obtenido las recomendaciones.
Acción:	El usuario selecciona ver la representación en escalado multidimensional de un miembro del grupo
Checkpoint 1:	El usuario visualiza las recomendaciones individuales del miembro seleccionado a la izquierda de la pantalla, junto con las grupales, y puede interactuar con la visualización

TEST 15: [RF 15. Ver la información de una película]

Pre-condiciones:	Se ha seleccionado el modo de visualización jerárquico o el modo escaldo multidimensional.
Acción:	El usuario selecciona ver la información de una película.
Checkpoint 1:	El usuario visualiza la información de la película seleccionada.

4.7.1. Resultado de los Casos de Prueba

TEST 1: [RF 1. Acceso a la aplicación]	
Checkpoint 1:	Ok
TEST 2: [RF 2. Establecer parámetros de configuración]	
Checkpoint 1:	Ok
TEST 3: [RF 3. Generación de grupos]	
Checkpoint 1:	Ok
Checkpoint 2:	Ok
TEST 4: [RF 4. Obtención de recomendaciones]	
Checkpoint 1:	Ok
Checkpoint 2:	Ok
Checkpoint 3:	Ok
Checkpoint 4:	Ok
TEST 5: [RF 5. Seleccionar el modo de visualización]	
Checkpoint 1:	Ok
TEST 6: [RF 6. VISUALIZACION JERÁRQUCIA de GRUPO (VJG) – Ver de manera jerárquica las recomendaciones para el grupo]	
Checkpoint 1:	Ok
TEST 7: [RF 7. VISUALIZACION JERÁRQUCIA INDIVIDUAL (VJI) – Ver de manera jerárquica las recomendaciones para un usuario del grupo]	
Checkpoint 1:	Ok
TEST 8: [RF 8. VJ vs Individual (VJvI) – Ver la aportación de un usuario del grupo frente a las recomendaciones generales del grupo]	
Checkpoint 1:	Ok
TEST 9: [RF 9. VJ vs Grupo (VJvG) – Ver influencia de las recomendaciones generales del grupo sobre un miembro de él]	
Checkpoint 1:	Ok
TEST 10: [RF 10. VISUALIZACIÓN DE LINEAS a GRUPO (VLG) - Ver las recomendaciones para el grupo]	
Checkpoint 1:	Ok
TEST 11: [RF 11. VISUALIZACIÓN DE LINEAS INDIVIDUAL (VLI) - Ver las recomendaciones para un usuario del grupo]	
Checkpoint 1:	Ok
TEST 12: [RF 12. VL-C – Combinar recomendaciones]	
Checkpoint 1:	Ok
TEST 13: [RF 13. VISUALIZACIÓN ESCALADO MULTIDIMENSIONAL a GRUPO (MDSG) - Ver las recomendaciones para el grupo]	
Checkpoint 1:	Ok
TEST 14: [RF 14. MDSI – Comparar las recomendaciones del grupo con las de un usuario]	
Checkpoint 1:	Ok
TEST 15: [RF 15. Ver la información de una película]	
Checkpoint 1:	Ok

Tabla 4.8. Resultado de los casos de prueba.

4.6. Despliegue y Mantenimiento

Dado que la aplicación desarrollada en este TFG es un prototipo, no se desplegará de manera comercial, así el manteamiento consistirá en sucesivas iteraciones de mejora y desarrollo del prototipo con fines de investigación

CAPÍTULO 5
CONCLUSIONES

5. CONCLUSIONES

5.1. Conclusión

Para terminar, y una vez acabada tanto la redacción de este documento escrito como el desarrollo de la aplicación objetivo de este trabajo, se pueden extraer varias conclusiones interesantes que ponen de manifiesto el estudio llevado a cabo para alcanzar los objetivos propuestos en el Capítulo 1.

Este proyecto nació de una proposición de proyecto por parte del Doctor Luís Martínez López, consistente en el Estudio de interfaces gráficas para Sistemas de Recomendación a Grupos.

El primer propósito que se ha perseguido ha sido conocer en profundidad la situación actual de los Sistemas de Recomendación y sobre todo de las técnicas de visualización para estos. Para ello se ha realizado un análisis de las diferentes alternativas gráficas que podrían implementarse y se eligieron las que consideraban mejor para la finalidad del proyecto.

Otro objetivo perseguido y logrado durante el desarrollo ha sido la construcción de una interfaz siguiendo una filosofía de diseño centrado en el usuario, donde lo importante es crear un producto visual intuitivo que les aporte una experiencia satisfactoria, logrando que realicen las tareas con el mínimo esfuerzo posible.

También se han de destacar las técnicas de ingeniería del software utilizadas en el análisis y diseño previos a la implementación, las cuales ayudan a la fijación de los objetivos y requisitos que debe respetar el proyecto, lo que permite abordar las fases de desarrollo con unas metas específicas y planteadas previamente.

Eventualmente hay que añadir que, aunque las funcionalidades del sistema estén totalmente operativas, ha quedado margen para seguir evolucionando el sistema, tanto en el lado de la aplicación como en el servidor de recomendaciones (aunque este no pertenezca a nuestro ámbito).

Desde mi punto de vista personal he de decir que me ha sido muy placentero el poder desarrollar un proyecto software de estas características ya que me ha

permitido poner en práctica muchos de los conceptos adquiridos a lo largo de este máster.

5.2. Reflexión

La realización de un Trabajo Fin de Máster supone una prueba de fuego para cualquier universitario, ya que se trata de un proyecto prácticamente en solitario y con unas dimensiones considerables que conllevan a que el autor demuestre que está preparado para afrontar la vida laboral, la vida profesional, con los conocimientos adquiridos durante todo el máster.

Estos conocimientos no sólo son técnicos; la teoría y la práctica son importantes, pero más aún lo es la capacidad de reaccionar ante un problema, de buscar soluciones, de no conformarse con cualquier cosa, de desenvolverse ante una situación difícil o delante de otras personas defendiendo tus propias ideas, conocimientos o soluciones, etc. Aptitudes que sin duda se van consiguiendo con los años y que durante la vida universitaria es cuando se empieza a ser realmente consciente de ellas.

Todos los conocimientos adquiridos en todas y cada una de las asignaturas, no sólo de la universidad, sino también del instituto, del colegio; todos los profesores que han dejado su huella, su experiencia, su sabiduría…; todos los trabajos, exámenes y exposiciones realizadas, sin olvidar los nervios que suponían; todas las experiencias vividas, buenas y malas; todos los éxitos alcanzados y, por supuesto, los errores cometidos; todo, absolutamente todo, han hecho que este Trabajo fin de Máster se haya convertido en toda una realidad.

La realización de una aplicación para un problema para mí desconocido hasta hace algunos meses como los sistemas de recomendación no habría sido posible de no ser por los conocimientos y conceptos adquiridos durante el máster.

La resolución de errores, la búsqueda de soluciones, el sólo conformarse con lo mejor y el buscar siempre más, no habría sido posible de no ser por los profesores que han dado lo mejor de ellos mismos para hacer a los demás mejores. Gracias.

Y, por supuesto, la documentación realizada no habría sido posible, de nuevo, sin aquellos profesores que se fijan hasta en el más mínimo espacio y para los que es difícil alcanzar la perfección, pero siempre hay que intentarlo al máximo.

