



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 6

Trabajo de Diploma para Optar por el Título de  
Ingeniero en Ciencias Informáticas

Título: Plugin de recomendación híbrido para la plataforma AGORAV, integrado al Sistema Gestor de  
Procesos de Medias.

Autor: Omar Castañeda Infante

Tutor: Ing. Miguel Morciego Varona

La Habana, Junio del 2015

“Año 56 de la Revolución”



## Declaración de autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

Autor

Omar Castañeda Infante

\_\_\_\_\_

Tutor

Miguel Morciego Varona

A portrait of Albert Einstein, showing his characteristic wild hair and mustache. He is wearing a dark suit jacket over a light-colored shirt and a dark tie. The background is a wooden wall with a faint circular chalkboard-like pattern. The quote is overlaid on the left side of the image.

**“SOMOS ARQUITECTOS DE  
NUESTRO PROPIO DESTINO”**

*-Albert Einstein*

## Datos de contacto

**Tutor:** Ing. Miguel Morciego Varona.

**Categoría Científica:** Ingeniero.

**Correo electrónico:** mmorciego@uci.cu

**Síntesis del Tutor:** Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2014.

Lleva 1 año de graduado.

## Agradecimientos.

A mis padres Deisy y Omar por apoyarme siempre, por la dedicación y atención que han tenido conmigo durante toda mi vida, su apoyo incondicional en mis estudios. A mi hermana Daily que a pesar de ser menor en edad es la mayor en madures. A mi hermanito Randy por estar siempre jugando y molestando. A mi madrastra Mairelys y a mi hermana de crianza Daimarelys. A mis dos bellas sobrinas Emily y Daenerys. A toda mi familia que me han brindado su apoyo de una forma u otra, y un agradecimiento muy especial a mi abuelo Elio que siempre lo voy a recordar.

A todos los amigos que he conocido en mi tiempo de estudiante. A los que se graduaron, a los que se fueron sin terminar, a los que todavía no se han graduado y a los que conocí en cada nuevo grupo. A los amigos de mi pueblo que siempre me han ayudado, y siempre me decían "¿Cuándo te piensas graduar?".

A mi amigo Alejandro y sus padres que me brindaron siempre su ayuda, les digo, muchas gracias mi hermano.

A mi tutor Miguel, que a pesar del poco tiempo de tutoría, me brindo su ayuda en todo momento y contribuyo mucho a mejor mi tesis, muchas gracias.

A mi oponente Grethell por ser paciente conmigo.

A la decana Yeleny Zulueta que me brindo una importante segunda oportunidad, siempre lo voy a recordar, a ti muchas gracias.

...y a todos los que de una forma u otra han contribuido a mi formación profesional y personal, que no serán olvidados nunca, a todos, muchas gracias.

Omar



Dedico mi tesis a mis padre, a mis hermanos, a mis sobrina, a mis abuelos, a los viejos y nuevos amigos, a todos los profesores que han influido aunque sea un poco en mi vida profesional y personal.

A todos muchas gracias por formar parte de mi vida.

## Resumen

Los sistemas de recomendación en los últimos tiempos se han convertido en herramientas muy comunes capaces de filtrar información disponible en la web. Estos sistemas utilizan los perfiles de los usuarios para mostrarle la información de mayor interés para ellos simplificándoles su búsqueda. La plataforma AGORAV para la publicación de archivos multimedia es un producto patentado por el centro de desarrollo GEYSED que realiza las recomendaciones al usuario utilizando indistintamente dos sistemas de recomendación uno basado en contenido y otro de filtrado colaborativo, no siempre seleccionando el más adecuado según las características propias del usuario. La presente investigación se plantea el objetivo de implementar un plugin para el Sistema Gestor de Procesos de Media que permita obtener recomendaciones a partir de una hibridación de los sistemas de recomendación basados en contenido y colaborativos. Para lograrlo se analizaron los sistemas de recomendación anteriormente mencionados así como los algoritmos de hibridación más apropiados. Se obtuvo como resultado un plugin para obtener recomendación híbrida, diseñado bajo una arquitectura de tres capas y validado con diferentes tipos de pruebas de software.

## Índice de Contenido

Capítulo 1 Fundamentación Teórica. ....	5
1.1 Conceptos asociados al dominio del problema. ....	5
1.1.1 Recomendación. ....	5
1.1.2 Sistemas de Recomendación. ....	5
1.1.3 Sistema de Recomendación de Filtrado Colaborativo. ....	6
1.1.4 Sistema de Recomendación de Filtrado Basado en Contenido. ....	7
1.1.5 Sistemas de Recomendación Híbridos. ....	8
1.2 Análisis del sistema de recomendación de filtrado colaborativo en AGORAV. ....	9
1.3 Análisis del sistema de recomendación de filtrado basado en contenido en AGORAV. ....	12
1.4 Sistemas de Recomendación existentes. ....	15
1.4.1 REJA. ....	15
1.4.2 FAB. ....	15
1.4.3 Módulo de recomendación híbrida para la plataforma Inter-nos 2.0. ....	16
1.5 Metodologías de desarrollo. ....	17
1.6 Selección de Herramientas. ....	19
1.6.1 Lenguaje de Modelado Unificado. ....	19
1.6.2 Herramienta CASE Visual Paradigm. ....	19
1.6.3 Lenguaje de programación. ....	19
1.6.4 Entorno Integrado de Desarrollo. ....	20
1.6.5 Sistema Gestor de Base de Datos PostgreSQL. ....	20
Capítulo 2 Análisis y diseño del sistema de recomendación híbrido. ....	22
2.1 Modelo de dominio. ....	22
2.2 Requisitos. ....	23
2.2.1 Requisitos funcionales. ....	23
2.2.2 Requisitos no funcionales. ....	24
2.3 Descripción del sistema de recomendación híbrido. ....	25
2.3.1 Diagrama de caso de uso del sistema de recomendación Híbrido. ....	27
2.3.2 Descripción textual del caso de uso: Generar Recomendación. ....	28
2.4 Patrón arquitectónico. ....	30



2.5 Patrones de diseño.....	31
2.5.1 Patrones de diseño GoF.....	31
2.5.2 Patrones de diseño GRASP.....	31
2.6 Diagrama de clase del diseño.....	32
2.6.1 Descripción de las clases del diseño.....	35
Capítulo 3 Implementación y Pruebas del sistema de recomendación híbrida.....	36
3.1 Modelo de datos.....	36
3.2 Diagrama de componente.....	39
3.3 Diagrama de despliegue.....	40
3.4 Estándar de codificación.....	42
3.5 Pruebas.....	43
3.5.1 Pruebas de integración.....	44
3.5.2 Pruebas de Rendimiento.....	44
Conclusiones Generales: .....	47
Recomendaciones.....	48
Bibliografía.....	49

A mediados de los '90 surge Internet como un medio de comunicación global, facilitando a las personas la comunicación y el intercambio de información. En la actualidad Internet es la mayor fuente de información utilizada por muchas personas. Con el paso del tiempo la cantidad de información que se publica en Internet ha tenido un aumento considerable, complicando el trabajo de los usuarios a la hora de buscar información de su interés, debido a la sobrecarga existente en la red (Mario Marrello, 1999).

La toma de decisiones puede aparecer en cualquier contexto de la vida cotidiana, ya sea a nivel profesional, sentimental, familiar, etc. El proceso, en esencia, permite resolver los distintos desafíos a los que se debe enfrentar una persona o una organización. A la hora de tomar una decisión, entran en juego diversos factores. En un caso ideal, se apela a la capacidad analítica (también llamada de razonamiento) para escoger el mejor camino posible; cuando los resultados son positivos, se produce una evolución, un paso a otro estadio, se abren las puertas a la solución de conflictos reales y potenciales. Cualquier toma de decisiones debería incluir un amplio conocimiento del problema que se desea superar, ya que solo luego del pertinente análisis es posible comprenderlo y dar con una solución adecuada (Definición, 2008). Cuando no se tiene conocimiento de que alternativa elegir se tienen en cuenta las opiniones de otras personas.

Los sistemas de recomendación surgen como agentes para ayudar en la toma de decisiones. Con el paso del tiempo estos sistemas se han convertido en herramientas fundamentales y bastante comunes, capaces de filtrar la información disponible en un sitio web. Estos sistemas utilizan los datos de los usuarios para mostrar información de mayor interés para ellos, permitiendo encontrar nuevos contenidos de una forma más rápida y eficiente (Taghipour, y otros, 2008).

En Cuba se encuentra la Universidad de las Ciencias Informáticas (UCI), la cual tiene centros dedicados al desarrollo de software, entre los que se encuentran el centro Geoinformática y Señales Digitales (GEYSED). En este centro se desarrolló la plataforma AGORAV para la publicación de archivos multimedia. Estos archivos multimedia son procesados antes de ser publicado y para resolver estas tareas se utilizan plugins desarrollados con diferentes tecnologías. Para integrar estos plugins se utiliza el Sistema Gestor de Procesos de Media (SGPM). El SGPM utiliza tecnología

distribuida para administrar una granja de servidores en los que se instalan estos plugins, de esta forma la carga de procesamiento de los archivos multimedia cae en estos servidores.

Para la recomendación de archivos multimedia se implementaron dos sistemas de recomendación:

- Sistema de recomendación basado en contenido, el cual utiliza el perfil del usuario y sus intereses presentes en archivos multimedia que hayan sido evaluados por él. Además de las valoraciones del usuario, también utiliza la información descriptiva de cada producto. Este sistema recomienda productos similares a los que les han gustado en el pasado. Algunos de los problemas que presentan este tipo de sistema es el problema del usuario nuevo, donde los usuarios nuevos en el sistema como no tienen un perfil, no se les puede generar recomendaciones. Otra de sus limitantes es la sobre especialización ya que las recomendaciones son generadas por las evaluaciones hechas por el usuario, por lo tanto el sistema no va a ser capaz de sorprender al usuario, es decir que las recomendaciones siempre van a tener las mismas características. Además estos sistemas tienen un análisis limitado del contenido; como las recomendaciones dependen del perfil del usuario, se hace necesario que el usuario tenga muchas evaluaciones para que las recomendaciones tengan mayor valor para él.
- Sistema de recomendación basado en filtrado colaborativo, el cual basa su recomendación en la creación de grupos de usuarios con similitudes entre ellos. Si a una porción del grupo le gusta un archivo multimedia entonces es de suponer que al resto del grupo también le guste. Pero estos sistemas tienen problemas como el del usuario nuevo anteriormente descrito, el de archivo multimedia nuevo ya que en estos sistemas como las recomendaciones dependen de las evaluaciones hechas a cada uno de los archivos multimedia y los archivos nuevos no tiene ninguna evaluación, se hace difícil que sean recomendados. La escasez de puntuaciones o *sparsity*, también es considerada un problema que aparece cuando el número de archivos multimedia sea mucho mayor que el número de usuarios registrados.

Cuando un usuario se autentica en la plataforma AGORAV, esta no es capaz de identificar cuál de las dos técnicas utilizar para generar recomendaciones. Si esta elección no es correcta entonces la

recomendación no tendrá la calidad requerida. Por lo que se plantea como **problema a resolver**: ¿Cómo identificar que técnica de recomendación arroja mejores resultados para el usuario activo?

Para darle solución a dicho problema se propone como **objetivo general**: Implementar un plugin para el Sistema Gestor de Procesos de Media que permita obtener recomendaciones a partir de una hibridación de los sistemas de recomendación basados en contenido y colaborativos. Por lo cual se define como **objeto de estudio**: los sistemas de recomendación híbridos, enmarcados en el **campo de acción**: los sistemas de recomendación híbridos para la plataforma AGORAV.

Como guía para el desarrollo del objetivo general se plantean las siguientes **tareas de la investigación**:

1. Caracterización de los diferentes tipos de sistemas de recomendación.
2. Análisis del campo de acción para la concepción asociada a la fundamentación teórica de la investigación.
3. Definición de las tecnologías y los requerimientos funcionales necesarios para la elaboración de la propuesta de solución.
4. Diseño e implementación de la propuesta de solución.
5. Elaboración de los artefactos asociados a la propuesta de solución.
6. Realización de las pruebas necesarias para validar la propuesta de solución.

