

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 6**

**Módulo de recomendación híbrido para la  
plataforma Inter-nos 2.0.**

**Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas**

**Autor: Yarisel Rojas Castellanos.**

**Tutor: Ing. Eridniel Suárez Contreras.**

**Co-tutor: Ing. Yunier Albrecht Delgado.**

**“Ciudad de la Habana, junio de 2013**

**Año 55 del Triunfo de la Revolución”**



*Los grandes espíritus siempre han encontrado la violenta oposición de las mentes mediocres. Estos últimos no pueden entender que un hombre no se someta irreflexivamente a los prejuicios hereditarios sino que emplee honestamente y con coraje su inteligencia.*

*Albert Einstein*

## Declaración de autoría

Declaro ser la única autora del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas y al centro GEySED los derechos patrimoniales del mismo, con carácter exclusivo.

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

Yarisel Rojas Castellanos

Ing. Eridniel Suarez Contreras

\_\_\_\_\_

Firma del Autor

\_\_\_\_\_

Firma del Tutor

Ing. Yunier Albrecht Delgado

\_\_\_\_\_

Firma del Co-tutor

### **Generales del diplomante**

**Nombre y Apellidos:** Yarisel Rojas Castellanos.

**Sexo:** Femenino.

**Grupo:** 6504.

**Correo electrónico:** [yrojas@estudiantes.uci.cu](mailto:yrojas@estudiantes.uci.cu)

### **Generales del tutor**

**Nombre y Apellidos:** Eridniel Suarez Contreras.

**Categoría docente:** Instructor.

**Centro de trabajo:** Universidad de las Ciencias Informáticas.

**Título de la especialidad de graduado:** Ingeniero en Ciencias Informáticas.

**Año de graduación:** 2008.

**Institución donde se graduó:** Universidad de las Ciencias Informáticas.

**Correo electrónico:** [esuarez@uci.cu](mailto:esuarez@uci.cu)

### **Generales del Co-tutor**

**Nombre y Apellidos:** Yunier Albrecht Delgado.

**Categoría docente:** Instructor.

**Centro de trabajo:** Universidad de las Ciencias Informáticas.

**Título de la especialidad de graduado:** Ingeniero en Ciencias Informáticas.

**Año de graduación:** 2007.

**Institución donde se graduó:** Universidad de las Ciencias Informáticas.

**Correo electrónico:** [yalbrecht@uci.cu](mailto:yalbrecht@uci.cu)

*A mis padres, por su amor, su dedicación y todo el sacrificio que han hecho por mí.*

*A mi papá, Yoel Rojas, por enseñarme que no hay satisfacción más grande en el mundo que ser lo que uno quiere ser y que el éxito no se consigue trabajando al máximo, sino disfrutando al máximo en todo lo que se trabaje.*

*A mi mamá: Yudik Castellanos por mostrarme que todo está a nuestro alcance, solo tenemos que trazarnos los objetivos correctos sin titubear jamás y que la vida se hizo para vivirla, no para tenerla de reliquia.*

*A mis abuelos: Fila, Migdalia, Bolito y Lorenzo, por ser la alegría de mi vida.*

*A mis tíos, por su preocupación y atención: William, Marciana, Alicia, Anibal, Rogelio, Regla, Niñita y Leobaldo.*

*A mis primos: Carlos, Dayron y Ernesto, por su cariño y confianza.*

*A mi prima Daniela y a mi sobrina Alicia, por ese amor sin medida.*

*A mis amigas, casi hermanas, Máyuri y Marizet, por soportarme como nadie y por brindarme su cariño y alegría.*

*A Yaima y Elaine por ser más que amigas, más en este último período de tensión y por las noches en vela que pasaron junto a mí.*

*A Ramón, por las noches en vela esperando a que terminase de trabajar y por tanto amor y a sus padres por toda su atención.*

*A las personas que han estado a mi lado y me han consentido en algún momento de la carrera, como son: Falcón, Marcial, Eliober, Edgar, Jose, Darcy, Oscar, Nathalie, Darian, Kirenia, Yeilen, Addiel, Luis Manuel, Yaimy, Manuel Alejandro, Orlando Yacsell, Aylín, Roberto, Carlos, Tony, Aramis, Susana, Osiel, Roberto Carlos, Wilber.*

*A las amistades que han disfrutado de mi compañía tanto como yo de la suya: Yíssel, Tomás, Darien, Ander, Yasmanny, Machado, Osay, Ronaldo, Alexei, Asiel, Darinca, Diannet, Frank Ramirez, Guillermo, Javier de León, Juan Carlos, Julio, Lisbet, Osmar, Yanet y Yicel Blanco, Yoander, Paul, Raul.*

*A los profesores cuyo profesionalismo ha hecho que mi interés por la carrera se haya convertido en el deleite de estudiarla: Angel Dayan, Frank Torres, Daniel Sampedro, David Silva, Sixto, Jose Carlos, Carlos Enrique.*

*Al resto de mi familia, por su cariño.*

*A Erid, por ser mi tutor, compañero, profesor, asesor y miles de calificativos más, pero ninguno con honor tan merecido como el de amigo.*

*A mi co-tutor Albrecht, por brindarme su amistad y todos sus consejos profesionales que permitieron mi preparación.*

*A los miembros del tribunal, por demostrar grato interés en el trabajo de diploma y al oponente por su preocupación por el mismo.*

*A todos los que de una forma u otra contribuyeron a que cada uno de mis sueños se haya hecho realidad, gracias.*

*A mis padres, por enseñarme que el verdadero triunfo no está en lo que se estudia  
sino en el placer de aprender y de expresar los conocimientos.*

*A mis abuelos, por mostrarme que todo objetivo tiene trabas en el camino,  
pero mientras más difícil, mayor satisfacción se obtiene en su logro.*

## **Resumen**

La red de datos se ha convertido en un medio para la publicación y distribución de contenido multimedia; sin embargo, con el progreso de la industria audiovisual, ha aumentado la cantidad de información disponible y su dispersión, provocando que los individuos acudan a fuentes de información externas, para la toma de decisiones en cuanto a los contenidos que cumplan con sus preferencias. Los sistemas de recomendación constituyen una alternativa de naturaleza proactiva, a esta problemática, se encargan de brindar a los usuarios propuestas personalizadas a partir de las preferencias y opiniones proporcionadas por los mismos. En la presente investigación se ha desarrollado un módulo de recomendación para la plataforma de distribución de contenido multimedia Inter-nos 2.0. La solución posee un enfoque híbrido, integrado por métodos de recomendación colaborativos y basados en contenido, a partir de la utilización de varias técnicas de hibridación asociativas como: hibridación en cascada, conmutada, mezclada y por pesos. El módulo de recomendación se encuentra desagregado en varios módulos según sus responsabilidades, controlados por un módulo principal que es el encargado de manejar las operaciones. Además, se cuenta con un módulo de comunicación que permite la utilización de la biblioteca de funciones de recomendación *Recommender Lib*, para su utilización con fines no comerciales. La integración del módulo de recomendación a la plataforma, permite minimizar el tiempo empleado por cada usuario en las búsquedas de un material, debido a la facilidad para acceder a los contenidos multimedia de su preferencia, desde la portada principal.

## **Palabras clave**

Filtrado de elementos, hibridación, Inter-nos 2.0, personalización, preferencias, sistemas de recomendación.

## Tabla de contenidos

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1. Descripción actual del dominio del problema .....	6
1.2. Situación problemática.....	7
1.3. Bases conceptuales asociadas al dominio del problema .....	7
1.3.1. Preferencias .....	8
1.3.2. Recomendación.....	8
1.3.3. Perfil del usuario .....	9
1.4. Sistemas de recomendación.....	9
1.4.1. Entradas de los sistemas de recomendación.....	10
1.4.2. Métodos de generación de recomendaciones.....	10
1.4.3. Salidas de los sistemas de recomendación .....	12
1.5. Sistema de recomendación híbrido.....	12
1.5.1. Funcionamiento del método basados en contenido .....	13
1.5.2. Funcionamiento del método colaborativo.....	14
1.5.3. Técnicas de hibridación .....	17
1.6. Análisis de soluciones existentes.....	19
1.6.1 Fab.....	20
1.6.2. REJA.....	20
1.6.3. Avatar.....	20
1.6.4. Sistema de recomendación colaborativo para la plataforma VideoWeb.....	21
1.6.5. Sistema de Recomendación Basado en Contenido para la Plataforma VideoWeb.....	21
1.7. Conclusiones parciales .....	21

CAPÍTULO 2: SELECCIÓN DE HERRAMIENTAS Y TECNOLOGÍAS.....	23
2.1. Metodología de desarrollo de software .....	23
2.2. Lenguaje de modelado unificado .....	24
2.3. Herramienta CASE Visual Paradigm.....	24
2.4. Entorno de desarrollo integrado NetBeans .....	25
2.5. Lenguaje de programación PHP 5.3. ....	26
2.6. Biblioteca de funciones de recomendación <i>Recommender Lib.</i> .....	26
2.7. Sistema gestor de base de datos PostgreSQL 9.1.....	27
2.8. Sistema gestor de contenidos Drupal 6. ....	27
2.9. Conclusiones parciales .....	28
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	29
3.1. Modelo de dominio .....	29
3.1.1. Descripción de los principales eventos del entorno.....	29
3.1.2. Objetos del dominio .....	29
3.1.3. Diagrama de clases del dominio .....	30
3.2. Requisitos.....	30
3.2.1. Requisitos funcionales.....	31
3.2.2. Requisitos no funcionales .....	33
3.3. Modelo de casos de uso .....	34
3.3.1. Diagrama de casos de uso .....	34
3.3.2. Descripción de los casos de uso.....	35
3.4. Modelo de análisis .....	40
3.4.1. Diagramas de clases del análisis.....	41
3.4.2. Diagramas de secuencia .....	41

3.5. Estilos arquitectónicos .....	42
3.5.1. Patrones arquitectónicos .....	43
3.6. Patrones de diseño .....	44
3.6.1. Patrones GOF .....	44
3.6.2. Patrones GRASP .....	45
3.7. Modelo de diseño .....	46
3.7.1. Diagrama de clases del diseño .....	47
3.8. Conclusiones parciales .....	48
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA .....	49
4.1. Modelo de datos .....	49
4.2. Modelo de implementación .....	52
4.2.1. Diagrama de componentes .....	52
4.3. Modelo de despliegue .....	53
4.4. Pruebas del módulo .....	54
4.4.1. Pruebas de caja blanca .....	55
4.4.2. Pruebas de caja negra .....	57
4.5. Conclusiones parciales .....	60
CONCLUSIONES GENERALES.....	62
RECOMENDACIONES .....	63
BIBLIOGRAFÍA.....	64

## Índice de figuras, tablas y ecuaciones.

### Índice de Figuras.

Figura 1: Modelo de dominio.....	30
Figura 2: Diagrama de casos de uso del sistema.....	34
Figura 4: Diagrama de clases del análisis del caso de uso Mostrar Recomendación.....	41
Figura 6: Diagrama de secuencia del caso de uso Mostrar Recomendación. ....	42
Figura 7: Funcionamiento del patrón arquitectónico MVC.....	44
Figura 8: Diagrama de clases de diseño del CU Mostrar Recomendación.....	47
Figura 9: Modelo de datos. ....	50
Figura 10: Diagrama de componentes. ....	53
Figura 11: Diagrama de despliegue .....	54

### Índice de Tablas.

Tabla 1: Mostrar Recomendación. ....	35
Tabla 2: Historial de navegación del usuario.....	51
Tabla 3: Información sobre los grupos de medias.....	51
Tabla 4: Información asociada a los grupos y los usuarios. ....	52
Tabla 5: Información asociada a las configuraciones del módulo de recomendación.....	52
Tabla 6: Diseño de caso de prueba de caja negra para el caso de uso Administrar configuración de recomendación colaborativa. ....	59
Tabla 7: Descripción de las variables de los casos de uso Administrar configuración de recomendación colaborativa. ....	59
Tabla 8: Matriz de datos del caso de uso Administrar configuración de recomendación colaborativa. ....	60

### Índice de Ecuaciones.

Ecuación 1: Cálculo de la similitud entre cada par de usuarios. (Olvera-Lobo, 2011) .....	15
---	----

## Introducción

Desde la concepción de Joseph Carl Robnett Licklider, en agosto de 1962, de *“una red interconectada globalmente a través de la que cada persona pudiera acceder desde cualquier lugar a datos y programas”*, ya se evidenciaba la necesidad del hombre de encontrar un mecanismo para diseminar información y un medio para la colaboración e interacción entre los individuos y sus ordenadores, independientemente de su ubicación geográfica. Estas fueron las bases de lo que hoy se conoce como “internet”, la cual a su vez es una herramienta de difusión mundial de información (Aroche, 2006).

La era actual está fundamentada sobre las tecnologías informáticas, la mayoría de los negocios, transacciones, comercio y hasta el aprendizaje están enfocados a internet, causa por la cual la información en la red de redes ha crecido exponencialmente, lo que provoca a su vez que los usuarios pasen gran parte de su tiempo conectados, con diversos fines. Estudios realizados en el 2012 por la *Internet World Stats* (Estadísticas mundiales de internet) arrojaron porcentajes altos del uso de la red global, donde se evidenció que aproximadamente la mitad de la población conectada a la red la utiliza con fines de entretenimiento y un apreciable porcentaje para consumo de información (IWS, 2012). Diariamente surgen variedades de empresas virtuales cuyo objetivo es satisfacer las demandas de los usuarios en cuanto al comercio electrónico y el ocio, además de sacar ganancias de las mismas, entre las que se encuentran los sitios de distribución de contenido multimedia (Buendía, 2010).

Desde 1895 con la concepción de la cinematografía y años más tarde su integración con el sonido, los medios audiovisuales como el cine y la televisión han adquirido una posición privilegiada con respecto a otras formas de comunicación; ofreciendo información y entretenimiento, lo que representa la fórmula perfecta para acercar todo tipo de contenidos a un público diverso (ASOCAE, 2008). Sin embargo, el desarrollo de internet y las tecnologías de la información y las comunicaciones (TIC), están transformando exponencialmente la industria audiovisual, estableciendo nuevas estrategias para la creación y distribución de contenidos multimedia. Esta tendencia permite disponer de canales de televisión en internet con programaciones durante las 24 horas del día o a petición de los clientes, en los casos donde se presentan mediante portales Web, lo que provoca que la red se convierta en la nueva ventana de consumo audiovisual, en la que cada usuario elige entre extensas ofertas de contenidos disponibles en múltiples formatos. (Peña, 2007)

El desarrollo progresivo del internet, aparejado a las nuevas tendencias de visualización de los contenidos en línea (on-line), como una forma directa de consumo, crea una variedad extensa de productos que

deben satisfacer las necesidades del usuario final. En este ámbito, los individuos se enfrentan a nuevos escenarios caracterizados por la existencia de una voluminosa cantidad de información, por lo que las recomendaciones de terceros son una opción a la que acuden para tratar de solucionar los problemas derivados de su limitado conocimiento, respecto a categorías o alternativas en el consumo de materiales. Para dar solución a esta problemática, mejorar el tratamiento de los clientes y satisfacer sus necesidades, surgen los sistemas de recomendación, encargados de brindar a los usuarios propuestas personalizadas de los contenidos o servicios ofertados. (Cruz, 2010)

En Cuba se estimula la investigación científico-técnica en aras de impulsar los avances tecnológicos del país y como parte del proceso nacional de desarrollo socio-económico, a pesar de la brecha tecnológica que separa a los países subdesarrollados de las potencias industrializadas (Utria, 2007). Como exponente fundamental en este sector se encuentra la Universidad de las Ciencias Informáticas (UCI), donde existen varios proyectos productivos, entre ellos el de Catalogación y Publicación de Medias dentro del centro de Geo-informática y Señales digitales (GEYSED). El mismo tiene como producto en vías de despliegue la versión 2.0 de la plataforma de Inter-nos, cuyo objetivo general se centra en la transmisión de video y audio digital utilizando tecnologías de transmisión mediante flujos de bits (también conocido por su término en inglés como *streaming*); además de incorporar gestores de usuarios, de contenido audiovisual e informativos, los cuales facilitan la gestión de la información.

Los contenidos en la plataforma son mostrados a los usuarios mediante una determinada organización previamente diseñada y de igual representación para todos. Por consiguiente, al navegar por los mismos se tienen que realizar búsquedas de los materiales que se necesitan, a pesar de responder en la mayoría de los casos, a características similares como tipo, género y otras clasificaciones. Las búsquedas de contenidos audiovisuales pueden volverse complejas, debido a que los usuarios, en ocasiones, no conocen los datos para realizar una consulta específica del material a través del buscador. La complejidad para encontrar un contenido multimedia de preferencia para el usuario propicia un aumento del tiempo empleado en las búsquedas del mismo y una considerable disminución de la fidelidad hacia la plataforma, propiciando que la misma no se ajuste a las nuevas tendencias de personalización de la información, para brindar un producto que satisfaga directamente las necesidades de cada usuario.

De las consideraciones anteriormente planteadas se deriva como **problema a resolver**: Los contenidos de la Plataforma Inter-nos 2.0 no son presentados de acuerdo a las preferencias particulares de los usuarios. Para desarrollar una solución a dicho problema la presente investigación define como **objeto de**

**estudio:** Los sistemas de recomendación, enmarcado en el **campo de acción:** Los sistemas de recomendación híbridos para la plataforma Inter-nos 2.0.

Atendiendo a la situación problemática descrita, el problema científico y el campo de acción definidos, el presente trabajo de diploma se estructura y desarrolla en función del siguiente **objetivo general:** Desarrollar un módulo de recomendación de contenidos a partir de la integración de un sistema de recomendación colaborativo y otro basado en contenido, para la Plataforma Inter-nos 2.0. Como **idea a defender** se propone: El desarrollo de un módulo de recomendación híbrido para la plataforma Inter-nos 2.0 permitirá presentar los contenidos publicados de acuerdo a las preferencias particulares de los usuarios.

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

1. Análisis de soluciones existentes referentes a los sistemas de recomendación híbridos.
2. Determinación de las tendencias y tecnologías actuales factibles para el desarrollo del módulo.
3. Determinación del tipo de algoritmo a utilizar en el desarrollo del módulo de recomendación teniendo en cuenta las características del negocio de Inter-nos 2.0.
4. Desarrollo de los artefactos y la documentación según la metodología de desarrollo seleccionada.
5. Desarrollo del módulo a partir de las funcionalidades identificadas.
6. Integración del módulo implementado al sistema Inter-nos 2.0.
7. Realización de pruebas de sistemas y pruebas unitarias para validar el correcto funcionamiento del módulo desarrollado.