Este TFM ha supuesto todo un reto para mí desde el primer momento en el que surgió la idea sobre cómo encaminarlo. Gracias a su desarrollo he aprendido muchos nuevos conocimientos, pero sobre todo, he adquirido aptitudes que me van a servir durante toda mi vida. En definitiva, la realización de este TFM me ha exigido dar lo mejor de mí en todo lo aprendido y vivido hasta ahora para empezar a construir sobre estos cimientos lo que va a ser el resto de mi vida.

ANEXO I
Manual de Instalación

6. ANEXO I. Manual de Instalación

En este manual vamos a explicar cómo instalar la aplicación en un PC con Windows (64 bits) cualquiera. El proceso es bastante sencillo y no supondrá ningún problema para el usuario.

Lo primero que tenemos que hacer es ejecutar el archivo "instalar_tfm.exe", que podemos encontrar en el CD incluido en el proyecto.

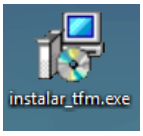

Figura 6.1 Icono instalador.

Una vez ejecutado, Windows nos pedirá permiso para que la aplicación realice cambios en nuestro pc.

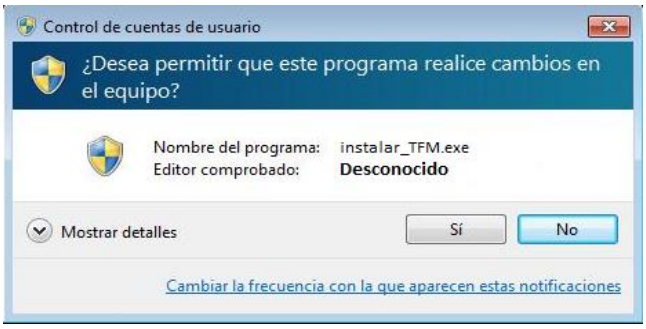

Figura 6.2 Solicitud permiso de Windows.

Pulsamos en Si, y se abrirá el asistente de instalación de la aplicación. Lo primero que nos solicitará es la ruta en la que queremos instalarlo. Se la indicamos como se muestra a continuación:

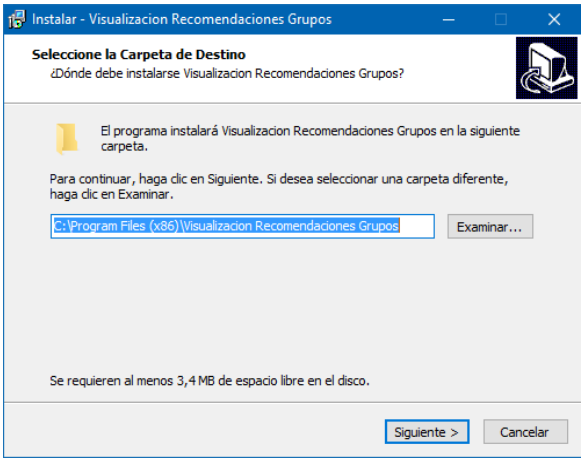

Figura 6.3 Seleccionar destino instalación.

Pulsamos en siguiente, y nos solicitará si queremos crear un acceso directo en el escritorio. Elegimos la opción que queramos y pulsamos en siguiente:

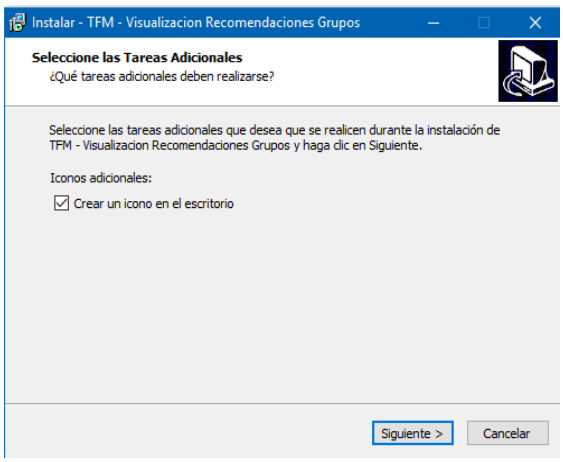

Figura 6.4 Crear acceso directo en el escritorio.

En la siguiente ventana que nos aparece simplemente tendremos que pulsar en Instalar, y comenzará el proceso de instalación de nuestra aplicación, como se muestra a continuación:

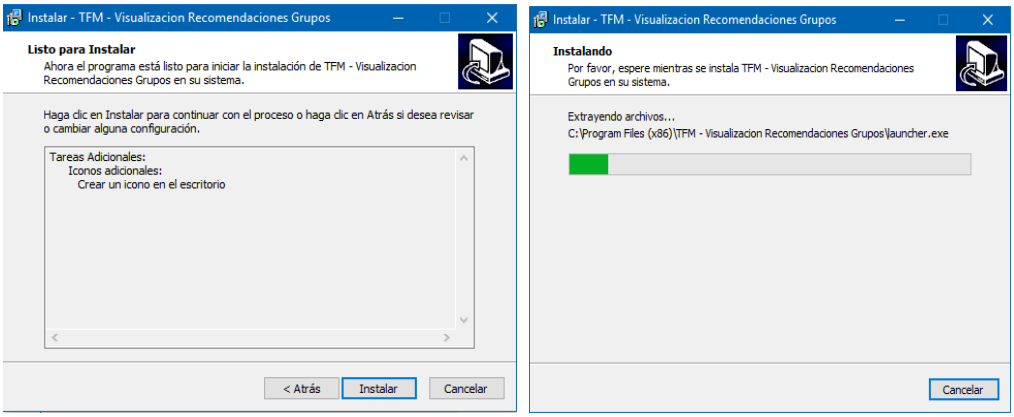

Figura 6.5 Proceso de instalación de la aplicación.

Por último pulsaremos en Finalizar, y la aplicación estará lista para su uso.

Figura 6.6 Finalizar instalación de la aplicación.

Pero antes de ejecutarla deberemos comprobar si tenemos instalado en nuestro equipo el entorno Java JRE en su última versión para 64 bits. Si no lo tuviéramos instalado, lo que tendremos que hacer es ejecutar el archivo "jre-8u121-windows-x64.exe", que podemos encontrar en el CD incluido en el proyecto.

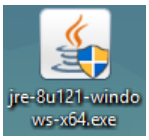

Figura 6.7 Icono instalador Java JRE.

Una vez aceptado el permiso de Windows para que la aplicación realice cambios en nuestro PC, lo primero que nos indicará el asistente de instalación es si queremos cambiar la ruta en la que queremos instalarlo o directamente proceder a la instalación, como se muestra a continuación:

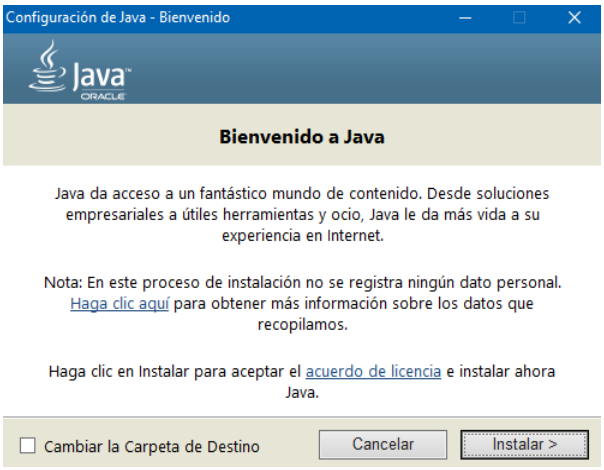

Figura 6.8 Comenzar instalación Java JRE.

Una vez pulsado en instalar comenzará el proceso de instalación del JRE de Java, como se muestra a continuación:

Figura 6.9 Proceso instalación Java JRE.