**Métodos teóricos:**

Los métodos teóricos permiten revelar las relaciones esenciales del objeto de investigación, no observables directamente. Participan en la etapa de asimilación de hechos, fenómenos y procesos y en la construcción del modelo e hipótesis de investigación (Alvarez de Zayas, y otros, 2000). A continuación se detallan los métodos empleados:

- **Analítico – sintético**: la utilización de este método permitió la comprensión y funcionamiento de los sistemas de recomendación híbridos a través del análisis de documentos, libros,

artículos y otras fuentes bibliográficas de diferentes autores, lo que permitió definir las características principales de estos sistemas.

- **Inductivo – deductivo:** este método permitió caracterizar los componentes fundamentales de los sistemas de recomendación híbridos existentes en el mundo, para así deducir el algoritmo que mejor se ajusta a las necesidades de la investigación.
- **Modelación:** con este método se realizó el proceso de diseño mediante la abstracción de sus elementos fundamentales utilizando un lenguaje de modelado unificado.

## Capítulo 1 Fundamentación Teórica.

En el presente capítulo se abordan conceptos asociados al dominio del problema, se hace un análisis de las principales características de los sistemas de recomendación de filtrado colaborativo y de filtrado basado en contenido. Además de reflejar un conjunto de particularidades de las tecnologías, metodología y herramientas a utilizar.

### 1.1 Conceptos asociados al dominio del problema.

Para una mejor comprensión del problema se hace necesaria la definición de algunos conceptos fundamentales a tener presente en el desarrollo de un sistema de recomendación híbrido.

#### 1.1.1 Recomendación.

*“Recomendación es autoridad, representación o calidad por la que algo se hace más apreciable y digno de respeto”* (RAE, 2013). *“Recomendación es la acción y la consecuencia de recomendar (sugerir algo, brindar un consejo). Una recomendación, por lo tanto, puede tratarse de una sugerencia referida a una cierta cuestión”* (Definición, 2008). Recomendación es un consejo o sugerencia que se le ofrece a una persona, con el objetivo de ayudarla a solucionar un problema o a tomar una decisión.

#### 1.1.2 Sistemas de Recomendación.

Diferentes autores han establecido definiciones acerca del concepto de sistema de recomendación, por ejemplo: *“Los sistemas de recomendaciones son herramientas que generan recomendaciones sobre un determinado objeto de estudio, a partir de las preferencias y opiniones dadas por los usuarios”* (Herrera-Viedma, y otros, 2012).

*“Un sistema de recomendación es un amigo virtual cuyos gustos son una mezcla de los gustos de miembros de la comunidad de usuarios, con gustos similares a los de uno mismo”* (Galán Nieto, 2007).

Los sistemas de recomendación son algoritmos diseñados para ayudar a los usuarios en la búsqueda de información de su interés. Estos se apoyan en el perfil del usuario, creado con las evaluaciones hechas por el propio usuario.

Algunos aspectos que se deben considerar sobre el diseño de los Sistemas de Recomendación son: (Resnick, y otros, 1997)

- **Representación de las recomendaciones:** Los contenidos de una evaluación pueden venir dados por un único bit (recomendado o no) o por comentarios de texto sin estructurar.
- **Expresión de las recomendaciones:** Las recomendaciones pueden ser introducidas de forma explícita o implícita.
- **Aspectos de identificación de la fuente:** Las recomendaciones pueden realizarse de forma anónima, identificando la fuente, o usando un pseudónimo.
- **Forma de agregar las evaluaciones:** Se refiere a cómo se van a ir agregando las evaluaciones disponibles sobre los ítems<sup>1</sup> de cara a generar las recomendaciones.
- **Uso de las recomendaciones:** Las recomendaciones se pueden usar de distintas formas. Por ejemplo, se podrían mostrar los ítems en forma de lista ordenada según las recomendaciones asociadas a cada uno, o que a la hora de visualizarlos se muestre además su recomendación.

### **1.1.3 Sistema de Recomendación de Filtrado Colaborativo.**

Los sistemas de recomendación colaborativos son aquellos en donde la recomendación se realiza basándose en un conjunto de similitudes entre usuarios. Si a un conjunto del grupo de usuarios les gusta un determinado elemento, es de suponer que ese mismo elemento guste al resto de los usuarios de ese grupo (Palomares Carrascosa, 2009).

---

<sup>1</sup> Según las bibliografías estudiadas, ítem es utilizado como sinónimo de objeto.

Galán Nieto dice que un sistema de recomendación basado en filtrado colaborativo “*encuentra usuarios con gustos similares a los de otro determinado y recomendar a este cosas que desconoce pero que gustan a aquellos con los que se tiene similitud*”. Además de ser “*un amigo virtual cuyos gustos son una mezcla de los gustos de miembros de la comunidad de usuarios con gustos similares a los de uno mismo*” (Galán Nieto, 2007).

Un sistema de recomendación basado en filtrado colaborativo encuentra gustos similares en los usuarios del sistema. Esta información la utiliza para crear grupos de usuarios y de esta forma recomendarle a un usuario específico los objetos que mejor han sido evaluados por su grupo.

### **1.1.4 Sistema de Recomendación de Filtrado Basado en Contenido.**

Los sistemas de recomendación basados en contenido aprenden del perfil del usuario, en función de las características presentes en los objetos que el usuario haya evaluado y tratan de recomendar productos similares a los que le han gustado en el pasado. Estos sistemas emplean valoraciones de los usuarios sobre los productos tomando a cada usuario como un individuo aislado. La información de los productos puede ser de distinta naturaleza (Burke, 2002).

Los sistemas de recomendación basados en el contenido generan recomendaciones teniendo en cuenta la valoración que el usuario ha hecho de los ítems en el pasado. Así pues, el sistema tiene en cuenta las valoraciones que el usuario realiza de los ítems y las utiliza para recomendar los nuevos que tienen características similares y han sido evaluados positivamente por el usuario (Seguido Font, 2009).

Este sistema crea un perfil de usuario basado en las características de los ítems que han sido evaluados por el propio usuario en el pasado. A través de este perfil recomienda ítems con características similares a los ya evaluados. Este sistema con el tiempo va construyendo un perfil de usuario más específico, lo que trae consigo que las recomendaciones tengan más probabilidades de ser certeras.



### 1.1.5 *Sistemas de Recomendación Híbridos.*

Los sistemas híbridos explotan características de los sistemas Basados en Contenido y Colaborativos, debido a la naturaleza complementaria de ambos. Lo que se busca es sobrellevar los inconvenientes de ambos sistemas para obtener mejores recomendaciones (Huecas, y otros, 2010).

Existen diferentes técnicas de hibridación utilizadas para compensar las debilidades de una técnica de recomendación con las fortalezas de otra; con el fin de mejorar la calidad de la recomendación que se va a presentar al usuario. Las diferentes técnicas utilizadas son (Burke, 2002):

- **Weighted (Ponderado):** los resultados de varias recomendaciones se combinan en una sola respuesta.
- **Switching (Cambio):** el sistema cambia de técnica de recomendación de acuerdo a la situación.
- **Mixed (Mixto):** se dan los resultados de diferentes sistemas de recomendación al mismo tiempo.
- **Feature Combination (Combinación de características):** las recomendaciones de diferentes orígenes de datos se combinan en un único algoritmo de recomendación.
- **Cascade (Cascada):** un sistema de recomendación refina la salida de otro.
- **Feature Augmentation (Aumentar las características):** la salida de un sistema de recomendación, se usa como característica de entrada de otro.
- **Meta-level (Meta-nivel):** el modelo aprendido por un sistema de recomendación, se usa como entrada para otro.

Es importante considerar que algunas de las estrategias de los sistemas híbridos, díganse: ponderado, cambio, mixto y combinación de características poseen la propiedad asociativa, lo que permite aplicar los diferentes sistemas de recomendación en cualquier orden sin alterar el resultado de las recomendaciones, sin embargo, en las restantes como son: cascada, aumento de

características y meta-nivel, se debe tener en cuenta el orden de los algoritmos a aplicar, pues esto implicaría resultados diferentes (Rojas Castellanos, 2013).

Se selecciona una técnica asociativa con vista a integrar nuevos sistemas de recomendaciones en futuras versiones, sin tener que prestar atención al orden en que se ejecutan con respecto a los ya implementados.

## **1.2 Análisis del sistema de recomendación de filtrado colaborativo en AGORAV.**

*“Las técnicas de filtrado colaborativo son un método de recomendación especialmente popular, gracias en gran medida a los buenos resultados obtenidos en dominios como el comercio electrónico. Al estar basadas en las preferencias de otros usuarios con gustos similares, la calidad de las recomendaciones es particularmente elevada, pues al final son personas quienes evalúan cada producto”* (Formoso López, 2013).

Estos sistemas pueden ser agrupados en dos clases generales (Castellano Torres, 2007):

- **Algoritmos basados en memoria** (o en vecindad, o heurísticos).
- **Algoritmos basados en modelos.**

El sistema implementado en la plataforma AGORAV utiliza el algoritmo basado en memoria. Estos algoritmos son esencialmente heurísticos que realizan predicciones basadas en una colección completa de ítems valorados previamente por el usuario. Es decir, el valor de una puntuación no conocida ‘ $v$ ’ para un usuario ‘ $u$ ’ sobre un ítem ‘ $i$ ’ se calcula como un agregado de las valoraciones de otros usuarios (generalmente, los  $K$  más parecidos) para el mismo ítem ‘ $i$ ’ (Castellano Torres, 2007).

La información que se está manejando consta de una serie de ítems, una serie de usuarios, y una serie de valoraciones de estos usuarios sobre aquellos ítems, se puede concluir que el espacio del problema viene definido como una matriz de usuarios frente a ítems, en la que cada celda representa la puntuación de un usuario referida a un ítem específico (Tabla 1) (Sarwar, y otros, 2000).

Tabla. 1: Representación del espacio de un problema asociado al filtrado colaborativo (Castellano Torres, 2007).

<b>Usuarios- ítems</b>	<b>Dire Straits</b>	<b>Metallica</b>	<b>The Beatles</b>	<b>Depeche Mode</b>	<b>Bee Gees</b>
<b>Alex</b>	4	1	9	6	10
<b>Ricardo</b>	7	9	6	10	
<b>Eva</b>	5	2	9	7	8
<b>Pedro</b>	6	3		6	7

En base a esto, encontrar buenos ítems, implica predecir el grado en el que un ítem gustará a un usuario que no tiene puntuación asociada a ese ítem, mediante el uso de una serie de valoraciones aportadas anteriormente por un grupo de usuarios (Castellano Torres, 2007).

Para realizar esta predicción el algoritmo de filtrado colaborativo cuenta con 3 partes principales (Castellano Torres, 2007):

- **Establecer similitud entre vecinos** (matriz usuarios-ítems).
- **Selección de vecinos.**
- **Realizar predicción.**

Para establecer la similitud entre vecinos, se tiene que definir una medida que nos permita evaluar el grado de parecido que tienen entre ellos. Para realizar este cálculo se utiliza el **coeficiente de coseno**. Este método supone que la similitud entre dos usuarios representados como vectores en el espacio viene dado por el coseno del ángulo que forman. Hay que destacar que en este algoritmo los valores que se utilizan son solo los positivos (Albín Rodríguez, 2009). La expresión para su cálculo es la siguiente:

$$s(u, x) = \cos(\vec{v}_u, \vec{v}_x) = \frac{\sum_j (v_{u,j}) * (v_{x,j})}{\sqrt{\sum_j (v_{u,j})^2} \sqrt{\sum_j (v_{x,j})^2}}$$

Fig. 1: Coeficiente del coseno

Para la selección de los vecinos se utiliza el método **K-nn** (K nearest neighbors) o de vecinos más cercanos. Actualmente este valor es fijo, K = 5.

Una vez se hallan elegido los vecinos más cercanos, se deben combinar las predicciones de esa vecindad para realizar una recomendación. La forma más sencilla de combinar estas puntuaciones es calcular una media con las valoraciones, sin embargo una de las aproximaciones más utilizadas es calcular una **suma media ajustada** de las puntuaciones, utilizando las correlaciones como pesos. Este sistema utiliza la suma media ajustada (WA – weighted adjusted sum) porque tiene en cuenta que diferentes usuarios pueden utilizar diferentes escalas de puntuación, es decir, diferentes usuarios pueden puntuar siempre o muy alto, o muy bajo, o siempre en los extremos, o sólo términos medios. Para ello en WA se utilizan desviaciones de la media para el usuario correspondiente (Castellano Torres, 2007). Para este cálculo se utiliza la siguiente fórmula:

$$p(u, i) = \overline{v(u)} + \frac{\sum_{n=1}^k s(u, x) (v(x, i) - \overline{v(x)})}{\sum_{n=1}^k |s(u, x)|}$$

Fig. 2: Suma media ajustada

Al terminar los cálculos, las recomendaciones generadas por el mismo son guardadas en la base de datos. Cuando un usuario se autentica en el sistema, son mostradas las recomendaciones que ya han sido calculadas previamente y guardadas en la base de datos.