Con el cumplimiento satisfactorio de las tareas trazadas se esperan obtener como **posibles resultados:** la documentación resultante de la investigación realizada, los artefactos resultantes del análisis y diseño del módulo, el módulo de recomendación híbrido para la plataforma Inter-nos 2.0, su código fuente y la documentación sobre las pruebas realizadas.

Para abordar la realidad conceptual y práctica de los sistemas de recomendación, en el transcurso de la investigación se utilizaron varios **métodos científicos**, clasificados en teóricos y empíricos.

Entre los **métodos teóricos** empleados se encuentran:

**Analítico – sintético:** este método permitió desglosar los componentes fundamentales de los sistemas de recomendación y sus relaciones para su comprensión total, así como sintetizar los rasgos principales de su funcionamiento, definiendo los ajustables a la plataforma Inter-nos 2.0.

**Análisis histórico – lógico:** para un análisis del objeto de estudio se utilizó este método con el objetivo de alcanzar una mayor visión del desarrollo evolutivo de los sistemas de recomendación, así como su funcionamiento lógico, permitiendo un alcance conceptual de la estructura de los mismos.

**Inductivo – deductivo:** el proceso de inducción permitió generalizar las características presentes en otros sistemas de recomendación utilizados en el mundo, para deducir conclusiones asociadas a las necesidades de la investigación.

**Modelación:** se utilizó este método en el modelado de la implementación del módulo, como forma de abstracción de la realidad, con el objetivo de tener una aproximación que permitió verificar el flujo de los procesos de generación de una recomendación.

Se ha empleado como **método empírico:**

**Experimento:** como forma de validar la estrategia de recomendación se realizaron, mediante la experimentación, una serie de pruebas al módulo desarrollado, creando y adaptando las condiciones en un ambiente específico para la obtención de resultados que permitió verificar el cumplimiento del objetivo planteado.

El presente trabajo de diploma está conformado por cuatro capítulos:

**Capítulo 1 Fundamentación teórica:** Se establecen las bases conceptuales asociadas al dominio del problema. Se realiza un estudio de las características fundamentales de los métodos de recomendación colaborativos y basados en contenido, además de su integración en un sistema de recomendación híbrido. También se efectúa un análisis de las soluciones existentes en el mundo y en Cuba, enfatizando en las orientadas a los sistemas de distribución de contenido audiovisual.

**Capítulo 2 Selección de herramientas y tecnologías:** Se presenta la metodología de desarrollo de software *RUP*, la herramienta *CASE Visual Paradigm* y el lenguaje de modelado *UML* a utilizar para el desarrollo del módulo, así como otras tecnologías a utilizar en la implementación del mismo, entre las que se encuentran el entorno de desarrollo integrado *NetBeans*, el lenguaje de programación *PHP*, el sistema gestor de base de datos PostgreSQL y el sistema gestor de contenido Drupal.

**Capítulo 3 Análisis y Diseño del sistema:** Se especifican los requisitos funcionales y no funcionales que debe cumplir el módulo a desarrollar. Se realiza la descripción de casos de uso, así como una propuesta

de los diagramas resultantes del análisis y diseño del módulo. Además de la selección y especificación de los patrones arquitectónicos y de diseño a utilizar en la solución.

**Capítulo 4 Implementación y pruebas:** Se abordan aspectos referentes a la implementación del módulo, como son los diagramas de componentes y despliegue. Además, de exponer los resultados arrojados por las pruebas realizadas, como forma de verificar la efectividad del algoritmo implementado y para comprobar el correcto funcionamiento de la solución.

## **CAPÍTULO 1: Fundamentación teórica**

El objetivo del presente capítulo es definir las características fundamentales de los sistemas de recomendación, para ello se presentan en orden lógico las bases conceptuales, permitiendo una mayor comprensión del objeto de estudio. Se caracterizan además los sistemas de recomendación híbridos, así como algunas de las soluciones existentes en Cuba y el resto del mundo.

### **1.1. Descripción actual del dominio del problema**

El proyecto de Catalogación y Publicación de Medias, perteneciente al centro de GEYSED, desarrolló la plataforma Video Web. Dicha plataforma es un sistema informático formado por componentes de hardware y software que permite la transmisión de video y audio digital mediante una red de datos. Este producto tiene varias personalizaciones las cuales son ajustadas a los negocios donde vayan a ser implantadas. Una de estas personalizaciones es la plataforma Inter-nos 2.0, la cual se encarga de la publicación de materiales audiovisuales, posibilitando su consumo a una parte considerable de la comunidad de la Universidad de las Ciencias Informáticas (UCI).

Los usuarios que navegan por los contenidos publicados en la plataforma, interactúan con una variada cantidad de información, díganse películas, documentales, videos musicales, materiales audiovisuales de carácter docente, además de los artículos de contenido informativo, como son las noticias y los avisos; todos ellos a su vez organizados por tipologías definidas en el sub-sistema de presentación y gestión de contenidos. (Pérez, 2010) De los archivos multimedia se almacenan los descriptores definidos por el subsistema de catalogación, como son género, autor, país y otros en dependencia del tipo de publicación.

Al acceder al portal web Inter-nos 2.0, los usuarios pueden visualizar el contenido publicado y autenticarse para votar por un contenido multimedia o actualizar su perfil. La aplicación está desarrollada en el sistema de gestión de contenido Drupal, el que posibilita, entre diversas funcionalidades, obtener registros sobre la cantidad de veces que un usuario accede a un contenido multimedia determinado, las votaciones que ha realizado al mismo, así como otras actividades realizadas sobre este. Estos registros pueden ser visualizados por el administrador del sistema y son almacenados en tablas de la base de datos, en las que también se registran datos como las direcciones de las páginas solicitadas por el usuario en una actividad y la hora y la fecha en que se realizó, características que no son utilizadas actualmente, lo que desaprovecha las posibilidades que brinda la herramienta para personalizar la interfaz de la plataforma, en cuanto a las preferencias de cada usuario.

## **1.2. Situación problemática**

En la plataforma Inter-nos 2.0 los contenidos son mostrados de acuerdo a políticas definidas por los administradores del sistema, ya sea por fechas de publicación o por promocionar algunos contenidos de interés, pero estas políticas son independientes de las preferencias de cada usuario y mantienen además la misma representación para todos. Cuando un usuario quiere visualizar un contenido publicado en la plataforma, puede acceder al mismo mediante tres variantes: desde la portada principal, navegar por el menú hasta encontrar la categoría donde se encuentra el material que desea o utilizar el buscador. La probabilidad de que un usuario encuentre en portada el material de su preferencia es baja, lo que ocasiona que se haga uso de cualquiera de las dos variantes restantes.

Para encontrar un material, los usuarios deben acceder al menú, navegando por las distintas categorías definidas según los contenidos publicados hasta encontrar la que compone el contenido multimedia de su preferencia. Para encontrar un material mediante esta variante el usuario debe tener una noción de la categoría definida para el material que busca. Otra opción disponible es utilizar el buscador, presente en la página principal, el cual lista los archivos multimedia que se corresponden con los criterios de búsqueda especificados. Para utilizar esta opción el usuario debe conocer de algunas características del material que desea, dígame autor, año de producción u otros criterios definidos.

Ambas variantes descritas anteriormente implican tanto esfuerzo, como conocimiento previo por el usuario de características del material que desea visualizar, esto propicia, en el peor de los casos, un aumento del tiempo empleado en la visualización de un material, lo que puede ser traducido en disminución a largo plazo de la fidelidad hacia la plataforma. Se considera que con una gestión eficaz de los contenidos publicados en base a las preferencias de cada usuario se pueda brindar un servicio adaptado a las necesidades de cada consumidor, lo que permitiría que los mismos se sientan identificados con la plataforma.

## **1.3. Bases conceptuales asociadas al dominio del problema**

Como estrategia para la conceptualización de los sistemas de recomendación, en el presente apartado se realiza un estudio de los principales conceptos que se abordan en el objeto de estudio.

### **1.3.1. Preferencias**

El término preferencia permite señalar la ventaja o primacía que algo o alguien tiene sobre otra cosa o persona. Dichas preferencias pueden surgir por distintos motivos, como el valor, el merecimiento o los intereses personales. Estas definiciones representan su significado en general, pero a su vez en el sector de la economía las preferencias de un consumidor determinan su consumo. Las preferencias generales de la sociedad son decisivas para el éxito de un comercio y, en general, para la estructura del mercado. (Definicion.de, 2012)

El concepto de preferencia describe la disyuntiva a la que se ven enfrentados los consumidores cuando tienen que elegir entre varios productos que les son ofertados. Teniendo como base que el consumidor puede realizar una comparación entre dos o más productos de consumo, puede resultar: que el mismo sea indiferente a los productos que está considerando o que considere uno con mayor preferencia que otros. (Colell, 1995)

En el ámbito del problema planteado las preferencias se basan en la aceptación o predilección que tiene un usuario hacia un material audiovisual publicado en la plataforma, como también la inclinación hacia algunas categorías, quienes a su vez definen un grupo de materiales audiovisuales.

### **1.3.2. Recomendación**

La definición general de “recomendación” es expresada como el consejo que se da a una persona por considerarse ventajoso o beneficioso para la misma (RAE, 2012). Enmarcado en la informática, las recomendaciones intentan proporcionar sugerencias significativas sobre los recursos que los usuarios puedan encontrar interesantes y útiles. Existen varios tipos de recomendaciones, entre ellas se encuentran las recomendaciones personales, díganse de amigos, familiares, asesores de ventas y las impersonales, que son las definidas por la comunicación comercial masiva y sistemas expertos. (Cruz, 2010) La automatización del proceso de obtener sugerencias de otras personas se basa en las recomendaciones personalizadas. Estas sugerencias pueden extraerse mediante las calificaciones que un usuario le da a los recursos a través de votaciones o mediante su historial de navegación. (Konstan, 2004)

Atendiendo al dominio del problema, las recomendaciones pueden ser definidas como la selección de  $n$  archivos multimedia, cuyas categorías estén dentro de un rango estrecho de similitud con las preferencias del usuario autenticado en el sistema, en ese instante de tiempo.

### 1.3.3. Perfil del usuario

El perfil del usuario es un formulario que permite seleccionar los datos correspondientes a los elementos a recomendar, que mejor se corresponden con las preferencias del usuario. Los tipos de información que se pueden contener en el perfil del usuario son (Font, 2009):

- **Preferencias del usuario:** contiene una relación de los tipos de elementos que el usuario puede o no estar interesado.
- **Historial del usuario:** se almacenan las interacciones que el usuario ha tenido con el sistema, como las valoraciones de los elementos, estos aspectos pueden ser utilizados por el sistema para asimilar gustos y preferencias del usuario.

La plataforma cuenta con la funcionalidad de gestionar datos del perfil de usuario, accesible para los usuarios autenticados, además almacena los datos gestionados en un registro asociado a la tabla usuario. Este perfil muestra dinámicamente los descriptores asociados a las medias, de modo que el usuario pueda definir su predilección por cada uno de ellos. Como posible solución para el almacenamiento de la información del historial del usuario, puede ser la concepción de un registro asociado a la tabla usuario almacenada en la base de datos. Este registro recogerá las preferencias del usuario a partir de las actividades realizada por el mismo en la plataforma.

### 1.4. Sistemas de recomendación

Los sistemas de recomendación se pueden definir como un tipo específico de filtro de información, cuyo objetivo es mostrar elementos al usuario que le sean relevantes o de interés. Se entiende por filtro de información un sistema que elimina información inadecuada o no deseada de un flujo de información de forma automática o semiautomática, para ser presentada a los usuarios. (Font, 2009)

Estos sistemas se han ido consolidando como potentes herramientas para ayudar a reducir la sobrecarga de información a la que se enfrentan los usuarios en los procesos de búsqueda. Ayudan a filtrar los elementos de información recuperados, usando distintas técnicas para identificar aquellos elementos que mejor satisfacen las preferencias o necesidades de los usuarios. (Herrera, y otros, 2011)

A lo largo de los años la definición de los sistemas de recomendación ha ido evolucionando, estando ligada a los avances de las nuevas técnicas o ideas que van surgiendo en su estudio. Algunas de sus definiciones más completas fueron formuladas en el 2005, una de ellas define que: “un sistema de recomendación es una solución de software típico, usada en el comercio electrónico para brindar servicios

personalizados, ayuda a los clientes a encontrar productos que les gustarían comprar, proporcionando recomendaciones basadas en sus preferencias” (Kim, et al., 2005). Además, la determinada por los autores Sung-Hwan Min, Ingo Han los conceptualiza como: “el filtrado de información que aplica técnicas de análisis de datos para el problema de ayudar a los clientes a encontrar los productos que les gustaría adquirir, realizando una predicción de calificaciones semejantes o un listado de productos a recomendar para un determinado cliente.” (Han, et al., 2005)

Con el análisis de ambas definiciones se puede conceptualizar un sistema de recomendación como: una herramienta que brinda a los usuarios sugerencias sobre productos que cumplen con sus preferencias, utilizando una o varias técnicas para generar las recomendaciones. Estas técnicas se basan en los datos que se almacenan sobre los elementos que recomiendan, conocimiento del entorno en el que se desarrollan o por el comportamiento de la comunidad que interactúa con el sistema. Los datos asociados a cada técnica o método de recomendación son denominadas como entradas del sistemas.

#### **1.4.1. Entradas de los sistemas de recomendación**

En las entradas del sistema de recomendación la información sobre los usuarios puede venir dada de dos formas, que no son mutuamente excluyentes. Una de estas formas es **por extensión**, la cual se refiere a información que se tenga sobre las experiencias pasadas del usuario con respecto a los elementos encontrados, es lo que también se conoce como **navegación implícita**, pues el usuario no es consciente de estos seguimientos. La otra forma es la información expresada **intencionalmente**, también se le llama **navegación explícita** y consiste en que el usuario expresa intencionalmente sus preferencias, esta forma se evidencia en los datos recogidos en el perfil de usuario. (Herrera, y otros, 2011)

Para el sistema de recomendación de la plataforma Inter-nos 2.0 puede adoptarse la obtención de información tanto implícita como explícita. Para ello se puede hacer uso de los datos del historial de navegación del usuario y del perfil del mismo con que cuenta la plataforma.

#### **1.4.2. Métodos de generación de recomendaciones**

Para generar las recomendaciones existen diferentes métodos clasificados en diversos criterios como: las fuentes que los financian, según los recursos de información que utilizan, según su movilidad, según su funcionamiento, según la aplicación de modelos simbólicos y según los atributos que deben exhibir. (Cruz, 2010) Todas estas clasificaciones separan los métodos de recomendación en grupos definidos según características propias para su estudio o utilización. Para el desarrollo de la investigación se han

seleccionado los conjuntos definidos para los métodos de recomendación según el criterio de funcionamiento, donde se delimitan en:

- **Sistemas de recomendación basados en contenido:** los sistemas de recomendación basados en contenido se encargan de encontrar productos similares a aquellos que le han gustado al usuario, en función del contenido de los mismos. El objetivo final será recomendar a un usuario, aquellos productos que más se aproximen a sus preferencias, obtenidas bien explícitamente (mediante un formulario o cuestionario) o implícitamente (analizando los registros de enlaces visitados, artículos vistos o votados). (Campos, et al., 2011)
- **Sistemas de recomendación colaborativos:** los sistemas de recomendación de filtrado colaborativo, también denominados sistemas de recomendación sociales, acumulan una base de datos de calificaciones de elementos, dadas por un conjunto de usuarios con preferencias similares. Para predecir las preferencias que puede tener un usuario sobre los elementos que no ha visto aún, se buscan sub-conjuntos de dicho grupo que hayan visto elementos no visitados por el usuario y estos le son recomendados; es de suponer que ese mismo elemento guste al usuario porque comparte preferencias similares al resto del grupo. (Salazar, 2006)
- **Sistemas de recomendación basados en conocimiento:** los sistemas de recomendación basados en conocimiento realizan las sugerencias de los elementos fundamentadas en inferencias sobre las necesidades de los usuarios y sus preferencias. Para ello se utiliza información sobre cómo un elemento específico responde a una necesidad en particular del usuario y a su vez, la razón sobre la relación entre la necesidad y una posible recomendación. Estas características hacen que este tipo de sistema de recomendación no necesite un gran cúmulo de datos sobre los elementos o usuarios, pero necesita especialistas que provean al sistema de este conocimiento. (Font, 2009)
- **Sistemas de recomendación basados en la utilidad:** los sistemas de recomendación basados en utilidad crean una función de utilidad para cada elemento, la cual interviene directamente en el proceso de recomendación. La ventaja de este método es que permite evaluar elementos no atribuibles al producto o elemento en sí, estos serían aspectos como la fiabilidad de un proveedor o la disponibilidad de un elemento. (Font, 2009)
- **Sistemas de recomendación demográficos:** los sistemas de recomendación demográficos 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 contenido, con la excepción de que las similitudes están calculadas a partir de la utilización de información demográfica en lugar de valoraciones de los elementos. (Font, 2009)

- **Sistemas de recomendación híbridos:** los sistemas de recomendación híbridos combinan diversos métodos de recomendación. Este tipo de sistemas de recomendación puede ser desarrollado a partir de cualquier combinación de varias técnicas, como consecuencia se atenúan las desventajas de cada una de ellas y se combinan los beneficios de las recomendaciones de ambos métodos. (Font, 2009)

Como forma de hacer uso de la funcionalidad de gestionar datos del perfil y la persistencia de trazas que posibilita el gestor de contenidos Drupal, como se explica en la descripción del dominio, se considera factible la utilización de un método de recomendación basado en contenido. Conjuntamente, la plataforma cuenta con la funcionalidad *votinapi*, que permite almacenar los votos realizados por cada usuario a los archivos multimedia, permitiendo adquirir información sobre medias más valoradas por grupos de usuarios que comparten preferencias similares, facilitando la utilización de un método por filtrado colaborativo. Para aumentar la precisión de las recomendaciones finales y para eliminar las limitaciones de ambos métodos de recomendación, que no hayan sido subsanadas con las características del dominio, se propone su hibridación, lo que permite además, la utilización en conjunto de las características que provee la plataforma.