Por último, si todo ha ido bien y se ha instalado correctamente, se nos mostrará la siguiente ventana:

Figura 6.10 Finalización instalación Java JRE.

Ahora ya si podemos ejecutar nuestra aplicación desde el acceso directo que se habrá creado en el escritorio:

Figura 6.11 Icono de la aplicación instalada.

ANEXO II
Manual de Usuario

7. ANEXO II. Manual de Usuario

A lo largo de este manual vamos a describir cómo acceder a toda la funcionalidad de la aplicación para permitir realizar un uso correcto de la misma. Para ello, vamos a seguir un orden de ejecución de acuerdo al orden en que se han colocado los modos de visualización disponibles.

7.1. Configuración de la aplicación

Para utilizar la aplicación primero deberemos configurarla. Para ello, una vez ejecutada, pulsaremos en el botón configuración, situado en el menú principal:

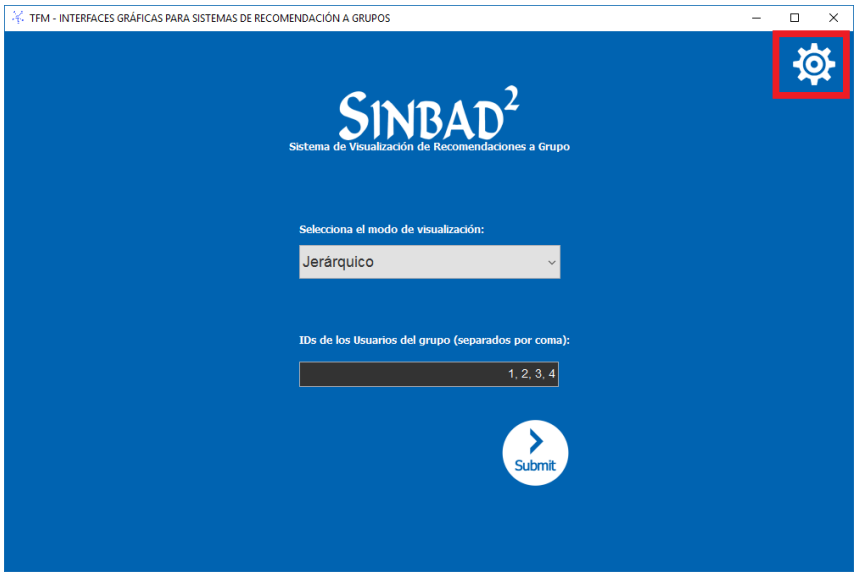

Figura 7.1 Acceder a la configuración.

Una vez dentro de la ventana de configuración deberemos introducir la URL del servicio de recomendaciones, tanto la URL del método para obtener las recomendaciones grupales, como el método para obtener las recomendaciones individuales.

También, desde esta ventana, podremos indicar el número de recomendaciones que queremos obtener y visualizar en cada uno de los modos de visualización.

Por último, volveremos al menú principal pulsando en el botón 'Home' y los cambios se guardaran automáticamente.

Figura 7.2 Parámetros de configuración de la aplicación.

7.2. Modo de visualización jerárquico

Para acceder a este primer método de visualización, lo seccionaremos en el menú desplegable de modos, en el menú principal de la aplicación. Después, indicaremos los usuarios que conformaran el grupo, separados por coma, y para el que se generarán las recomendaciones que se visualizarán.

Por último pulsaremos en el botón de comenzar para iniciar el proceso:

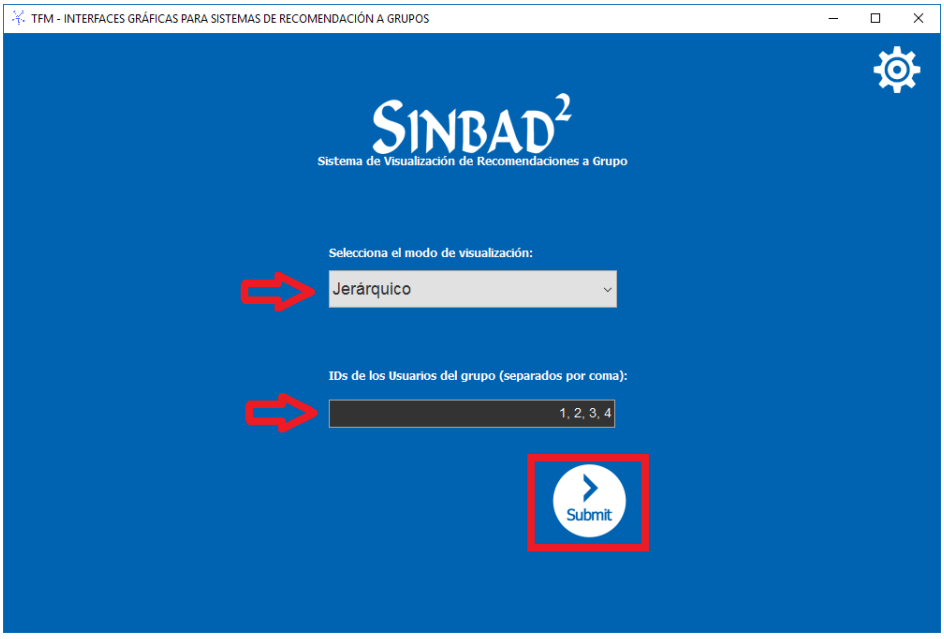

Figura 7.3 Selección modo de visualización jerárquico.

Ahora se comenzarán a solicitar las recomendaciones para el grupo indicado. Una vez obtenida la información se construirá un modelo necesario para visualizar las recomendaciones.

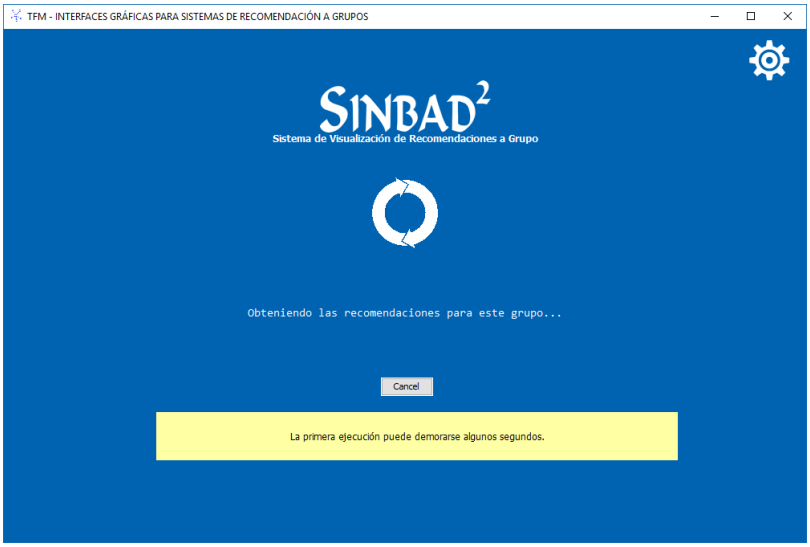

Figura 7.4 Carga de las recomendaciones en modo de visualización jerárquico.

Y una vez terminado el proceso, se mostrará la pantalla principal del modo de visualización jerárquico, y ya estaremos listos para interactuar con él.

El primer paso que deberemos hacer es elegir, desde el círculo central, si queremos ver las recomendaciones del grupo, o las recomendaciones de alguno de los miembros de manera individual.