### 1.3 Análisis del sistema de recomendación de filtrado basado en contenido en AGORAV.

El sistemas de recomendación de filtrado basado en contenido que utiliza AGORAV recomienda ítems que son similares a los ítems que previamente valoró el usuario otorgándoles un valor de 1 a 5 a la media que votó (Ver Tabla 2).

Tabla. 2: Ejemplo de Usuarios otorgándole una votación a un archivo multimedia.

Usuario/Media	The Croods	El Vuelo	Jack Reacher	Río 2
<b>Thalia</b>		3	4	5
<b>Yandy</b>		2	5	5
<b>Oniel</b>	2	4	2	
<b>Danger</b>	4		5	3

Estos ítems se definen con un valor de importancia calculado a través de sus descriptores. El método utilizado para calcular el valor de importancia es:

- **Reglas de Asociación:** Se intenta descubrir relaciones entre los ítems, que luego pueden ser utilizadas para recomendación. Los ítems se comparan en base al comportamiento pasado de los usuarios respecto a ellos (Seguido Font, 2009). Además permite que los ítems tengo un número variado de atributos o características.

A continuación se explica cómo es aplicada la regla de asociación al sistema de recomendación basado en contenido.

En la Tabla 3 se tiene las medias que han sido votadas por el usuario y sus descriptores.

Tabla. 3: Tabla de los archivos multimedia con sus descriptores y la votación recibida por parte del usuario a recomendar.

Id_media	Lista de los descriptores del archivo multimedia	Votación
1	Título: Jurassic World, Director: Steven Spielberg, Actor: Chris Pratt, País: Estados Unidos, Género: Aventuras, Género: Ciencia ficción	5

2	Título: Transformers 4, Director: Steven Spielberg, Actor: Mark Wahlberg, País: Estados Unidos, Género: Ciencia Ficción	4
3	Título: Lone Survivor , Director: Peter Berg, Actor: Mark Wahlberg, País: Estados Unidos, Género: Thriller	2

- **Id\_media:** identificador del archivo multimedia votado por el usuario.
- **Listado de los descriptores:** lista de todos los descriptores que se tienen del archivo multimedia. Ejemplo: Tipo de archivo multimedia: película, Director: Steven Spielberg, etc.
- **Votación:** es el valor que el usuario le otorga al archivo multimedia.

Se quiere obtener como resultado un conjunto de archivos multimedia que sean de su agrado utilizando la expresión: X -> Y donde 'X' es el consecuente y 'Y' el resultante.

Medidas de evaluación de las reglas de asociación:

- **Soporte de las reglas:** la cantidad de veces que aparece el descriptor entre el total de los descriptores.
- **Confianza de las reglas:** suma de las votaciones dadas al descriptor entre la cantidad de veces que aparece el descriptor.

En la tabla 4 aparece un ejemplo del cálculo del valor de los descriptores.

Tabla. 4: Tabla para calcular el valor de importancia de un descriptor.

Nombre del Descriptor	Votaciones dadas al descriptor	Confianza	Soporte	Valor de importancia
Director: Steven Spielberg (A)	5,4	4.5	0.15	4.65
Actor: Chris Pratt(B)	5	5	0.08	5.08
País: Estados Unidos(C)	5,4,2	3.67	0.23	3.9
Género: Aventuras(D)	5	5	0.08	5.08

Género: Ciencia ficción(E)	5,4	4.5	0.15	4.65
Actor: Mark Wahlberg(F)	4,2	3	0.15	3.15
Director: Peter Berg(G)	2	2	0.08	2.08
Género: Thriller(H)	2	2	0.08	2.08

Calcular el valor de importancia de un conjunto de archivos multimedia no vistos por el usuario.

Tabla. 5: Tabla para el valor de importancia de un archivo multimedia.

Id_media	Lista de los descriptores	T	Descriptores comunes	S	Valor de Importancia
4	W,C,F,A	4	$C(3.9)+F(3.15)+A(4.65)$	11.7	2,925
5	L,M,N	3		0	0
6	Z,B,D,A,H	5	$B(5.08)+D(5.08)+A(4.65)+H(2.08)$	21.54	4,308
7	O,P,Q,A	4	$A(4.65)$	4.65	1,1625
8	A,B,C,D,E	5	$A(4.65)+B(5.08)+C(3.9)+D(5.08)+E(4.65)$	23.36	4,672

- **Id\_media:** identificador del archivo multimedia.
- **T:** cantidad de descriptores de la media.
- **Descriptores comunes:** son los descriptores que ya han sido votados por el usuario en una o varias ocasiones.
- **S:** suma de todos los descriptores comunes.
- **Valor de Importancia:** es la división de  $S / T$ .

Al finalizar el algoritmo se tienen todas las medias con su valor de importancia, solo queda elegir los 'n' archivos multimedia con mayor valor y mostrarlos.

## **1.4 Sistemas de Recomendación existentes.**

En este epígrafe se hace un estudio a grandes rasgos de las principales características de algunos sistemas de recomendaciones híbridos.

### **1.4.1 REJA.**

Es un sistema de recomendación de restaurantes que está orientado a usuarios que, ante la disyuntiva de qué restaurante elegir dentro de una determinada zona, facilitan algunas preferencias sobre restaurantes conocidos y el sistema les recomienda otros similares. REJA es un sistema de recomendación híbrido que combina dos modelos, por un lado un modelo colaborativo para usuarios habituales del sistema y por otro, un modelo basado en conocimiento para aquellos usuarios que han interactuado poco con el sistema. Este sistema de recomendación es capaz de generar recomendaciones a partir de las descripciones de un restaurante propuesto por el usuario, comparándolo con el resto de restaurantes almacenados en la base de datos, utilizando técnicas y mecanismos pertenecientes al ámbito de la lógica difusa (Barranco, 2008).

La hibridación en este sistema es entre un modelo colaborativo y uno basado en conocimiento con información lingüística. Además es un sistema específicamente diseñado para recomendar restaurantes, por lo que no es factible utilizar este sistema.

### **1.4.2 FAB.**

Balavanovic y Shoham en 1997 crearon un sistema híbrido, basado en contenido y filtrado colaborativo, que recomienda al usuario páginas Web (Fab). Combinando ambas técnicas Fab reduce las limitaciones de cada una de ellas. El sistema modela el perfil del usuario basado en el contenido de los análisis (cuando un usuario califica una página) y compara estos perfiles para determinar similitudes entre usuarios para una recomendación colaborativa. De esta manera el usuario recibirá páginas, tanto las que ha calificado relevantes (con respecto a su perfil) como las que han recibido calificaciones altas por usuarios con un perfil similar al suyo (vecinos cercanos). La arquitectura de Fab se encuentra formada por tres componentes principales: agentes de colección, que seleccionan páginas de un tema específico; agentes de selección, los cuales encuentran páginas



para un usuario específico, y un router central que los conecta. Cada agente mantiene un perfil del usuario basado en las palabras que contienen las páginas Web, que el usuario ha calificado (Fernández Ramírez, 1998).

Este sistema recomienda específicamente páginas web por lo que tampoco es factible, ya que el sistema que se desea implementar debe recomendar archivos multimedia, aunque el análisis de este sistema permitió incorporar la creación de grupos de usuarios con el sistema basado en contenido para generar las recomendaciones en el sistema basado en filtrado colaborativo.

En este sistema se tiene una hibridación entre un basado en contenido y filtrado colaborativo para la recomendación de páginas web. De este sistema se toma la creación de grupos de vecinos a través de los perfiles de los usuarios generados a partir de las características de las páginas web que han sido evaluadas por el mismo.

### ***1.4.3 Módulo de recomendación híbrida para la plataforma Inter-nos 2.0.***

Este módulo combina los sistemas de recomendación basado en contenido y filtrado colaborativo aplicando diferentes técnicas de hibridación, que son configuradas por el administrador del sistema para cada usuario. El sistema analiza los datos asociados al perfil del usuario y al historial de navegación del usuario, almacenados en la base de datos. Luego realiza una intersección entre ellos y selecciona el contenido audiovisual resultante de dicha intersección, ordenándolo descendientemente según la cantidad de veces que han sido visualizados. El sistema emite un informe conformado por tres partes. En la primera se encuentran el id de los archivos multimedia resultantes de la intersección entre el perfil y el historial del usuario, en las otras dos los ids de los archivos multimedia no presentes en la intersección. Después se analizan los grupos de contenido multimedia que se corresponden más con las preferencias del usuario para obtener un listado de todos los archivos multimedia del grupo que no han sido valorados por el usuario.

Al ser integrado a la plataforma Inter-nos 2.0 el módulo de recomendación híbrida atentó contra el rendimiento de la misma, ya que los algoritmos empleados realizan cálculos complejos que consumen muchos recursos de cómputo. Además de que a esta plataforma concurren frecuentemente un número considerable de usuarios.

De lo anterior se concluye que es muy ventajoso la hibridación entre un sistema de recomendación basado en contenido y uno de filtrado colaborativo. También se concluye que para garantizar un mejor rendimiento de la plataforma es valorable que la implementación del módulo de recomendación híbrido se convierta en un plugin para el SGPM lo que evitaría que el cálculo de las recomendaciones se realizara sobre la plataforma.

### **1.5 Metodologías de desarrollo.**

En la actualidad existen diversas herramientas, procesos y métodos que permiten simplificar y agilizar la construcción de un software. Las metodologías de desarrollo son específicamente las guías que detallan paso a paso el proceso de conformación de un producto informático.

Partiendo del hecho que se tiene experiencia en el desarrollo de sistema de recomendación, la criticidad no es alta y las funcionalidades del sistema pueden ir cambiando paulatinamente se decide utilizar un enfoque de gestión ágil.

Dentro de las metodologías ágiles destaca Programación Extrema (XP) la cual se recomienda para proyectos de corto plazo y Open Up que se centra en articular la arquitectura para facilitar la colaboración técnica. XP plantea que se trabaje sobre la base de la colaboración mutua con el cliente y donde no exista más documentación que el propio código. Este proceso define cuatro actividades del marco de trabajo: planeación, diseño, codificación y prueba (Ver Fig. 3).

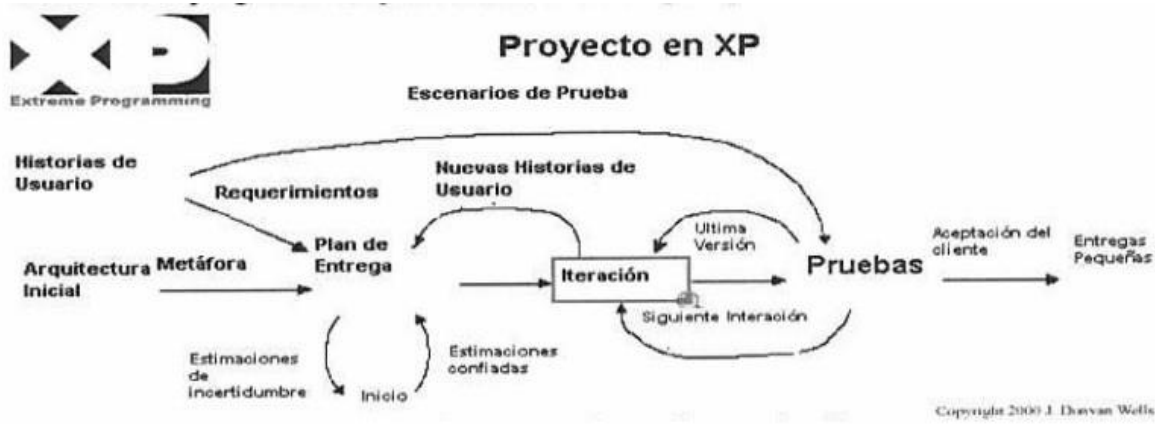


Fig. 3: Ciclo de vida de la metodología XP (Sánchez González, 2004).