#### **1.4.3. Salidas de los sistemas de recomendación**

Las salidas de los sistemas de recomendación están enfocadas a mostrar las recomendaciones finales al usuario. Para ello pueden utilizarse diferentes representaciones, como una lista de sugerencias, predicciones del grado de satisfacción asociadas a cada recomendación o las votaciones realizadas a cada elemento. (Herrera, y otros, 2011) Para mostrar las recomendaciones en la plataforma, se presentarán en la portada principal una lista de sugerencias con las medias de mayor preferencia para el usuario.

### **1.5. Sistema de recomendación híbrido**

En la presente investigación se define como sistema de recomendación híbrido a la integración de los métodos de recomendación colaborativo y basado en contenido. A continuación se explica el

funcionamiento de cada uno y sus principales limitaciones, concluyendo con las distintas técnicas de hibridación existentes.

### 1.5.1. Funcionamiento del método basados en contenido

El funcionamiento del filtrado basado en contenido se basa en la selección de aquellos elementos que aún no han sido visitados por el usuario, teniendo en cuenta las características de cada uno de ellos. (Font, 2009) Enmarcadas en el campo de acción se refiere a los descriptores definidos para cada media que carga el módulo de perfil de usuario dinámicamente, así como la visualización de elementos con las mismas características, estas preferencias arrojan un resultado que permite definir si los elementos con estas características tienen mucho o poca aceptación para el usuario, lo que se traduce en poder recomendarle o no el elemento en cuestión.

#### Inferencias sobre el conocimiento

Una parte fundamental de los sistemas de recomendación es el procesado de los datos (preferencias de los usuarios), permitiendo la extracción de conclusiones que puedan ser traducidas en recomendaciones. Estos algoritmos crean una función que modela los intereses de los usuarios. Por consiguiente, dado un modelo de usuario y un elemento nuevo, la función predice si el elemento será del agrado del usuario o no. A continuación se presentan diferentes formas de inferir en el conocimiento almacenado en el sistema de recomendación (Campos, et al., 2011):

- **Personalización del usuario:** el sistema proporciona una interfaz donde el usuario puede aportar las preferencias de sus gustos. Una vez que el usuario ha proporcionado sus preferencias, el sistema simplemente busca los elementos que encajan con los parámetros indicados por el usuario.
- **Reglas de asociación (*Associative rules*):** se intenta descubrir relaciones entre los elementos, que luego pueden ser utilizadas para la recomendación. Los elementos se comparan en base al comportamiento pasado de los usuarios respecto a ellos.
- **Método del vecino más cercano:** permite clasificar un nuevo dato basándose en observaciones conocidas o pasadas, se utiliza para determinar valores de un elemento sin valorar. Para ello, el nuevo elemento que carece de valoración, se compara (mediante el uso de los atributos del elemento) con los otros elementos ya valorados. Mediante una función de similitud, se obtienen los que más se corresponden con el nuevo elemento.

- **Feedback relevante:** método que propone obtener información sobre los elementos recomendados al usuario cuando éste ha realizado una búsqueda.
- **Bayesiano naive (Naïve Bayes):** método de clasificación probabilístico, que se utiliza para clasificar un nuevo elemento dentro de las preferencias del usuario. Dado un nuevo elemento y un conjunto de usuarios, se calcula la probabilidad de que el elemento tenga cavidad dentro de cada grupo de preferencias. Arrojan un importante valor cuantitativo que define una medida probabilística de la importancia de las variables en el problema.

Los diferentes algoritmos para inferir conocimiento permiten realizar recomendaciones a partir de las preferencias de usuarios. La personalización del usuario, basada en el perfil con que cuenta la plataforma y el método del vecino más cercano, para la realizar sugerencias de las últimas publicaciones, posibilitan la obtención de recomendaciones utilizando datos almacenados sobre los archivos multimedia y su correspondencia con las preferencias del usuario.

### Problemas generales

Los métodos de recomendación basados en contenido presentan los siguientes inconvenientes: (Font, 2009)

- **Limitación por el análisis del contenido:** el sistema de recomendación se ve limitado por las propiedades que están explícitamente asociadas al elemento, lo que hace necesario tener las suficientes propiedades de cada uno para un correcto funcionamiento.
- **Sobre-especialización:** aparece cuando el sistema de recomendación sólo muestra al usuario elementos similares a los que ya ha visto antes.
- **Usuario nuevo:** un usuario tiene que haber valorado algunos elementos antes de que el sistema pueda tener registrados sus gustos y preferencias para poderle recomendar.

La limitación por el análisis de contenido no presupone un problema en el sistema de recomendación a desarrollar, para ello se utilizan los descriptores almacenados en la plataforma, asociados a cada media. El resto de los inconvenientes pueden ser atenuados e incluso eliminados con el desarrollo de un sistema de recomendación híbrido.

#### 1.5.2. Funcionamiento del método colaborativo

Los sistemas de recomendación que utilizan técnicas de filtrado colaborativo se adaptan al siguiente funcionamiento (Konstan, 2004):

- Los usuarios elaboran valoraciones sobre elementos.
- Se analizan las valoraciones realizadas por los usuarios y se establecen grupos (vecinos cercanos) de usuarios con preferencias similares u elementos con características similares.
- Una vez obtenidos los grupos, se realizan las recomendaciones al usuario activo teniendo en cuenta sus vecinos más cercanos y sus correspondientes valoraciones.

Los algoritmos de recomendación colaborativa pueden basarse en usuarios o en elementos. Estos algoritmos se basan en el producto punto entre dos vectores y en las fórmulas de correlación. El grado de cercanía entre dos vectores, es decir la similitud entre dos personas o elementos, se puede obtener calculando el coseno del ángulo que forman o por la correlación de Pearson (Olivera, 2011).

### **Basados en usuarios**

Las recomendaciones se realizan buscando usuarios con gustos similares. En un sistema de recomendación con **m** usuarios y **n** recomendaciones, se define el vector  $U_i$  como aquel cuyas

coordenadas corresponden al número de veces que el usuario **i** utilizó cada elemento. El sistema debe calcular la similitud entre cada par de usuarios para generar una matriz de similitud.

$$\cos(U_i, U_j) = \frac{U_i * U_j}{|U_i| |U_j|}$$

**Ecuación 1: Cálculo de la similitud entre cada par de usuarios.** (Olivera, 2011)

Para determinar cuáles usuarios son similares a un usuario particular, se debe seleccionar un umbral, del cual los usuarios obtenidos se denominan vecinos cercanos. (Macias, et al., 2011) El inconveniente que presenta este algoritmo es que las relaciones han de ser constantemente recalculadas, lo que es muy costoso en procesamiento cuando el sistema tiene millones de usuarios. (Olivera, 2011)

### **Basados en elementos**

El principio de este algoritmo es básicamente el mismo que el algoritmo basado en usuarios, la diferencia radica en que se buscan similitudes entre elementos, en lugar de buscar similitudes entre usuarios. Las recomendaciones se realizan considerando elementos que tienen valoraciones similares por parte de muchos usuarios. Este método puede ser implementado para millones de elementos y de usuarios. (Olivera, 2011)

Según las características de los algoritmos anteriormente expuestos, se puede definir que los algoritmos basados en elementos son menos costosos en procesamiento que los algoritmos basados en usuario, debido a que la similitud entre elementos es menos variable que la similitud entre usuarios. Además, la plataforma Inter-nos 2.0 cuenta con mayor número de usuarios que de elementos, lo que define una disminución de los cálculos a realizar si se utilizan algoritmos basados en elementos.

Los sistemas de recomendación basados en filtrado colaborativo se clasifican en **basados en memoria o heurísticos**, los cuales realizan sus predicciones teniendo en cuenta todos los elementos evaluados o información disponible en el sistema y los **basados en el modelo**, los que utilizan un conjunto de valoraciones para crear un “modelo”, el cual es utilizado a posteriori para crear predicciones de valoraciones. Generalmente los algoritmos de filtrado colaborativo basado en memoria son utilizados en sistemas de recomendación comerciales y los basados en modelos son asociados a los sistemas de recomendación investigativos. (Sun, 2012)

En la plataforma Inter-nos 2.0 se almacena la información necesaria para generar las predicciones de las recomendaciones, directamente en la base de datos. Esto posibilita realizar estos cálculos sin la necesidad de generar un modelo para ello, haciendo uso de algoritmos basados en memoria. Una vez realizados los grupos de medias a partir del cálculo de la similitud, se procede a la generación de las recomendaciones a partir de la selección de los archivos multimedia más votadas del grupo que mejor se corresponde con las preferencias del usuario.

### **Problemas generales**

Los métodos de recomendación por filtrado colaborativo presentan los siguientes inconvenientes:

- **Escasez:** Los métodos colaborativos necesitan una cantidad cuantiosa de usuarios puntuando por elementos similares, para de esta forma poder calcular los grupos de vecinos y, en base a ellos, realizar las recomendaciones. Si en la base de datos se tienen pocos usuarios o pocas puntuaciones por parte de cada usuario, la matriz de puntuaciones será muy escasa y los cálculos de vecindad no pueden ser realizados con exactitud, obteniendo recomendaciones de baja calidad. (Feria, 2012)
- **Escalabilidad:** Los métodos de recomendación colaborativos que utilizan algoritmos de cálculo de los k vecinos más cercanos (knn, *K-nearest neighbors*) para obtener la similitud entre usuarios, presentan el problema de escalabilidad. Definido porque estos algoritmos son costosos

computacionalmente y su coste crece linealmente cuanto mayor sea el número de usuarios, de elementos y de datos asociados a estos. (Jeffrey, y otros, 2011)

- **Problema del elemento o usuario nuevo:** los elementos nuevos, que tienen muy pocas o incluso, ninguna puntuación no van a ser recomendados prácticamente nunca. De la misma forma, los nuevos usuarios en el sistema recibirán muy pobres predicciones debido a que ellos han puntuado muy pocos elementos y se hace difícil encuadrarlos en algún grupo de vecinos. Estos dos hechos muestran que estos sistemas de recomendación requieren un cierto tiempo antes de empezar a hacer predicciones y recomendaciones ciertamente relevantes y acertadas. (Macias, y otros, 2011)

El portal de la plataforma Inter-nos 2.0, es el sitio oficial de distribución de contenido multimedia de la universidad, lo que define que mantiene una alta concurrencia de usuarios ya familiarizados con la plataforma y a su vez con su sistema de votaciones, quedando así resuelto el problema de la escasez. El problema de escalabilidad se redujo a partir de la utilización del algoritmo para el cálculo de las  $k$  medias más cercanas (*k-means*) para obtener la similitud entre los archivos multimedia, pues la cantidad de medias que maneja la plataforma es en gran medida menor que la cantidad de usuarios y sus características no varían. Aun así, la hibridación permite que si el método por filtrado colaborativo no arroje ningún resultado, se puedan obtener las recomendaciones a partir del método basado en contenido, permitiendo minimizar el resto de las limitaciones.

### 1.5.3. Técnicas de hibridación

Para la integración de varios métodos de recomendación en un mismo sistema de recomendación, se hace necesario asumir una estrategia que posibilite una combinación fructífera, estas estrategias se pueden clasificar en (Font, 2009):

#### **Por pesos (*weighted*)**

En la combinación por pesos, el valor de la recomendación de un elemento se obtiene, ponderando los diferentes resultados obtenidos por los sistemas de recomendación. Ocasionalmente el resultado de una recomendación no se puede ponderar porque el sistema de recomendación utilizado no ofrece un valor que expresa el grado de similitud o nivel de agrado del elemento, en estos casos en lugar de ponderar los resultados de las recomendaciones, se realiza la unión o intersección de los resultados obtenidos por los diferentes sistemas de recomendación y los elementos resultantes se denominan candidatos.

### **Conmutados (*switching*)**

Los sistemas de recomendación híbridos conmutados utilizan un criterio para establecer qué sistema de recomendación utilizar en cada momento. Existen dos posibilidades para este caso:

- Se determina qué resultados mostrar a partir de los resultados obtenidos por los sistemas de recomendación involucrados.
- Se selecciona el sistema de recomendación a utilizar antes de procesar ninguna información.

### **Mezclados (*mixed*)**

La combinación de varios sistemas de recomendación mediante el mezclado permite realizar las recomendaciones simultáneamente de más de un método, lo que arroja como resultado una lista de varias recomendaciones.

### **Combinación de propiedades (*feature combination*)**

La combinación de propiedades adapta las propiedades o rasgos de un tipo de sistema de recomendación a otro tipo diferente de sistema de recomendación para ser utilizadas posteriormente. Un método para ello puede ser añadir reglas de aprendizaje colaborativo a un sistema de recomendación basado en contenido. La combinación de propiedades no utiliza un solo sistema de recomendación, sino que realiza una combinación de fuentes de conocimiento. Su estrategia se basa en tomar la lógica de las recomendaciones de un tipo de técnica en lugar de utilizar un componente o artefacto que la implementa, lo que posibilita que sólo exista un sistema de recomendación, el cual adquiere conocimiento o propiedades de una fuente asociada a otro sistema de recomendación.

### **En cascada (*cascade*)**

La hibridación en cascada utiliza un método de recomendación para elaborar una lista de posibles recomendaciones y a esta lista de recomendaciones se le aplica un segundo algoritmo de recomendación, posibilitando refinar con un sistema de recomendación las recomendaciones dadas por otro sistema de recomendación. Se puede utilizar un sistema de recomendación basado en contenido, para obtener una lista con todos los elementos que se correspondan a las preferencias del usuario y refinar esta con un sistema de recomendación colaborativo, permitiendo seleccionar los elementos que mejor valoración popular tengan, lo que posibilita obtener recomendaciones de mejor calidad.

### **Aumento de cualidades (*feature augmentation*)**

El aumento de cualidades combina la utilización de dos sistemas de recomendación, uno contribuye realizando recomendaciones que aportan información complementaria sobre un elemento y luego estas recomendaciones son utilizadas por un segundo sistema de recomendación como elementos de entrada del mismo. En síntesis, los resultados de una técnica o sistema de recomendación son usados como elementos o cualidades de entrada de otro sistema de recomendación.

### **Meta niveles (*meta-level*)**

En los sistemas híbridos que utilizan meta niveles el modelo aprendido por un sistema de recomendación es usado como fuente de entrada para otro sistema de recomendación. El enfoque se asemeja al de “aumento de cualidades” en el hecho de que un sistema de recomendación contribuye en la entrada de información de otro sistema de recomendación, pero en el caso de los meta niveles, la contribución del primer sistema de recomendación reemplaza completamente la fuente original de conocimiento.

Es importante considerar que algunas de las estrategias de los sistemas híbridos, díganse: por pesos, conmutados, mezclados y combinación de propiedades presentan 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: en cascada, aumento de cualidades y meta niveles, se debe tener en cuenta el orden de los algoritmos a aplicar, pues esto implicaría resultados diferentes.

Con vistas de mantener un mayor alcance en la administración del sistema de recomendación, se considera factible la utilización de dos o más estrategias de hibridación, brindando además la posibilidad al administrador de seleccionar cuál utilizar. Entre las estrategias que podrían ser desarrolladas se encuentran la hibridación conmutada, mezclada y por pesos. Se seleccionan métodos asociativos, para posibilitar la integración de nuevos métodos de generación de recomendaciones en futuras versiones del módulo de recomendación, sin tener que prestar interés en el orden en que se ejecuten con respecto a los ya implementados.

## **1.6. Análisis de soluciones existentes**

La función principal de los sistemas de recomendación se basa en la personalización de la experiencia del consumidor en lo relacionado con la presentación de los productos ofrecidos en un sitio web. Varios

comerciantes y compañías han obtenido resultados con la creación de sistemas de recomendación. A continuación se describen las características de los sistemas de recomendación analizados:

### **1.6.1 Fab**

Es un sistema orientado a la recomendación de páginas web que combina el uso de información por extensión con el enfoque colaborativo, lo que se define como un sistema de recomendación híbrido integrado por métodos colaborativos y basado en contenido. 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 agente central de procesamiento 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, 2010)

Para su funcionamiento el agente recolector encuentra páginas web sobre un tema específico, estas páginas son enviadas al agente central o núcleo del sistema de recomendación; el cual compara los perfiles de usuario y envía los elementos que coinciden con las preferencias de los usuarios correspondientes a cada perfil. El agente seleccionador descarta algunas páginas que el usuario ya ha visto y las direcciones web repetidas, luego el usuario recibe las recomendaciones y las puntúa. Cualquier página web altamente puntuada pasa a ser recomendada a los vecinos más cercanos (extraídos por el sistema de recomendación colaborativo). (Olivera, 2011)

### **1.6.2. 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 propuestas 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)

### **1.6.3. Avatar**

Avatar es un sistema de recomendación híbrido de contenidos multimedia basado en agentes. Este sistema híbrido combina tres técnicas de recomendación: dos de filtrado colaborativo y una basada en

contenido. El sistema de recomendación se divide en tres módulos: el Agente Bayesiano (basado en filtrado colaborativo) que utiliza la información referente a los programas vistos y las preferencias de los usuarios para realizar las recomendaciones, el Agente Semántico (basado en contenido) que utiliza información actualizada sobre las preferencias de los usuarios y conocimiento del dominio, y el Agente de Perfiles (basado en filtrado colaborativo) que compara distintos perfiles de usuario para realizar las predicciones. Las recomendaciones de los tres agentes son combinadas utilizando técnicas de redes neuronales. (Busquet, 2008)

#### **1.6.4. Sistema de recomendación colaborativo para la plataforma Video Web**

El sistema de recomendación colaborativo, es un módulo creado para la plataforma VideoWeb, con el objetivo de proveer un servicio que cumpla con las nuevas tendencias de personalización de las ofertas. Es un sistema de recomendación de filtrado colaborativo que utiliza el algoritmo K-nn para realizar el proceso de la recomendación, además de crear grupos de usuarios con preferencias similares, teniendo en cuenta las votaciones realizadas al material audiovisual visualizado por los mismos. Está desarrollado haciendo uso de las tecnologías que se utilizaron en el desarrollo de la plataforma VideoWeb. (Feria, 2012)

#### **1.6.5. Sistema de Recomendación Basado en Contenido para la Plataforma VideoWeb.**

El sistema de recomendación basado en contenido, está encapsulado en un módulo para la plataforma VideoWeb, es capaz de generar recomendaciones basándose en las preferencias de los usuarios que acceden a la misma. Utiliza técnicas de recomendación basadas en el contenido de los archivos multimedia que se publican en la plataforma. El sistema utiliza técnicas para la extracción de preferencias tanto explícitas como implícitas, haciendo uso de los datos almacenados en el perfil de usuario y en el historial de navegación respectivamente. (Brown, 2012)

### **1.7. Conclusiones parciales**

Mediante el análisis de las bases conceptuales enfocadas al dominio del problema, se pudo constatar las dificultades que presentan los usuarios al navegar por la plataforma Inter-nos 2.0 para encontrar un material de su preferencia. Estas dificultades, son propiciadas por el cúmulo de contenidos multimedia publicados en la plataforma, definiendo que la búsqueda de un material puede ser agotadora para los usuarios e incluso provocar un rechazo hacia la misma. Se analizaron además, los métodos de recomendación y se seleccionaron los que mejor se ajustaron al dominio del problema. Se definió que, a

partir de la posibilidad de la plataforma para la obtención de las acciones del usuario, la persistencia de los datos del perfil y la funcionalidad de votaciones a las medias, se hizo uso de los métodos de recomendación colaborativo y basado en contenido que podrán beneficiarse de dichas características. Ambos métodos tienen limitaciones que fueron atenuadas a partir de su hibridación, concretada con técnicas asociativas, que permitan la inclusión de otros métodos de recomendación en futuras versiones del módulo.