Primero vamos a ver las del grupo y para ello haremos clic en la parte derecha del círculo.

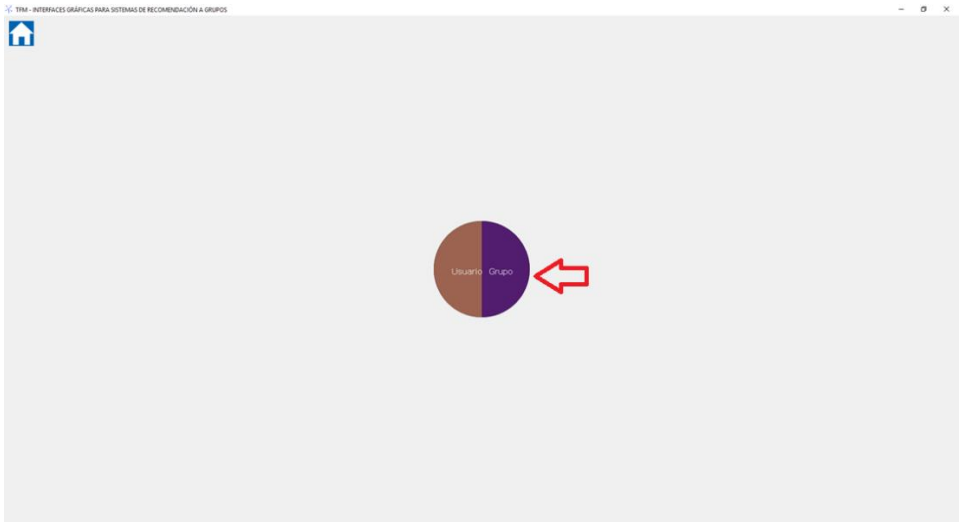

Figura 7.5 Ver las recomedaciones de grupo en modo de visualización jerárquico.

Ahora podremos navegar por los distintos niveles de clasificación de manera jerárquica, haciendo clic en las subcategorías que queramos ver. La anchura de cada subcategoría viene representada por la cantidad de películas que hay en ella.

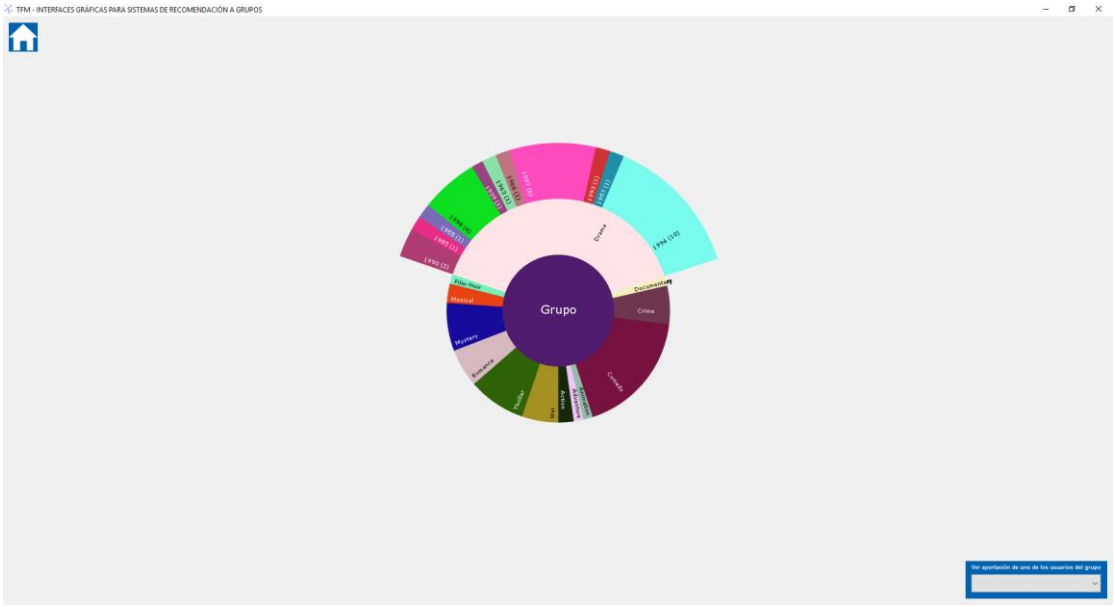

Figura 7.6 Navegación por modo de visualización jerárquico.

Si en algún momento, sin importar el nivel en que estemos, queremos comparar las recomendaciones del grupo con las de algún miembro de él, es decir, ver el grado de aportación de un miembro del grupo a la recomendación global, pulsaremos en el selector de usuarios en la esquina inferior derecha y elegiremos el usuario del que queremos visualizar la aportación que realiza:

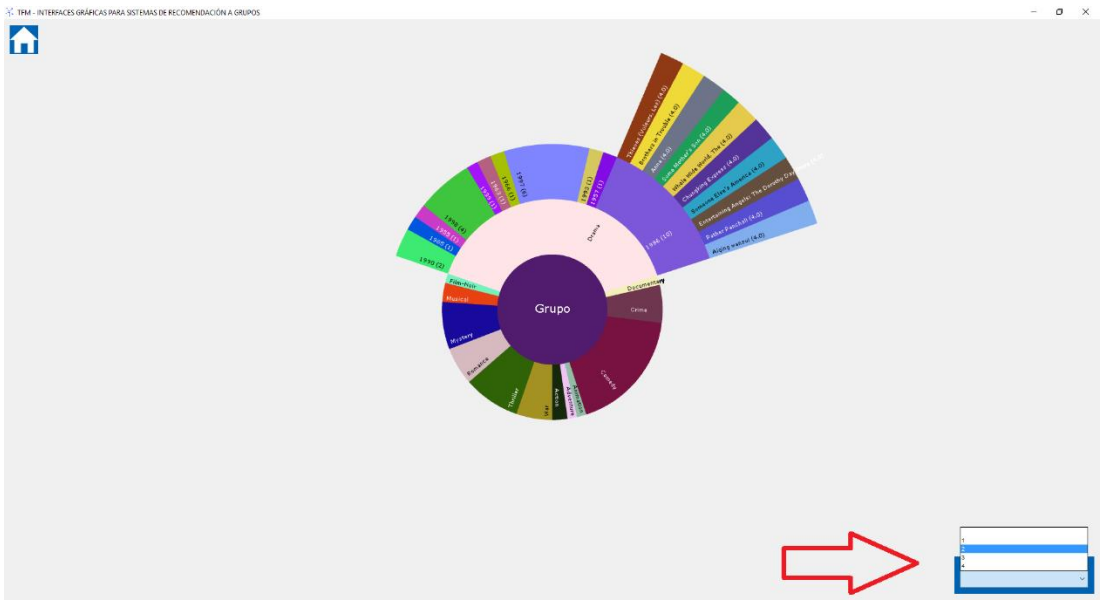

Figura 7.7 Seleccionar un miembro del grupo para ver su aportación.

La aportación se mostrará en rojo encima del grafico de grupo, siendo la anchura de esta cobertura para cada subcategoría la cantidad de películas recomendadas para el usuario, y en comparación con las del grupo. Para ocultar esta visualización comparativa pulsaremos en “Ocultar aportación”.