La metodología Open Up es un proceso unificado que aplica propuestas de gestión ágil, tratando de ser manejable en relación con RUP pues mantiene sus características esenciales: centrado en la arquitectura, dirigido por Casos de Uso (CU) y un desarrollo iterativo e incremental, dentro del ciclo de vida de un proyecto de software. El ciclo de vida en Open Up se comporta como se muestra a continuación:

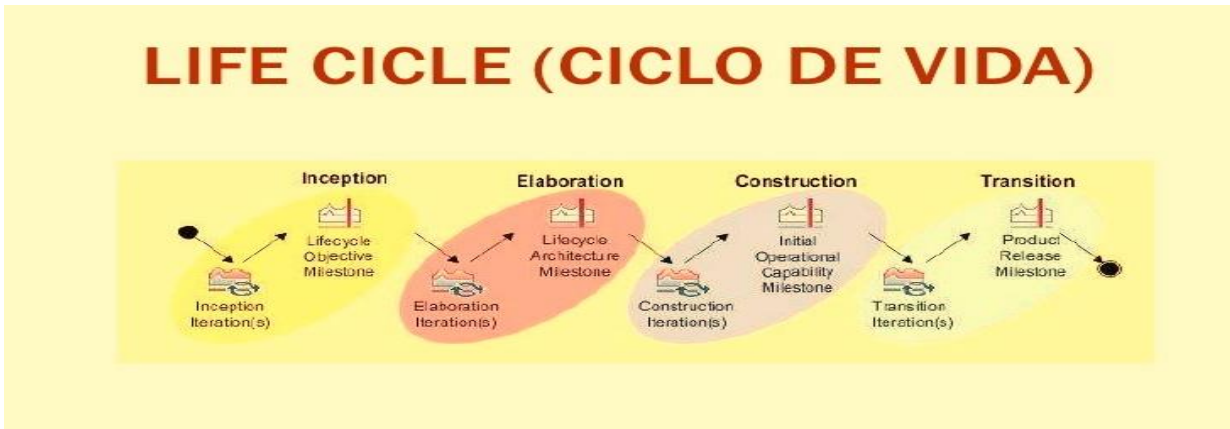


Fig. 4: Ciclo de vida de la metodología Open UP (Eclipse, 2015).

Para un mejor desarrollo del producto se selecciona Open Up, primero porque XP es muy costosa en tiempo en caso de fallar, en tanto Open Up permite disminuir las probabilidades de fracaso e incrementar las probabilidades de éxito, y segundo esta metodología de desarrollo de software de

código abierto está diseñada para equipos pequeños. Además es preciso centrarse en la arquitectura para lograr una eficiente integración de la solución que se propone y que esta sea escalable y flexible, también aporta un desarrollo iterativo e incremental.

## **1.6 Selección de Herramientas.**

En este epígrafe se mencionan las herramientas seleccionadas, entre las que se encuentra el lenguaje de modelado y la herramienta CASE (del inglés CASE Herramienta de Ingeniería de Software Asistida por Computadora), los entornos de desarrollo integrado, los lenguajes de programación, el sistema gestor de base de datos y el sistema gestor de contenidos.

### ***1.6.1 Lenguaje de Modelado Unificado.***

Se selecciona el Lenguaje Unificado de Modelado (UML por sus siglas en inglés) ya que es el más conocido y utilizado en la actualidad. Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Su objetivo es lograr que, además de describir con cierto grado de formalismo tales sistemas, puedan ser entendidos por los usuarios de aquello que se modela (Varas Contreras, y otros, 2003).

### ***1.6.2 Herramienta CASE Visual Paradigm.***

Definido el lenguaje de modelado, se selecciona como herramienta CASE Visual Paradigm. Visual Paradigm es una herramienta CASE de modelado del visual UML, de plataforma vertical, multiplataforma y fácil de utilizar. Facilita la interoperabilidad con otras herramientas CASE y la mayoría de los principales IDE (Entorno de Desarrollo Integrado) lo cual resalta el proceso entero de desarrollo Modelo-Código-Despliegue (Visual\_Paradigm, 2014). Esta herramienta permitirá un diseño organizado de todos los artefactos generados en el proceso de desarrollo del sistema.

### ***1.6.3 Lenguaje de programación.***

Se selecciona el lenguaje de programación C++ para el desarrollo de esta investigación debido a que el SGPM y todas los plugins desarrollados para él están implementados en este lenguaje y para

lograr compatibilidad es necesario que este plugin también lo esté. Además C++ es un lenguaje de alto rendimiento lo cual es factible para el sistema, ya que todos los plugins implementados en él SGPM son de procesamiento.

### **1.6.4 Entorno Integrado de Desarrollo.**

Qt-Creator es un entorno de desarrollo integrado que proporciona herramientas para diseñar y desarrollar aplicaciones e interfaces de usuarios una vez y desplegarlas a través de varios sistemas operativos móviles, de escritorio y la implementación de plugins. Entre las principales características de Qt-Creator se puede encontrar que posee un avanzado editor de código C++, posee ayuda sensible al contexto, depurador visual, resaltado y autocompletado de código y proporciona herramientas para llevar a cabo sus tareas a lo largo de todo el ciclo de vida de desarrollo de la aplicación (Qt\_Project, 2015). Basado en las características y ventajas antes vistas y teniendo en cuenta que el lenguaje de programación utilizado es C++, se utiliza Qt-Creator en su versión 4.8 para el desarrollo del plugin.

### **1.6.5 Sistema Gestor de Base de Datos PostgreSQL.**

PostgreSQL es un sistema de base de datos libre y de código abierto, que se caracteriza por brindar estabilidad, potencia y facilidad de administración. Entre las propiedades más destacadas se encuentra el Acceso Concurrente Multiversión el cual permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos (Martinez, 2009). Se elige trabajar con PostgreSQL 9.1 por ser una herramienta pública y multiplataforma, además de tener en cuenta que la plataforma AGORAV trabaja utilizando dicho gestor.

## **Conclusiones del capítulo.**

En este capítulo se analizaron los principales conceptos asociados a los sistemas de recomendación. Además de un estudio de los sistemas de recomendación colaborativos y basados en contenido, donde se expusieron las principales ventajas y desventajas de estos sistemas. Además se expusieron características generales de sistemas de recomendación existentes en el mundo, que no representan una solución al problema pero fue la base para elaborar una idea general de cómo se

desea representar la solución final.

## Capítulo 2 Análisis y diseño del sistema de recomendación híbrido.

El siguiente capítulo tiene el objetivo de presentar la descripción del modelo de dominio, se definirán los requisitos funcionales y no funcionales que deberá cumplir el sistema propuesto. Además se generan los diagramas de casos de usos del sistema, los de secuencia y así como los de clases del diseño.

### 2.1 Modelo de dominio.

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. El modelo del dominio muestra las clases conceptuales o vocabulario del dominio, muestra una vista parcial, o abstracción, e ignora detalles sin interés. Por tanto, el modelo del dominio podría considerarse como un diccionario visual de las abstracciones relevantes, vocabulario del dominio e información del dominio (Larman, 2003).

A continuación, la Fig. 5 presenta el modelo de dominio de la presente investigación:

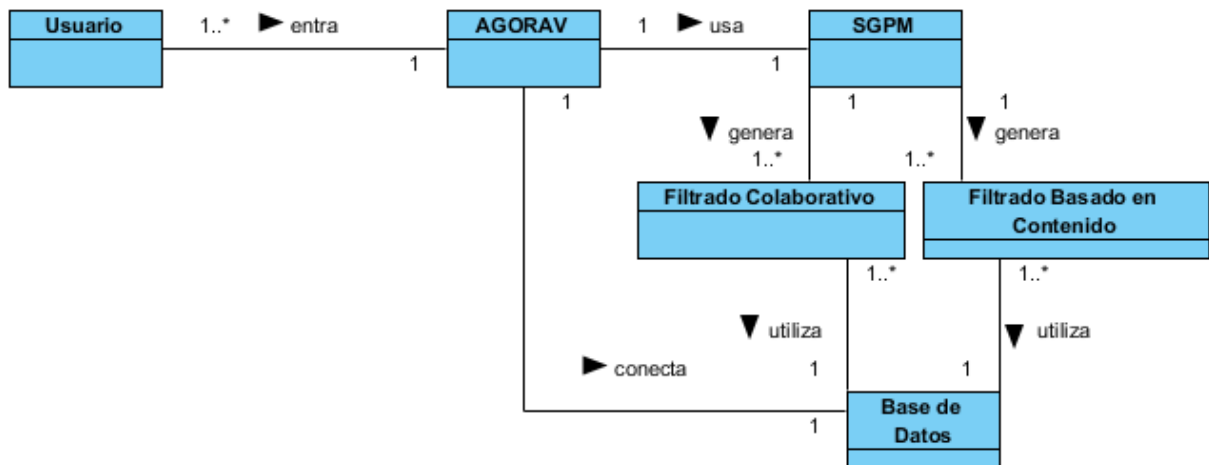


Fig. 5: Modelo de dominio.

El usuario se conecta en la plataforma AGORAV y esta le notifica al SGPM que debe calcular las recomendaciones del usuario conectado. El SGPM genera las recomendaciones mediante un filtrado colaborativo o un filtrado basado en contenido sobre las medias publicadas en la plataforma que no han sido vistas por el usuario las cuales están almacenadas en la base de datos.

- **Usuario:** persona que visita la plataforma en busca de un archivo multimedia de su agrado.
- **AGORAV:** plataforma de publicación de archivos multimedia desarrollada en el centro productivo GEYSED.
- **SGPM:** Sistema Gestor de Procesos de Media, es una aplicación que permite integrar aplicaciones con diferentes tecnologías.
- **Filtrado Colaborativo:** algoritmo para realizar recomendación utilizando solo las características del usuario al que se le realiza la recomendación.
- **Filtrado Basado en Contenido:** algoritmo para realizar recomendaciones utilizando un conjunto de usuarios que son muy parecidos al usuario que se le desea hacer la recomendación.
- **Base de Datos:** sistema que se utiliza para almacenar datos de forma organizada y de fácil acceso.

## 2.2 Requisitos.

Los requisitos se han convertido en un punto clave en el desarrollo de las aplicaciones informáticas. Para garantizar el éxito en cierta medida de un proyecto de software hay que realizar una buena definición, especificación o administración de requisitos.

### 2.2.1 *Requisitos funcionales.*

Los requisitos funcionales son declaraciones de los servicios que debe propiciar el sistema o sea definen las capacidades, condiciones o funciones que el sistema debe ser capaz de satisfacer (Sommerville, 2007). A continuación se detallan los requisitos funcionales del sistema a construir:



**RF1- Obtener el historial del usuario.** El sistema para generar una recomendación debe obtener todo el historial del usuario.

**RF2- Generar la lista de recomendaciones.** El sistema debe ser capaz de generar/crear una lista de recomendaciones para cada usuario.

**RF3- Guardar las recomendaciones.** El sistema debe ser capaz de guardar las recomendaciones generadas.

**RF4- Actualizar el historial del usuario.** El sistema debe ser capaz de actualizar el historial del usuario cada vez que este haga un cambio.

### **2.2.2 Requisitos no funcionales.**

Los requisitos no funcionales por lo general se aplican a sistemas como un todo. Estos representan limitaciones en los servicios o funciones que ofrece el sistema (Sommerville, 2007). Esta investigación arrojó como requisitos no funcionales los siguientes:

#### **RNF1- Requisitos de apariencia o interfaz externa.**

Las recomendaciones deben aparecer automáticamente una vez que el usuario se haya autenticado, visible en todo momento por el usuario en la parte inferior de la interfaz visual de la plataforma.

#### **RNF2- Restricciones del diseño e implementación.**

- Qt-Creator 4,8 como entorno de desarrollo integrado.
- PostgreSQL 9,1 como sistema gestor de base de datos.
- Visual Paradigm for UML 8,0 como herramienta CASE.
- C++ como lenguaje de programación utilizado en el desarrollo del plugin.

#### **RNF3- Software.**

PC Servidor:

- Sistema Operativo GNU/Linux Ubuntu10.4 o superior.
- Plataforma de procesamiento<sup>2</sup>.
- PostgreSQL 9.1.