Con el análisis de las soluciones existentes, se concluyó que algunos de los sistemas de recomendación analizados están creados para un contexto específico, distinto al dominio en el que se desarrolla la problemática planteada. En el caso del sistema de recomendación Avatar, que cumple con las mismas características de la plataforma Inter-nos 2.0, carece de una interfaz de comunicación que permita su integración con el sistema gestor de contenido Drupal. De igual forma los módulos creados para la plataforma VideoWeb no se pueden utilizar en su totalidad, porque a pesar de que ambas plataformas están creadas con las mismas tecnologías, Inter-nos 2.0 es una personalización enfocada a otro negocio, lo que conlleva a cambios en el modelo de datos. De estos módulos se aprovecharon algunas características de los algoritmos de filtrado colaborativo y filtrado basado en contenido que implementan cada uno respectivamente.

## **CAPÍTULO 2: Selección de herramientas y tecnologías**

En el presente capítulo se realiza un estudio de las herramientas y tecnologías factibles para utilizar en el desarrollo del módulo. Se selecciona la metodología de desarrollo de software que guiará el proceso de desarrollo de la solución, además del lenguaje de modelado y la herramienta CASE que permitirá realizar los diagramas correspondientes al diseño. Se analizan y seleccionan las tecnologías a utilizar en la implementación, entre las que se encuentran el entorno de desarrollo integrado, el lenguaje de programación, el sistema gestor de Base de datos y el sistema gestor de contenidos.

### **2.1. Metodología de desarrollo de software**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayudan y guían el desarrollo de un producto de software. Se refiere a un marco de trabajo (*framework*) que es utilizado para estructurar, planear y controlar el proceso de desarrollo de sistemas informáticos, además de definir el conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto. Se pueden clasificar en dos grupos generales: las pesadas (o tradicionales) y las ligeras (o ágiles). (Jacobson, 2000)

Las metodologías tradicionales están orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y requieren de una documentación considerable. Se ajustan más a proyectos grandes, con mayor alcance de tiempo. Por otra parte las metodologías ágiles están orientadas a la interacción con el cliente y el desarrollo incremental del software, para ello se muestran versiones parcialmente funcionales del software en intervalos cortos de tiempo, para que el cliente pueda evaluar y sugerir cambios en el producto según se va desarrollando. (Silva, 2003)

#### **Proceso Unificado de Software**

Entre las metodologías tradicionales más utilizadas está el Proceso Unificado de Software (*RUP* por sus siglas en inglés). Entre sus principales características se encuentra que es dirigido por casos de uso, lo que define una guía en los procesos de diseño, implementación y prueba, que responde directamente a los requisitos funcionales del software. Otra característica importante es la de ser centrada en la arquitectura, que permite tener una perspectiva clara del sistema completo, además de ser iterativo e incremental, definiendo que por cada iteración se creen incrementos del producto final. (Jacobson, 2000)

Se cuenta con la perspectiva de que el módulo a desarrollar pueda ser utilizado en otras personalizaciones del producto Video-Web, por lo que se hace necesaria una cuantiosa documentación de todo el proceso para facilitar su posterior integración con otros negocios. Una de las formas de lograr un registro detallado de la solución es la utilización de la metodología de desarrollo de software RUP, la que será utilizada para documentar los procesos sustanciales en el desarrollo del módulo.

## **2.2. Lenguaje de modelado unificado**

Lenguaje de Modelado Unificado (*UML* por sus siglas en inglés) es un lenguaje para la modelación de un sistema con tecnologías orientadas a objetos. Además permite visualizar, especificar, construir y documentar la solución. (Jacobson, 2000) UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación y esquemas de bases de datos. (Ryan, 2009) Presenta una serie de características que lo sitúan como un lenguaje de modelado factible para ser utilizado en el modelado de la solución, entre las que se encuentran: viabilidad en la corrección de errores, modelación de estructuras complejas, posibilidad de conectarse con lenguajes de programación, además de permitir documentar varios artefactos en un proceso de desarrollo. Por la descripción anterior queda seleccionado UML como lenguaje de modelado a utilizar.

## **2.3. Herramienta CASE Visual Paradigm**

Herramienta de Ingeniería de Software Asistida por Computadora (*CASE* por sus siglas en inglés), son diversas aplicaciones informáticas o métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información y permiten aumentar la productividad en el desarrollo de software, reduciendo el costo de las mismas en términos de tiempo y dinero. Las herramientas CASE proporcionan un beneficio para las organizaciones al facilitar el proceso de revisión ya que éste proporciona bases para las definiciones y estándares para los datos. Estas herramientas incluyen las capacidades de generación interna, lo que contribuye a modificar el sistema por medio de las especificaciones más que por los ajustes al código fuente. (Cordero, 2007)

Visual Paradigm es una herramienta CASE que incluye las capacidades de modelar diagramas de clases, realizar ingeniería inversa, generar código desde diagramas y generar documentación. Esta herramienta proporciona tutoriales del lenguaje de modelado UML, demostraciones interactivas del mismo y proyectos

UML; presenta licencia gratuita y comercial. Este software es fácil de instalar y actualizar y compatible entre ediciones.

Visual Paradigm posee una capacidad de integración con diferentes entornos de desarrollo integrado, entre los que se encuentran NetBeans, JDeveloper y Eclipse. (VP.org, 2007) Cuenta con otras características como exportación de diagramas en imágenes jpg, png y svg, además de ser una herramienta multiplataforma que emplea las últimas notaciones del lenguaje de modelado UML. El Visual Paradigm soporta el ciclo de vida completo del desarrollo de software y una de las licencias bajo la que se encuentra publicado es gratuita, lo que facilita su utilización sin gastos adicionales. (Cordero, 2007)

Según las características estudiadas en las herramientas CASE investigadas, se considera factible la utilización del Visual Paradigm. Entre las características que llevaron a su selección se encuentran la capacidad de integración con el entorno de desarrollo integrado NetBeans, soporte para las últimas notaciones del lenguaje de modelado UML, además de su capacidad de generación de código en PHP a partir de modelos.

#### **2.4. Entorno de desarrollo integrado NetBeans**

Entorno de desarrollo integrado (*IDE* por sus siglas en inglés) es un programa informático compuesto por un conjunto de herramientas de programación, díganse un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (*GUI* por sus siglas en inglés). Puede dedicarse a uno o varios lenguajes de programación, además de proveer un entorno de trabajo amigable para la mayoría de estos. (Farfán, y otros, 2010)

El entorno de desarrollo integrado *NetBeans* es una herramienta multilenguaje escrita en java, con un enfoque modular. En la actualidad está publicada bajo una licencia libre. Como una de sus características importantes destaca la posibilidad de crear aplicaciones de escritorio y webs, además de proveer para estas últimas una amplia gama de características visuales, aspecto de gran utilidad en el desarrollo del módulo. (Cerde, 2009) Entre los lenguajes de programación para los que brinda soporte se encuentra PHP, lenguaje en el cual ha sido desarrollada la plataforma de Inter-nos 2.0, así como los módulos que utiliza. El *NetBeans* reconoce las funcionalidades del núcleo de Drupal, permitiendo su utilización en las aplicaciones que sean desarrolladas con este *IDE*.

## **2.5. Lenguaje de programación PHP 5.3.**

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevados a cabo por las computadoras. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Existen lenguajes de programación de escritorio y lenguajes de programación para la web. Los lenguajes de programación para la web se clasifican en lenguajes del lado del servidor y lenguajes del lado del cliente. En el presente epígrafe se describe el lenguaje de programación para la web seleccionado para el desarrollo del módulo. (Lanzillotta, 2008.)

El lenguaje de programación pre-procesador de hipertexto (*PHP* por sus siglas en inglés), es un lenguaje interpretado de alto nivel, diseñado originalmente para la creación de páginas web dinámicas con acceso a información almacenada en bases de datos. Entre sus ventajas se encuentra el consumo de pocos recursos, por lo que generalmente se ejecuta con rapidez y no tiende a ralentizar el resto de los procesos del sistema, además de su capacidad de conectividad al utilizar un sistema de extensiones modular hacia interfaces de varios tipos, tales como librerías gráficas, *XML* y encriptación. Tiene soporte para una considerable cantidad de sistemas gestores de bases de datos, entre los que se encuentra PostgreSQL, gestor utilizado para el desarrollo del módulo. Es un producto de código abierto y se encuentra publicado bajo una licencia totalmente libre. (Mah, 2009)

La plataforma Inter-nos 2.0 está desarrollada con el CMS Drupal, el cual utiliza el lenguaje de programación PHP, siendo este un factor importante en la selección de este lenguaje para el desarrollo del módulo. Además, posibilita mantener una compatibilidad con los demás módulos que utiliza la plataforma.

## **2.6. Biblioteca de funciones de recomendación *Recommender Lib*.**

*Recommender Lib* es una biblioteca de funciones que implementa varios métodos de recomendación permitiendo a su vez configurarlos, usarlos y evaluarlos. (SINBAD, 2012) El sistema *Recommender Lib* tiene dos modos de funcionamiento: uno, el uso de sistemas de recomendación y otro, la evaluación de sistemas de recomendación. Está compuesta por varias técnicas de recomendación, que pueden ser utilizadas en el modo de uso de sistemas de recomendación, como son: la de filtrado colaborativo, filtrado basado en contenido y filtrado híbrido, proporcionando una estructura común para todas ellas, lo que

permite utilizarlas de manera transparente, además de añadir otros métodos de recomendación a la biblioteca de funciones.

La funcionalidad de evaluación de sistemas de recomendación permite realizar las evaluaciones con cualquier conjunto de datos del que se disponga. Para ello incorpora una interfaz de evaluación de algoritmos de recomendación según el tipo de algoritmo que se desee evaluar, ya que cada tipo tiene una serie de requerimientos (propiedades del conjunto de datos) y restricciones (medidas de evaluación aplicables) diferentes. (*Recommender Lib*, 2012)

## **2.7. Sistema gestor de base de datos PostgreSQL 9.1**

Un sistema de gestión de base de datos (*DBMS* por sus siglas en inglés) es un conjunto de programas que permiten crear, mantener y definir base de datos para diversas aplicaciones, además de administrar y gestionar la información contenida en estas. (Lockhart, 1996)

PostgreSQL es un sistema de gestión de base de datos de código abierto, ofrece soporte al lenguaje SQL92/SQL3 y extensibilidad de tipos de datos. Incorpora los conceptos de clases, herencias y funciones que posibilita la fácil extensión del sistema, además de otras características que aportan potencia y flexibilidad como son: las restricciones, los disparadores, las reglas y la integridad transaccional. Las características anteriores lo sitúan como una base de datos objeto-relacional. (Lockhart, 1996) Por la condición de ser una herramienta pública y multiplataforma, además de tener en cuenta que la plataforma Inter-nos 2.0 trabaja utilizando dicho gestor, siendo necesario mantener una compatibilidad para la próxima integración del módulo con la misma, se decide utilizar PostgreSQL como gestor de base de datos en su versión 9.1.

## **2.8. Sistema gestor de contenidos Drupal 6.**

Sistema Gestor de Contenidos (*CMS* por sus siglas en inglés) es una herramienta que permite la gestión de contenidos en un sitio web con actualizaciones periódicas, apariencia uniforme y diseño centrado en el usuario. Su principal objetivo es proveer al desarrollador de una herramienta para la construcción de aplicaciones web, que manipulen contenidos de forma dinámica minimizando la necesidad de conocimientos técnicos en cuanto a la programación. (Formantín, y otros, 2007) Como la plataforma de Inter-nos 2.0 está desarrollada con el CMS Drupal, el módulo a desarrollar debe cumplir con la misma

característica para facilitar su integración. A continuación se describen algunas características de dicho CMS.

Drupal es un sistema gestor de contenido que permite, tanto a individuos como a comunidades de usuarios, la publicación de contenidos en una red de datos, de manera sencilla y sin tener conocimientos de lenguajes de marcas. Está desarrollado dentro de la filosofía de software libre y de código fuente abierto, publicado bajo licencia GPL (*General Public License*), licencia que permite el uso y modificación del código para desarrollar software libre. Su capacidad modular y su arquitectura consistente permiten la extensibilidad de funcionalidades en los sitios web, además de la interacción de módulos creados por cualquier desarrollador con el núcleo del sistema y con otros módulos creados por el resto de la comunidad. (Formantín, y otros, 2007)

El módulo de recomendación estará desarrollado para el núcleo (*core*) 6 de Drupal independientemente de la versión, lo que establece que se utilizarán las funcionalidades encapsuladas por Drupal para dicha versión del núcleo.

## **2.9. Conclusiones parciales**

Con la investigación de cada una de las tecnologías y herramientas analizadas, se obtuvo una guía que rigió la documentación del proceso de desarrollo de software definida por la metodología de software RUP, además se generaron los diagramas definidos por dicha metodología empleando el lenguaje de modelado UML y haciendo uso de la herramienta CASE Visual Paradigm. La integración de estas herramientas y tecnologías, permitió mantener un registro del esquema del módulo, al mismo tiempo de corregir errores en la fase de análisis y diseño y así entrar en la fase de implementación con un modelado refinado de la solución propuesta.

El entorno de desarrollo integrado NetBeans, posibilitó viabilizar el desarrollo en la fase de implementación a partir de su integración con el núcleo de Drupal, lo que permitió hacer uso de las funciones del mismo que se utilizaron en el desarrollo del módulo. Además, con el análisis de las tecnologías con las que fue implementada la plataforma Inter-nos 2.0, se definieron las que se debían utilizar en el módulo de recomendación, posibilitando su compatibilidad con los demás módulos de la plataforma; en este caso están el lenguaje de programación PHP 5.2, el gestor de base de datos PostgreSQL 9.1 y el sistema gestor de contenidos Drupal 6.

## **CAPÍTULO 3: Análisis y diseño del sistema**

En el presente capítulo se presenta una descripción del modelo de dominio. Se definen los requisitos funcionales y no funcionales que deberá cumplir el módulo a desarrollar, así como el diagrama de casos de uso por el que estará guiado su desarrollo. Se presentan además los diagramas correspondientes a las disciplinas de análisis y diseño definidos por la metodología RUP y se especifican los patrones arquitectónicos y de diseño que se emplearon en la solución propuesta.

### **3.1. Modelo de dominio**

El modelo de dominio o modelo conceptual permite de manera visual, mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software, puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. A través de los diagramas UML y por su capacidad para representar conceptos, el modelo de dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema en cuestión. (Larman, 1999)

A raíz de la poca estructura y claridad de los procesos de negocios, lo que dificulta la realización de un completo modelado del negocio, se decide hacer uso de un modelo de dominio que permita comprender y describir las clases más importantes dentro del contexto del negocio de la plataforma Inter-nos 2.0.

#### **3.1.1. Descripción de los principales eventos del entorno.**

La plataforma Inter-nos 2.0 cuenta con una interfaz principal en la cual se publican los archivos multimedia, estos son presentados según las categorías de destacados, últimas publicaciones, más vistos y más votados. Además brinda otras opciones que permiten al usuario acceder a los contenidos audiovisuales como son el menú, que mantiene diversos niveles de organización definidos por tipologías y el buscador, que lista los contenidos que se corresponden con el criterio de búsqueda que sea seleccionado.

#### **3.1.2. Objetos del dominio**

- **Usuario:** persona que interactúa con la plataforma Inter-nos 2.0 para visualizar archivos multimedia de su preferencia.

- **Plataforma Inter-nos 2.0:** sistema informático formado por componentes de hardware y software que permite la transmisión de video y audio digital a través de una red de datos.
- **Archivo multimedia:** contenido audiovisual publicado en la plataforma.
- **Menú:** permite organizar, mediante diversos niveles, los archivos multimedia agrupándolos por tipologías.
- **Tipologías:** clasificación que se le da a un archivo multimedia según su tipo.
- **Buscador:** permite acceder a una lista de contenido multimedia mediante un criterio de búsqueda.

### 3.1.3. Diagrama de clases del dominio

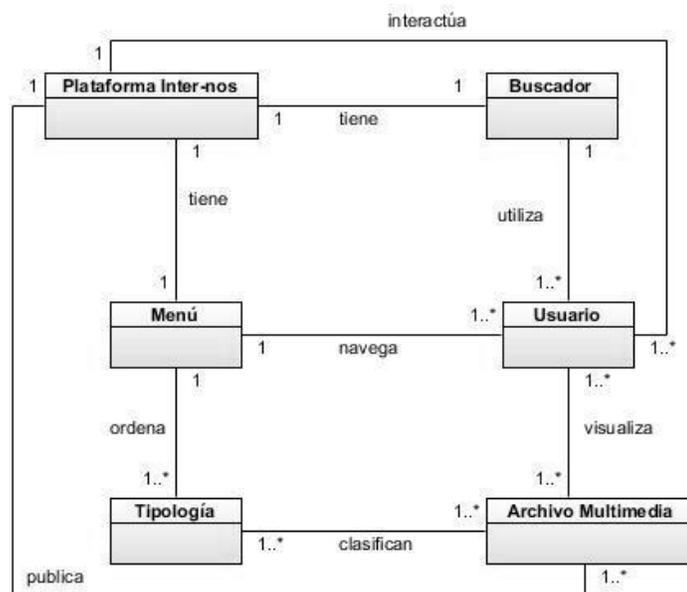


Figura 1: Modelo de dominio.