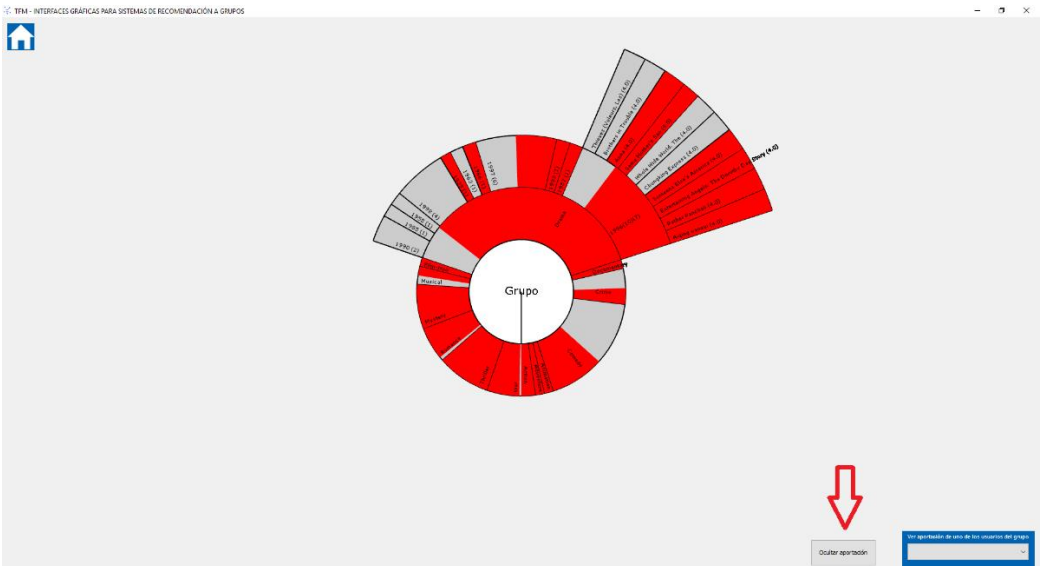

Figura 7.8 Ver y ocultar aportación de un miembro del grupo.

Ahora vamos a volver al nivel 0 de nuestro gráfico jerárquico para ver las recomendaciones de un miembro del grupo de manera individual. Para ello pulsamos en el círculo central interior, y el gráfico se contraerá.

Ahora, pera visualizar las recomendaciones de un miembro del grupo haremos clic en la parte izquierda del círculo.

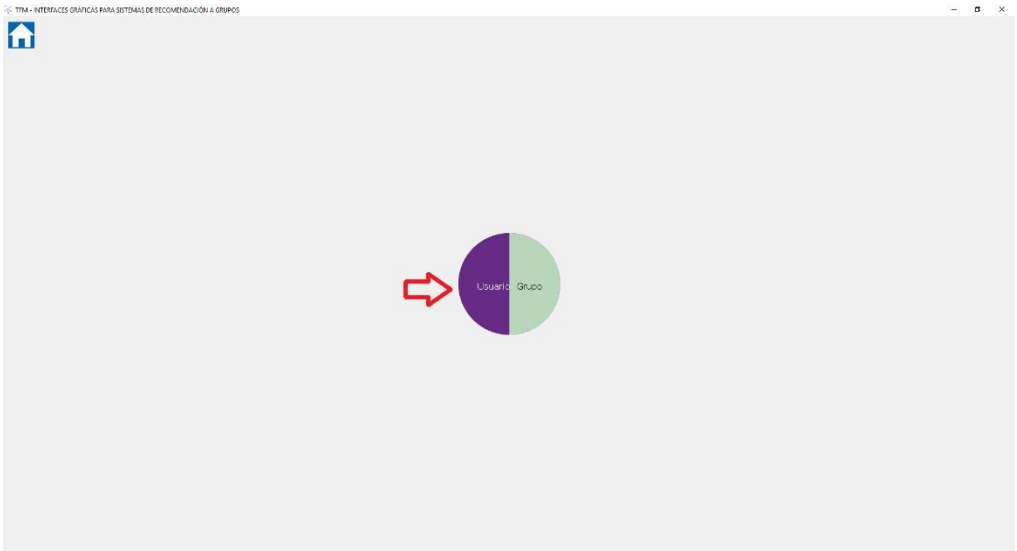

Figura 7.9 Ver las recomedaciones individuales en modo de visualización jerárquico.

Ahora pulsaremos en el selector de usuarios en la esquina inferior derecha para elegir el usuario del que queremos visualizar sus recomendaciones:

Figura 7.10 Seleccionar miembro del grupo para visualizar.

De igual modo que para el grupo, ahora podremos navegar por los distintos niveles de clasificación de manera jerárquica, haciendo clic en las subcategorías que queramos ver.

Si en algún momento, sin importar el nivel en que estemos, queremos comparar las recomendaciones del miembro con las de su grupo, es decir, ver cómo influye dicho miembro del grupo a la recomendación global, pulsaremos en el botón "Ver influencia del miembro en el grupo" en la esquina inferior izquierda:

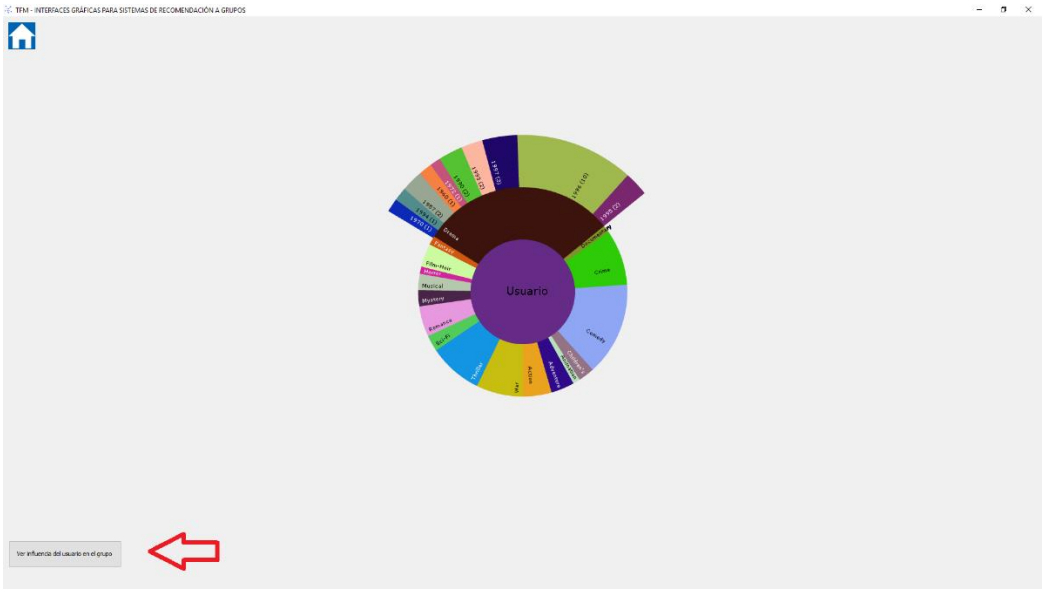

Figura 7.11 Ver la influencia del grupo en el miembro seleccionado.

La influencia se mostrará en rojo encima del grafico del miembro del grupo, siendo la anchura de esta cobertura para cada subcategoría la cantidad de películas recomendadas para el grupo, y en comparación con las del miembro de este. Para ocultar esta visualización comparativa pulsaremos en “Ocultar influencia”.