### 2.3 Descripción del sistema de recomendación híbrido.

Este algoritmo se basa en los descriptores de los ítems, dándole un valor de importancia a cada uno de estos según los usuarios que han votado por ellos. Las 4 etapas por las que pasa este algoritmo son las siguientes:

1. La primera etapa es el cálculo de los valores de importancia de los descriptores. Este proceso se ejecuta cada vez que un usuario realiza una votación por una media.

Para calcular el valor de importancia se utiliza el algoritmo de **Reglas de Asociación** explicado en el **epígrafe 1.3**.

Los valores calculados a los descriptores son almacenados en una tabla “descriptores”. Esta tabla se actualiza cada vez que el usuario vota por una media. Este proceso ayuda a que una parte del algoritmo ya esté pre-calculado, por tal motivo el tiempo de ejecución del algoritmo disminuye.

2. Cálculo de las recomendaciones basadas en contenido, aquí solo se utilizan las características del propio usuario a recomendar.

Este algoritmo toma los valores pre-calculados en la primera etapa para darle un valor de importancia a todas las medias que están publicadas en la plataforma, ordenándolas de forma descendente y tomando las 10 primeras.

3. Cálculo de las recomendaciones colaborativas, en este sistema de recomendación se utilizó las características de un grupo de vecinos del usuario a recomendar.

Este algoritmo tiene tres fases de ejecución:

---

<sup>2</sup> Es el sistema donde se instalan los plugins de procesamientos.

## Capítulo 2: Análisis y diseño del sistema de recomendación híbrido.

- La primera fase es encontrar la similitud de cada uno de los usuarios de la plataforma con el usuario a recomendar.

Tomamos los descriptores y los valores de los descriptores del usuario a recomendar:

Tabla. 6: Tabla que muestra el valor de importancia de cada descriptor del usuario a recomendar.

<b>Nombre del Descriptor</b>	<b>Valor de importancia</b>
Director: Steven Spielberg	4.65
Actor: Chris Pratt	5.08
País: Estados Unidos	3.9
Género: Aventuras	5.08
Género: Ciencia ficción	4.65
Actor: Mark Wahlberg	3.15
Director: Peter Berg	2.08
Género: Thriller	2.08

También se buscan los descriptores del usuario a comparar:

Tabla. 7: Tabla que muestra el valor de importancia de los descriptores del usuario a comparar.

<b>Nombre del Descriptor</b>	<b>Valor de importancia</b>
Director: Steven Spielberg	4.6
Actor: Paul Walker	5.08
País: Estados Unidos	3.9
Género: Ciencia ficción	4.08
Actor: Tom Hanks	3.15
Director: Quentin Tarantino	2.08
Actor: Mark Wahlberg	3.08

Luego de establecer la comparación, se obtienen los siguientes valores:

Tabla. 8: Tabla de los descriptores comunes en los dos usuarios.

<b>Nombre del Descriptor</b>	<b>Valor de Importancia</b>	<b>Valor de Importancia</b>
------------------------------	-----------------------------	-----------------------------

	(Usuario a recomendar)	(Usuario a comparar)
Director: Steven Spielberg	4.65	4.6
País: Estados Unidos	3.9	3.9
Género: Ciencia ficción	4.65	4.08
Actor: Mark Wahlberg	3.15	3.08
<b>Total</b>	16.35	15.66

Estos valores son comparados y se dividen el mayor entre el menor. El valor obtenido de este proceso es el parecido que tienen ambos usuarios. Este proceso se realiza para todos los usuarios del sistema, obteniendo una lista con el id del usuario y el valor de parecido que tiene con el usuario a recomendar.

- La segunda fase es sacar los k-vecinos parecidos que hasta el momento es un valor fijo.
  - En la tercera y última fase se eligen los 10 archivos multimedia que han sido visualizados por la mayor cantidad de los k-vecinos del usuario.
4. En esta etapa se lleva la recomendación a todos los usuarios del sistema. Como el algoritmo se basa en los descriptores de los archivos multimedia, todos los usuarios que han realizado al menos una votación pueden darle a todas las medias publicadas una evaluación basada en sus descriptores. Esto quiere decir que se va a tener una tabla “medias” con los *ids* de las medias y el valor de importancia dado por todos los usuarios del sistema.
  5. Esta es la última etapa donde se realiza la hibridación de las recomendaciones dadas por las 3 etapas anteriores. La técnica a utilizar será “Ponderado”, es decir, escogiendo las 10 recomendaciones que más se repiten en los sistemas antes ejecutados.

### **2.3.1 Diagrama de caso de uso del sistema de recomendación Híbrido.**

Un modelo de caso de uso es una aproximación de las funciones previstas del sistema y su entorno. Son utilizados para comprender fácilmente los requisitos funcionales, los cuales definen las funcionalidades que serán inicializadas por uno de los actores que intervienen en el modelo (S. Pressman, 2010).

**Actor del sistema:**

Un actor del sistema es aquella persona, máquina o software que inicializa un caso de uso o sea que interactúa con el sistema. En la solución que se propone el actor sería el SGPM que es el responsable de generar las recomendaciones que se le mostrarán al usuario.

Tabla. 9: Tabla de Casos de Uso.

Referencia a requisitos	Nombre de los casos de uso	Prioridad
RF1	Obtener historial del usuario	Secundario
RF2, RF3	Generar recomendaciones	Crítico
RF4	Actualizar historial del usuario	Crítico

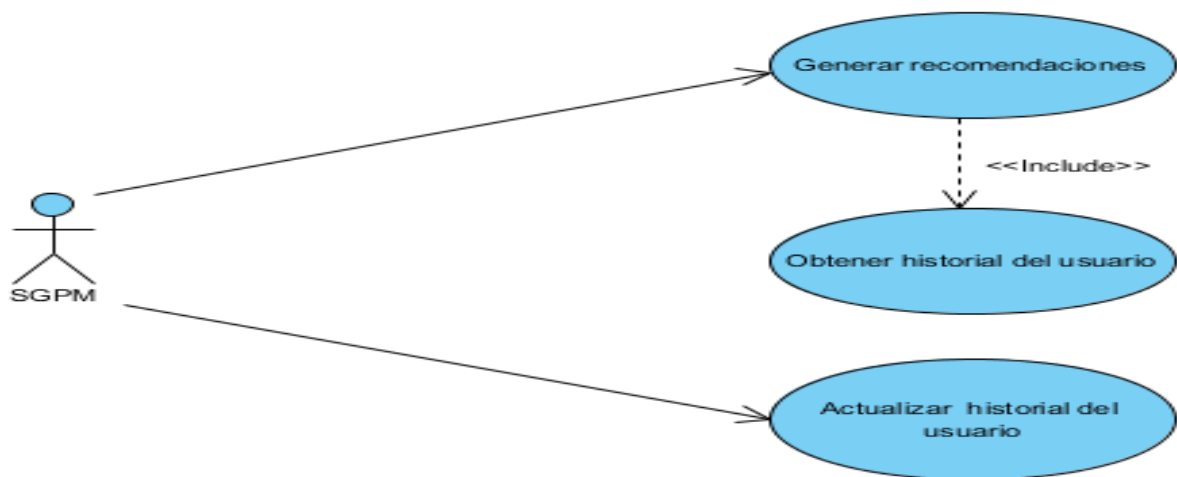


Fig. 6: Diagrama de Casos de Uso.

### 2.3.2 Descripción textual del caso de uso: Generar Recomendación.

El usuario se autentica en el sistema e inmediatamente se le muestran las recomendaciones. Si el usuario descarga, visualiza o vota por alguna media, se actualizan los valores de los descriptores de esa media al usuario que realizó la acción. Cuando el usuario sale del sistema, entonces se generan las recomendaciones del usuario y se guardan en la base de datos hasta que el usuario vuelva a entrar en el sistema.

Tabla. 10: Caso de uso crear las recomendaciones del usuario.

<b>Objetivo</b>	<i>Generar recomendaciones</i>
-----------------	--------------------------------

<b>Actores</b>	<i>SGPM (inicia)</i>	
<b>Resumen</b>	<i>Una vez que el SGPM es notificado de que un usuario se autenticó, el sistema debe generar las recomendaciones para ese usuario.</i>	
<b>Complejidad</b>	<i>Alta</i>	
<b>Prioridad</b>	<i>Crítico</i>	
<b>Precondiciones</b>		
<b>Postcondiciones</b>	<i>Se genera una lista de recomendaciones para el usuario</i>	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Generar Recomendaciones&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Notifica que se conectó un usuario	
2.		Obtiene los datos relacionados al historial de usuario existente en la base de datos.(Ver CU Obtener historial del usuario)
3.		Ejecuta el algoritmo de hibridación para generar la lista de recomendaciones del usuario.
4.		Obtiene los archivos multimedia que estén publicados y no hayan sido visualizados por el usuario activo.
5.		Realiza el cálculo de predicción para los archivos multimedia no vistos por el usuario.
6.		Genera una lista de recomendación para el usuario activo.
7.		Guarda las recomendaciones generadas en la base de datos.
<b>Flujos alternos</b>		
<b>Nº Evento &lt;No se generan recomendaciones&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
3.1		No guarda ninguna información en la base de datos.

<b>Relaciones</b>	<b>CU Incluidos</b>	Obtener historial del usuario
	<b>CU Extendidos</b>	No procede

## 2.4 Patrón arquitectónico

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica administrando una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada (Peláez, 2009).

En la presente investigación se selecciona el patrón en 3 capas. Garlan y Shaw definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior (Garlan, y otros, 1994). Dentro de las ventajas que destaca el uso de este patrón se encuentra la reutilización de capas, además facilita la estandarización y tiene contención de cambios de una o pocas capas.

Otras características que identifican al patrón seleccionado son que (Peláez, 2009):

- Describe la descomposición de servicios de forma que la mayoría de la interacción ocurre solamente entre capas vecinas.
- Las capas de una aplicación pueden residir en la misma maquina física (misma capa) o puede estar distribuido sobre diferentes computadores (n-capas).
- Los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces muy bien definidas.
- Este modelo ha sido descrito como una “pirámide invertida de re-uso” donde cada capa agrega responsabilidad y abstracción a la capa directamente sobre ella.

## 2.5 Patrones de diseño.

Christopher Alexander dice: “Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo ni siquiera dos veces de la misma forma” (Alexander, y otros, 1977). A pesar de que Alexander estaba hablando de patrones en edificios y ciudades, lo que dice es cierto acerca de los patrones de diseño orientado a objetos. Nuestras soluciones se expresan en términos de objetos e interfaces en lugar de paredes y puertas, pero en la base de dos tipos de patrones es una solución a un problema en un contexto (Gamma, y otros, 1994).

### 2.5.1 Patrones de diseño GoF.

- **Singleton:** El patrón singleton consiste en crear una instancia de un objeto y solo una, para toda nuestra aplicación. Sería como una especie de variable global que almacena el objeto (Escolar, 2012). En este caso es utilizado en la clase acceso a datos.
- **Observador:** Define una dependencia de uno-a-muchos entre objetos de forma que, cuando un objeto cambia de estado, se notifica a los objetos dependientes para que se actualicen automáticamente (Guerra Sánchez, 2008). El plugin de recomendación híbrida notifica al SGPM el porcentaje de ejecución que lleva la tarea a través de la clase **I\_Observador\_Ejecucion**.

### 2.5.2 Patrones de diseño GRASP.

- **Controlador:** El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es el controlador quien recibe los datos del usuario y quien los envía a las distintas clases según el método llamado (García Carmona , 2012). En la clase **CC\_Generar\_Recomendacion** se ve en evidencia su uso.
- **Experto:** El patrón experto dice que la responsabilidad de la creación de un objeto o la



implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada, es decir, disminuye el acoplamiento (García Carmona, 2012). En la clase **CC\_Generar\_Recomendacion** se ve en evidencia el uso de este patrón.

- **Creador:** El patrón creador nos ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Este patrón nos dice que la nueva instancia podrá ser creada por una clase si:
  - Contiene o agrega la clase.
  - Almacena la instancia en algún sitio (por ejemplo una base de datos).
  - Tiene la información necesaria para realizar la creación del objeto (es 'Experta').
  - Usa directamente las instancias creadas del objeto (García Carmona , 2012).

Este patrón es utilizado en la clase **CC\_Generar\_Recomendacion** al cual se le pasan por parámetros los datos necesarios.