## 3.2. Requisitos

La principal tarea de la ingeniería de requisitos consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema, de manera que le garantice minimizar los problemas relacionados con el desarrollo de software. Estas especificaciones son denominadas requisitos, los que se encuentran divididos en dos grandes grupos: los funcionales y los no funcionales. (Falgueras, 2003)

Luego de haber analizado el dominio del problema y definir puntos comunes en algunas de las soluciones existentes, resulta imprescindible caracterizar las funciones, capacidades o condiciones que el módulo de recomendación debe ser capaz de cumplir.

### **3.2.1. Requisitos funcionales**

Los requisitos funcionales enuncian la definición de los servicios que un sistema debe proveer o su comportamiento ante las diferentes entradas y situaciones que le sean presentadas. (Pressman, 1988)

#### **RF1 Analizar historial del usuario**

Descripción: El sistema debe analizar el historial de navegación del usuario, adquiriendo la información de los materiales audiovisuales que este ha visitado o reproducido con mayor frecuencia.

#### **RF2 Analizar perfil de usuario**

Descripción: El sistema debe extraer los datos almacenados en el perfil de usuario, seleccionando los de mayor importancia, para ser utilizados posteriormente en el proceso de recomendación.

#### **RF3 Generar recomendación utilizando algoritmo de filtrado basado en contenido**

Descripción: El sistema debe ser capaz de generar recomendaciones haciendo uso de la información extraída del perfil de usuario y de los datos obtenidos tras analizar el historial de navegación, las mismas contendrán los  $n$  primeros archivos multimedia preferidos por el usuario en cuestión.

#### **RF4 Extraer preferencias de usuario**

Descripción: El sistema debe obtener de la base de datos un listado con todos los votos realizados por el usuario a los archivos multimedia, extrayendo de los mismos las preferencias del usuario.

#### **RF5 Agrupar los contenidos multimedia para el filtrado colaborativo**

Descripción: El sistema debe permitir al administrador crear grupos con características similares y guardarlos en la base de datos.

#### **RF6 Generar recomendación utilizando algoritmo de filtrado colaborativo**

Descripción: El sistema debe proporcionar al usuario, basándose en las preferencias del grupo al que es afín y a los archivos multimedia ya valorados por él mismo, una lista con los materiales audiovisuales que no hayan sido vistos por él todavía y que pueden ser de su agrado.

#### **RF7 Obtener listado de recomendación mediante algoritmo de hibridación.**

Descripción: Luego de adquirir los listados de recomendación de los módulos de filtrado colaborativos y filtrado basado en contenido el sistema debe obtener un listado con las recomendaciones más afines al usuario utilizando uno de los algoritmos de hibridación.

#### **RF8 Mostrar recomendación**

Descripción: El sistema debe permitir mostrar a los usuarios los archivos multimedia de su preferencia, obtenidos por el sistema de recomendación.

#### **RF9 Insertar configuraciones de algoritmo de hibridación a utilizar.**

Descripción: El sistema debe permitir al administrador seleccionar el algoritmo de recomendación a utilizar y configurarlo según sea el caso.

#### **RF10 Actualizar configuraciones de algoritmo de hibridación a utilizar.**

Descripción: El sistema debe permitir al administrador actualizar los parámetros de configuración previamente seleccionados.

#### **RF11 Insertar configuraciones de comunicación.**

Descripción: El sistema debe permitir al administrador configurar los parámetros necesarios para establecer la comunicación con la biblioteca de funciones *Recommender Lib* para obtener las recomendaciones a través de la misma.

#### **RF12 Actualizar configuraciones de comunicación.**

Descripción: El sistema debe permitir al administrador actualizar los parámetros de comunicación previamente seleccionados.

#### **RF13 Generar recomendación utilizando la biblioteca de funciones *Recommender Lib***

Descripción: El sistema debe enviar a la biblioteca de funciones las medias y los usuarios almacenados en la base de datos y el id del usuario, la biblioteca de funciones obtendrá una lista con los materiales audiovisuales que no hayan sido vistos por él y que pueden ser de su agrado.

### **3.2.2. Requisitos no funcionales**

Los requisitos no funcionales definen las propiedades y restricciones del sistema a construir o sobre los procesos que lo constituirán; son de valiosa importancia, dado que su incumplimiento puede hacer inútil al sistema. Estos están clasificados según el tipo de restricción que se quiera implementar. (Pressman, 1988)

#### **Requisitos de apariencia o interfaz externa**

Descripción: El bloque en el que se muestran las recomendaciones debe aparecer automáticamente una vez que el usuario se halla autenticado en el sistema, con una visibilidad constante para el mismo. Además será mostrado en la parte inferior de la plataforma. Los mensajes de error mostrados al administrador contendrán todas las variables que interactúen en los mismos, así como una descripción y serán mostrados en la entrada del menú administrar, sub-menú reportes.

#### **Restricciones de diseño**

Descripción: El producto de software final estará diseñado según la concepción de la arquitectura cliente-servidor en sistemas de software. Se deben respetar los estándares definidos por Drupal para la codificación del código, además de utilizar el estándar de codificación UTF-8 para la creación de tablas en la base de datos PostgreSQL.

#### **Restricciones de implementación**

Descripción: El lenguaje de programación que se debe utilizar para el desarrollo del módulo es PHP en su versión 5.3. Se debe hacer uso de la biblioteca de funciones *Recommender Lib* como una alternativa para realizar recomendaciones.

#### **Restricciones de software**

Descripción:

##### **PC clientes**

Para las PCs-clientes se recomienda un navegador Mozilla Firefox 3.5.5, pero se puede hacer uso de otro navegador que cumpla con los estándares W3C (World Wide Web Consortium).

## Servidores

Se requiere un servidor web con el módulo PHP 5 configurado y la extensión pgsql incluida. Se requiere un sistema gestor de base de datos PostgreSQL 9.1.

### 3.3. Modelo de casos de uso

Un modelo de caso de uso es una aproximación de las funciones previstas del sistema y su entorno, además sirve como un contrato entre el cliente y los desarrolladores. Los casos de uso se utilizan como hebra de unión a lo largo del desarrollo del sistema. El modelo de caso de uso es el resultado de la disciplina de requisitos y se utiliza como entrada para la disciplinas de Análisis y Diseño. (Jacobson, 2000)

#### 3.3.1. Diagrama de casos de uso

Un caso de uso representa una clase de funcionalidad dada por el sistema como un flujo de eventos. También se puede definir como una secuencia de interacciones que se desarrollan entre un sistema y sus actores en respuesta a un evento, que es iniciado por un actor principal sobre el propio sistema. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema, al mostrar cómo reaccionan a eventos que se producen en su ámbito o en él mismo. (Pressman, 1988)

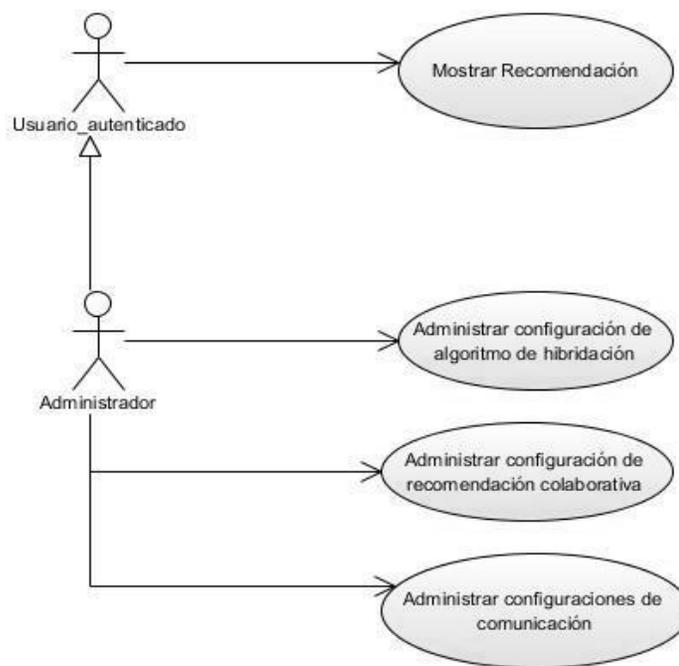


Figura 2: Diagrama de casos de uso del sistema.

En el diagrama de casos de uso del sistema (Figura 3), el caso de uso Mostrar recomendación es inicializado una vez que el usuario está autenticado en el sistema. Además, se modeló un actor administrador encargado de administrar todas las configuraciones del módulo de recomendación.

### 3.3.2. Descripción de los casos de uso

La descripción de los casos de uso permite definir una serie de especificaciones que no se pueden detallar en el modelo de casos de uso. Estas especificaciones se utilizan como información de entrada en la disciplina de análisis y diseño, lo que conlleva que mientras más se detallan los elementos que componen cada caso de uso, más factible se hará el modelado de los diagramas correspondientes al análisis de la solución.

A continuación se presenta la descripción del caso de uso más significativo del módulo de recomendación (Tabla. 1), para ver el resto dirigirse al [Anexo 1](#).

**Tabla 1: Mostrar Recomendación.**

Caso de Uso:	Mostrar Recomendación	
Actores:	Usuario_autenticado	
Resumen:	Este caso de uso se inicia una vez que el usuario se ha autenticado en la plataforma. El caso de uso termina con la generación de una recomendación con los archivos multimedia preferidos por el usuario.	
Precondiciones:	El usuario debe haberse autenticado en la plataforma.	
Poscondiciones:	Se genera un listado con los archivos multimedia preferidos por el usuario.	
Referencias	RF1,RF2,RF3,RF6,RF7,RF8,RF13	
Prioridad	Crítico	
Flujo Normal de Eventos		
Mostrar Recomendación		
Acción del Actor	Respuesta del Sistema	

1. El usuario se autentica en la plataforma	2. El sistema obtiene las configuraciones predefinidas para la recomendación híbrida. Ver descripción del CU Administrar configuración del algoritmo de hibridación.
	3. El sistema verifica el tipo de recomendación predefinida, en caso de que las configuraciones se hallan establecido mediante algoritmo de hibridación se ejecuta el siguiente flujo de eventos:
Escenario #1: Recomendación mediante algoritmo de hibridación	
Acción del Actor	Respuesta del Sistema
	1. El sistema obtiene las trazas de navegación del usuario revisando cuáles han sido los archivos multimedia más vistos y reproducidos por este.
	2. El sistema emite un informe con el id del usuario, el id de los archivos multimedia más vistos o reproducidos por el usuario, así como la cantidad de veces que los ha reproducido o visualizado.
	3. El sistema almacena dicho informe en el historial del usuario en la base de datos para posteriormente ser utilizado en el proceso de recomendación.
	4. El sistema analiza los datos asociados al perfil de usuario y al historial de navegación del usuario, almacenados en la base de datos y se realiza una comparación entre ellos.

	<p>5. El sistema realiza una intersección entre los datos del perfil y los datos asociados al historial de navegación del usuario y selecciona el contenido audiovisual resultante de dicha intersección, ordenándolo descendientemente según la cantidad de veces que han sido visualizados.</p>
	<p>6. El sistema selecciona los archivos multimedia pertenecientes al historial de navegación del usuario que no se encuentran en la intersección de este con el perfil. Igual que en el paso anterior estos datos son ordenados descendientemente según la cantidad de veces que han sido visualizados.</p>
	<p>7. El sistema selecciona los archivos multimedia pertenecientes al perfil del usuario que no se encuentran en la intersección de este con el historial de navegación. Igual que en el paso anterior estos datos son ordenados descendientemente según la cantidad de veces que han sido visualizados.</p>
	<p>8. 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.</p>
	<p>9. El sistema busca los grupos de contenidos multimedia se corresponde más con las preferencias del usuario.</p>

	10. El sistema obtiene un listado de todos los archivos multimedia del grupo que no han sido valorados por el usuario.
	11. El sistema selecciona los archivos multimedia que se deben recomendar al usuario a través de la hibridación del listado de recomendaciones BC y el listado de recomendaciones por FC, para ello utiliza el algoritmo de hibridación seleccionado por el administrador (hibridación conmutada, mezclada o por pesos) o el seleccionado por defecto. (hibridación por pesos).
	12. El sistema muestra en los diferentes bloques de contenidos de la plataforma (“Contenido Principal” (A), “Más Vistos” (B), “Más Votados” (C)) los materiales audiovisuales seleccionados por la recomendación.
Flujos alternos: Escenario #1	
Acción del Actor	Respuesta del Sistema
	4.1. No se cuenta con ningún dato del historial de navegación del usuario con el que el sistema pueda generar una recomendación. La recomendación se genera con la información del perfil de usuario, en caso de que este lo tenga.
	4.2. No se cuenta con ningún dato en el perfil del usuario para generar una recomendación. Se emite un informe en blanco.
	9.1. El sistema no encuentra un grupo de medias que se corresponda con las preferencias del usuario. Se emite un informe en blanco.

	11.1. Si los listados de recomendación BC y FC se encuentran vacíos se realiza la recomendación a partir de los archivos multimedia más vistos por la comunidad.
<b>Escenario #2: Recomendación mediante biblioteca de funciones <i>Recommender Lib</i></b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema analiza las configuraciones de comunicación almacenadas en la base de datos, permitiendo establecer con las mismas la comunicación con la biblioteca de funciones. Ver CU Administrar configuraciones de comunicación.
	2. El sistema obtiene de la base de datos un listado con todos los datos asociados al perfil del usuario.
	3. El sistema obtiene un informe desde la base de datos con el id del usuario, el id de los archivos multimedia más vistos o reproducidos por el usuario, así como la cantidad de veces que los ha reproducido o visualizado.
	4. El sistema obtiene un listado con el id de los archivos multimedia almacenados en la base de datos.
	5. El sistema se conecta con la biblioteca de funciones y envía los datos del perfil de usuario, el informe del historial y el listado de los archivos multimedia, con estos datos la biblioteca de funciones genera una recomendación.

	6. El sistema muestra en los diferentes bloques de contenidos de la plataforma (“Contenido Principal” (A), “Más Vistos” (B), “Más Votados” (C)) los materiales audiovisuales seleccionados por la recomendación.
Flujos alternos: Escenario #2	
Acción del Actor	Respuesta del Sistema
	5.1. El sistema no se puede obtener una recomendación de la biblioteca de funciones. Se emite un informe de error y se almacena en la base de datos. Se realiza la recomendación a partir de los archivos multimedia más vistos por la comunidad.
Prototipo de Interfaz para ambos escenarios	
<u><a href="#">Ver Anexo 3.</a></u>	

### 3.4. Modelo de análisis

El modelo de análisis, es la primera representación técnica de un sistema y establece una base para la creación del diseño del software. Estructura los requisitos de un modo que facilita su comprensión, preparación, modificación y en general, su mantenimiento en un formato que se corresponda con el área de conocimiento de los diseñadores del software. Este constituye una abstracción del modelo de diseño y está conformado por las clases interfaz, clases controladoras y clases entidad. (Jacobson, 2000)

A continuación se presenta el diagrama de clases del análisis del caso de uso más significativo del módulo de recomendación (Figs. 4 y 5), para ver el resto dirigirse al [Anexo 6.](#)

### 3.4.1. Diagramas de clases del análisis.

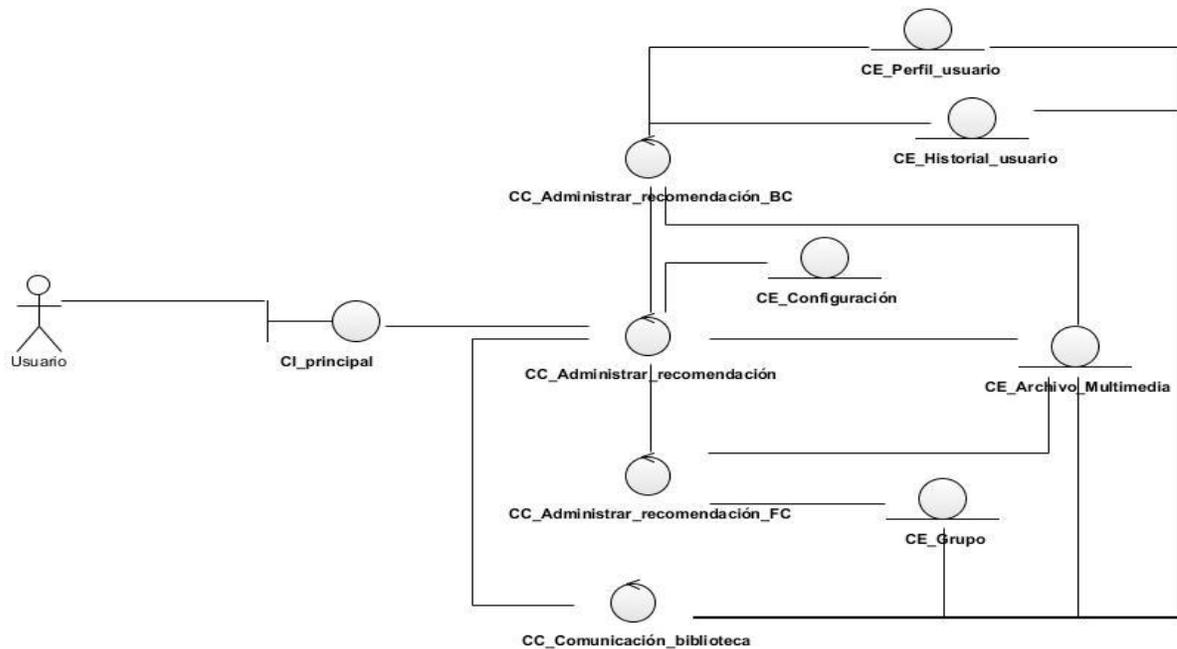
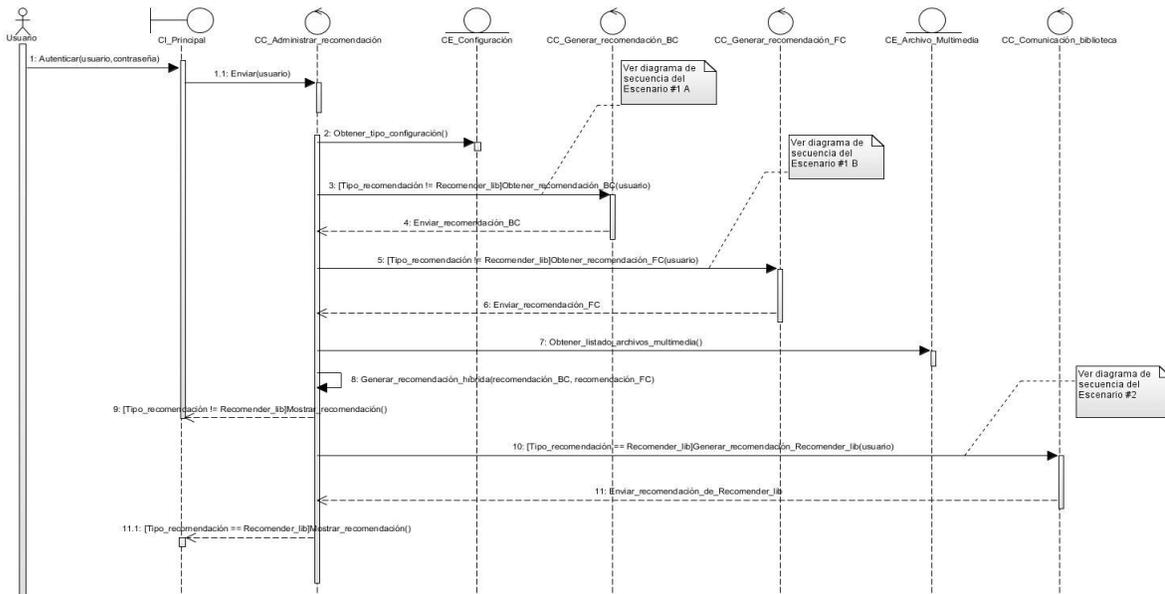


Figura 3: Diagrama de clases del análisis del caso de uso Mostrar Recomendación.

### 3.4.2. Diagramas de secuencia

Los diagramas de secuencia muestran una interacción, que representa la secuencia de mensajes entre las instancias de clases, componentes, subsistemas o actores. (Jacobson, 2000) A continuación se presenta el diagrama de secuencias del caso de uso más significativo del módulo de recomendación (Figs. 6 y 7), para ver el resto dirigirse al [Anexo 8](#). El diagrama de secuencia del caso de uso Mostrar Recomendación ha quedado separado por escenarios por la magnitud del flujo de eventos, para ver los demás escenario debe dirigirse al [Anexo 7](#).



**Figura 4: Diagrama de secuencia del caso de uso Mostrar Recomendación.**

### 3.5. Estilos arquitectónicos

Los estilos arquitectónicos son una familia de sistemas de software en términos de su organización estructural. Expresan componentes y las relaciones entre estos, con las restricciones de su aplicación y composición, así como las reglas para su construcción. Se considera como un tipo particular de estructura fundamental para un sistema de software, conjuntamente con un método asociado que especifica cómo construirlo. Estos estilos incluyen información acerca de cuándo usar la arquitectura que describe, sus invariantes y especializaciones, así como las consecuencias de su aplicación. (Camacho, y otros, 2004)

Para el desarrollo del módulo se hace uso de la familia de estilos de llamada y retorno, estos enfatizan en la escalabilidad. Son los estilos más generalizados en sistemas a gran escala. Incluye una estructura implícita de subsistemas y está dividido en varios sub-estilos, como son: programa principal o subrutina, los sistemas basados en llamadas a procedimientos remotos, los sistemas orientados a objeto y los sistemas jerárquicos en capas. (Kicillof, y otros, 2004)

La estructura principal del módulo de recomendación está basada en el sub-estilo de sistemas jerárquicos en capas, también denominados arquitecturas estratificadas. Su funcionamiento se basa en diferentes capas, donde cada una realiza operaciones específicas. En la capa externa, los componentes sirven a las operaciones de interfaz de usuario. En la capa interna, los componentes realizan operaciones de interfaz

del sistema. Las capas intermedias proporcionan servicios de utilidad y funciones de software de aplicaciones. (Kicillof, y otros, 2004), esto permite un alto nivel estructural para el módulo, así como asignación de funciones específicas en cada capa.

En un plano más abarcador, donde el módulo de recomendación interactúa con la biblioteca de funciones *Recommender Lib* mediante interfaces de comunicación, cuyas funcionalidades intervienen en el proceso de recomendación, se hace uso del sub-estilo arquitectónico programa principal o subrutina. El funcionamiento del módulo parte de la descomposición de funcionalidades en una jerarquía de control, donde un programa principal llama a un número de componentes del programa, que estos a su vez pueden llamar a otros componentes. (Kicillof, y otros, 2004) La definición de esta estructura admite un mayor nivel de desagregación de los módulos utilizados, permitiendo la expansión del sistema con pequeñas modificaciones al módulo de recomendación principal y sin afectar los restantes módulos que lo componen.

### 3.5.1. Patrones arquitectónicos

Los patrones arquitectónicos o patrones de arquitectura ofrecen soluciones a problemas de arquitectura de software. Describen los elementos y sus relaciones, junto a un conjunto de restricciones sobre cómo pueden ser utilizados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor. (Venete, 2011)

Para el desarrollo de la aplicación se decide hacer uso del patrón arquitectónico Modelo – Vista – Controlador (MVC). Patrón de arquitectura de software encargado de separar el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario, en tres clases diferentes. (Venete, 2011)

- **Modelo:** El modelo administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones para cambiar el estado (habitualmente desde el controlador).
- **Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo, cada una teniendo asociado un componente controlador.

- **Controlador:** Es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario, solicitando los datos al modelo y entregándolos a la vista, para que esta a su vez los presente al usuario.

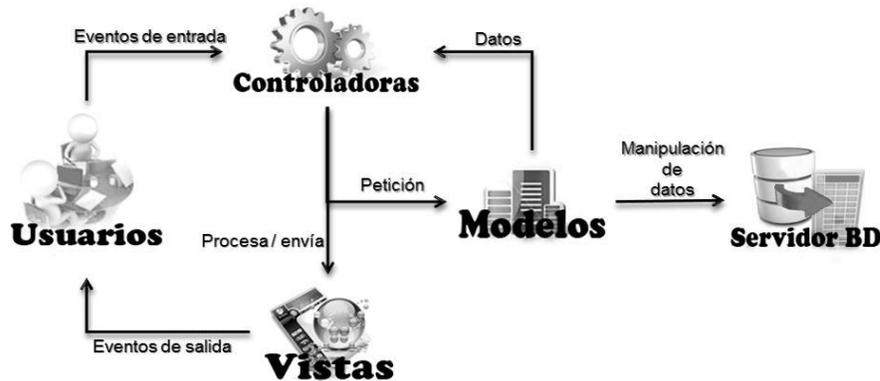


Figura 5: Funcionamiento del patrón arquitectónico MVC.

### 3.6. Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Estos patrones brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. (Tedeschi, 2009)

Los patrones de diseño se encuentran divididos en dos grupos, los *General Responsibility Assignment Software Patterns* o Patrones Generales de Asignación de Responsabilidad de Software (GRASP) que establecen los principios fundamentales de la asignación de responsabilidades a objetos y los llamados *Gang of Four* o Banda de los Cuatro (GOF) que se clasifican en tres grupos (estructurales, creacionales y de comportamiento). Los patrones estructurales tratan la combinación de clases, su relación y la formación de estructuras de alta complejidad. Por otra parte los creacionales tratan la creación de instancias, mientras que los de comportamientos tratan la interacción y la cooperación entre clases. (Tedeschi, 2009)

#### 3.6.1. Patrones GOF

- **Fachada (Facade):** Patrón estructural a nivel de objetos. Su propósito es proporcionar una interfaz unificada de alto nivel que, representando a todo un subsistema, facilite su uso. La fachada

satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas. (Permuy, 2006) El uso de este patrón se evidencia en la interfaz de comunicación con que cuenta el módulo comunicacion\_rl con la biblioteca de funciones *Recommender Lib*. Esta interfaz es la clase comunicacion\_vista.php, la que permiten acceder a las funcionalidades de la biblioteca de funciones.

- **Observador (Observer):** Este patrón define una dependencia del tipo uno a muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes. (Permuy, 2006) Este patrón se utiliza en la realización de la recomendación al usuario. Una vez que el usuario es autenticado en el sistema, la función recomendacion\_hibrida(), en dependencia del historial de navegación del usuario, los datos almacenados en su perfil y las valoraciones hechas a archivos multimedia, realiza una recomendación. Dependiendo de este cambio de estado es que se actualizan los bloques de visualización de la página principal con los contenidos multimedia de la nueva recomendación.
- **Puente (Bridge):** Es utilizado para desacoplar una abstracción de su implementación, de manera que ambas puedan ser modificadas independientemente sin necesidad de alterar por ello la otra. Tanto la abstracción como su implementación deben ser extensibles por subclases. (Tedeschi, 2009) Este patrón se evidencia en la creación de los módulos que conforman el Sistema de Recomendación (recomendacion\_hibrida, recomendacion\_bc, recomendacion\_fc y comunicacion\_rl), cada uno de ellos puede ser modificado independientemente sin que las funcionalidades de los otros se vean afectadas.
- **Acción (Command):** Patrón de comportamiento a nivel de objetos, su propósito es encapsular en un objeto la acción que satisface una petición, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma. (Tedeschi, 2009) La utilización de este patrón se puede apreciar en cada uno de los módulos creados, cada uno implementan los ganchos (funciones que le indican al núcleo de Drupal qué operación va a realizar y sobre qué módulo la hará) necesarios para su funcionamiento.

### 3.6.2. Patrones GRASP

- **Experto:** Este patrón tiene como objetivo principal asignar una responsabilidad al experto en información. (Permuy, 2006) Se evidencia el uso de este patrón en los módulos que componen el sistema de recomendación, implementados bajo el patrón arquitectónico MVC (Modelo-Vista-

Controlador). En este modelo a cada clase se le asigna una tarea determinada según la responsabilidad que esta posee en el funcionamiento del módulo, ya sea, mostrando la información correspondiente, accediendo a la base de datos del sistema o gestionando las operaciones a realizar.

- **Creador:** Este patrón tiene como objetivo principal asignar a las clases responsabilidad de crear una instancia de otra, soportando mayor claridad en el código y posibilitando el encapsulamiento y la reusabilidad. (Tedeschi, 2009) Un claro ejemplo de la utilización de este patrón se refleja en el módulo `recomendacion_hibrida` en la clase `recomendacion_hibrida_module`, que tiene la responsabilidad de crear instancias de la clase `recomendacion_hibrida_vista` para poder acceder a la información que se necesita para realizar la recomendación.
- **Bajo acoplamiento:** Es un patrón que tiene como principal objetivo asignar una responsabilidad para mantener el bajo acoplamiento, o sea, mantener las clases lo menos ligadas posibles. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios y también más reutilizables, acrecentando la oportunidad de una mayor productividad. (Tedeschi, 2009) El uso de este patrón se encuentra reflejado en todos los módulos que componen el sistema de recomendación, debido a que el nivel de dependencias entre las clases es muy bajo, debido a que en cada clase implementada por cada módulo se crea una instancia sólo de las clases que se necesite. De manera tal que cuando se produzca alguna modificación en algunas de ellas la repercusión en las otras sea mínima.
- **Controlador:** es un patrón que sirve como intermediario entre una interfaz y el algoritmo que la implementa, de tal forma es la que recibe los datos del usuario y los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, lo que permite aumentar la reutilización de código y a la vez tener un mayor control. (Permuy, 2006) Este patrón es evidenciado en la clase `recomendacion_hibrida_controladora` en la cual se encuentran los métodos que se encargan de la lógica del negocio y esta a su vez, pide información a `recomendacion_hibrida_modelo` y envía datos a `recomendacion_hibrida_vista`.

### 3.7. Modelo de diseño

El diseño de software desarrolla un modelo de instrumentación o implantación basado en los modelos conceptuales desarrollados durante el análisis del sistema. Estos modelos implican diseñar la decisión

sobre la distribución de datos y procesos. Debe proporcionar una idea completa de lo que es el software. (Silberschatz, y otros, 2007)

### 3.7.1. Diagrama de clases del diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases del diseño crean el diseño conceptual de la información que se manejará en el sistema, así como sus relaciones, permitiendo lograr una mejor interpretación en el momento de la implementación. (Silberschatz, y otros, 2007)

A continuación se presenta el diagrama de clases del diseño general del módulo de recomendación (Fig. 13).

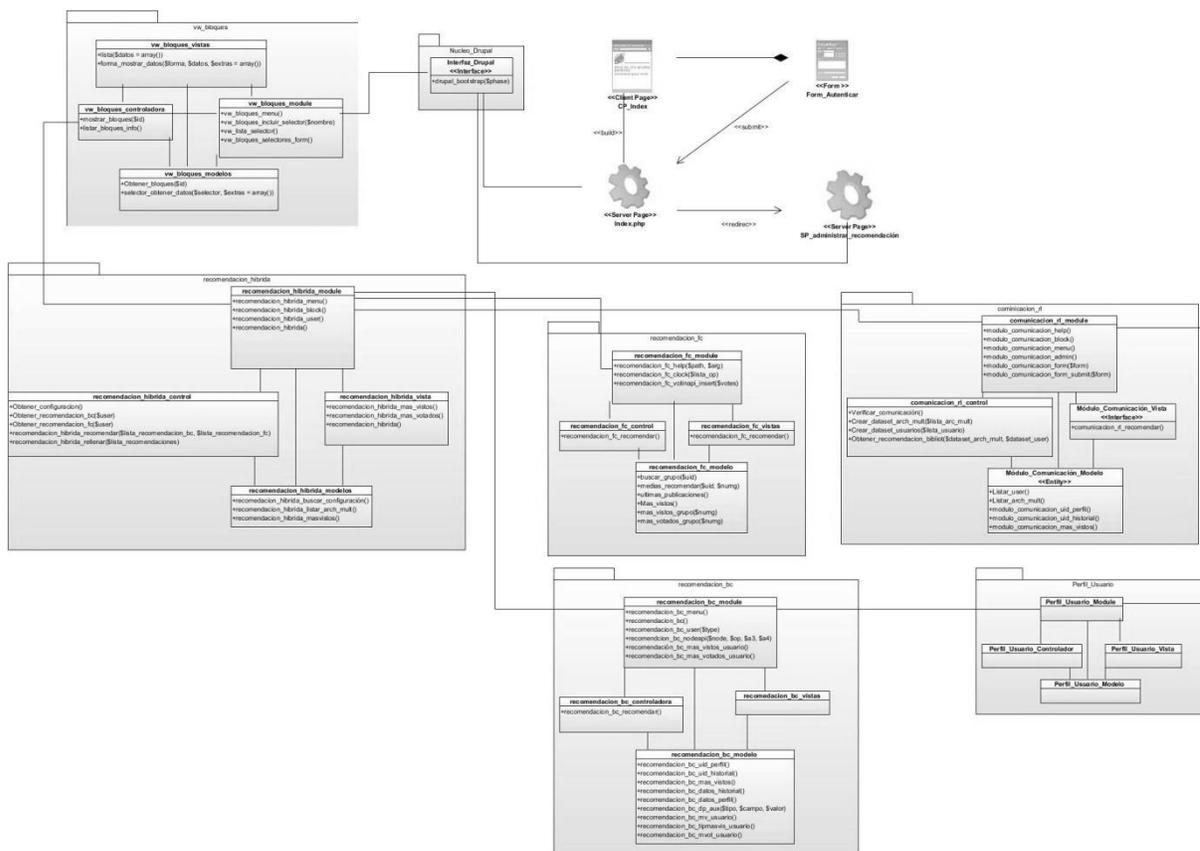


Figura 6: Diagrama de clases de diseño del CU Mostrar Recomendación.

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

**recomendacion\_hibrida\_module:** Es una interfaz entre el núcleo de Drupal y el módulo `recomendacion_hibrida`. La misma implementa una serie de funciones (*hooks*) predefinidos por Drupal para ser utilizadas durante una petición. Ejemplo de estas: son los *hooks* para crear formularios y para definir los permisos que permiten el acceso de los usuarios al módulo.

**recomendacion\_hibrida\_vista:** Su objetivo es construir y validar formularios para interactuar con el usuario.

**recomendacion\_hibrida\_controlador:** Contiene una serie de funciones encargadas de procesar información referente a la generación de las recomendaciones, recibe las acciones generadas por el usuario mediante su interacción con la vista y gestiona los eventos desencadenados. Además accede a los modelos para actualizarlos según la funcionalidad solicitada por el usuario.

**recomendacion\_hibrida\_modelo:** Contiene las funciones de acceso a los datos almacenados físicamente en las tablas de la base de datos.

### **3.8. Conclusiones parciales**

Con el modelado del dominio se concretó que los conceptos de archivos multimedia, usuario, tipografía y plataforma Inter-nos 2.0 intervinieron directamente en el proceso de recomendación. Se seleccionaron los casos de uso arquitectónicamente significativos como el de mostrar recomendación y administrar configuraciones de recomendación colaborativa; además, se realizaron los diagramas correspondientes al análisis y diseño de los mismos, registrando la trazabilidad del proceso de desarrollo de software a partir de su descripción. De esta fase se obtuvo, un diagrama de diseño conformado por tres módulos que se encargarán de cada técnica de recomendación y un módulo principal que se encargará de controlar dichas funcionalidades. Cada uno de estos módulos responde al estilo arquitectónico modelo-vista-controlador y las clases que los conforman, mantienen las relaciones y las responsabilidades definidas por los patrones de diseño seleccionados. Se modeló la comunicación con la biblioteca de funciones haciendo uso del patrón GOF: fachada, además se definió utilizar el patrón GOF: observador para la actualización del bloque de recomendaciones en la interfaz principal de la plataforma. La definición del modelo de diseño, utilizando patrones arquitectónicos y de diseño, permitió obtener una representación basada en soluciones ya probadas y con un esquema de relaciones que se ajusta a las necesidades del dominio del problema.

## **CAPÍTULO 4: Implementación y prueba**

En el presente capítulo se abordarán aspectos referentes a las disciplinas de implementación y prueba definidas por la metodología de software RUP. Se hace referencia a las tablas que interactúan en el proceso de la recomendación, mostrando su distribución en el modelo de datos. Además, se realiza el diagrama de componentes, permitiendo tener una visión más amplia de la organización y dependencias lógicas de los componentes por los que está compuesto el módulo. Conjuntamente se modela la distribución física de las funcionalidades en sus respectivos nodos, representados en el diagrama de despliegue. Finalmente, se verifica el correcto funcionamiento del módulo desarrollado aplicando las pruebas pertinentes para ello.

### **4.1. Modelo de datos**

Para dar solución a los requisitos funcionales definidos para el módulo de recomendación, se agregaron varias tablas a la base de datos de la plataforma Inter-nos 2.0. A continuación se muestra el modelo de datos definido, el mismo estará compuesto por las tablas creadas y sus relaciones, así como las relaciones de estas con otras tablas ya existentes en la base de datos que intervienen directamente en el proceso de recomendación.

#### **Listado de tablas:**

- abp\_perfil\_usuario
- grupo
- grupo\_usuario
- configuracion
- abp\_historial\_usuario
- películas
- series
- documentales
- videos\_musicales
- docentes
- archivo\_multimedia
- publicación\_am

- tipologías\_am
- node
- node\_counter
- votingapi\_cache
- votingapi\_vote

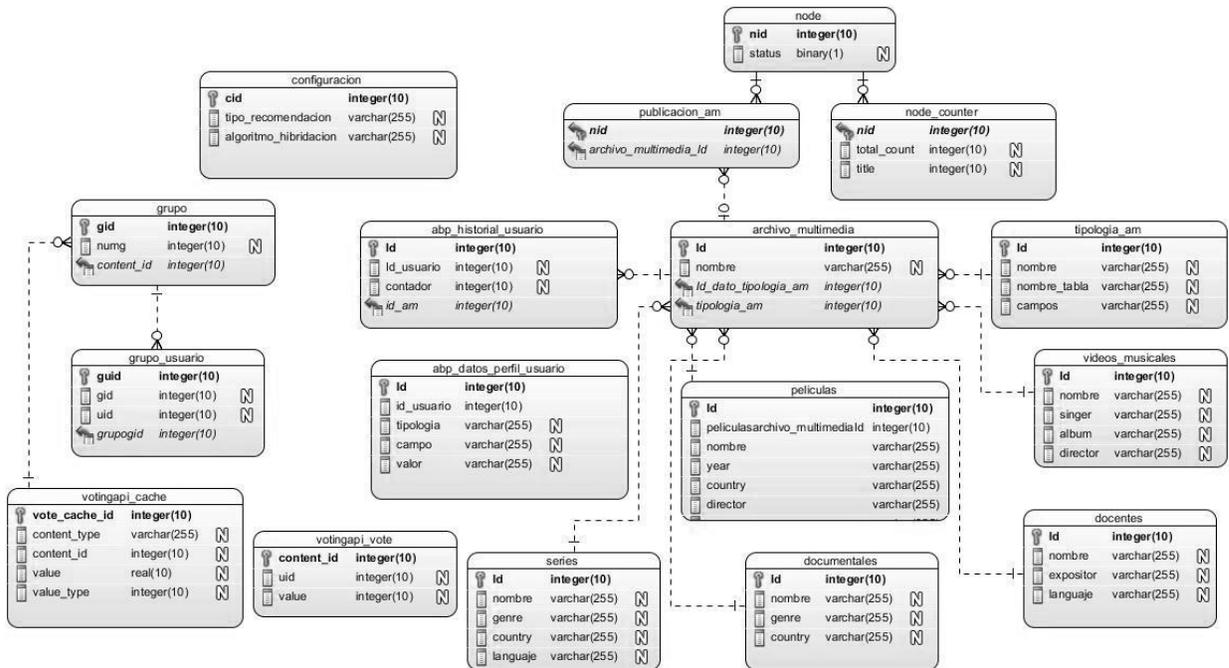


Figura 7: Modelo de datos.

A continuación se describen las tablas generadas por los módulos de recomendación implementados. Estas tablas utilizan información almacenada en algunas de las tablas representadas en el modelo de datos, que almacenan información sobre los archivos multimedia.

Nombre: abp_historial_usuario		
Descripción: Esta tabla almacena la información referente al usuario y las preferencias de este sobre los materiales audiovisuales existentes en el sistema.		
Atributo	Tipo	Descripción
Id	integer(10)	Valor numérico auto incremental que identifica cada uno de los campos de esta tabla.

id_usuario	integer(10)	Valor numérico que identifica al usuario.
tipología	varchar(255)	Clasificación del archivo multimedia seleccionado (series, películas, videos musicales, docentes o documentales).
campo	varchar(255)	Esta columna guarda el nombre del campo de los descriptores seleccionados por el usuario del archivo multimedia, por ejemplo, de la tipología películas se pueden seleccionar los campos (director, año o país).
valor	varchar(255)	Valor textual del campo seleccionado, por ejemplo, en caso de seleccionarse el campo “año” correspondiente a la tipología “películas”, valor tendría como dato el año en que se filmó la película.

**Tabla 2: Historial de navegación del usuario.**

Nombre: grupo		
Descripción: Esta tabla almacena la información referente a los grupos de medias creados para ser utilizados en el algoritmo de filtrado colaborativo.		
Atributo	Tipo	Descripción
gid	integer(10)	Valor numérico auto incremental que identifica cada uno de los campos de esta tabla.
numg	integer(10)	Valor numérico que identifica el número del grupo para dicho conjunto de medias.
content_id	Integer(10)	Valor numérico que identifica la media perteneciente al grupo.

**Tabla 3: Información sobre los grupos de medias.**

Nombre: grupo_usuario		
Descripción: Esta tabla almacena la información referente a la relación de muchos a muchos entre la tabla grupos y la tabla usuario.		
Atributo	Tipo	Descripción
guid	integer(10)	Valor numérico auto incremental que identifica cada uno de los campos de esta tabla.
gid	integer(10)	Valor numérico que identifica el número del grupo.

uid	Integer(10)	Valor numérico que identifica a los usuarios que han votado por las medias que compone dicho grupo.
-----	-------------	---

**Tabla 4: Información asociada a los grupos y los usuarios.**

Nombre: configuracion		
Descripción: Esta tabla almacena la información referente a las configuraciones del módulo de recomendación.		
Atributo	Tipo	Descripción
cld	integer(10)	Valor numérico auto incremental que identifica cada uno de los campos de esta tabla.
tipo_recomendacion	varchar(255)	Especifica si las recomendaciones se realizarán por el algoritmo de hibridación o la biblioteca de funciones.
algoritmo_hibridacion	varchar(255)	Especifica el algoritmo de hibridación a utilizar para realizar las recomendaciones.

**Tabla 5: Información asociada a las configuraciones del módulo de recomendación.**

## 4.2. Modelo de implementación

El modelo de implementación muestra la distribución de las clases del diseño en componentes, estos componentes están definidos en ficheros de código fuente, ejecutables y librerías. El modelo de implementación describe también la organización de los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación, además de sus relaciones con otros componentes. (Pressman, 2007)

### 4.2.1. Diagrama de componentes

Los componentes representan los tipos de elementos de software que entran en la elaboración de aplicaciones informáticas, y las relaciones de dependencia se utilizan para indicar que un componente utiliza los servicios ofrecidos por otros componentes. (Pressman, 2007) A continuación se muestra el diagrama de componente del módulo de recomendación desarrollado.

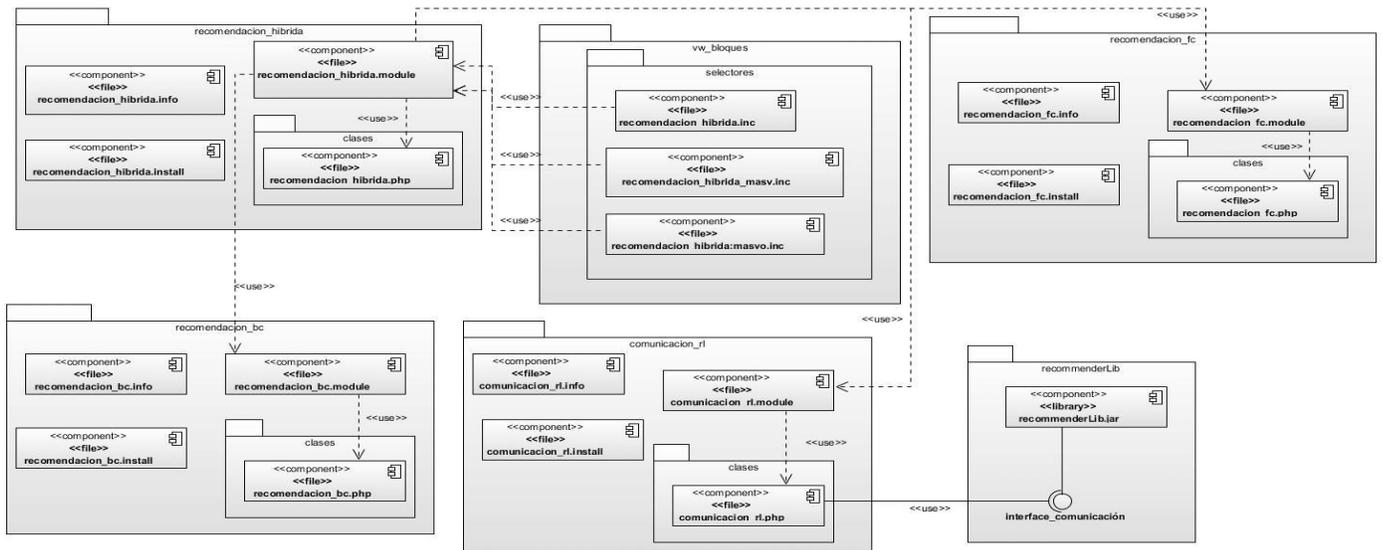


Figura 8: Diagrama de componentes.

El diagrama de componentes que se muestra contiene los siguientes ficheros:

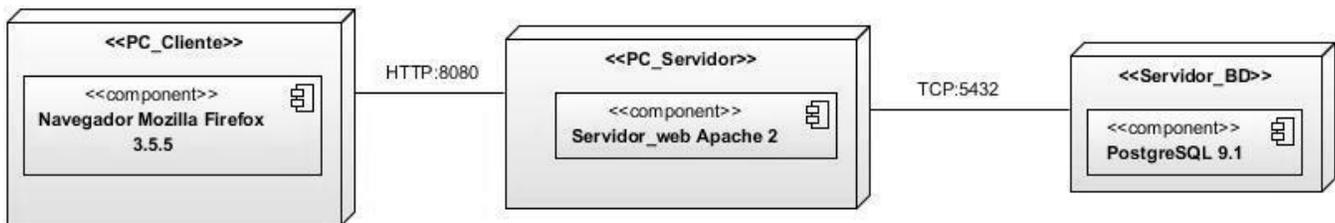
- **.module**: fichero que contiene funciones ganchos (*hooks*) predefinidos por Drupal para ser llamadas predeterminadas durante una petición.
- **.info**: fichero que define los metadatos del módulo.
- **.install**: fichero que codifica las operaciones necesarias para el funcionamiento inicial del módulo como la creación del modelo de datos.
- **.php**: fichero en el que se encuentran las clases controladoras, vistas y modelas de cada módulo desarrollado.
- **.inc**: fichero que incluye los selectores encargados de la información que se le mostrará al usuario.

### 4.3. Modelo de despliegue

El modelo de despliegue describe la distribución física del sistema, así como la distribución de las funcionalidades del módulo de recomendación en cada nodo de cómputo. Cada nodo representa un recurso de cómputo y poseen relaciones que representan medios de comunicación entre ellos. (Pressman, 2007)

A continuación se muestra el modelo de despliegue del módulo de recomendación, el cual está conformado por una pc cliente para la visualización de las recomendaciones por el usuario. La pc cliente

se comunica con el servidor web mediante el protocolo HTTP (*Hypertext Transfer Protocol*). A su vez este nodo se comunica con el servidor de base de datos que es el encargado de manejar toda la información referente al usuario y los archivos multimedia existentes en la plataforma Inter-nos 2.0. Esta relación se realiza mediante el protocolo de comunicación TCP por el puerto 5432.



**Figura 9: Diagrama de despliegue**

#### **4.4. Pruebas del módulo**

Las pruebas de software son el proceso encargado de verificar que el software cumpla criterios específicos de calidad, corrección, completión, exactitud, eficacia y eficiencia, además permite validar su buen funcionamiento y la satisfacción del cliente final. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad.

Las pruebas de unidad están dirigidas hacia los componentes testeables más pequeños del software. Es aplicable a los componentes representados en el modelo de implementación para verificar que los flujos de control y de datos estén cubiertos, además de que funcionen como se espera. También se encargan de comprobar el correcto funcionamiento de un módulo de código, lo que permite verificar que cada uno de los módulos funcione correctamente por separado. Estas pruebas pueden orientarse tanto a los métodos de caja blanca, como de caja negra. (Pressman, 2007)

**Pruebas de Caja Blanca:** las pruebas de caja blanca se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Su procedimiento se basa en escoger distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y verificar que se devuelvan los valores de salida adecuados. (Glenford, 2004)

**Prueba de Caja Negra:** las pruebas de caja negra se enfocan en las entradas que recibe el sistema y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. Dichas pruebas son llevadas a cabo sobre la interfaz del software, es decir, de cada una de sus funcionalidades,

proporcionando un juego de datos y analizando las salidas para ver si concuerdan con las esperadas. (Steven, 2001)

Luego de un análisis de la estructura del módulo de recomendación, se considera factible la realización de pruebas de caja blanca al caso de uso mostrar recomendación y pruebas de caja negra a los restantes casos de uso. Dicha selección está definida porque el caso de uso mostrar recomendación cuenta con dos algoritmos encargados de generar las recomendaciones y sus resultados son hibridados por un tercer algoritmo, la integración de estos tres algoritmos debe ser testeada para verificar cada uno de sus flujos de ejecución, además de que los restantes casos de uso del módulo cuentan con una interfaz que permite la entrada de un juego de datos de prueba, lo que facilita la comprobación de dichas funcionalidades mediante pruebas de caja negra.

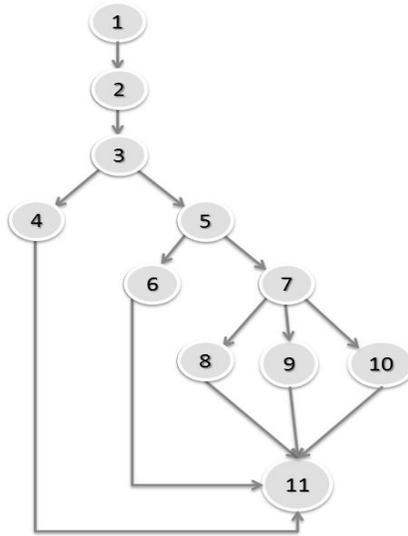
#### **4.4.1. Pruebas de caja blanca**

Las pruebas de caja blanca son aplicables a varios niveles como unidad, integración y sistema. Habitualmente se aplican a las unidades de software. Su cometido es comprobar los flujos de ejecución dentro de cada unidad (función, clase, módulo) pero también pueden testear los flujos entre unidades durante la integración, e incluso entre subsistemas, durante las pruebas de sistema. (Dr.Rodríguez, 2012)

Al emplear los métodos de prueba de caja blanca, el ingeniero del software podrá derivar casos de prueba que garanticen que todas las rutas independientes dentro del módulo se han ejercitado por lo menos una vez. Con la aplicación de esta prueba también se puede verificar que se ejecuten todos los bucles dentro de sus límites operacionales y las estructuras de datos internos para asegurar su validez. En la aplicación de esta prueba se utilizó la técnica de la ruta básica (Dr.Rodríguez, 2012)

Código fuente de `recomendacion_hibrida_recomendar()`: ver **Anexo 9**

**Paso número 1:**



**Paso número 2:**

Complejidad ciclomática:  $\text{Aristas} - \text{Nodos} + 2 = 14 - 11 + 2 = 5$

**Paso número 3:**

Caminos independientes:

$C|1 = 1 - 2 - 3 - 4 - 11$

$C|2 = 1 - 2 - 3 - 5 - 6 - 11$

$C|3 = 1 - 2 - 3 - 5 - 7 - 8 - 11$

$C|4 = 1 - 2 - 3 - 5 - 7 - 9 - 11$

$C|5 = 1 - 2 - 3 - 5 - 7 - 10 - 11$

**Paso número 4:**

Caso de prueba para:  $C|1 = 1 - 2 - 3 - 4 - 11$

Entrada: void (\$uid (usuario no autenticado))

Resultado esperado: Generar recomendación a partir de más vistos por la comunidad.

Resultado de la prueba: Satisfactorio.

Caso de prueba para: C|5 = 1 – 2 – 3 – 5 – 7 – 10 – 11

Entrada: void (\$uid (usuario autenticado)), (\$conf['tipo\_recomendacion']==1'(Tipo de recomendación definida por recomendación híbrida)), (\$hib['algoritmo']== '3'(Tipo de algoritmo de hibridación definido por mezcla))

Resultado esperado: Generar recomendación a partir de la mezcla de las recomendaciones por las preferencias de los usuarios, su historial y de las recomendaciones de las medias votadas por estos.

Resultado de la prueba: Satisfactorio.

#### **4.4.2. Pruebas de caja negra**

Las pruebas de caja negra son pruebas funcionales que parten de los requisitos funcionales para diseñar pruebas que se aplican sobre el sistema, sin necesidad de conocer como está construido por dentro. Se emplean un determinado conjunto de datos de entrada y se observan las salidas que se producen, para determinar si la función se está desempeñando correctamente en el sistema bajo prueba. Las herramientas básicas son observar la funcionalidad y contrastar con la especificación. (Víctor, 2012)

Para la realización de dichas pruebas se utilizó la técnica de partición equivalente, esta técnica divide el campo de entrada de un sistema en clases de datos que permitan derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores, reduciendo así el número de casos de prueba a desarrollar para detectar un error genérico. (Víctor, 2012)

#### **Caso de uso:**

Administrar configuración de recomendación colaborativa

#### **Descripción General:**

Este caso de uso se inicia una vez que el administrador selecciona configurar algoritmo de filtrado colaborativo en el panel de configuración. El caso de uso se termina cuando el sistema guarda las configuraciones en la base de datos, así como haber agrupado las medias.

#### **Condiciones de Ejecución:**

El usuario debe haberse autenticado en la plataforma como administrador y debe haber seleccionado la opción de configurar algoritmo colaborativo.

#### **Secciones a probar en el Caso de Uso:**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Configurar algoritmo de filtrado colaborativo	EC1.1: Configurar algoritmo de filtrado colaborativo con valores numéricos dentro del rango establecido.	El sistema muestra una ventana con el campo para entrar el número de grupos a crear. El administrador inserta el número de grupos y selecciona el botón guardar. El sistema valida el número de grupos y agrupa las medias almacenadas en la base de datos según los grupos definidos y las valoraciones de los usuarios.	Inicio Inter-nos2/ admin/settings/recomendacion_fc
	EC1.2: Configurar algoritmo de filtrado colaborativo con valores numéricos fuera del rango establecido.	El sistema muestra una ventana con el campo para entrar el número de grupos a crear. El administrador inserta el número de grupos y selecciona el botón guardar. El sistema valida el número de grupos y muestra el mensaje de error definido: "Número no permitido de grupos".	Inicio Inter-nos2/ admin/settings/recomendacion_fc

	EC1.3: Configurar algoritmo de filtrado colaborativo con valores alfanuméricos.	El sistema muestra una ventana con el campo para entrar el número de grupos a crear. El administrador inserta el número de grupos y selecciona el botón guardar. El sistema valida el número de grupos y muestra un mensaje de error: "Sólo valores numéricos".	Inicio Inter-nos2/ admin/settings/recomendacion_fc
--	---	---	---

**Tabla 6: Diseño de caso de prueba de caja negra para el caso de uso Administrar configuración de recomendación colaborativa.**

**Descripción de las variables:**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	grupos	textfield	no	Campo para entrar la cantidad de grupos para agrupar las medias.

**Tabla 7: Descripción de las variables de los casos de uso Administrar configuración de recomendación colaborativa.**

**Matriz de datos:**

Id del escenario	Escenario	Grupo	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Configurar algoritmo de filtrado colaborativo	( 3 ) V	El sistema muestra una ventana con el campo para entrar el número de grupos a crear. El administrador inserta el número de grupos y selecciona el botón	Satisfactorio

	con valores numéricos dentro del rango establecido.		guardar. El sistema valida el número de grupos y agrupa las medias almacenadas en la base de datos según los grupos definidos y las valoraciones de los usuarios. El sistema muestra un mensaje de confirmación: "se han creado satisfactoriamente los grupos de medias".	
EC 1.2	Configurar algoritmo de filtrado colaborativo con valores numéricos fuera del rango establecido.	( 1 ) I	El sistema muestra una ventana con el campo para entrar el número de grupos a crear. El administrador inserta el número de grupos y selecciona el botón guardar. El sistema valida el número de grupos y muestra el mensaje de error definido: "Número no permitido de grupos".	Satisfactorio
EC 1.3	Configurar algoritmo de filtrado colaborativo con valores alfanuméricos.	(cinco) I	El sistema muestra una ventana con el campo para entrar el número de grupos a crear. El administrador inserta el número de grupos y selecciona el botón guardar. El sistema valida el número de grupos y muestra un mensaje de error: "Sólo valores numéricos".	Satisfactorio

**Tabla 8: Matriz de datos del caso de uso Administrar configuración de recomendación colaborativa.**

#### **4.5. Conclusiones parciales**

Con la realización del presente capítulo se analizó la encapsulación de las clases del diseño en los componentes de código fuente, de esta representación se obtuvo un alcance de la interacción entre estos,

así como las relaciones de interfaces de comunicación con los restantes componentes del módulo. Además, se realizó una concepción del modelo de datos, definiendo las relaciones de las tablas creadas por el módulo de recomendación con las restantes tablas de la plataforma, lo que permitió reflejar su nivel de integración. Tras la definición de los componentes, se pasó a disponer de su distribución en los nodos de cómputo definidos en el diagrama de despliegue, para así comenzar con las pruebas que permitieron la validación de las funcionalidades del módulo de recomendación y su integración con la plataforma Inter-nos 2.0. Como completamiento de las pruebas de software, se definieron los casos de prueba para aquellos escenarios en los que ocurren interacciones bidireccionales con el usuario, a través de los cuales se verificaron las respuestas del sistema para cada flujo de ejecución, obteniendo la validación de los requisitos funcionales del módulo.

## Conclusiones generales

El desarrollo del proceso investigativo, basado en el objeto de estudio definido por este trabajo de diploma, estableció la concepción de un módulo de recomendación híbrido, integrado por los métodos de recomendación basado en contenido y colaborativo, utilizando para ello diferentes técnicas de hibridación asociativas, como la conmutada, por peso y mezclada. El desarrollo de dicho módulo estuvo guiado por la metodología de software RUP, que permitió generar los artefactos que documentaron la solución, para continuar con el desarrollo futuro de nuevas versiones, si fuese necesario. Además, se emplearon para su desarrollo las tecnologías utilizadas en la creación de la plataforma Inter-nos 2.0, como forma de mantener la compatibilidad con los restantes módulos ya desarrollados. Se concluyó utilizar la biblioteca de funciones *Recommender Lib* como alternativa en el proceso de generación de recomendaciones, manteniendo esta opción configurable por el administrador. Se definieron los conceptos asociados al dominio del problema para una mejor comprensión de los procesos que se manejaban en la plataforma. Con la definición de dichos procesos se identificaron las funcionalidades del módulo de recomendación, así como las características que permitiesen su integración a Inter-nos 2.0.

Con la integración del módulo de recomendación a la plataforma Inter-nos 2.0, la interfaz principal cuenta con una nueva funcionalidad que permite presentar los contenidos publicados de acuerdo a las preferencias particulares de cada usuario, lo que le propicia un valor agregado a la misma viabilizando el proceso de búsqueda de un material. De esta forma, la plataforma se ajusta a las tendencias de personalización de las ofertas que utilizan los sitios de distribución de contenido multimedia, para mantener la fiabilidad de sus clientes.

## **Recomendaciones**

Las recomendaciones del presente trabajo de diploma están encaminadas a:

- Desarrollar otros criterios de configuración para la generación de recomendaciones definidas por los bloques de más votados, más vistos y más leídos.
- Desarrollar un módulo de recomendación que utilice métodos basado en conocimiento que aporten mayor exactitud al proceso.
- Desarrollar un modelo para la generación de explicaciones para el módulo de recomendación híbrido.

## Bibliografía

- Aroche, Stephanie Falla. 2006.** Maestros del Web. *La Historia de la Internet*. [En línea] 02 12, 2006. [Citado el: 10 26, 2012.] <http://www.maestrosdelweb.com/editorial/internethis/>.
- ASOCAE. 2008.** Cultura Educativa. *Historia de la Cinematografía*. [En línea] Obra bajo licencia de Creative Commons, 2008. [Citado el: 10 26, 2012.] [http://www.cultureduca.com/cine\\_indice\\_hist1.php](http://www.cultureduca.com/cine_indice_hist1.php).
- Bamshad, Mobashe y Sarabjot, Singh Anand. 2005.** Intelligent Techniques for Web Personalization. Edinburgh, Scotland, UK : 19th International joint conference on artificial intelligence(IJCAI 2005), 2005. Vol. 3, 13.
- Barranco, M.J. 2008.** REJA: un sistema de recomendación de restaurantes basado en técnicas difusas. *Acta de conferencia*. Jaén : VII Congreso “Turismo y Tecnologías de la Información y las Comunicaciones” Turitec, 2008.
- Brown, Pérez Alexander Jesús. 2012.** *Desarrollo de un Sistema de Recomendación Basado en Contenido para la Plataforma VideoWeb*. La Habana,Cuba : s.n., 2012.
- Buendía, Lola. 2010.** Nuevas formas de distribución de contenido audiovisual. *Nuevas Tecnologías by Suite101* . [En línea] 08 12, 2010. [Citado el: 10 26, 2012.] <http://www.nuevastecnologias/nuevas-formas-de-distribucion-de-contenido-audiovisual-a23117.html>.
- Busquet, Victor Codina. 2008.** Sistema de recomendación personalizada de contenido multimedia. *Revista*. Madrid : s.n., 2008.
- Camacho, Erika y Cardeso, Fabio. 2004.** *ARQUITECTURAS DE SOFTWARE*. 2004.
- Campos, Luis M y Fernández, Luna Juan M. 2011.** *Uso de conocimiento estructurado en un sistema de recomendación basado en contenido*. Andalucía : Trabajo respaldado por el Ministerio de Educación y Ciencia y la Consejería de Innovación, 2011. TIN2005-02516.
- Castells, Pablo, Díez, Fernando y Pulido, Estrella. 2010.** *Recuperación y almacenamiento de información en la web*. Madrid : Universidad Autónoma de Madrid, 2010.
- Cerda, Felipe. 2009.** *NetBeans el unico IDE que necesitas*. 2009.
- Claypool, M. 1999.** Combining content-based and collaborative filters in an online newspaper. 1999.
- Colell, Mas Whinston. 1995.** *Microeconomic Theory*. New York : Oxford University Press, 1995.

- Cordero, Infante Antonio. 2007.** *Visual Paradigm For Uml.* 2007.
- Cruz, Constanza Claudia. 2010.** Sistema de recomendación en el Comercio Electrónico y la E-educación. Bogotá (Colombia) : s.n., 2010. Vol. Vol. 8, Criterio Libre N-12. ISSN 1900-0642.
- . 2010. Sistemas de recomendación en el Comercio Electrónico y la E-educación. 2010. págs. 161-182.
- Definicion.de. 2012.** Definiciones. *Preferencia.* [En línea] 2012. [Citado el: 11 26, 2012.] <http://definicion.de/preferencias>.
- Dr.Rodriguez, Eduardo T. 2012.** *Estrategias y técnicas de prueba del software.* s.l. : 24 de octubre, 2012.
- Falgueras, Campderrich Benet. 2003.** *Ingeniería de Software.* s.l. : Editorial UOC, 2003.
- Farfán, Gunther y Arias, Jesús. 2010.** Scridb. [En línea] 2010. <http://es.scribd.com/doc/61498499/25trabajode-Scridb-Visual-Basic-2>.
- Feria, Álvarez Liset. 2012.** *Desarrollo de un sistema de recomendación colaborativo para la plataforma VideoW eb.* La Habana, Cuba : s.n., 2012.
- Fernández, Ramírez M. L. 2010.** Sistemas de recomendación. *Tesis Licenciatura. Ingeniería en Sistemas Computacionales.* [En línea] Universidad de las Américas-Puebla., Diciembre 2010. [Citado el: 11 25, 2012.] [ict.udlap.mx/people/lulu/documento/capitulo4.html](http://ict.udlap.mx/people/lulu/documento/capitulo4.html).
- Font, Seguido Miguel. 2009.** *Tesis de maestría-Sistemas de recomendación para webs de información sobre la salud.* Cataluña : Universidad Politécnica de Cataluña, 2009.
- Formantín, Pérez Yusleidys y Sierra, Peraza Indira. 2007.** *Guía para el trabajo con el CMS Drupal.* La habana, Cuba : s.n., 2007.
- Glenford, J. Myers. 2004.** *The Art of Software Testing, Second Edition.* 2004.
- Han, S.H y Min, I. 2005.** Detection of the customer time-variant pattern for improving recommender systems. *Expert Systems with Applications.* 2005. Vol. vol. 28.
- Herrera, Viedma Enrique, Porcel, Carlos y Hidalgo, Lorenzo. 2011.** Sistemas De Recomendaciones: Herramientas para el Filtrado de Información en Internet. Granada : Depto. de Ciencias de la Computación e Inteligencia Artificial, 2011. Vol. vol.1, 4.
- IWS. 2012.** Internet Word Stats. *Internet Usage Statistics.* [En línea] 06 30, 2012. [Citado el: 10 26, 2012.] <http://www.internetworldstats.com/stats.htm>.

- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000.
- Jeffrey, Parsons Paul Ralph y Gallagher, Katherine. 2011.** *Using Viewing Time to Infer User Preference in Recommender Systems*. Canada : Confidential to the authors and Memorial University until published, 2011.
- Kicillof, Nicolás y Reynoso, Carlos. 2004.** *Estilos y Patrones en la Estrategia de Arquitecturade Microsoft*. UNIVERSIDAD DE BUENOS AIRES : s.n., 2004.
- Kim, Y.S., y otros. 2005.** Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites. *Expert Systems with Applications*. 2005. Vol. vol. 28.
- Konstan, J., Terveen, J. y Riedl, Herlocker R. 2004.** Evaluating Collaborative Filtering. 2004.
- Lanzillotta, Analía. 2008..** mastermagazine.info. Definición de Leguajes de Programación. 2008.
- Larman, Craig. 1999.** *UML Y PATRONES*. México : s.n., 1999.
- León, Hernández Rolando Alfredo. 2011.** *El proceso de investigación científica*. Ciudad de La Haban : Editorial Universitaria del Ministerio de Educación Superior, 2011. ISBN 978-959-16-1307-3.
- Liu, Bing. 2011.** Web Data Mining. University of Illinois, Chicago : Data-Centric Systems and Applications, 2011. Vol. Second Edition, 2011932320. ISBN 978-3-642-19459-7.
- Lockhart, Thomas. 1996.** *El equipo de desarrollo de PostgreSQL*. 1996.
- Macias, Mervyn y De la Rosa, Freddy. 2011.** Recomendaciones con Filtrado Colaborativo Basado en Usuario y en Ítem Aplicando el Paradigma Map-Reduce. Guayaquil, Ecuador : Centro de Investigación Científica y Tecnológica, 2011.
- Mah, Paul. 2009.** Disadvantages of Web Development Using PHP. ITBusinessEdge. [En línea] 2009. <http://www.itbusinessedge.com/cm/blogs/mah/disadvantages-of-web-development-using-php/?cs=33397..>
- MCs.Saburit, Borrell Arelys. 2012.** Centro de Ayuda de la Biblioteca Virtual en Salud. *¿Qué es una recomendación?* [En línea] 2012. [Citado el: 11 26, 2012.] <http://www.bvsite.sld.cu/>.
- Minería de uso Web aplicada a registros de navegación por Internet.* **Suárez, Horacio Darian, Boda, Grass y Rosete, Alejandro. 2012.** No. 3, La habana, Cuba : Revista Cubana de Ingeniería, 2012, Vol. Vol. III. ISSN 2223 -1781.

- Morales, de Castillo J.M. 2010.** *Modelo semántico-difuso de un sistema de recomendaciones de información para bibliotecas digitales universitarias*. Granada : Dpto. Inteligencia Artificial y Ciencias de la Computación, 2010.
- Nobrega, Maria D. 2012.** Curso Sistemas de Información II. *Herramientas CASE*. [En línea] 2012. [Citado el: 01 08, 2013.] [http://curso\\_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php](http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php).
- Olivera, Lobo María Dolores. 2011.** Sistemas de recuperación de información en la Web: Nuevos servicios de búsqueda y recuperación colaborativa. *pdf*. Granada : Universidad de Granada, 2011.
- Peña, Fernández Emilio. 2007.** Tendencias en el consumo audiovisual. *Cambios en los medios tradicionales: participación y personalización*. [En línea] 03 07, 2007. [Citado el: 10 26, 2012.] <file:///D:/Documentos/01-Tesis/02-Introducci%C3%B3n/Paginas%20webs/tendencias-en-el-consumo-audiovisual%28coment%29.html>.
- Pérez, Tejada Yakelín de la Caridad. 2010.** Implementación del subsistema Gestión y Presentación de Contenidos de la Plataforma VideoWeb. *Trabajo de diploma*. La habana, Cuba : s.n., 2010.
- Permuy, Bellas Fernando. 2006.** *Introducción al Diseño con Patrones*. Universidad de A Coruña : s.n., 2006.
- Pressman, S. Roger. 1988.** *Ingeniería de Software*. s.l. : Mc Graw-Hill, 1988.
- . **2007.** *Ingeniería del Software*. 2007.
- RAE. 2012.** REAL ACADEMIA ESPAÑOLA. *recomendación*. [En línea] 2012. [Citado el: 11 25, 2012.] <http://lema.rae.es/drae/>.
- Recommender Lib. 2012.** Recommender Lib: *The recommender system engine*. [pdf] Universidad de Jaén, Italia : *Recommender Lib*, 2012.
- Ryan, A. Dave. 2009.** *SistemasdeInformacion2. Lenguaje Unificado de Modelado (UML)*. 2009.
- Salazar, J. Erika y Ortega, L. Oscar. 2006.** Sistema de Búsqueda Personalizada y Recomendación de Documentación Científica. Universidad de Antioquia, Medellín, Colombia : Revista Iberoamericana de Inteligencia Artificial, 2006. Vol. vol.1, No.30. ISSN: 1137-3601.

- Sánchez, H. O. Alejandres, Member, Student y IEEE. 2011.** *Sistemas de recomendación en ambientes organizacionales: estado del arte y tendencias futuras.* Cuernavaca Morelos, México : IX Congreso Internacional sobre Innovación y Desarrollo Tecnológico, 2011.
- Silberschatz, F. Henry y Abraham, Korth &. 2007.** *Análisis y Diseño de Sistemas.* Segunda Edicion. : Editora Mc Graw Hill, 2007.
- Silva, Pérez Andy. 2003.** *Metodologías de desarrollo de software. Su aplicación y uso.* 2003.
- SINBAD. 2012.** Personalización y los Sistemas de Recomendación. Jaén : Universidad de Jaén, 2012.
- Steven, R. Rakitin. 2001.** *Software Verification and Validation for Practitioners and Managers, Second Edition.* 2001.
- Sun, Hui-Feng, ACM y Jun-Liang, Chen. 2012.** JacUOD: A New Similarity Measurement for Collaborative Filtering. China : JacUOD, 2012. ISBN11390-012-1301-5.
- Tedeschi, Nicolás. 2009.** *Patrones de Diseño.* s.l. : TACC II, 2009.
- Utria, D. Rubén. 2007.** Globalización y Desarrollo Científico y Tecnológico: El gran reto para los países latinoamericanos. La Habana : LA HOJARASCA, 2007. Vol. I, 28.
- Venete, Adriana. 2011.** *Introducción a los Patrones de Arquitectura.* Universitat Jaume I (UJI) : s.n., 2011.
- Víctor, Gómez Adán. 2012.** *Software Testing and ALM done right.* 2012.
- VP.org. 2007.** *Visual Paradigm for the Unified Modeling Language: VP-UML 6.0 User's Guide.* 2007.