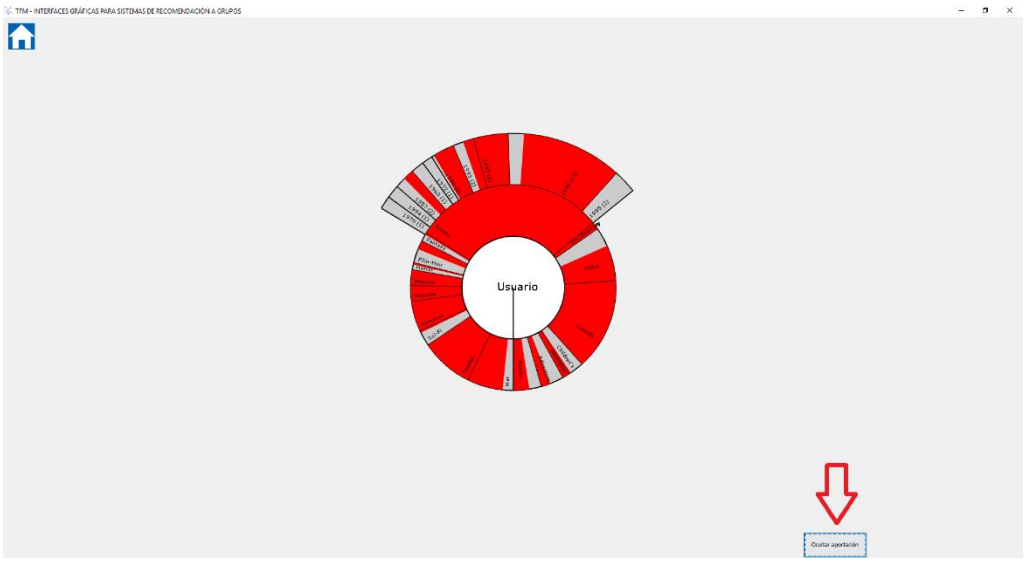

Figura 7.12 Ocultar la influencia del grupo en el miembro seleccionado.

Tanto en la modalidad para grupo como para miembro individual, si navegamos hasta el último nivel de la jerarquía obtendremos las películas recomendadas.

Si pulsamos en cualquiera de ellas se solicitará al servidor externo de información de películas los datos de la película seleccionada. Si se encuentra, se mostrará la información en un panel a la derecha.

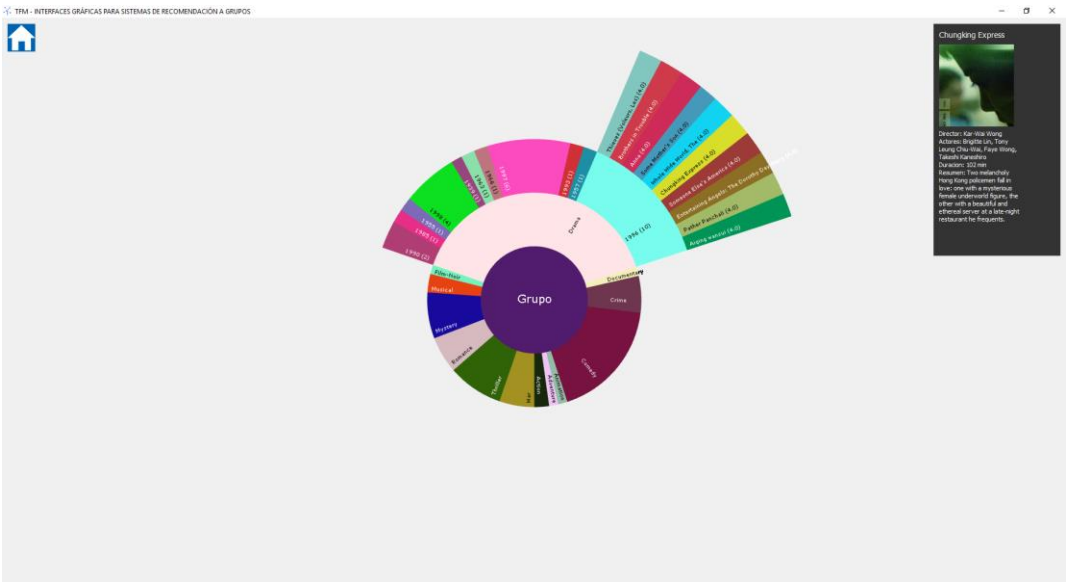

Figura 7.13 Ver informacion de una película en modo de visualización jerárquico.

7.3. Modo de visualización en Gráfico de Líneas

Para acceder a este segundo método de visualización, lo seccionaremos en el menú desplegable de modos, en el menú principal de la aplicación. Después, indicaremos los usuarios que conformaran el grupo, separados por coma, y para el que se generarán las recomendaciones que se visualizarán.

Por último pulsaremos en el botón de comenzar para iniciar el proceso de solicitud de las recomendaciones para el grupo indicado. Una vez obtenida la información se construirá un modelo necesario para visualizar las recomendaciones.

Figura 7.14 Selección y carga del modo de visualización en gráfico de líneas.

Y una vez terminado el proceso, se mostrará la pantalla principal del modo de visualización en gráfico de líneas, y ya estaremos listos para interactuar con él.

Lo primero que se nos mostrará en este método es la línea perteneciente a las recomendaciones para el grupo:

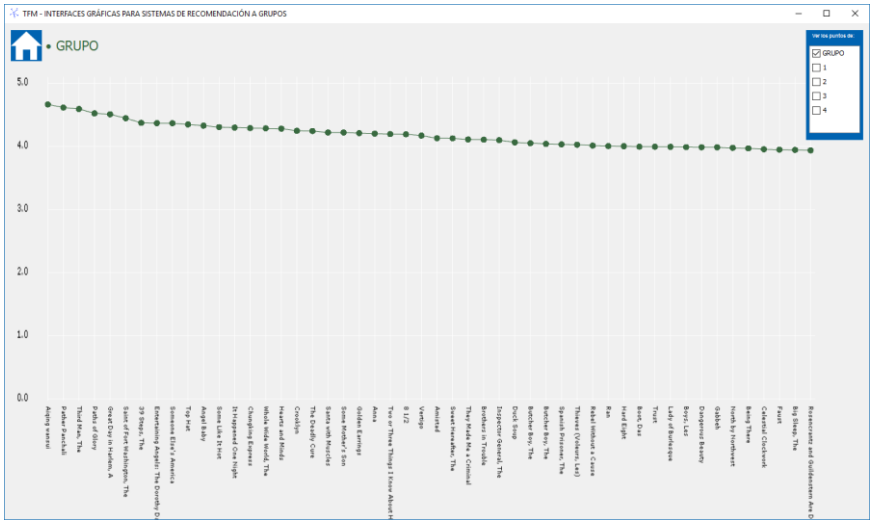

Figura 7.15 Línea de recomendaciónes del grupo.

Ahora, desde el menú lateral derecho podremos interactuar con el gráfico añadiendo o eliminado líneas del gráfico.

Las líneas corresponden a la línea grupal y a las líneas pertenecientes a cada uno de los miembros del grupo, y se podrán combinar como el usuario desee.