- **Bajo Acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda, de tal forma que, en caso de producirse una modificación en alguna de ellas, tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases (García Carmona , 2012). Este patrón es usado en todo el sistema ya que el cambio en una clase afecta lo menos posible al resto.
- **Alta Cohesión:** Nos dice que la información que almacena una clase debe de ser coherente y debe estar, en la medida de lo posible, relacionada con la clase (García Carmona , 2012).

### 2.6 Diagrama de clase del diseño.

El diagrama de clase es el diagrama principal del diseño. En él la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Siempre buscando satisfacer las necesidades de implementación. A continuación se muestra el diagrama de clase del diseño del

plugin “**plugins\_Recomendacion\_Hibrida**” a implementar (Ver figura 5).

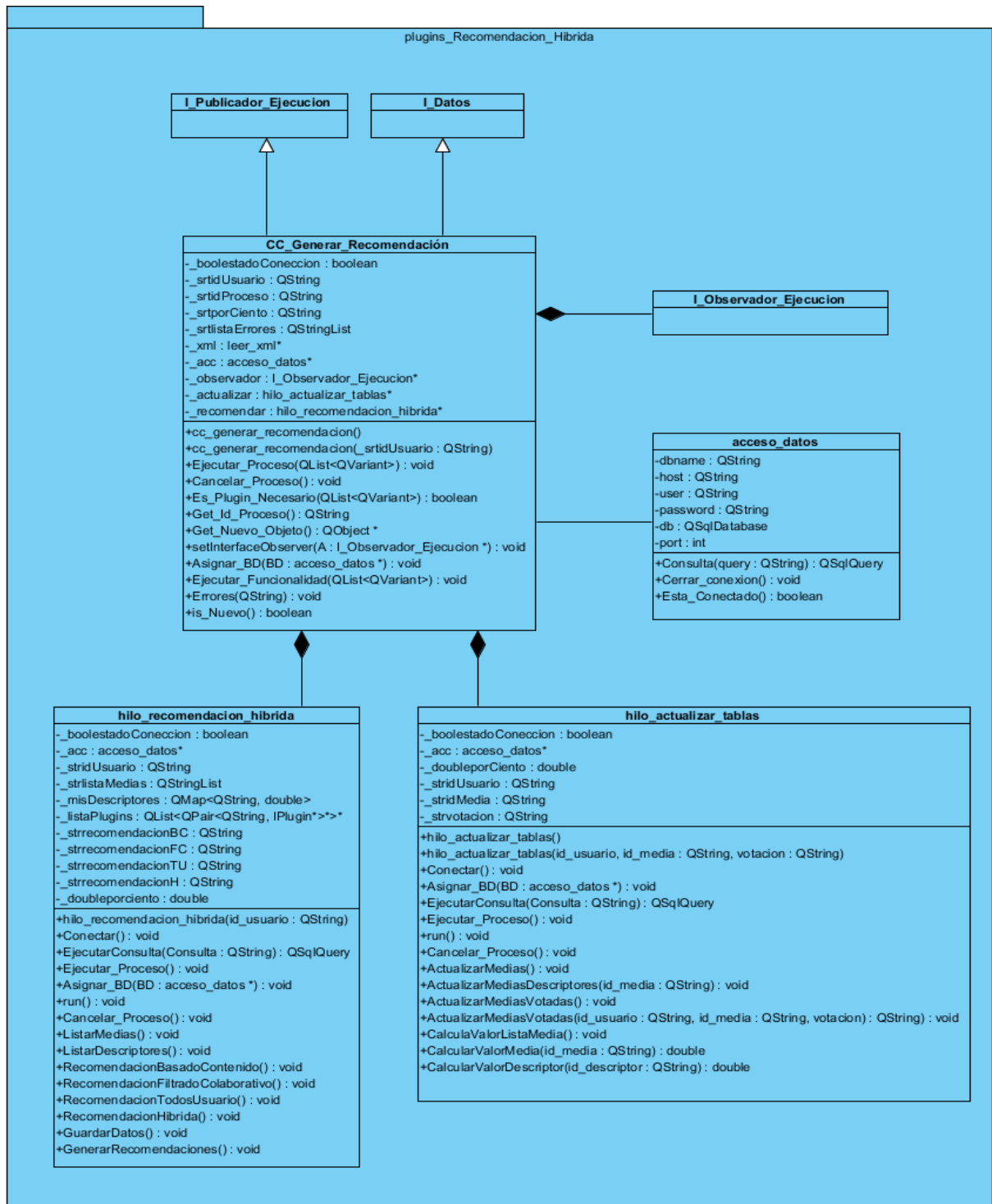


Fig. 6: Diagrama de clases del diseño.

### **2.6.1 Descripción de las clases del diseño.**

**CC\_Generar\_Recomendación:** es la clase que controla la realización de recomendaciones y la actualización de las tablas que son utilizadas para realizar las recomendaciones en el plugin, permite ejecutar las recomendaciones concurrentes mediante la creación de hilos. Esta clase es también la encargada de la comunicación con el Sistema Gestor de Procesos de Medias a través de las interfaces **I\_Plugin\_Ejecucion** e **I\_Publicador\_Ejecucion**.

**hilo\_recomendacion\_hibrida:** esta clase corresponde a un hilo de proceso que calcula las recomendaciones híbridas.

**hilo\_actualizar\_tablas:** esta clase es la encargada de mantener actualizadas las tablas que guardan los datos necesarios para el cálculo de las recomendaciones.

## **Conclusiones parciales.**

La generación del modelo de dominio proporcionó una entrada para definir los 4 requisitos funcionales del sistema así como los no funcionales. También se realizó el diagrama de casos de uso del sistema permitiendo comprender de una mejor forma las funcionalidades a implementar, dejando listo el diseño de las clases definidas. Además se trazó la arquitectura del sistema sustentada en los estilos y patrones arquitectónicos.

## Capítulo 3 Implementación y Pruebas del sistema de recomendación híbrida.

En el presente capítulo se tratan aspectos referentes a las disciplinas de implementación y prueba definidas por la metodología de software Open Up. Se realiza una representación de las tablas que intervienen en el cálculo de las recomendaciones. También se muestran los diagramas de componentes para una mejor comprensión de la organización y dependencias lógicas de los mismos y el diagrama de despliegue para una representación de la distribución física de las funcionalidades.

### 3.1 Modelo de datos.

Para el desarrollo del plugin fueron necesarias varias tablas con el objetivo de mejorar el tiempo de ejecución del algoritmo. A continuación se muestran las nuevas entidades.

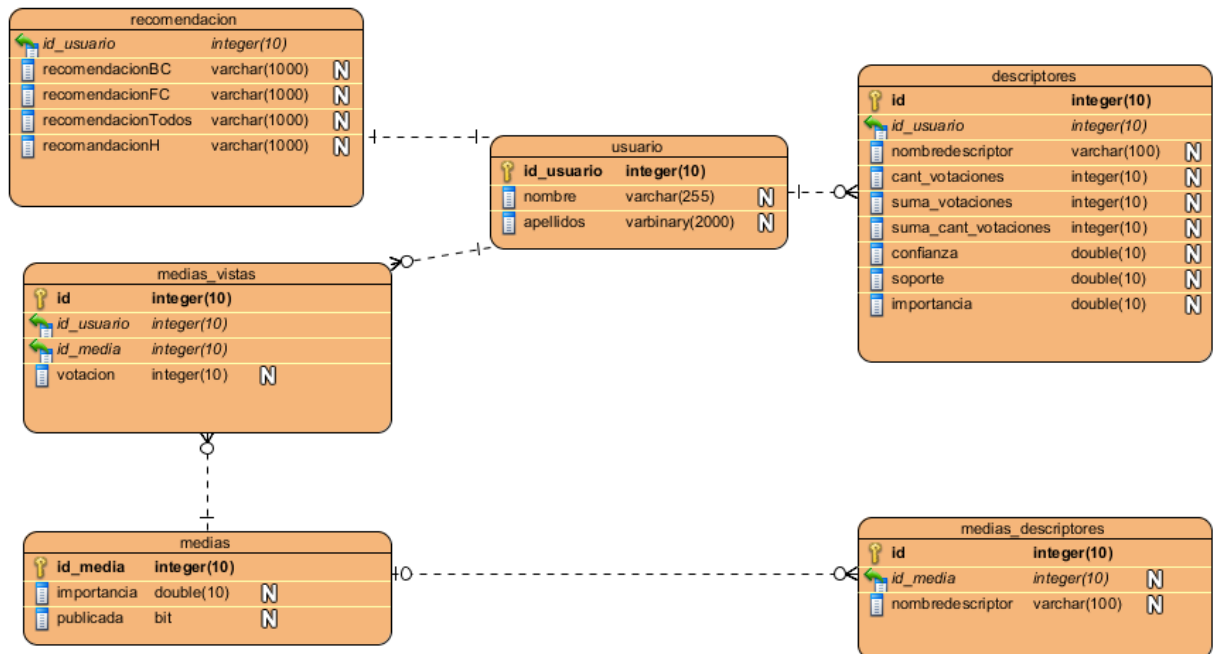


Fig. 7: Modelo de datos.

A continuación se describen las tablas creadas para el sistema de recomendación híbrido.

### Capítulo 3: Implementación y Pruebas del sistema de recomendación híbrida.

Tabla. 11: Descripción de la tabla recomendación.

<b>Nombre:</b> recomendación		
<b>Descripción:</b> En esta tabla se almacenarán las diferentes recomendaciones calculadas para generar la recomendación híbrida.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_usuario	Integer	Valor numérico que identifica al usuario.
recomendacion_BC	Varchar(1000)	Lista de la recomendación basado en contenido.
recomendacion_FC	Varchar(1000)	Lista de la recomendación basado en filtrado colaborativo.
recomendacion_Todos	Varchar(1000)	Lista de la recomendación de todos los usuarios del sistema.
recomendacion_H	Varchar(1000)	Lista de la recomendación híbrida.

Tabla. 12: Descripción de la tabla descriptores.

<b>Nombre:</b> descriptores		
<b>Descripción:</b> En esta tabla se almacena para cada usuario todos los descriptores de las medias vistas y se le calcula el valor de importancia de cada descriptor.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	Integer	Valor numérico incremental que identifica todos los campos de la tabla.
id_usuario	Integer	Valor numérico que identifica al usuario.
nombredescriptor	Varchar(100)	Nombre del descriptor.
cant_votaciones	Integer	Cantidad de veces que el usuario ha votado por una media donde aparece este descriptor.
suma_votaciones	Integer	Suma de las votaciones que el usuario le ha dado a la media donde aparece dicho descriptor.

### Capítulo 3: Implementación y Pruebas del sistema de recomendación híbrida.

suma_cant_votaciones	Integer	Suma de las cant_votaciones de todos los descriptores.
Confianza	Double	La confianza se calcula dividiendo suma_votaciones entre cant_votaciones.
Soporte	Double	La confianza se calcula dividiendo cant_votaciones entre suma_cant_votaciones.
Importancia	Double	La importancia es la suma del soporte más la confianza.

Tabla. 13: Descripción de la tabla medias\_vistas.

<b>Nombre:</b> medias_vistas		
<b>Descripción:</b> En esta tabla se almacena para cada usuario todas las medias que ha evaluado y la evaluación dada a la misma.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	Integer	Valor numérico incremental que identifica todos los campos de la tabla.
id_usuario	Integer	Valor numérico que identifica al usuario.
id_media	integer	Valor numérico que identifica el archivo multimedia.
Votación	Integer	Evaluación cuantitativa dada por el usuario al archivo multimedia.

Tabla. 14: Descripción de la tabla medias.

<b>Nombre:</b> medias		
<b>Descripción:</b> En esta tabla se almacenan todos los id de los archivos multimedia que están o han sido publicados en la plataforma.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_media	Integer	Valor numérico que identifica el archivo multimedia.

### Capítulo 3: Implementación y Pruebas del sistema de recomendación híbrida.

Importancia	double	Evaluación calculada sobre todos los archivos multimedia, a través de las evaluaciones dadas por todos los usuarios del sistema a los descriptores o características de la misma.
Publicada	bit	Este valor afirma si un archivo multimedia está publicado o no.

Tabla. 15: Descripción de la tabla medias\_descriptores.

<b>Nombre:</b> medias_descriptores		
<b>Descripción:</b> En esta tabla se almacena para cada archivo multimedia los descriptores o características del mismo.		
Atributo	Tipo	Descripción
id	Integer	Valor numérico incremental que identifica todos los campos de la tabla.
id_media	Integer	Valor numérico que identifica al archivo multimedia.
nombredescriptor	Varchar(100)	Nombre del descriptor.

### 3.2 Diagrama de componente.

Un componente es una parte física y reemplazable de un sistema que se conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces.

Un diagrama de componentes incluye a los componentes y archivos que se utilizan para ensamblar y hacer disponible el sistema físico. Muestra como el sistema está dividido en componentes y las dependencias entre ellos. Provee una vista arquitectónica de alto nivel del sistema. Ayuda a los desarrolladores a visualizar el camino de la implementación, permitiendo tomar decisiones respecto a las tareas de implementación. La figura 8 muestra el diagrama de componentes de la solución que se propone.



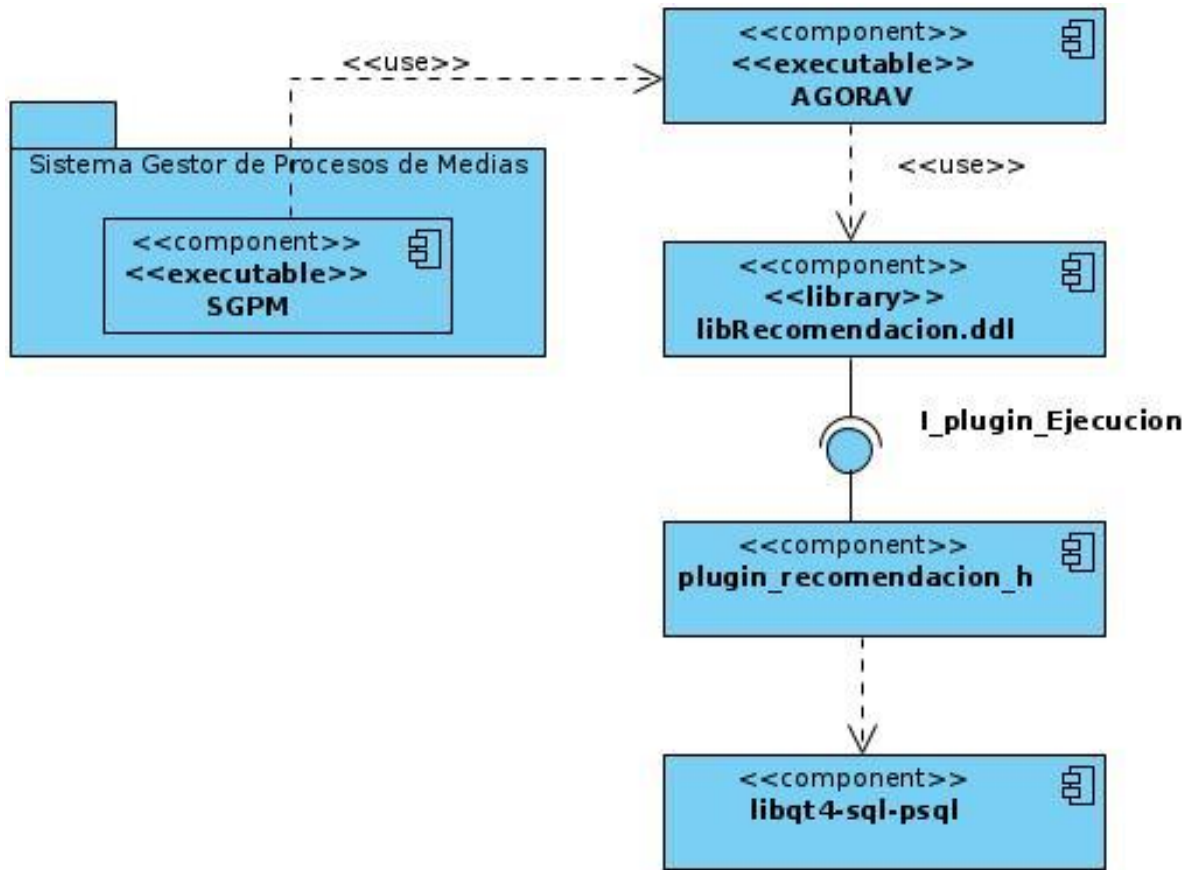


Fig. 8: Diagrama de componente.

### 3.3 Diagrama de despliegue.

El diagrama de despliegue permite modelar la disposición física o topología de un sistema. También contiene los nodos que forman la topología de hardware sobre la que se ejecuta el sistema y la distribución de sus componentes en los diferentes nodos. También muestra el hardware usado y los componentes instalados en el hardware, así como las conexiones físicas entre el hardware y las relaciones entre componentes. En la siguiente figura se muestra como estarán distribuidos los nodos de la solución propuesta.

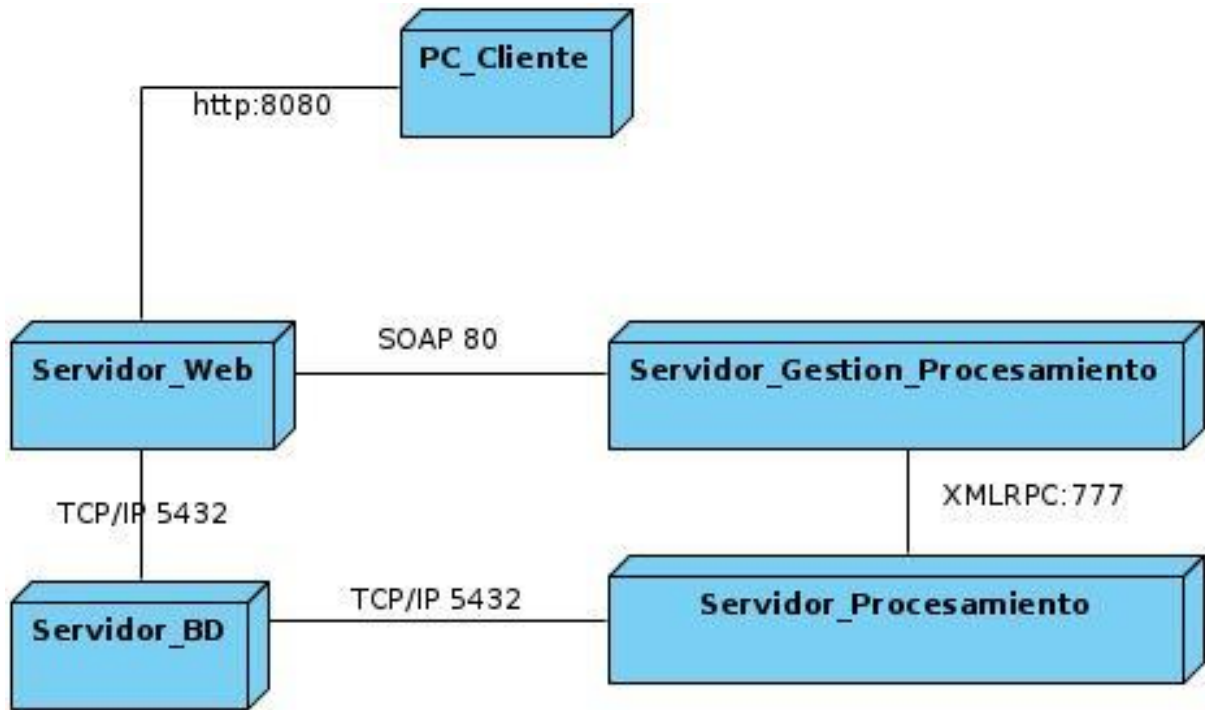


Fig. 9: Diagrama de despliegue.

Este modelo está compuesto primeramente por una PC cliente que es donde el usuario se conecta, y en la cual se le muestran las recomendaciones, esta PC se conecta al servidor web mediante el protocolo HTTP (*Hypertext Transfer Protocol* o Protocolo de Transferencia de Hipertexto). El servidor web, se comunica con el servidor de base de datos a través del protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol* o Protocolo de Control de Transmisión/Protocolo de Internet) por el puerto 5432. En esta base de datos se guarda toda la información referente a los usuarios y a los archivos multimedia publicados. El servidor web también se comunica con el Sistema Gestor de Procesos de Medias solicitando la generación de recomendaciones a través del protocolo SOAP (*Simple Object Access Protocol*) por el puerto 8080, mientras que la conexión Sistema Gestor de Procesos de Medias, servidor web se realiza por el protocolo XML-RPC por el puerto 777. El Sistema Gestor de Procesos de Medias se comunica con los servidores de procesamiento (servidores esclavos) para generar las recomendaciones. Estos servidores de procesamiento necesitan conectarse a la base de datos para obtener datos que necesita para la generación de recomendaciones utilizando el protocolo TCP/IP por el puerto 5432.

### 3.4 Estándar de codificación.

Un estándar de codificación comprende todos los aspectos de la generación de código. Los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez (Microsoft, 2003).

Los estándares de codificación utilizados en el Plugin de recomendación híbrido tienen como objetivo lograr una arquitectura y un estilo consistente, con el cual el sistema resulta fácil de entender y por supuesto fácil de mantener.

**Indentación:** Las normas de Indentación se centran en la lectura y comprensión del código por las personas. Estas normas deben seguir las siguientes reglas:

- Siempre se indenta 4 espacios el código encerrado entre llaves '{' y '}'.
- El primer corchete está en el mismo renglón que la instrucción.
- A partir de la última llave '}' se termina la indentación.
- Las líneas deben tener una longitud menor a 80 caracteres.

La Fig. 11 muestra un ejemplo de la indentación del código.

```
double hilo_actualizar_tablas::CalcularValorMedia(QString id_media)
{
    double importancia = 0.0;
    if(this->_boolestadoConeccion){
        QSqlQuery query = this->_acc->Consultar
            ("SELECT id_descriptor FROM medias_descriptores WHERE id_media = "+id_media+");
        while(query.next()){
            importancia += CalcularValorDescriptor(query.value(0).toString());
        }
    }
    return importancia;
}
```

Fig. 10: Indentación del código.

**Comentarios:** Los comentarios son notas cortas explicando el código con el objetivo de aportar

mayor información a las personas que lo leen. Los comentarios de línea deben escribirse de forma clara, en el caso de las funcionalidades significativas o clases se usarán comentarios de bloque al inicio de estos. Debe evitarse comentar por cada línea de código (Ver Fig. 11).

```
//En este metodo se generan las actualizaciones de las tablas utilizadas por el sistema de recomendación híbrido
void hilo_recomendacion_hibrida::GenerarRecomendaciones()
{
    if(this->_boolestadoConeccion){
        ListarMedias();
        ListarDescriptores();
        RecomendacionBasadoContenido();
        RecomendacionFiltradoColaborativo();
        RecomendacionTodosUsuario();
        RecomendacionHibrida();
    }
}
```

Fig. 11: Comentarios del código.

**Convenciones de nombres:** El objetivo fundamental es nombrar las clases y métodos de las clases y compresión de forma estándar. En ambos casos se usará el idioma español y el estilo CamelCase. (Ver la Fig. 11).

### 3.5 Pruebas.

Las pruebas son prácticas a realizar en diversos momentos de la vida del sistema informático para verificar: el correcto funcionamiento de los componentes del sistema, el ensamblaje entre los distintos componentes, el funcionamiento de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica, el funcionamiento del sistema integrado de hardware y software en el entorno de operación. Validan que el sistema cumple con el funcionamiento esperado y permite al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. La prueba es un proceso que intenta proporcionar confianza en el software (Sommerville, 2007).

En este epígrafe se describen las principales pruebas realizadas al plugin.

#### 3.5.1 Pruebas de integración.

Este tipo de pruebas consiste en construir el sistema a partir de los distintos componentes y probarlo con todos integrados poniendo énfasis en la interacción y no en el funcionamiento individual del sistema (Fernández Peña, 2011).

Estas pruebas deben realizarse progresivamente. El foco de atención es el diseño y la construcción de la arquitectura de software, que en este caso específico es una arquitectura basada en componente lo que prevé la reutilización e integración del sistema. El plugin desarrollado que generará las recomendaciones será puesto en la plataforma de codificación del SGPM ubicada en uno o varios nodos esclavos del propio SGPM. Estableciendo una comunicación a través del protocolo SOAP predefinida entre los componentes del gestor.

#### 3.5.2 Pruebas de Rendimiento

Las pruebas de rendimiento se basan en comprobar que el sistema puede soportar el volumen de carga definido en la especificación, es decir, hay que comprobar la eficiencia.

El sistema de recomendación híbrido utiliza 3 algoritmos diferentes que calculan recomendaciones por separado para luego mezclarlas en una sola. En la tabla 16 se muestran los tiempos de respuestas de cada uno de los algoritmos desarrollados en el plugin de recomendación híbrida. En el caso del sistema de recomendación de filtrado colaborativo, en el sistema se agregaron 10 usuarios con un promedio entre 500 y 1000 medias visualizadas por cada uno, garantizando que tengan al menos 200 medias que hayan sido visualizadas por más de 5 usuarios.

Tabla. 16: Primera iteración.

Agoritmos	Id	Cant. Mim. DAM	Cant. total de AM	AMV	Tiempo min. (mseg)	Tiempo max. (mseg)	Tiempo prom. (mseg)
SRBC	1	10	1000	1000	20000	60000	40000
				2000	23000	85500	54250
SRC				1000	45000	65700	55350

### Capítulo 3: Implementación y Pruebas del sistema de recomendación híbrida.

				2000	65500	95500	80500
<b>SRT</b>				1000	20	200	110
				2000	100	400	250
<b>SRH</b>				1000	100	110	105
				2000	100	120	110

Como se refleja los valores de tiempo obtenidos, no son funcionales en un mercado que exige cada vez más de la rapidez de respuesta de los sistemas informáticos.

En una segunda iteración, una vez optimizado el algoritmo, cambiando los tipos de datos de QList a QMap, y disminuyendo las consultas a la base de datos se lograron los resultados que se muestran a continuación:

Tabla. 17: Segunda iteración.

<b>Agoritmos</b>	<b>Id</b>	<b>Cant. Mim. DAM</b>	<b>Cant. total de AM</b>	<b>AMV</b>	<b>Tiempo min. (mseg)</b>	<b>Tiempo max. (mseg)</b>	<b>Tiempo prom. (mseg)</b>
<b>SRBC</b>	1	10	1000	1000	200	1000	600
				2000	300	1500	900
<b>SRC</b>				1000	400	1000	700
				2000	650	1150	900
<b>SRT</b>				1000	20	200	110
				2000	100	400	250
<b>SRH</b>				1000	100	110	105
				2000	100	120	110

Donde los valores de tiempo de respuesta disminuyen considerablemente al punto de ser prácticamente despreciables al usuario.

### **Conclusiones parciales**

El modelo entidad relación y la descripción de las tablas garantiza una mejor comprensión del almacenamiento de la información en la base datos, en tanto el diagrama de componente muestra una vista de implementación abreviada. La implementación fue regida por estándares de codificación que posibilitan una legible lectura e interpretación del código. Los resultados arrojados por las pruebas realizadas no aseguran ciento por ciento que el sistema no tenga errores pero si brinda confianza sobre la solides del mismo, brindando un producto funcional al usuario final

## **Conclusiones Generales:**

Una vez realizada la investigación se concluye lo siguiente:

- La fundamentación teórica de los sistemas de recomendación permitió comprender el funcionamiento de los sistemas de recomendación basado en contenido y de los sistemas de recomendación de filtrado colaborativo, así como definir una forma de hibridación a partir de estos, una vez seleccionadas las herramientas y tecnologías apropiadas.
- Se identificaron 3 tipos de requisitos no funcionales y 4 requisitos funcionales reflejados en el diagrama de casos de uso del sistema que muestra una visión global de las características de la solución propuesta.
- El diseño del sistema de recomendación híbrido muestra como quedarían implementadas las clases y las relaciones entre estas, bajo una arquitectura en 3 capas y el uso de los patrones GRASP y GOF.
- La estrategia de prueba definida para validar la propuesta de solución arrojó resultados favorables, priorizando las pruebas de rendimiento donde los valores del tiempo de respuesta fueron satisfactorios para el usuario.



## Recomendaciones.

1. Integrar la base de datos de la aplicación a la base de datos central de la plataforma AGORAV.
2. Utilizar una variante donde el valor de  $k$  no sea fijo para mejorar las recomendaciones del filtrado colaborativo.

## Bibliografía

**Albín Rodríguez, Pedro Antonio. 2009.** *Sistema de recomendación colaborativo basado en algoritmos de filtrado mejorados.* Jaén : s.n., 2009.

**Alexander, Christopher, Ishikawa, Sara y Silv, Murray. 1977.** *A Pattern Language.* 1977.

**Alvarez de Zayas, Carlos M y Sierra Lombardía, Virginia M. 2000.** *LA INVESTIGACION CIENTIFICA EN LA SOCIEDAD DEL CONOCIMIENTO.* 2000.

**Burke, Robin. 2002.** *Hybrid Recommender Systems: Survey and Experiments.* 2002.

**Castellano Torres, Emilio José. 2007.** *EVALUACIÓN DEL USO DE ALGORITMOS COLABORATIVOS PARA ORIENTAR ACADÉMICAMENTE AL ALUMNADO EN BACHILLERATO.* Jaén : s.n., 2007.

**Definición. 2008.** Definicion.de. *Definicion.de.* [En línea] 2008. [Citado el: 05 de 06 de 2015.] <http://definicion.de/toma-de-decisiones/>.

—. **2008.** Definición.de. *Definición.de.* [En línea] 2008. [Citado el: 10 de 03 de 2015.] <http://definicion.de/recomendacion/>.

**Eclipse. 2015.** Eclipse. *Eclipse.* [En línea] The Eclipse Foundation, 2015. [Citado el: 10 de 06 de 2015.] [http://epf.eclipse.org/wikis/openupsp/openup\\_basic/deliveryprocesses/openup\\_basic\\_process,\\_0uyGoMlgEdmt3adZL5Dmdw.html](http://epf.eclipse.org/wikis/openupsp/openup_basic/deliveryprocesses/openup_basic_process,_0uyGoMlgEdmt3adZL5Dmdw.html).

**Escolar, Fernando. 2012.** Programando en .net . *Programando en .net .* [En línea] 10 de 09 de 2012. [Citado el: 20 de 05 de 2015.] <http://programandonet.com/web/patrones-diseno-singleton/>.

**Fernández Peña, Juan Manuel. 2011.** *Ingeniería de Software I: Pruebas de Integración.* 2011.

**Fernández Ramírez, M. L. 1998.** *Interactive\_and\_Cooperative\_Technologies\_Lab.* [En línea] 1998. [Citado el: 20 de 03 de 2015.] <http://ict.udlap.mx/people/lulu/documento/capitulo4.html>.

**Formoso López, Vreixo. 2013.** *Técnicas eficientes para la recomendación de productos basadas en filtrado colaborativo.* Coruña : s.n., 2013.

**Galán Nieto, Sergio Manuel. 2007.** *Filtrado Colaborativo y Sistemas de Recomendación*. Madrid, España : s.n., 2007.

**Gamma, Erich, y otros. 1994.** *Disegn Patterns: Elements of Reusable Object-Oriented Software*. 1994.

**García Carmona , Juan. 2012.** Juan García Carmona. *Juan García Carmona*. [En línea] 7 de 09 de 2012. [Citado el: 20 de 05 de 2015.] <http://juan-garcia-carmona.blogspot.com/2012/09/grasp-alta-cohesion-y-bajo-acoplamiento.html>.

—. **2012.** Juan García Carmona . *Juan García Carmona* . [En línea] 7 de 09 de 2012. [Citado el: 20 de 05 de 2015.] <http://juan-garcia-carmona.blogspot.com/2012/09/grasp-controlador.html>.

—. **2012.** Juan García Carmona . *Juan García Carmona* . [En línea] 7 de 09 de 2012. [Citado el: 20 de 05 de 2015.] <http://juan-garcia-carmona.blogspot.com/2012/09/grasp-creador.html>.

**García Carmona, Juan. 2012.** Juan García Carmona. *Juan García Carmona*. [En línea] 07 de 09 de 2012. [Citado el: 20 de 05 de 2015.] <http://juan-garcia-carmona.blogspot.com/search?q=experto>.

**Garlan, David y Shaw, Mary. 1994.** *An Introduction to Software Architecture*. Pittsburgh : s.n., 1994. CMU-CS-94-166.

**Guerra Sánchez, Esther. 2008.** *Patrones de Diseño*. Madrid : s.n., 2008.

**Herrera-Viedma , Enrique, Porcel, Carlos y Hidalgo, Lorenzo. 2012.** *hipertext.net*. *hipertext.net*. [En línea] 05 de 06 de 2012. [Citado el: 27 de 03 de 2015.] <http://www.upf.edu/hipertextnet/numero-2/recomendacion.html>.

**Huecas, Gabriel y Salvachúa, J. 2010.** *Filtros Colaborativos y Sistemas de Recomendación*. 2010.

**Larman, Craig. 2003.** *UML y Patrones 2da Edición*. s.l. : Prentice Hall, 2003.

**Mario Marrello, Pedro. 1999.** Asociación Internacional de Clubes de Leones. *Asociación Internacional de Clubes de Leones*. [En línea] 21 de 12 de 1999. [Citado el: 03 de 06 de 2015.] <http://www.leonismoargentino.com.ar/Info6.htm>.

**Martinez, Rafael. 2009.** PostgreSQL-es. *PostgreSQL-es*. [En línea] 2009. [Citado el: 24 de 04 de 2015.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).

### Capítulo 3: Implementación y Pruebas del sistema de recomendación híbrida.

**Microsoft. 2003.** msdn.microsoft.com. *msdn.microsoft.com*. [En línea] Microsoft, 2003. [Citado el: 24 de 05 de 2015.] <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.

**Palomares Carrascosa, Iván. 2009.** *Radio Online basada en un motor de filtrado colaborativo*. 2009.

**Peláez, Juan. 2009.** Arquitectura basada en capas. *Arquitectura basada en capas*. [En línea] 26 de 05 de 2009. [Citado el: 10 de 06 de 2015.] <http://www.juanpelaez.com/geekstuff/arquitectura/arquitectura-basada-en-capas/>.

**Qt\_Project. 2015.** Qt Creator. *Qt Creator*. [En línea] The Qt Company, 24 de 05 de 2015. [Citado el: 5 de 04 de 2015.] [https://wiki.qt.io/Category:Tools::QtCreator\\_Spanish](https://wiki.qt.io/Category:Tools::QtCreator_Spanish).

**RAE. 2013.** Real Academia Española. *Real Academia Española*. [En línea] 2013. [Citado el: 10 de 03 de 2015.] <http://lema.rae.es/drae/?val=Recomendacion>.

**Resnick, Paul y Varian, Hal R. 1997.** *Recommender Systems: COMMUNICATIONS OF THE ACM*. s.l. : Guest Editors, 1997.

**Rojas Castellanos, Yarisel. 2013.** *Módulo de recomendación híbrida para la plataforma Inter-nos 2.0*. La Habana : s.n., 2013.

**S. Pressman, Roger. 2010.** *Software Engineering: A Practitioner's Approach*. 2010.

**Sánchez González, Carlos. 2004.** SOURCEFORGE.NET. *SOURCEFORGE.NET*. [En línea] 28 de 09 de 2004. [Citado el: 10 de 06 de 2015.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.

**Sarwar, Badrul, y otros. 2000.** *Analysis of Recommendation Algorithms for ECommerce*. 2000. pág. 10.

**Seguido Font, Miguel. 2009.** *Sistemas de recomendación para webs de información sobre la salud*. Catalunya : s.n., 2009.

**Sommerville, Ian. 2007.** *Software Engineering*. 2007.

**Taghipour, Nima y Kardan, Ahmad. 2008.** *A hybrid web recommender system based on Q-learning*. New York, USA : s.n., 2008.

**Varas Contreras, Marcela y Urrutia Sepúlveda, Angélica. 2003.** *Bases de Datos Difusas Modeladas con UML*. Concepción, Chile : s.n., 2003.

### Capítulo 3: Implementación y Pruebas del sistema de recomendación híbrida.

**Visual\_Paradigm. 2014.** Visual Paradigm. *Visual Paradigm*. [En línea] Company Headquarters, 2014. [Citado el: 15 de 03 de 2015.] [http://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963\\_visualparadi.html](http://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html).