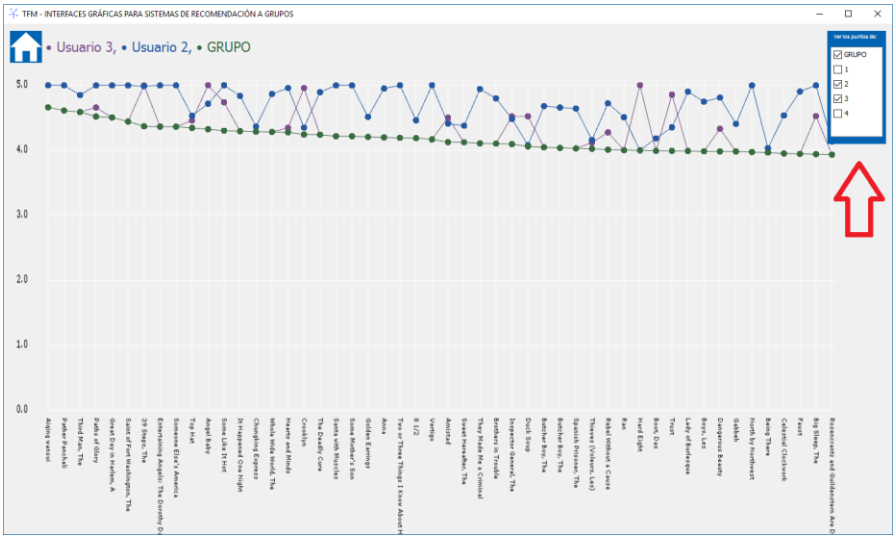

Figura 7.16 Combinación de varías líneas de recomendaciones.

7.4. Modo de visualización mediante Escalado Multidimensional

Para acceder a este último método de visualización, lo seleccionaremos en el menú desplegable de modos, en el menú principal de la aplicación. Después, indicaremos los usuarios que conformaran el grupo, separados por coma, y para el que se generarán las recomendaciones que se visualizarán.

Por último pulsaremos en el botón de comenzar para iniciar el proceso de solicitud de las recomendaciones para el grupo indicado. Una vez obtenida la información se construirá un modelo necesario para visualizar las recomendaciones.

Figura 7.17 Selección y carga del modo de visualización en gráfico de líneas.

Y una vez terminado el proceso, se mostrará la pantalla principal del modo de visualización mediante escalado multidimensional, y ya estaremos listos para interactuar con él.

Lo primero que se nos mostrará en este método es, a la derecha, la representación mediante puntos 2D de las recomendaciones para el grupo y, a la izquierda, la representación de las del primer miembro del grupo:

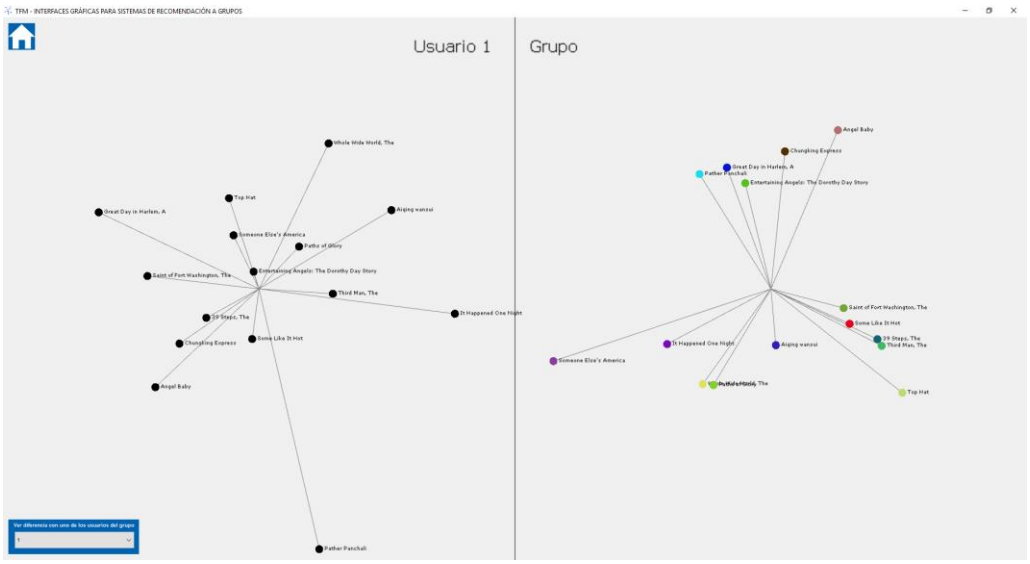

Figura 7.18 Comparativa recomendación grupal y miembro en MDS.

Si queremos cambiar el miembro del grupo visualizado en la parte izquierda, simplemente pulsaremos en el selector de usuarios de la esquina inferior izquierda y seleccionaremos el miembro del grupo que queremos visualizar junto al grupo, y de esta manera compararlos.

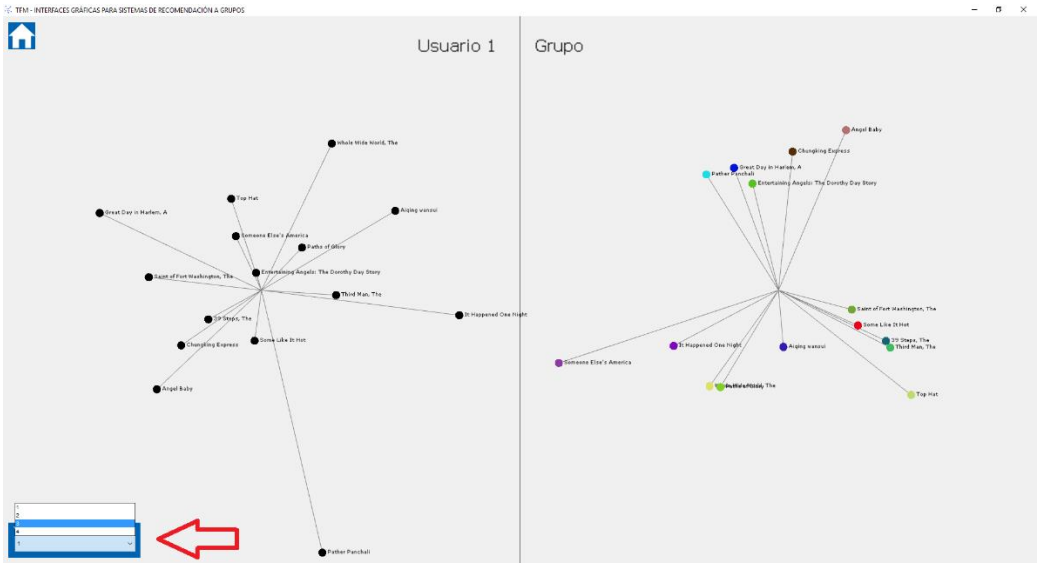

Figura 7.19 Cambiar miembro del grupo mostrado en MDS.

La parte izquierda de la visualización se actualizará automáticamente con el usuario seleccionado:

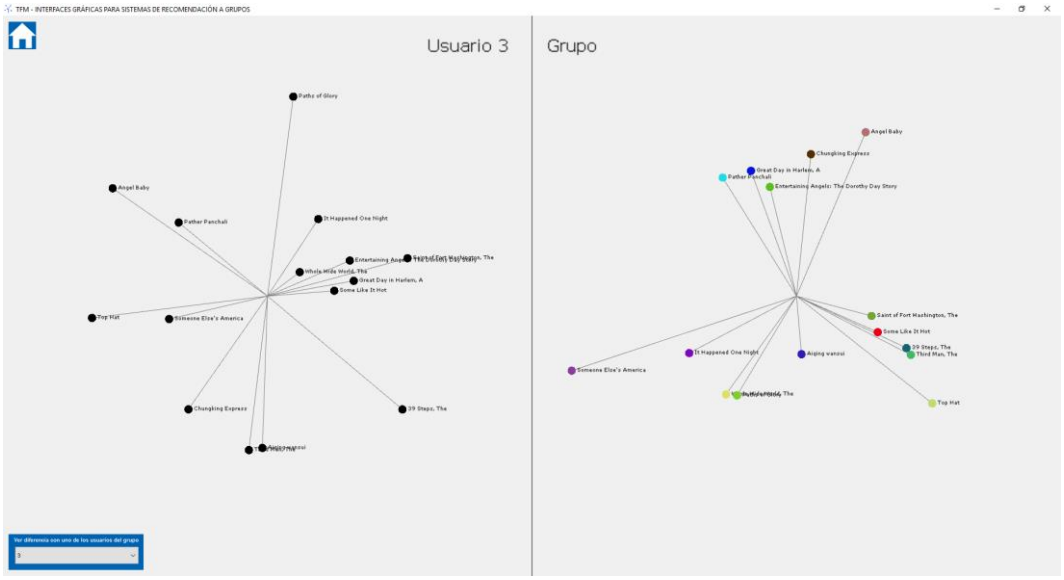

Figura 7.20 Actualización automática del miembro del grupo seleccionado.

Por último, si pulsamos en cualquiera de los puntos que corresponden a las películas, se solicitará al servidor externo de información de películas los datos de la película seleccionada. Si se encuentra, se mostrará la información en un panel a la derecha.

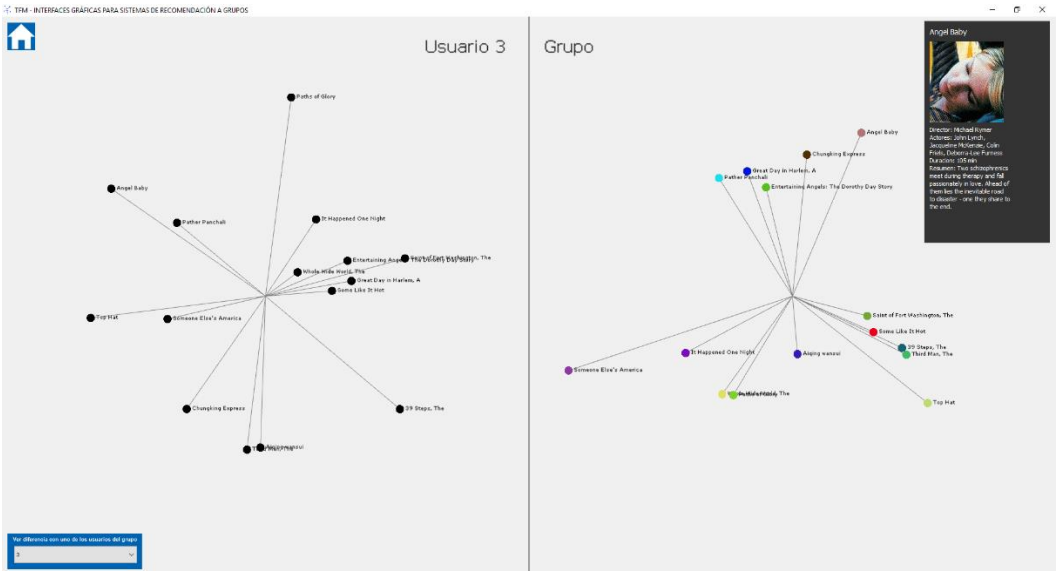

Figura 7.21 Ver informacion de una película en MDS.

Bibliografía

[1] Adomavicius, G. and A. Tuzhilin, (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.

[2] Battista, G., Eades, P. (1999), Graph Drawing: Algorithms for the Visualization of Graphs, Prentice Hall

[3] Bostandjiev, S., O'Donovan, J., Höllerer, T. (2012), TasteWeights: A Visual Interactive Hybrid Recommender System.

[4] Burke, R., (2002). Hybrid Recommender Systems: Survey and Experiments

[5] Carralero, C. (2015) Recommender inCloud. QualityObjects.

[6] Chen, N. (2013). Clustering and visualization of bankruptcy trajectory using self-organizing map. Expert Systems with Applications.

[7] Dix, A., Janet Finlay, Gregory D. Abowd, Russell Beale, (2003). "Human-Computer Interaction"

[8] Eades, P. (1984), "A Heuristic for Graph Drawing", Congressus Numerantium.

[9] Friendly, M. (2006). A brief history of data visualization. Handbook of Data Visualization.

[10] Farre, C. (2009) Sistemas de recomendación para webs de información.

[11] Galán, J. (2014). Metodología para la Generación de Explicaciones para Sistemas de Recomendación Sensibles al Contexto. CNIDT México.

[12] Galan, S. (2007). Filtrado Colaborativo y Sistemas de Recomendación. Universidad Carlos III Madrid.

[13] García, O., Guevara, A. (2005). Introducción a la Programación Gráfica con OpenGL. Universidad Ramón Llull.

[14] Gotz, D., Wen, Z. (2009), HARVEST: Situational Visual Analytics for the Masses

[15] Heckerman, D. (2001), Dependency Networks for Inference, Collaborative Filtering, and Data Visualization

[16] Herrera-Viedma, E., Porcel C., Hidalgo L., (2004). Sistemas de recomendaciones: herramientas para el filtrado de información en Internet.

[17] Huecas, G., and Salvachua, J., (2010) Generating Context-aware Recommendations using Banking Data.

[18] Kaufmann, M., Wagner, D. (2001). Drawing Graphs. Springer.

[19] Konstan, J., Riedl, J., (2012) "Deconstructing Recommender Systems"

[20] Middleton, S.E., De Roure, D., & Shadbolt, N.R. (2004), Ontology-based recommender systems. Handbook on Ontologies.

[21] Moya García, R. (2015). Mapas gráficos para la visualización de relaciones en sistemas de recomendación.

[22] O'Connor, M., Cosley, D., Konstan, J., and Riedl, J. (2001). PolyLens: A Recommender System for Groups of Users

[23] O'Donovan, J., Smyth, B. (2008), PeerChooser: Visual Interactive Recommendation

[24] Peis, E., Morales-del-Castillo, J. M., Delgado-López J. A. (2008). Sistemas de Recomendación. *Un análisis del estado de la cuestión*. Universidad de Granada.

[25] Quijano, L. (2010). Impacto de los factores y organizaciones sociales en los procesos de recomendación para grupos.

[26] Reenskaug, T., (1979). THING-MODEL-VIEW-EDITOR an Example from planning system.

[27] Reingold, E., Tilford, J. (1981), Tidier Drawings of Trees.

[28] Ricci, F., Rokach, L., and Shapira, B. (2010) "Recommender Systems Handbook"

[29] Rodríguez, E., (2013).Sistemas de recomendación sensibles al contexto

[30] Rolando Nuñez, E. (2012). Sistemas de Recomendación de Contenidos para libros inteligentes. Universidad de Oviedo.

[31] Shani, G., and Gunawardana, A., (2009). "Evaluating Recommendation Systems"

[32] Shneiderman, B., Plaisant, C., Designing the User Interface, Pearson 5ª edición.

[33] Sommerville, I. (2007). Ingeniería del Software. Editorial Addison Wesley.

[34] Sotelo, R., (2010). Recommendation of audiovisual contents for families and groups of friends.

[35] Stoica, I. (2003), Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications

[36] Tufte, E. (1983). The Visual Display of Quantitative Information. Graphics Press (vol. 2, núm. 9).

[37] Vassileva, J. (2015). Visualization and User Control of Recommender Systems. CRIWG 2015.

[38] Wang, W. (2016). Enhanced Group Recommender System and Visualization. University of Technology Sydney.

[39] Wielsen, J. (2012). Usability 101: Introduction to Usability.

[40] Wongsuphasawat, K., Gotz, D. (2012). Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization.