

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**FACULTAD 6**



“Sistema de recomendación basado en las preferencias  
de los usuarios para la plataforma PTARTV.”

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

Autor: Elvis Ferrera Cedeño

Tutor: Ing. Yandy León Núñez

# Declaración de Autoría

---

## DECLARACION DE AUTORIA

Declaro que soy el único autor del presente trabajo titulado: "Sistema de Recomendación basado en las preferencias de los usuarios para la plataforma PTARTV" y autorizo al Proyecto PTARTV de la Universidad de las Ciencias Informáticas hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 28 días del mes de junio del año 2011.

Autor:

---

Elvis Ferrera Cedeño

Tutor:

---

Ing. Yandy León Núñez

# Datos de Contacto

---

## DATOS DE CONTACTO

**Tutor:** Ing.Yandy León Núñez(ylnunez@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas en la UCI en el año 2008. Es Profesor Instructor y durante su trayectoria como profesor ha impartido las asignaturas Inteligencia Artificial y Programación Web. Trabajador del Centro GEYSED en el Proyecto Plataforma de Transmisión Abierta para Radio y Televisión donde se desempeña como líder de desarrollo.

# Agradecimientos

---

## AGRADECIMIENTOS

Gracias a mis padres y a mis hermanos por tanto esfuerzo, empeño y sacrificio para que yo pueda haber hecho este sueño realidad.

A mi tutor Yandy por haber tenido tanta paciencia conmigo y haber dedicado tanto empeño en la realización de la presente investigación.

A mi novia por estar siempre a mi lado, por su cariño, respeto y constancia.

A amigos como Alejandro, Sixto, Burgo, Odesa, Pulla por haberme apoyado en los momentos que más lo he necesitado.

A los compañeros del proyecto PTARTV por ser una familia más y hacer de mi una mejor persona.

A la Revolución por haberme permitido nacer en un país libre donde la educación es gratuita.

# Dedicatoria

---

## DEDICATORIA

Dedico el presente trabajo a las personas más importantes de mi vida, a mi madre, padre, hermanos y a toda mi familia en general.

Hablar de mi madre me costaría hacer una nueva tesis ya que las palabras no alcanzarían en esta breve sección pues ella me ha guiado con su fe, cariño, constancia, amor, en todas las etapas de mi vida. También ha sabido cuidar de sus hijos dando hasta la última gota de sudor para protegerlos. Sin duda alguna es madre antes que mujer.

De mi padre Romérico decir que ha sido un ejemplo, faro y guía a seguir siempre ha estado presente en los momentos que lo he necesitado, junto con mi madre me ha enseñado distinguir lo bueno y lo malo. Es un padre ejemplar con toda la de la ley porque aunque no me dio el ser ha sabido cuidar de mí desde que tenía 2 años de edad.

Mis hermanos también me han apoyado mucho, cuidando de los viejos en los momentos que he estado distante de la casa. Gracias por compartir la misma sangre, los mismos sufrimientos, las mismas alegrías, la misma familia.

# Resumen

---

## RESUMEN

En varias ocasiones las personas tienen que tomar decisiones sin tener el conocimiento previo o la experiencia necesaria sobre las opciones a elegir, por eso se apoyan en las recomendaciones que otras personas le puedan brindar. Esta falta de conocimientos y la cantidad de información existente, hace que en estos días vaya cobrando importancia el desarrollo de métodos automáticos de selección de elementos como películas, libros, páginas web, amigos, ajustándose a los gustos y preferencias de los usuarios.

Los sistemas de recomendación son utilizados en la actualidad para resolver este tipo de problemática. Estos sistemas de acuerdo a la información que manejan se clasifican en colaborativos, basados en conocimientos, basados en contenidos e híbridos. En la presente investigación se abordan los sistemas de recomendación basados en contenidos cuyo elemento principal es la función de similitud que se utiliza para inferir conocimientos. Teniendo en cuenta que la cantidad de información que tienen que manejar estos sistemas es excesivamente grande se propuso una función de similitud llamada Coeficiente de Beltin la cual permitió que disminuyera el tiempo de respuesta de las recomendaciones que recibe el usuario implicado en dicho proceso.

La aplicación de esta solución aumentará el nivel de satisfacción de los usuarios que interactúan con el subsistema Web de la PTARTV (Plataforma de Trasmisión Abierta para Radio y Televisión), recomendándoles medias como películas, documentales, música que sean de su agrado y preferencia.

## Palabras Claves

Sistema, Recomendación, Información, subsistema, PTARTV, similitud, Beltin.

# Contenido

---

## Índice de Contenido

Capítulo1: Fundamentación teórica.....	15
Introducción .....	15
1.2 Sistema de Recomendación.....	15
1.3 Sistema de recomendación basado en contenido .....	16
1.3.1 Perfil de usuario .....	17
1.3.2 Contenido de media .....	18
1.3.3 Cálculo de similitud y nueva propuesta .....	18
1.4 Soluciones existentes.....	22
1.5 Tendencias y tecnología actuales para el desarrollo de la aplicación.....	25
1.5.1 Metodología de desarrollo RUP .....	25
1.5.2 Lenguaje de modelado UML .....	29
1.5.3 Herramienta CASE (Computer Aided Software Engineering) Visual Parading.....	30
1.5.4 Lenguaje de programación PHP5.....	31
1.5.5 Sistema gestor de base de datos Postgresql.....	32
1.5.6 Framework de desarrollo Symfony .....	32
1.5.7 Extjs .....	34
1.6 Conclusiones .....	35
Capítulo2: Características del sistema .....	36
2.1 Introducción .....	36
2.2 Modelo de dominio.....	36
2.3 Principales conceptos .....	37
2.4 Requisitos funcionales .....	38
2.5 Requisitos no funcionales .....	38
2.6 Diagrama de caso de uso del sistema.....	40
2.7 Descripción de los Casos de Uso.....	41
2.8 Conclusiones .....	47
Capitulo3: Análisis y diseño del sistema.....	49
3.1 Introducción .....	49
3.2 Modelo de análisis .....	49

# Contenido

---

3.3 Framework de Symfony .....	50
3.3.1 Patrones de Diseño Symfony .....	52
3.4 Modelo de Diseño .....	56
3.4.1 Diagrama de clase del Diseño.....	56
3.6 Conclusiones .....	61
Capítulo 4: Validación de la Solución Propuesta .....	62
4.1 Introducción .....	62
4.2 Modelo de Implementación .....	62
4.3 Modelo de Despliegue.....	65
4.4 Pruebas.....	66
4.5 Resultados de las pruebas.....	72
4.6 Conclusiones .....	72
Conclusiones Generales .....	73
Recomendaciones .....	74

# Índice de Figuras y Tablas

---

## Índice de Figuras

Figura 1 El RUP fases y flujos de trabajos .....	27
Figura 2 Iteraciones en RUP .....	28
Figura 3 UML .....	30
Figura 4 Visual Paradigm y los ID con los que se integra .....	31
Figura 5 Modelo de Dominio .....	37
Figura 6 Diagrama de Caso de Uso del Sistema .....	41
Figura 7 Clases de Análisis de Caso de Uso Pedir Recomendación .....	49
Figura 8 Clases de Análisis de Caso de Uso Crear Perfil .....	49
Figura 9 Clases de Análisis de Caso de Uso Modificar Perfil .....	50
Figura 10 Patrón Modelo Vista Controlador de Symfony .....	51
Figura 11 Diagrama de Clases de Diseño para el CU Modificar Perfil .....	58
Figura 12 Diagrama Clases de Diseño para el CU Crear Perfil .....	59
Figura 13 Diagrama de Clases del Diseño para el CU Pedir Recomendación .....	60
Figura 14 Diagrama de Componentes. Vista General .....	63
Figura 15 Diagrama de Componente. Paquete Modelo .....	64
Figura 16 Diagrama de Componente. Paquete Ext. ....	64
Figura 17 Diagrama de Componente. Paquete Módulos. ....	65

## Índice de Tablas

Tabla 1 Cálculo de similitud entre dos muestras usando el Coeficiente Simple .....	20
Tabla 2 Cálculo de similitud entre dos muestras usando el Coeficiente de Beltin .....	22
Tabla 3 Descripción de Caso de Uso Crear Perfil .....	43
Tabla 4 Descripción de Caso de Uso Modificar Perfil .....	45
Tabla 5 Descripción de Caso de Uso Pedir Recomendación .....	47
Tabla 6 Estereotipos .....	57
Tabla 7 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Crear Perfil de Usuario. ....	67
Tabla 8 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Modificar Perfil de Usuario. ....	68
Tabla 9 Descripción de las Variables de los casos de usos Crear Perfil de Usuario, Modificar Perfil de Usuario y Pedir Recomendación .....	69

# Índice de Figuras y Tablas

---

Tabla 10 Matriz de datos Caso de Uso Crear Perfil de Usuario. ....	69
Tabla 11 Matriz de datos Caso de Uso Modificar Perfil de Usuario. ....	70
Tabla 12 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Pedir Recomendación.....	71
Tabla 13 Matriz de datos Caso de Uso Pedir Recomendación.....	72

# Capítulo 1: Fundamentación Teórica

---

## Introducción

Desde sus inicios el hombre ha tenido la necesidad de comunicarse con sus semejantes para transmitir y adquirir conocimientos, de esta forma transformar la naturaleza y mejorar sus condiciones de vida. Su quehacer científico no ha cesado a lo largo de toda la historia y ha estado enmarcado fundamentalmente en el estudio de los medios de comunicación y difusión masiva.

Internet es un medio de comunicación y difusión creado por el hombre, que surge en 1969 en los Estados Unidos con el nombre de ARPANET (Investigación de Proyectos Avanzados de Red de Agencias) y es la red de computadora más grande del mundo, la cual posibilita que millones de personas se comuniquen para que intercambien puntos de vistas, creencias religiosas e ideas científicas. En esta red se publican varios tipos de información como noticias, libros películas, música, Web para que sean consultadas y consumidas por los usuarios<sup>1</sup>.

La cantidad de información que se publica en internet, aumenta considerablemente por cada año que pasa. En el 2006 este universo digital alcanzó la dimensión de 161 000 millones de de gigabytes esto equivale aproximadamente a 3 millones de veces la cantidad total de libros escritos en la historia de la humanidad o el equivalente a 12 pilas de libros extendiéndose a 93 millones de millas desde la Tierra al Sol, para el 2011 la información aumentará 7 veces de la antes escrita (1).

Los usuarios siempre están realizando búsquedas en internet, pero al existir esta gran cantidad de información pierden tiempo para encontrar lo que realmente necesitan, sintiéndose decepcionados, teniendo que acudir a otras personas para que los orienten en sus búsquedas. De esta forma surgen los sistemas de recomendación como herramienta que orientan a los usuarios en sus búsquedas y recomiendan contenido ajustado a sus gustos y preferencias cuando la cantidad de información es realmente grande.

Estos avances tecnológicos se han expandido en distintos países y por supuesto Cuba, que es la mayor de las Antillas, no se ha quedado atrás a pesar del bloqueo impuesto.

---

<sup>1</sup> Persona que hace uso de una computadora

# Capítulo 1: Fundamentación Teórica

---

Cuando triunfó la Revolución el Comandante en Jefe Fidel Castro Ruz se interesó por el desarrollo educacional de la sociedad para que los ciudadanos cubanos obtuvieran una cultura general integral. Fue así como dentro de la batalla de ideas surgió la Universidad de Ciencias Informáticas (UCI) en septiembre del 2002, la cual tiene como objetivo formar personal altamente calificado en la rama de la informática para desarrollar software confiable de forma eficiente y creativa.

En esta universidad existen varios centros de desarrollo de software con diferentes líneas investigativas. En la facultad 6 se encuentra el Centro de Geoinformática y Señales Digitales (GEYSED) y específicamente dentro del departamento de Señales Digitales se trabaja en una Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV). La misma es un proyecto de investigación y desarrollo enfocado en la realización de un producto para la mejora de las transmisiones televisivas y radiales que son realizadas en la Dirección de Televisión Universitaria en la UCI. La Plataforma cuenta con los siguientes subsistemas:

- Web, Transmisión, Programación
- Radial, Seguridad, Monitoreo
- Transferencia, Equipamiento, Gestión de Medias
- Producción y Reporte.

El subsistema Web de la plataforma, es el encargado de publicar al usuario final las medias, que con anterioridad fueron programadas por un usuario de la plataforma con permisos para ello (programador). En este proceso de programación, no intervienen las preferencias y gustos de los usuarios, sino que estos materiales son publicados prácticamente de forma aleatoria, por estrenos u otro tipo de criterio. No existen mecanismos que posibiliten al usuario elegir qué ver o qué publicar y de esta forma lograr una retroalimentación por parte de los administradores de la plataforma. Esta situación crea insatisfacciones en los usuarios finales, que son los que deben ser beneficiados para satisfacer sus necesidades. Por otra parte, actualmente en la plataforma se gestionan alrededor de 3000 ficheros de video, pertenecientes a la categoría películas solamente, lo cual quiere decir que esta cifra es bastante conservadora con respecto a la cantidad total que debe incluir el sistema cuando se tengan registrados todos los contenidos necesarios para que funcione correctamente. Esta cantidad abrumadora de medias imposibilita que el usuario realice búsquedas efectivas de algún contenido de su preferencia, o que pueda decidir en un momento

# Capítulo 1: Fundamentación Teórica

---

determinado qué media desea ver. En resumen, este sistema no centra su atención en los gustos y preferencias de los usuarios.

Teniendo en cuenta lo dicho anteriormente surge el siguiente problema a resolver: La carencia de procedimientos de retroalimentación en la plataforma PTARTV, a partir de las preferencias de los usuarios, genera insatisfacciones por parte de los clientes.

Dado el problema planteado se define como objeto de estudio: Los Sistemas de Recomendación y se define como campo de acción: Las técnicas de recomendación a partir de las preferencias de los usuarios en la plataforma PTARTV.

Para darle solución al problema planteado se trazó como objetivo general de la investigación: Desarrollar un sistema que permita la automatización de los procesos de recomendación en la plataforma PTARTV basado en las preferencias de los usuarios.

Se plantea como idea a defender: El desarrollo de un sistema automatizado de recomendación empleando algoritmos de Inteligencia Artificial en la plataforma PTARTV permitirá orientar y sugerir materiales a los clientes de una forma efectiva, elevando el grado de satisfacción de los mismos con dicha plataforma.

Para desarrollar estos objetivos se desarrollan las siguientes tareas de investigación:

1. Describir la evolución histórico-lógica de los sistemas de recomendaciones.
2. Caracterizar los algoritmos y técnicas de inteligencia artificial a utilizar en el desarrollo del sistema.
3. Determinar las tendencias y tecnologías actuales factibles para el desarrollo de la aplicación.
4. Caracterizar el proceso de recomendaciones aplicado a la plataforma PTARTV.
5. Desarrollar los artefactos y la documentación según la metodología de desarrollo seleccionada.
6. Implementar las funcionalidades requeridas.
7. Validar la propuesta de solución.

Para la realización de estas tareas se emplearon diferentes métodos de investigación:

## **Métodos Teóricos:**

*Histórico-Lógico:* ayuda a caracterizar las soluciones existentes de Sistema de Recomendación. Lo que hace posible que se entienda la evolución de las mismas, para aprovechar los puntos en común de las

# Capítulo 1: Fundamentación Teórica

---

que existen actualmente con la que se pretende construir. Utilizando en el desarrollo de la que se realiza, conceptos teóricos existentes que sean de relevancia.

*Análítico-Sintético:* permite sintetizar la información necesaria para los Sistemas de Recomendación. Esto posibilita la comprensión de la misma, con ideas claras y concisas.

*Modelación:* en el desarrollo de la investigación se utiliza este método para hacer modelos que ofrecen la posibilidad de crear abstracciones para explicar la realidad. Se hace visible en el trabajo al crear modelos como el de implementación.

## Métodos Empíricos

*Observación:* con la utilización de este método se determinarán los rasgos imprescindibles en el desarrollo del Sistema de Recomendación para la Plataforma de Transmisión Abierta para Radio y Televisión.

Este documento tiene cuatro capítulos:

**Capítulo 1:** Describe algunas de las características de los sistemas de recomendación, surgimiento y desarrollo de estos, así como una referencia al estado del arte de los proyectos existentes en el mundo, semejantes al que se va a desarrollar. También, se realiza la selección de la metodología, tecnología y herramientas actuales, que se van a utilizar para el desarrollo del sistema.

**Capítulo 2:** Se ofrece el modelo de dominio de la aplicación, conjuntamente con la especificación de los requerimientos funcionales y no funcionales, y el modelo de casos de usos del sistema.

**Capítulo 3:** Muestra los resultados obtenidos en el desarrollo de los procesos de análisis y diseño del sistema, así como los diagramas que fueron necesarios para obtener una mayor claridad a la hora de elaborar la solución que se propone.

**Capítulo 4:** Contiene la solución del sistema propuesto reflejada en el modelo de implementación, también las pruebas que fueron realizadas a dicho sistema.

# Capítulo 1: Fundamentación Teórica

---

## Capítulo 1: Fundamentación teórica

### Introducción

El objetivo de este capítulo es identificar los conceptos y elementos teóricos que permitan entender de una forma asequible y clara, los elementos principales de un sistema de recomendación así como que sirvan de soporte para la realización y descripción del mismo. También, se da una panorámica del estado en que se encuentran en el mundo estas tecnologías. Además, se realiza una breve descripción sobre las principales metodologías y herramientas utilizadas en la solución propuesta.

### 1.2 Sistema de Recomendación

Un sistema de recomendación, es una herramienta que ayuda al usuario a obtener información que necesita según sus preferencias y gustos.

Existen varias clasificaciones para este tipo de sistemas, de acuerdo al contenido que manejan.

- **Basado en contenido (*contentbased*):** Generan recomendaciones al usuario teniendo en cuenta un ítem similar que él haya seleccionado. Por ejemplo si un usuario es afín a un libro de matemáticas, el sistema recomendará cualquier libro etiquetado con el tema de matemática (2).
- **Filtrado Colaborativo (*Collaborativefilter*):** Se les llama también sistemas de recomendación sociales. Si a un conjunto de usuarios de un grupo le gusta un determinado ítem es de suponer que a los demás usuario de ese grupo le guste también. En conclusión se le recomendará al usuario un ítem que a las personas con mismos gusto y preferencias les haya gustado en un pasado (2)
- **Recomendaciones basadas en conocimiento (*Knowledgebased*):** Las sugerencias de los ítems se basan en inferencias sobre las necesidades de los usuarios y sus preferencias. Para ello se utiliza conocimiento en donde se tiene información sobre cómo un ítem específico responde a una necesidad en particular del usuario y, por lo tanto, la razón sobre la relación entre la necesidad y una posible recomendación (2).
- **Sistema de recomendación híbrido (*HybridRecommender Systems*):** Contemplan varios tipos de sistemas de recomendación, el más común es el filtro colaborativo y el sistema basado en contenido. Como consecuencia se disminuyen las desventajas de ambos y se aumentan los beneficios de la recomendación (2).

# Capítulo 1: Fundamentación Teórica

---

La forma en que le llega el contenido a los sistemas de recomendación se llama retroalimentación, las cuales se describen a continuación:

**Explícita:** El usuario tiene que proporcionar información al sistema. Se puede ver como ejemplo un usuario que ha valorado una película con un número entre uno y cinco de acuerdo a su apreciación.

**Implícita:** El usuario proporciona al sistema muy poca o ninguna información.

El subsistema Web de la plataforma PTARTV no está aún liberado por lo que no recibe visitas de usuarios, no obstante en la base de datos de la plataforma existen 20 usuarios registrados para simular dichas visitas. Esto conllevó a que en la presente investigación se descartara la posibilidad de desarrollar un sistema de recomendación colaborativo hasta tanto se tengan más usuarios y valoraciones de dichos usuarios sobre las medias de la plataforma. Esta falta de información se conoce en los sistemas de recomendaciones colaborativos como arranque en frío puesto que el sistema no cuenta con datos para realizar recomendaciones y representa una de las principales desventajas de estos tipos de sistemas.

Por otra parte los sistemas de recomendaciones basados en conocimientos carecen del problema de arranque en frío ya que no tiene en cuenta las valoraciones de los usuarios, pero las recomendaciones adquiridas son de tipo estático, es decir, para usuarios con las mismas necesidades se obtienen las mismas recomendaciones, además se necesita de una ingeniería de conocimiento. Por el análisis antes hecho también se descarta el desarrollo de un sistema de recomendación basado en conocimiento para la presente investigación y se propone la realización de un sistema de recomendación basado en contenido el cual se describe a continuación.

## 1.3 Sistema de recomendación basado en contenido

Los sistemas de recomendación basados en contenido buscan objetos que se adaptan lo mejor posible a los gustos de los usuarios. El elemento principal de esta metodología es la medida de similitud que se encarga de determinar cuan similar es un objeto de otro (3).

La ventaja de estos sistemas con respecto a los demás existentes, es que se realiza la recomendación sin necesidad de que el usuario haya tenido un historial determinado. Además dado que cada usuario tiene un perfil, se adapta mejor a sus gustos y preferencias.

Las principales desventajas de estos sistemas son:

# Capítulo 1: Fundamentación Teórica

---

- **Sobre especialización:** consiste en que las recomendaciones dadas al usuario suelen ser muy similares ya que se basan en la misma información
- **Dependencia de objetos:** es difícil extraer buena información de los objetos para realizar un cálculo confiable de similitud.

Los usuarios que usarán el sistema de recomendación deberán tener un perfil, según la información que le hayan proporcionado al mismo. Esta información consiste en sus preferencias e información personal.

A continuación se muestra la composición del perfil de usuario con la información de preferencia y la información personal.

## 1.3.1 Perfil de usuario

Los atributos del perfil de usuario son los siguientes:

- **Ingredientes:** Es el atributo más importante de preferencia de usuario, es donde él usuario define con precisión sus gustos sobre un contenido de media, dándole un valor que oscila entre [0...100]. Los ingredientes que se utilizaron son:
  - Música, Terror, Drama
  - Violencia, Humor, Suspenso
  - Aventura, Amor, Infantil
- **Actores:** Lista de actores de la base de datos.
- **Directores:** Lista de directores de la base de datos.
- **Productores:** Lista de productores de la base de datos

Los actores, directores y productores tienen un peso asociado con un valor que oscila entre [0,10]. Este valor indica que relevancia poseen para los usuarios los atributos valorados.

La información personal de los usuarios presenta los siguientes atributos:

- **Nombre**
- **Apellidos**
- **Edad**

# Capítulo 1: Fundamentación Teórica

---

- **Ocupación**
- **Estado civil**
- **Gustos personales**

## 1.3.2 Contenido de media

Por cada media del sistema se define un contenido, el mismo se define como contenido de media. A continuación se brinda una descripción de los atributos que lo conforman.

- **Ingredientes**
- **Actores**
- **Directores**
- **Productores**

Estos atributos se definen de la misma manera que se definieron los del perfil de usuario. Los directores, actores y productores se consideran información opcional ya que pueden ser o no proporcionados por el usuario en su perfil, pero son obligatoriamente definidos en el contenido de media.

## 1.3.3 Cálculo de similitud y nueva propuesta

Para saber cuán parecido es un contenido de video con las especificaciones hechas por el usuario en su perfil hay que aplicar una medida de similitud. El cálculo de similitud consiste en hallar la distancia entre los ingredientes de ambos (contenido y perfil de usuario) y el cálculo de las coincidencias de información opcional asociada con su peso entre ambos también.

Por lo general para calcular la similitud que existe entre los ingredientes del perfil de usuario y los ingredientes de contenido de media se utiliza como medida de similitud una variación de la fórmula euclidiana que está dada como (3):

$$\delta = \sqrt{\frac{\sum_{i=1}^n [(1-(a_i-c_i)^2)*a_i]}{\sum_{i=1}^n a_i}} \quad [1]$$

En esta fórmula  $a_i$  representa el valor de un ingrediente del perfil del usuario y  $c_i$  el valor de un ingrediente de contenido de media con el cual se hace la comparación.

# Capítulo 1: Fundamentación Teórica

---

Cuando existe información opcional en el perfil del usuario se calcula la similitud hallando el número de coincidencias que existen entre los pesos de los actores, productores y directores de ese perfil y del contenido de media determinado, aplicándose la siguiente fórmula:

$$\beta = \sum_{i=1}^n P_i * N_i \quad [2]$$

En esta fórmula  $P_i$  es el peso asociado cada autor, productor y director, normalizado entre  $[0...1]$  y  $N_i$  es el número de coincidencias. Los valores de  $\beta$  están en el rango de  $[0...1]$  por lo que si en la fórmula [2] los valores de la suma son superiores a uno se coge el máximo que es uno.

El resultado final del cálculo de similitud se obtiene aplicando la medida siguiente:

$\gamma = \omega * \delta + (1 - \omega) * \beta$  [3] donde  $\gamma, \omega \in [0,1]$  y la variable  $\omega$  distribuye el peso entre la distancia calculada en la ecuación [1] y la coincidencias en la ecuación [2].

Los valores de  $\gamma$  cercanos a uno indican que el contenido de media es de interés al usuario cuyo perfil es procesado para la recomendación.

Seguidamente un ejemplo donde se expone el cálculo simplificado de similitud entre un perfil y un contenido de media:

Suponiendo que el cálculo de la distancia euclidiana aplicando la fórmula [1] entre el perfil del usuario y la media de contenido es de 0.6 y el número de coincidencia es 3 actores 1 productor y 1 director con un peso asociado a cada uno de 0.3, obtenemos un valor de  $\beta = 1,5$  aplicando la ecuación [2] dado que es mayor que 1 entonces  $\beta = 1$ , luego se le da un valor a  $\omega = 0.5$  el resultado de la similitud sería  $\gamma = 0.8$  aplicando la ecuación [3].

Los sistemas de recomendación tienen que tratar con una gran cantidad de contenido por lo que el consumo de memoria y de CPU es muy elevado. La optimización de los algoritmos para mejorar el rendimiento es uno de los campos que va cobrando importancia a medida que se desarrollan estos sistemas. En la presente investigación científica se propone una función para el cálculo de similitud en los sistemas de recomendación basados en contenido con el fin de mejorar el rendimiento del sistema propuesto. Antes de la proposición se da una definición formal del coeficiente de similitud y las condiciones que debe cumplir una función de esta índole.

# Capítulo 1: Fundamentación Teórica

---

Washington (1984) afirma que un coeficiente de similitud es una medida de semejanza de la estructura de dos comunidades (4).

Janson y Vegelius (1981) afirma que un coeficiente de similitud debe de cumplir para todas las especies  $X_x, X_y, X_z$  las siguientes propiedades (4):

- a)  $S(X_x, X_y) = S(X_y, X_x)$
- b)  $S(X_x, X_x) = 1$
- c)  $|S(X_x, X_y)| \leq 1$
- d)  $S(X_x, X_y) = 1 \Rightarrow S(X_x, X_z) = S(X_y, X_z)$

Legendre y Legendre (1983) afirma que el coeficiente de similitud simple se utiliza de la forma siguiente (4):

$$S(X_1, X_2) = \frac{\text{arreglo}}{n} \quad [4]$$

El arreglo, corresponde al número de atributos para los cuales ambas muestras tienen las mismas características o valor. Esto se puede observar claramente en la tabla siguiente donde hay un par de muestras descritas por 10 variables.

**Tabla 1 Cálculo de similitud entre dos muestras usando el Coeficiente Simple**

Variables										
Muestra $X_1$	3	1	8	9	5	8	2	5	0	1
Muestra $X_2$	2	1	5	2	3	8	1	8	0	1
Arreglo	0	1	0	0	0	1	0	0	1	1

Sumando todos los valores del arreglo, quedaría como  $0+1+0+0+0+0+1+0+0+1+1=4$  y  $S(X_1, X_2) = \frac{4}{10} = 0.4$

# Capítulo 1: Fundamentación Teórica

---

Suponiendo que la muestra  $X_1$  representa los valores de los ingredientes de un perfil de usuario y la muestra  $X_2$  los valores de ingredientes de un contenido de media, se puede aplicar la ecuación [4] y se obtendrían resultados bien parecidos a los obtenidos aplicando la ecuación [1].

Al hacer algunas observaciones, se evidencia claramente que el coeficiente de similitud simple es mucho más sencillo de implementar que la distancia euclidiana antes mostrada y el tiempo de ejecución es mucho menor. Esto implica que dicho coeficiente sea un candidato ideal para la implementación de los sistemas de recomendaciones basados en contenido, pero se detectó el problema siguiente:

Dado que en cada comparación el coeficiente de similitud simple establece dos valores (0 ó 1) para el arreglo resultante, esto conlleva a que se pierda mucha información ya que los valores deben ser iguales para asignar el máximo valor que es 1 en otro caso toma valor 0. Esto no tendría mucho sentido en el cálculo de similitud para los sistemas de recomendaciones basados en contenido, porque un perfil de usuario y un contenido de video estarían lo bastante lejos de ser similares. Por lo antes enunciado se propone una medida de similitud extendida del coeficiente de similitud simple llamada Coeficiente de Beltin, con la única diferencia que cuando los valores de dos muestras sean diferentes en vez de tomar como valor 0, tomar valores entre [0...1]. A continuación se muestra una definición más formal del Coeficiente de Beltin.

Definición 1:

sea  $S(X_i, X_j) = \frac{A}{n}$  (Lagendre y Lagendre (1983)) el coeficiente de similitud simple entre dos muestras

cualesquiera donde  $A = \sum_{k=1}^n T(X_i, X_j)_k$  y  $T(X_i, X_j)_k = \begin{cases} 1 & \text{si } X_i = X_j \\ 0 & \text{si } X_i \neq X_j \end{cases}$  además se tiene que

$\max(X_i, X_j) = \begin{cases} X_i & \text{si } X_i \geq X_j \\ X_j & \text{si } X_j \geq X_i \end{cases}$  es el máximo y  $\min(X_i, X_j) = \begin{cases} X_i & \text{si } X_i \leq X_j \\ X_j & \text{si } X_j \leq X_i \end{cases}$  es el mínimo, entonces se define

el Coeficiente de Beltin como,  $S(X_i, X_j) = \frac{A}{n}$  pero con un  $T(X_i, X_j)_k = \begin{cases} 1 & \text{si } X_i = X_j \\ \frac{\min(X_i, X_j)}{\max(X_i, X_j)} & \text{si } X_i \neq X_j \end{cases}$ .

Luego de definir el Coeficiente de Beltin se comprobará con las propiedades de Janson y Vegelius (1981), para especies  $X_i, X_j, X_h$ .

# Capítulo 1: Fundamentación Teórica

- a.  $S(X_i, X_j) = S(X_j, X_i)$  : si se aplica  $T(X_i, X_j)$  siendo  $X_i \neq X_j$  entonces el mínimo y el máximo siempre son los mismos.
- b.  $S(X_i, X_i) = 1$  : ( se comprueba con la Definición 1)
- c.  $|S(X_i, X_j)| \leq 1$  : (se comprueba con la Definición 1)
- d.  $S(X_i, X_j) = 1 \Rightarrow S(X_i, X_h) = S(X_j, X_h)$  : dado que  $S(X_i, X_j)=1$  solo si las especies  $X_i, X_j$  son totalmente similares por lo tanto  $S(X_i, X_h) = S(X_j, X_h) = 1$

**Tabla 2 Cálculo de similitud entre dos muestras usando el Coeficiente de Beltin**

Variables										
Muestra $X_1$	3	1	8	9	5	8	2	5	0	1
Muestra $X_2$	2	1	5	2	3	8	1	8	0	1
Arreglo	0,6	1	0,6	0,2	0,6	1	0,5	0,6	1	1

Con el Coeficiente de Beltin se pierde mucho menos información que con el coeficiente de similitud simple. En la última fila de la tabla 2 se muestran los valores del arreglo cuyo valor se calculó de la siguiente manera:  $2/3+1+5/8+2/9+3/5+1+1/2+5/8+1+1= 7,2$  y  $S(X_1, X_2) = \frac{7,2}{10} = 0,72$  .

Luego de haber dado una definición formal del Coeficiente de Beltin y haber expuesto sus ventajas, se propone el mismo como medida de similitud en el sistema de recomendación propuesto y dicha investigación científica.

## 1.4 Soluciones existentes

Los Sistemas de recomendación surgieron en la década del 90 dentro de los servicios de grupo de noticias, servicio de filtro de noticias que posibilitaba a la comunidad de usuarios acceder a las noticias que podían ser de su interés.

El primer sistema de recomendación que surgió fue nombrado Tapestry. Este sistema fue adoptado con el nombre de filtro colaborativo. En 1997 estos sistemas son llamados “sistemas de recomendación”, los

# Capítulo 1: Fundamentación Teórica

---

cuales han ido evolucionando y ligados siempre con el avance de las técnicas o ideas que iban surgiendo en diversas literaturas referentes a estos temas. También cabe destacar que en los últimos años los sistemas de recomendación han sido utilizados ampliamente en el ámbito de comercio electrónico (2).

Ejemplos de sistemas de recomendación son:

# Capítulo 1: Fundamentación Teórica

---

- **Fab**

En Fab, Balavanovic y Shoham [1997] crean un sistema híbrido, basado en contenido y filtrado colaborativo, que recomienda al usuario páginas Web. Combinando ambas técnicas Fab reduce las limitaciones de cada una de ellas. El sistema modela el perfil del usuario basado en el contenido de los análisis (cuando un usuario califica una página) y compara estos perfiles para determinar similitudes entre usuarios para una recomendación colaborativa. De esta manera el usuario recibirá páginas, tanto las que ha calificado relevantes (con respecto a su perfil) como las que han recibido calificaciones altas por usuarios con un perfil similar al suyo (vecinos cercanos) (5).

- **PHOAKS**

PeopleHelpingOneAnotherKnowStuff es un sistema de recomendación que reconoce y redistribuye recomendaciones de recursos de la Web buscando en mensajes electrónicos. Este sistema está basado en filtrado colaborativo, lo que hace posible que un grupo de personas hagan y reciban recomendaciones entre sí.

Se distingue de otros sistemas por dos características principales: el rol de especialización y re-uso. PHOAKS recomienda páginas de la Web, busca en los mensajes las opiniones que los participantes dejen acerca de estas páginas, y las selecciona si pasan ciertos requerimientos (5).

- **Referral Web**

Es un sistema interactivo para la construcción, visualización y búsqueda de redes sociales en páginas Web. Este sistema modela una red social mediante un grafo en cuyos nodos son representados los individuos y las conexiones entre nodos indican una relación directa descubierta entre ellos. Para identificar la relación directa entre individuos, se utilizan métodos tales como referencias encontradas en las páginas personales, referencias de co-autores en publicaciones técnicas, citas en las publicaciones y organigramas.

# Capítulo 1: Fundamentación Teórica

---

Referral Web no intenta crear nuevas comunidades sino más bien ayudar a los usuarios a hacer un uso más eficiente de sus redes existentes de colegas profesionales. Perteneciendo a una comunidad, el usuario puede descubrir contactos a gente o a información que de otra manera le estaría oculta (5).

- **Siteseer**

Es un sistema que utiliza los registros de páginas favoritas (*bookmarks*) de un usuario y la organización de éstos registros para la recomendación de páginas de la Web relevantes puesto que los registros representan interés en el contenido y su organización indica relevancia entre los elementos. Siteseer utiliza en método de filtrado colaborativo y recomienda al usuario las páginas de electrónicas de sus "vecinos cercanos". Estos sistemas de recomendaciones dan un ejemplo del sistema que se quiere lograr (5).

En Cuba la existencia de los sistemas de recomendaciones es casi nula, se propuso uno para la biblioteca de la Universidad de la Habana. Esta propuesta también aporta a este trabajo por lo que se estudió y se tiene en cuenta.

## 1.5 Tendencias y tecnología actuales para el desarrollo de la aplicación

El desarrollo de software no es una tarea fácil en estos tiempos ya que son muchas las empresas que se dedican su desarrollo. La calidad del software es sin duda alguna, necesaria para satisfacer las necesidades del cliente, por lo que el estudio de las tecnologías y metodologías de software es un elemento fundamental para que cualquier empresa tenga éxito en este campo.

A continuación se describen las herramientas y tecnologías escogidas para el desarrollo del sistema de recomendación propuesto en esta investigación.

### 1.5.1 Metodología de desarrollo RUP

El desarrollo de software es una tarea compleja, y prueba de ello es la existencia de varias propuestas metodológicas que inciden de distintas maneras en el proceso de desarrollo. Estas metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo de forma eficiente, se clasifican en dos grandes grupos, robustas o ágiles. Independientemente de la clasificación que puedan recibir, todas presentan ventajas y desventajas, que determina la utilización de una u otra en dependencia

# Capítulo 1: Fundamentación Teórica

---

del tipo de proyecto. Por encima de todo, su principal utilidad radica en organizar y guiar un equipo de proyecto.

Por su parte las metodologías ágiles, tales como Programación Extrema (XP, por sus siglas en inglés), Scrum, y Microsoft Solutions Framework (MSF), ofrecen una buena solución para aquellos proyectos de entorno volátil, donde los requisitos no se conocen con exactitud y pueden sufrir cambios durante el tiempo de vida del proyecto (6).

RUP (Rational Unified Process) es una metodología para asignar tareas en una empresa que define quién hace, qué hace y cómo lo hace. Consiste en un conjunto de actividades para transformar los requisitos de un usuario en un sistema software. Permite especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos. Está basado en componentes y utiliza el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) para representar los esquemas de un sistema de software y está centrado en el control de proceso, siendo muy usado en el desarrollo de proyectos de gran tamaño. Entre sus características principales están; dirigido por casos de uso, centrado en arquitectura e iterativo e incremental. El proceso de desarrollo de RUP se divide en cuatro fases donde cada una concluye con un hito bien definido (7).

**Inicio:** En esta fase se determina la visión del proyecto a realizar, alcance del mismo. Las iteraciones se centran específicamente en las actividades de modelado y requerimientos de la empresa.

**Elaboración:** Se establece una arquitectura base sólida. Las iteraciones se centran en el desarrollo de la base del diseño.

**Construcción:** Durante esta fase se lleva a cabo la construcción del producto por medio de una serie de iteraciones en las cuales se seleccionan algunos Casos de Uso, se redefine su análisis y diseño y se procede a su implementación y pruebas.

**Transición:** En esta fase se busca garantizar que se tiene un producto listo para ser entregado al usuario.

# Capítulo 1: Fundamentación Teórica

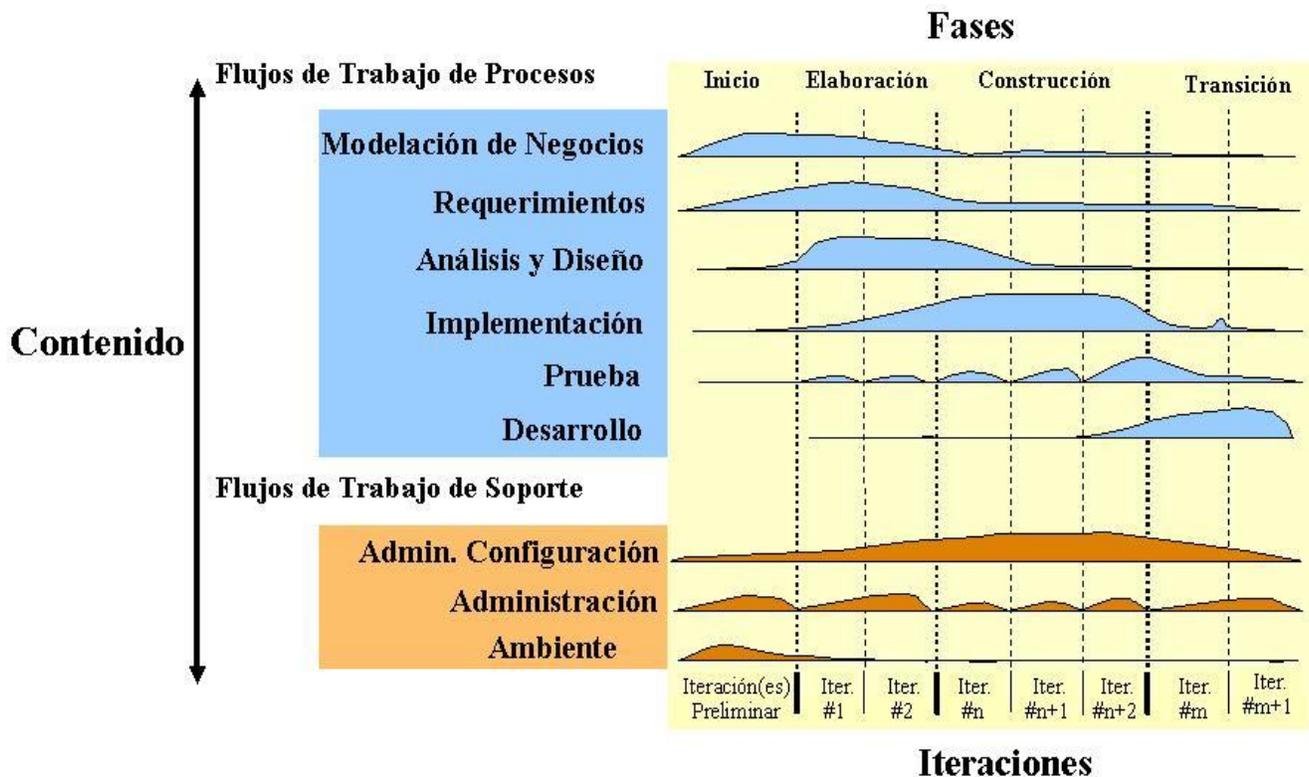


Figura 1 El RUP fases y flujos de trabajos (8).

Dentro de cada iteración de cada fase se llevan a cabo nueve flujos de trabajo, los cuales se explican a continuación.

**Modelado del negocio:** Se identifican los procesos de negocio. Con este flujo de trabajo se llega a un mejor entendimiento de la organización donde se va a implantar el producto. Los objetivos del modelado de negocio son:

- Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado.
- Entender el problema actual en la organización e identificar mejoras potenciales.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.

# Capítulo 1: Fundamentación Teórica

---

- Derivar los requisitos del sistema necesarios para apoyar a la organización.

**Requisitos:** Se identifican las restricciones que se imponen así como lo que el sistema debe hacer.

**Análisis y Diseño:** Describe cómo el programa será realizado y define cómo será programado.

**Implementación:** Define cómo estarán colocados los nodos y la ubicación en paquete de los objetos y clases.

**Prueba:** Se localizan los defectos del software.

**Despliegue:** Se entrega una versión operacional.

**Administración de configuración y cambio:** Describe el uso y la actualización concurrente de los elementos, el control de versiones entre otras actividades.

**Administración de proyecto:** Encargado de organizar el trabajo y de que el proyecto se termine en el tiempo planificado.

**Ambiente:** Describe los procesos y herramientas que soportarán el trabajo del proyecto.

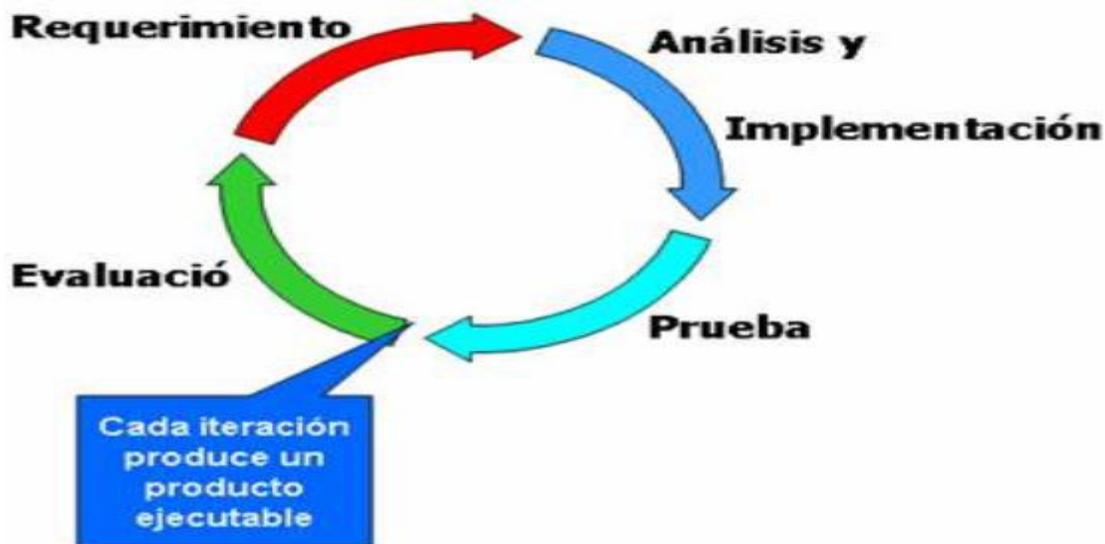


Figura 2 Iteraciones en RUP

# Capítulo 1: Fundamentación Teórica

---

La metodología RUP fue seleccionada para desarrollar el sistema de recomendación por ser la metodología que sustenta el desarrollo general del producto. Genera una gran cantidad de artefactos, lo cual representa una garantía para la continuidad del trabajo del proyecto y la futura implementación de las funcionalidades modeladas. Es una de las metodologías más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

## 1.5.2 Lenguaje de modelado UML

UML es un lenguaje de modelado que permite la representación conceptual y física de un sistema determinado así como documentar, visualizar y especificar los artefactos del mismo. Posee una notación muy especificada y permite modelar todas las fases de un proyecto informático. El éxito de los proyectos de desarrollo de aplicaciones o sistemas, se debe a los que sirven como enlace entre quien tiene la idea y el desarrollador. UML es una herramienta que cumple con esta función, ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien está involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo. De manera general presenta una serie de características como (9):

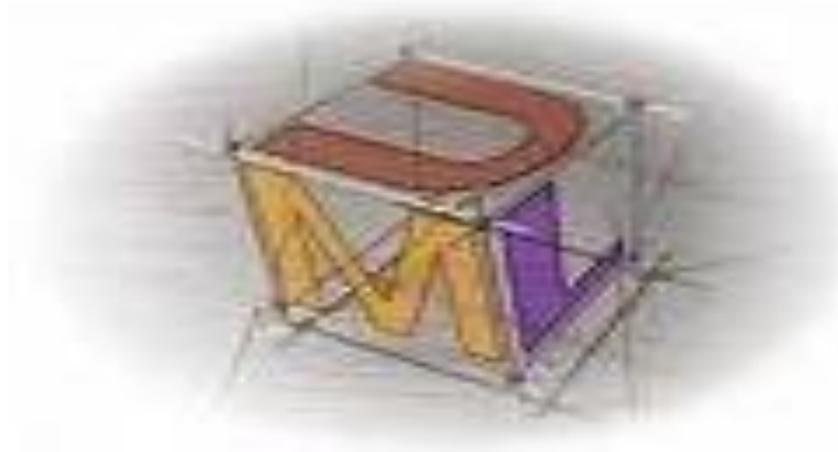
Tecnología orientada a objetos.

- Viabilidad en la corrección de errores.
- Desarrollo iterativo e incremental.
- Modela estructuras complejas.
- Permite especificar las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación.
- Modela el comportamiento del sistema mediante casos de uso, diagramas de secuencia y de colaboración, que sirven para evaluar el estado de este.
- Permite documentar varios artefactos de un proceso de desarrollo.

Los diagramas de UML se clasifican en estructuras estáticas (diagramas de clases, objetos y casos de uso), de comportamiento (diagrama de interacción de estado y actividad) y de implementación (diagrama de componente y despliegue).

# Capítulo 1: Fundamentación Teórica

---



**Figura3 UML (10)**

## **1.5.3 Herramienta CASE (Computer Aided Software Engineering) Visual Paradigm**

Visual Paradigm es una herramienta que sirve para realizar modelado UML y permite que se desarrollen productos de alta calidad para comunicar a los desarrolladores usando un lenguaje común. Otra característica importante que tiene es la capacidad de integración con diferentes IDE de desarrollo de software como son NetBeans (de Sun), JDeveloper (de Oracle), Eclipse (de IBM), JBuilder (de Borland). Presenta ingeniería inversa para JAVA, .NET, XML. Presenta otras características como exportación de imágenes jpg, png y svg. Las posibilidades que brinda el Visual Paradigm son muchas, esto hace que sea una de las herramientas de modelado más explotada por los desarrolladores de software (11).

# Capítulo 1: Fundamentación Teórica

---

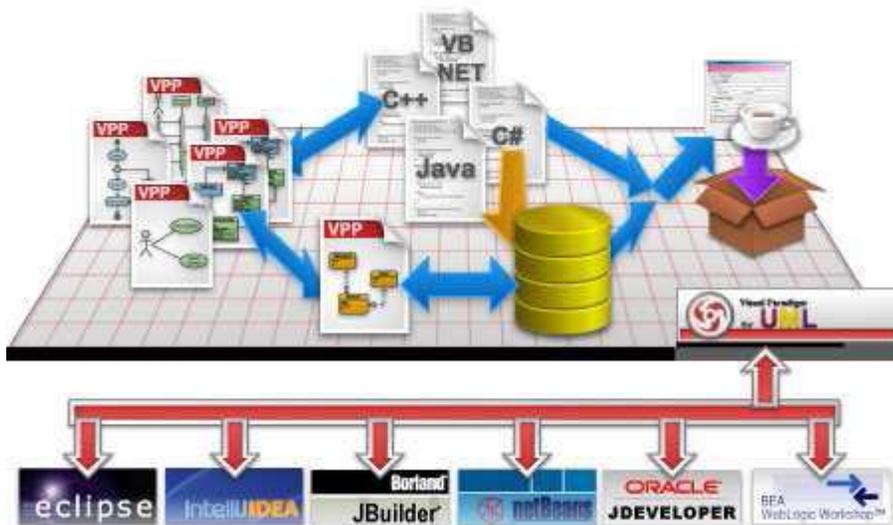


Figura 4 Visual Paradigm y los ID con los que se integra (12).

Esta herramienta fue escogida por las características enunciadas anteriormente, además de que el sistema de recomendación propuesto estará integrado al subsistema Web de la Plataforma PTARTV, cuyos artefactos y modelos han sido generados haciendo uso de la misma.

## 1.5.4 Lenguaje de programación PHP5

PHP es un lenguaje de alto nivel específicamente pensado para desarrollar páginas web por lo cual puede ser incrustado en páginas HTML. Es un lenguaje fácil de aprender. Su sintaxis es similar a C, Perl y Java. La versión PHP5 soporta programación orientada a objeto lo cual lo hace mucho más portable, asequible y manejable. Es uno de los lenguajes más usados en el mundo. Dispone de una gran cantidad de interfaces para diferentes sistemas de base de datos además puede conectarse de forma directa a estas (Mysql, Postgresql, Msq, Oracle, dbm, filepro). PHP está disponible para una gran cantidad de sistemas operativos como Unix, Linux, Windows, Free BSD. El código funciona sin tener que modificar los sistemas que lo ejecutan y dispone de acceso, por lo cual se puede agregar otras funcionalidades al lenguaje con total libertad.

Teniendo en cuenta la arquitectura de la plataforma PTARTV se escogió dicho lenguaje ya que permite la integración a la misma.

# Capítulo 1: Fundamentación Teórica

---

## 1.5.5 Sistema gestor de base de datos Postgresql

Los sistemas gestores de base de datos (SGBD), son un tipo de software que sirve de interfaz entre el usuario la base de datos y la aplicación que lo utiliza. El mismo se compone de un lenguaje de manipulación de datos DML, uno de definición de consulta DLL y de un lenguaje de consulta.

El Postgresql es un SGB de cual está bajo licencia BSD (Berkeley Software Distribution). Esta licencia tiene menos restricciones que la GPL ya que permite el uso de código fuente en software no libre. También corre en casi todos los sistemas operativos como Linux, Unix, Windows, BSDs, Mac OS y otros. Tiene una documentación muy bien organizada y se adapta fácilmente a las necesidades del cliente. Presenta un soporte para los lenguajes más usado como PHP, C, C++, Perl, Python. El Postgresql soporta todas las características de una base de dato profesional (funciones, secuencias, triggers, relaciones, reglas, tipo de datos defino por el usuario, vistas, etc.). El uso de este gestor de base de datos permite integrar el sistema de recomendación propuesto en la presente investigación con el subsistema Web de PTARTV.

## 1.5.6 Framework de desarrollo Symfony

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web (13).

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar

# Capítulo 1: Fundamentación Teórica

---

tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se muestran algunas de sus características:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- La capa de presentación utiliza plantillas y layouts que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del framework. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones.
- Los formularios incluyen validación automatizada y relleno automático de datos (“repopulation”), lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.
- La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.
- El sistema de enrutamiento y las URL limpias permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.

# Capítulo 1: Fundamentación Teórica

---

- El soporte de e-mail incluido y la gestión de APIs permiten a las aplicaciones web interactuar más allá de los navegadores.
- Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.
- Los plugins, las factorías (patrón de diseño “Factory”) y los “mixin” permiten realizar extensiones a medida de Symfony.
- Las interacciones con Ajax son muy fáciles de implementar mediante los helpers que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código (13).

El framework de desarrollo Symfony brinda muchas posibilidades para los programadores que lo deseen usar, esto debido a simplicidad y robustez. Estas características llevaron al equipo de desarrollo de la plataforma PTATV y a su arquitecto a seleccionarlo para la implementación de los subsistemas que se desarrollan en entorno web en dicha plataforma.

## 1.5.7 Extjs

ExtJs es un API escrito en Java Script con la finalidad de asistir el desarrollo de aplicaciones enriquecidas para internet. Entre las principales características de ExtJs resalta un fuerte paradigma basado en componentes soportado por recursos para la programación orientada a objetos en Java Script que facilitan la implementación de extensiones y aplicaciones de gran complejidad, el poseer un amplio conjunto de componentes configurables de alta calidad y una implementación transparente y sencilla para el trabajo con AJAX. (14)

ExtJs es usado como framework de presentación en el desarrollo de proyectos Web que conforman las RIA<sup>2</sup>, dado que sus características tributan a que pueda ser utilizado con estas finalidades. Para el desarrollo del presente trabajo se usará ExtJs v3.0, versión estable de dicho producto. Esta versión está bajo la licencia de la GPL.

---

<sup>2</sup> RIA: Rich Internet Application por sus siglas en Inglés, Aplicaciones enriquecidas para Internet

# Capítulo 1: Fundamentación Teórica

---

## 1.6 Conclusiones

Los sistemas de recomendación son un tipo de sistema que ayuda a los usuarios a seleccionar ítems de su interés. Existen diferentes tipos de sistemas de recomendación, entre ellos se encuentran el del filtro colaborativo, el basado en contenido, el basado en conocimiento e híbridos. En el presente capítulo se seleccionaron las herramientas necesarias que guiarán el proceso de desarrollo del sistema que se propone. Se fundamentó el uso del tipo de sistema de recomendación basado en contenido como solución a la problemática planteada, usando una medida de similitud llamada Coeficiente de Beltin.

# Capítulo 2: Características del Sistema

---

## Capítulo 2: Características del sistema

### 2.1 Introducción

En el presente capítulo se explica cómo se desarrolla el sistema de recomendación para el subsistema Web de la Plataforma de Transmisión Abierta para Radio y Televisión, generándose los artefactos necesarios, siguiendo la metodología de software seleccionada RUP. Se describirán los conceptos de los objetos del modelo de dominio, así como la especificación de los requisitos funcionales y no funcionales que el sistema debe cumplir. También se muestra una solución mediante el diagrama de caso de uso del sistema.

### 2.2 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 software. El modelo desarrollado no se trata de un conjunto de diagramas que describen clases de software u objetos de software con responsabilidades, sino que puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. Aprovechando las bondades de los diagramas UML 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 (15).

Se determinó desarrollar el modelo de dominio por la poca estructura y claridad de los procesos de negocios, por lo que se expone un marco conceptual y relaciones entre las definiciones. También se muestran los requisitos no funcionales y funcionales que son las capacidades que el sistema debe cumplir para satisfacer las necesidades del usuario, estos se utilizan para hacer el diagrama de caso de uso del sistema.

## Capítulo 2: Características del Sistema

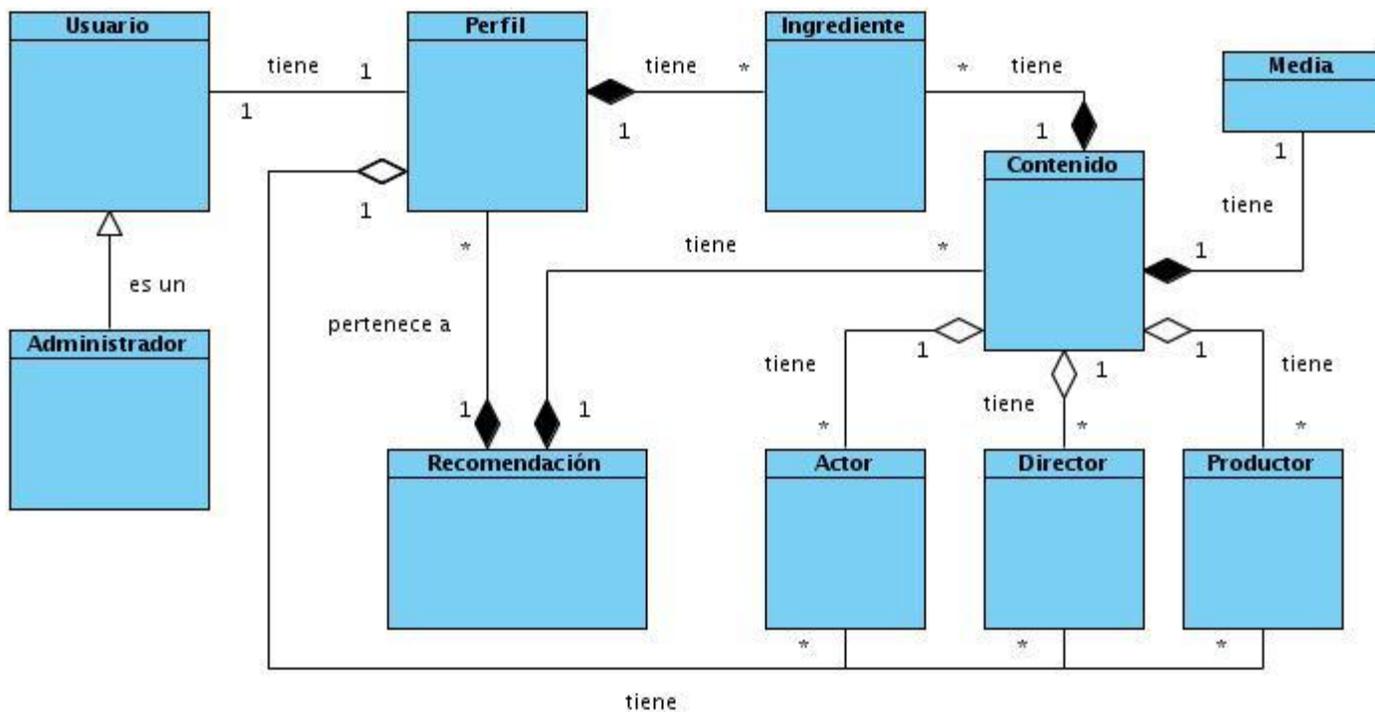


Figura 5 Modelo de Dominio

### 2.3 Principales conceptos

Para comprender mejor el Diagrama de Dominio a continuación se explican los conceptos siguientes:

- **Perfil:** Cada perfil pertenece a un usuario determinado donde se guarda la información personal del mismo, así como la descripción de sus gustos y preferencias. Estos gustos y preferencias se califican en ingredientes, actores, productores y directores.
- **Usuario:** Es la persona que interactúa con el sistema en busca de una recomendación orientada a sus gustos y preferencias.
- **Administrador:** Es un tipo de persona pero su función principal es la de agregar los contenidos al sistema.
- **Contenido:** Es la descripción de un producto determinado, en este caso se refiere a la descripción de una media, el cual está compuesto por ingredientes, actores, productores y directores.

# Capítulo 2: Características del Sistema

---

- **Ingrediente:** Está presente tanto en el perfil del usuario como en el contenido, es un calificador en el cual los usuarios pueden evaluar sus gustos y preferencias.
- **Media:** Puede ser un video o audio, ejemplo (música, películas, documentales).
- **Actor:** Está presente en el contenido como información obligatorio o en el perfil de forma opcional, describe el actor de una media.
- **Productor:** Está presente en el contenido como información obligatorio o en el perfil de forma describe un productor de una media.
- **Director:** Está presente en el contenido como información obligatorio o en el perfil de forma opcional, describe el director de una media.
- **Recomendación:** Es una cantidad de contenido determinado que el sistema de recomendación le brinda al usuario de acuerdo a sus gustos y preferencias.

## 2.4 Requisitos funcionales

Una vez realizado el modelo de dominio, es imprescindible caracterizar las funcionalidades que tendrá el sistema de recomendación de medias para la plataforma PTARTV. Estas funcionalidades se presentan como requisitos funcionales, los cuales no son más que las capacidades que el sistema debe tener para satisfacer las necesidades del usuario. A continuación se mencionan los requisitos del sistema propuesto.

- **R1:** Crear perfil de usuario.
- **R2:** Modificar perfil de usuario.
- **R3:** Pedir Recomendación.

## 2.5 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Son características que hacen al producto, atractivo, usable, rápido y confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Son

# Capítulo 2: Características del Sistema

---

importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. A continuación se mencionan los mismos.

## Requisitos de Usabilidad

- El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.

## Requisitos de Fiabilidad

- El sistema debe estar disponible todo el tiempo para sus usuarios, descontando el tiempo en que se encuentre en mantenimiento.
- El tiempo entre fallos no debe exceder los 3 meses.
- El tiempo medio de reparación en caso de fallos es de 7 días.

## Requisitos de Eficiencia

- El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento.
- Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

## Restricciones de diseño

- El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.
- Se deben emplear los estándares establecidos (diseño de interfaces, base de datos y codificación).

## Requisitos de Interfaz de usuario.

- El sistema debe tener una apariencia profesional y un diseño gráfico sencillo.
- El sistema debe ser intuitivo.

## Requisitos de Hardware Para las PCs clientes:

- Se requiere tengan tarjeta de red.
- Al menos 100 MB de memoria RAM.
- Se requiere al menos 150 MB de disco duro.

# Capítulo 2: Características del Sistema

---

- Procesador 512 MHz como mínimo.

## Para los servidores:

- El Servidor de Base de Datos tendrá como mínimo 1TB.
- Procesador Core 2Duo.

## Requisitos de Software

### Para las PCs clientes:

- Se requiere un Navegador como Mozilla Firefox 3.5.5 u otro navegador que cumpla con los estándares W3C<sup>3</sup> (World Wide Web Consortium).

### Para los Servidores:

- Sistema operativo GNU/Linux Ubuntu.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL 8.2 como Sistema Gestor de Base de Datos.

## 2.6 Diagrama de caso de uso del sistema

El diagrama de Caso de Uso del Sistema muestra la relación que existe entre los actores y los casos de uso del sistema, donde cada caso de uso representa una funcionalidad. Sirve como contrato entre los clientes y los desarrolladores. Se utiliza como punto de partida inicial para llegar a las actividades de análisis y diseño que posteriormente se realizarán en el desarrollo del software (16).

---

<sup>3</sup> Es un consorcio internacional que produce estándares para World Wide Web

# Capítulo 2: Características del Sistema

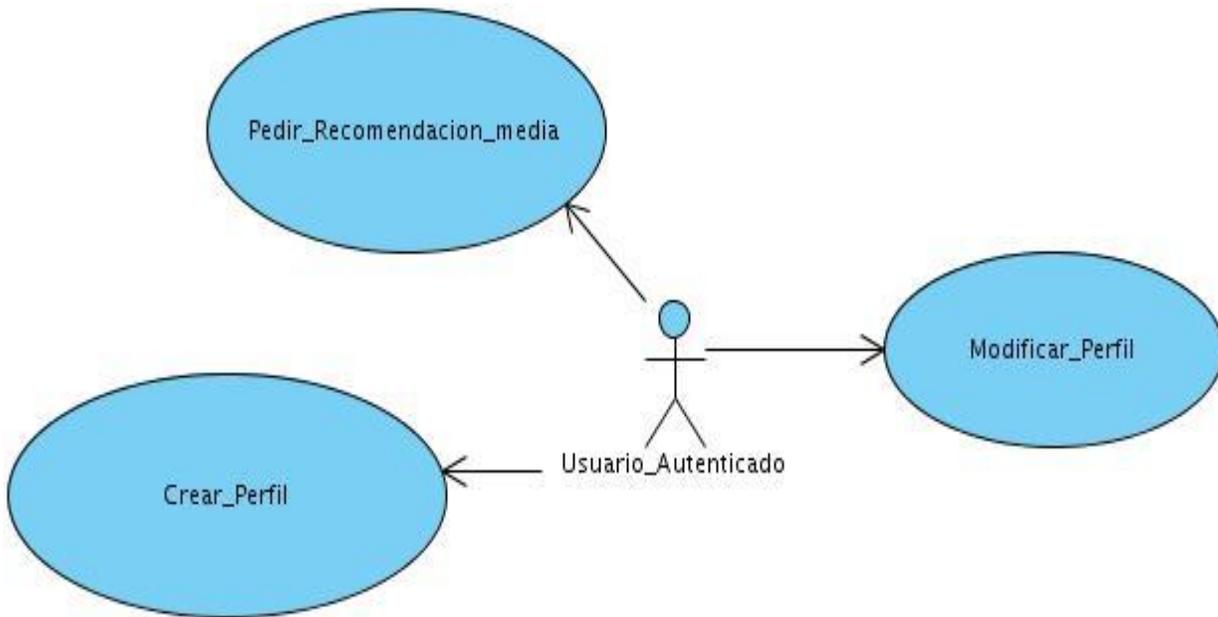


Figura 6 Diagrama de Casos de Uso del Sistema

## 2.7 Descripción de los Casos de Uso

Nombre del Caso de Uso:	Crear Perfil de usuario	
Actores:	Usuario	
Propósito:	Crear un nuevo perfil de usuario	
Resumen:	Comienza cuando el usuario accede al sistema para crear un nuevo perfil, luego el sistema muestra una interfaz para que el usuario introduzca los datos, el sistema valida los datos introducidos por el usuario y notifica al usuario si se creó o no el perfil.	
Referencias:	R1	
Tipo:	Crítico	
Precondiciones:		
Curso Normal de los eventos		
Acciones de los Actores	Respuesta del Sistema	
1- El caso de uso comienza cuando el usuario accede al sistema para crearse un perfil.		

## Capítulo 2: Características del Sistema

---

	2- El sistema muestra la interfaz al usuario.
3- El usuario introduce los datos en la interfaz.	
	4-El sistema valida los datos introducidos por el usuario.
	5-El sistema crea el perfil del usuario y muestra una notificación de la creación del mismo.
Curso Alterno de los eventos	
El usuario introduce los datos incorrectamente.	El sistema muestra un mensaje que dice que los datos no son correctos y se vuelve al paso 3.
Post condiciones	El nuevo perfil queda o no creado.

## Capítulo 2: Características del Sistema

Prototipo de Interfaz

The screenshot shows a web browser window titled "MI PERFIL" with a blue header bar containing navigation links: "Gestionar Perfil", "Administrar Contenido", and "Ayuda". The main content area contains a form with the following fields and options:

- Nombre: [Text input field]
- Apellidos: [Text input field]
- Edad: [Text input field]
- Ocupación: [Text input field]
- Estado civil:  Soltero
- Gustos Personales: [Text input field]
- Sexo:  Masculino

« Anterior Siguiente »

Tabla 3 Descripción de Caso de Uso Crear Perfil

Nombre del Caso de Uso:	Modificar Perfil de usuario
Actores:	Usuario
Propósito:	Modificar Perfil de usuario.

## Capítulo 2: Características del Sistema

---

Resumen:	Comienza cuando el usuario accede al sistema para modificar los datos de su perfil, luego el sistema muestra el perfil para que los datos sean modificados. El usuario modifica los datos de su perfil y el sistema muestra una notificación diciendo si los datos se modificaron correctamente o no.	
Referencias:	R2	
Tipo:	Crítico	
Precondiciones:	El usuario debe tener un perfil asociado.	
Curso Normal de los eventos		
Acciones de los Actores	Respuesta del Sistema	
1- El caso de uso comienza cuando el usuario accede al sistema para modificar su perfil.		
	2- El sistema muestra el perfil del usuario para que este modifique los datos correspondientes.	
3-El usuario modifica los datos necesarios.		
	4- El sistema valida los datos introducidos por el usuario.	
	5-El sistema guarda el perfil del usuario con los datos modificados.	
Curso Alternativo de los eventos		
El usuario introduce los datos incorrectamente.	El sistema muestra un mensaje que dice que los datos no son correctos y se vuelve al paso 3.	
Postcondiciones	El perfil del usuario queda o no modificado.	

## Capítulo 2: Características del Sistema

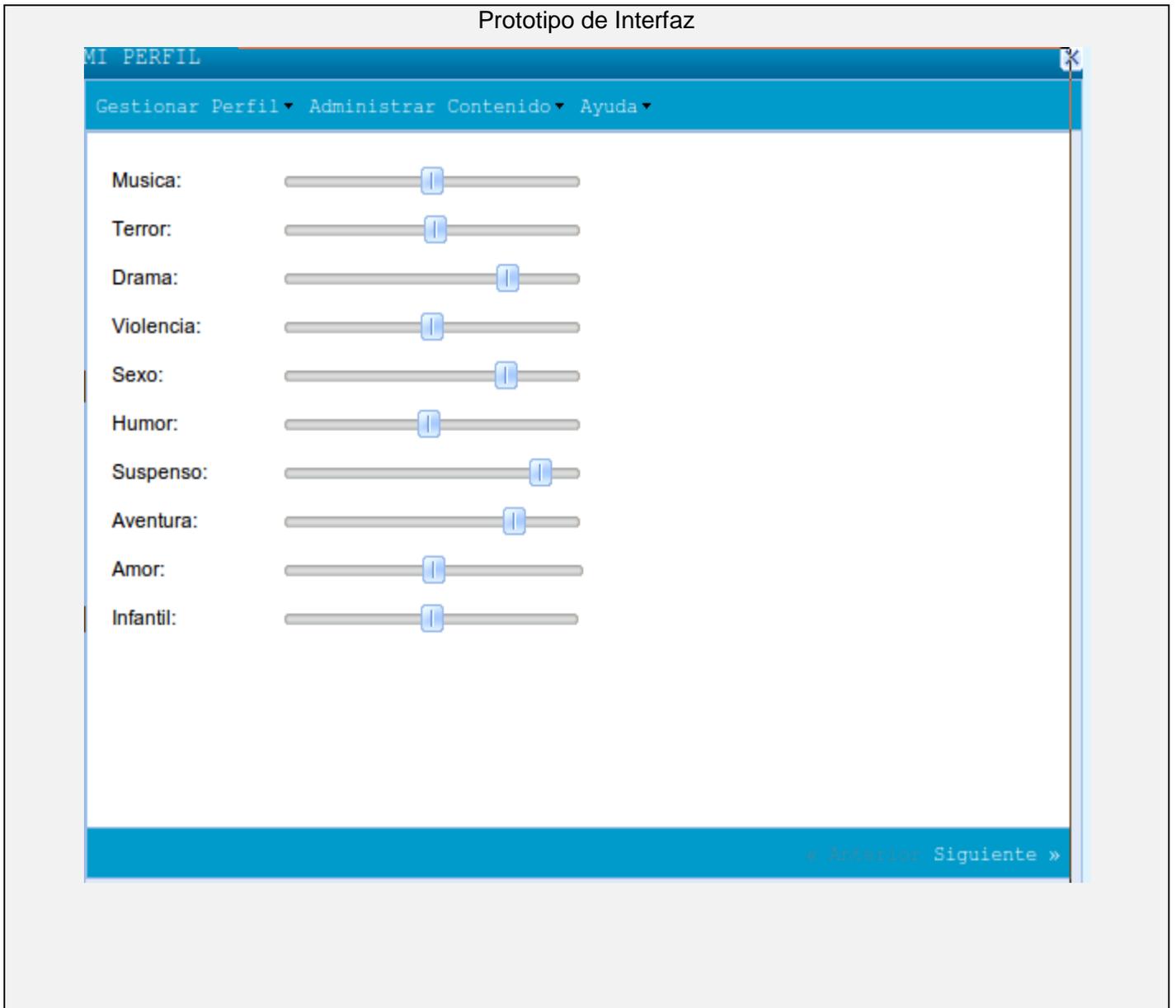


Tabla 4 Descripción de Caso de Uso Modificar Perfil

Nombre del Caso de Uso:	Pedir Recomendación
Actores:	Usuario
Propósito:	Recomendar al usuario
Resumen:	Comienza cuando el usuario accede al sistema para pedir una

## Capítulo 2: Características del Sistema

	recomendación, luego el sistema muestra una interfaz para que el usuario especifique la cantidad de contenido que desea obtener y una vez que el usuario introduzca dicha cantidad el sistema mostrará una lista con la cantidad de contenido especificada.
Referencias:	R3
Tipo:	Crítico
Precondiciones:	El usuario debe de tener un perfil asociado
Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- El caso de uso comienza cuando el usuario accede al sistema para pedir una recomendación.	
	2- El sistema muestra una interfaz para que el usuario especifique la cantidad de contenido que quiere recibir.
3- El usuario introduce la cantidad de contenido que quiere obtener en la recomendación.	
	4-El sistema verifica que la cantidad esté en un rango especificado.
	5-El sistema muestra una lista con la cantidad de contenido pedido por el usuario.
Curso Alternativo de los eventos	
El usuario introduce la cantidad de contenido fuera del rango especificado.	Se vuelve al paso 3.
Postcondiciones	Las recomendaciones para el usuario

# Capítulo 2: Características del Sistema

---

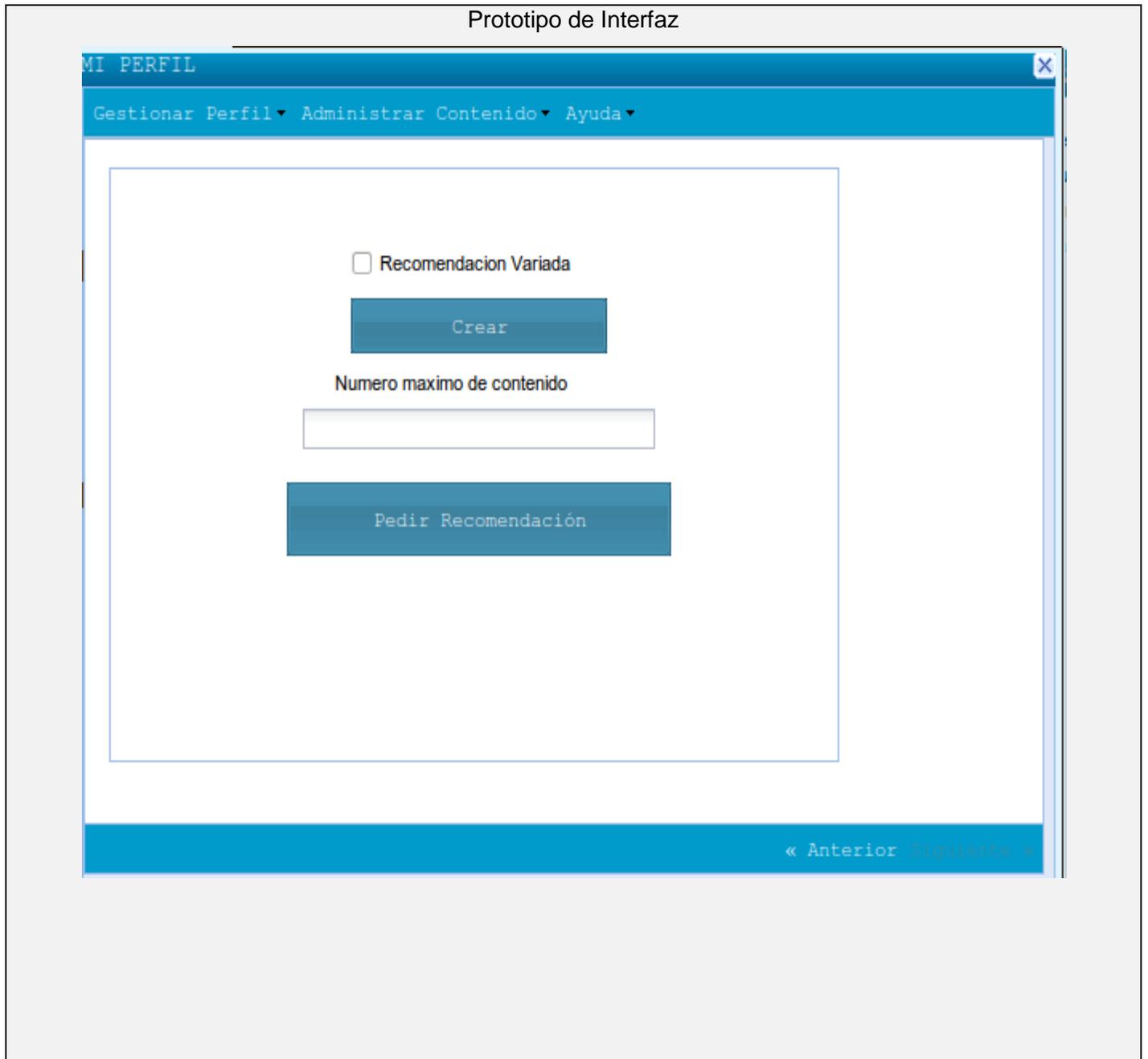


Tabla 5 Descripción de Caso de Uso Pedir Recomendación

## 2.8 Conclusiones

En este capítulo se realizó el modelo de dominio, el cual se utiliza para lograr la comprensión de los conceptos utilizados por los usuarios y con los que deberá trabajar en el sistema de recomendación. Se

## Capítulo 2: Características del Sistema

---

examinaron y valoraron todos los requisitos de software planteados por los usuarios finales del sistema. Como resultado principal de este capítulo, quedó plasmada la propuesta de solución detallada a través del diagrama de casos de usos del sistema, los cuales fueron descritos reflejando las posibles interacciones entre el usuario y la aplicación.

# Capítulo 3: Análisis y diseño del Sistema

## Capítulo 3: Análisis y diseño del sistema

### 3.1 Introducción

En este capítulo se construyen las clases de análisis y clases de diseño de la aplicación, también se realiza un análisis de los patrones de diseño y arquitectura usada por el framework de desarrollo Symfony para alcanzar mejores resultados del diseño que se realice.

### 3.2 Modelo de análisis

Con el objetivo de conseguir una comprensión más precisa y una descripción más detallada del sistema de recomendación propuesto se refinan y estructuran los requisitos obtenidos con anterioridad. Se realizan los diagramas de clases del análisis, los cuales muestran qué clases participan en las realizaciones de los distintos casos de uso del primer ciclo de desarrollo planteados en capítulo anterior. Estos diagramas se muestran a continuación.

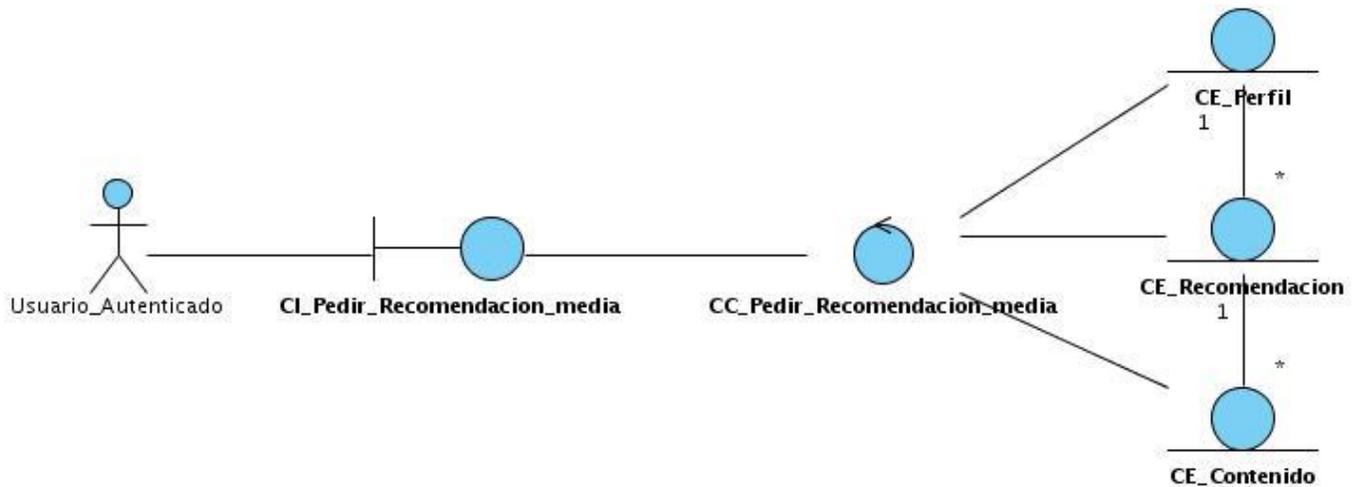


Figura 7 Clases de Análisis de Caso de Uso Pedir Recomendación

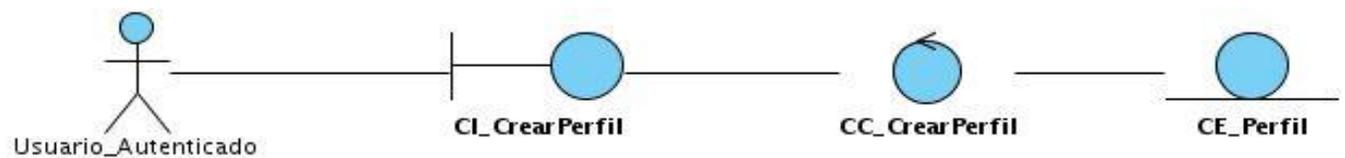


Figura 8 Clases de Análisis de Caso de Uso Crear Perfil

# Capítulo 3: Análisis y diseño del Sistema

---



Figura 9 Clases de Análisis de Caso de Uso Modificar Perfil

## 3.3 Framework de Symfony

Para realizar el modelo de diseño del sistema de recomendación se hace necesario conocer algunas características del framework de desarrollo Symfony. Para ello se analiza la arquitectura y las peculiaridades del mismo así como los patrones de diseño aplicado a su núcleo.

El framework de desarrollo Symfony está basado en el patrón arquitectónico Modelo Vista Controlador el cual está formado por tres niveles.

- El modelo, representa toda la información con la que trabaja el sistema.
- La vista, es la parte donde el usuario interactúa con el sistema.
- El controlador, procesa las interacciones con el usuario y los cambios que ocurren en el modelo o en la vista.

Esta arquitectura separa la presentación (vista) de la lógica de negocio (modelo) por lo que se consigue un mantenimiento más eficiente y sencillo de la aplicación. El modelo se encarga de abstraer la lógica de los datos haciendo que las vistas y las acciones sean independientes del gestor de base de datos utilizado. El controlador se encarga de aislar el modelo y la vista de los detalles del protocolo utilizado. Symfony realiza su propia implementación del Modelo Vista Controlador tomado lo mejor de la arquitectura, garantizando un desarrollo más rápido de las aplicaciones.

# Capítulo 3: Análisis y diseño del Sistema

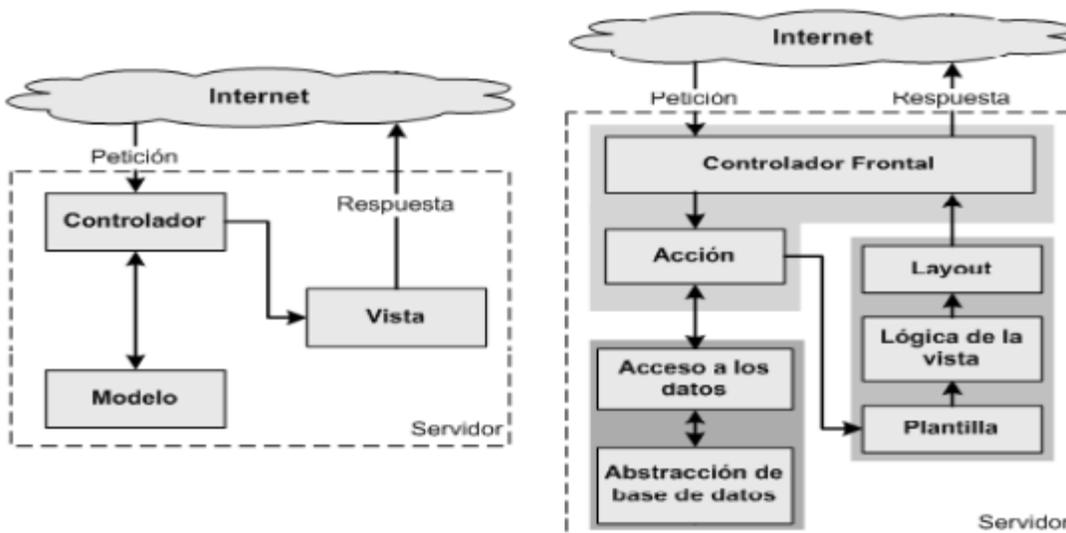


Figura 10 Patrón Modelo Vista Controlador de Symfony (13).

El controlador realiza múltiples tareas en las aplicaciones web por lo que se sobrecarga mucho. Por esta razón el controlador de Symfony se divide en controlador frontal que es único para cada aplicación y las acciones que incluyen el código específico del controlador de cada página. El controlador frontal ofrece un punto de entrada para cada aplicación y se encarga fundamentalmente del manejo de las peticiones del usuario, de la seguridad y de cargar la configuración de la aplicación (13).

Las páginas web presentan elementos que no sufren cambio en ningún momento a lo largo de toda la aplicación. Por este motivo en Symfony la vista se separa en un layout global en un grupo de páginas o en toda la aplicación y en una plantilla que se encarga de visualizar las variables definidas en el controlador. Para que la interacción de estos componentes se realice correctamente es necesario agregar cierto código llamado lógica de vista que determina el layout ha usar para cada plantilla (13).

La capa de modelo se divide en dos, la de acceso a datos y la de atracción a la base de datos. De esta forma las funciones que acceden a los datos no utilizan sentencias ni consultas que dependan de una base de datos sino de otras funciones que realizan las consultas. De esta forma si se cambia el gestor de base de datos solo es necesario realizar la capa de abstracción de la base de datos.

Las clases de la capa del modelo se generan automáticamente con la utilización del mapeo de objetos a base datos (ORM, por sus siglas en inglés "Object-Relational-Mapping"). La abstracción de la base de

# Capítulo 3: Análisis y diseño del Sistema

---

datos es completamente transparente para el programador, de esta manera si se cambia el sistema gestor de bases de datos en cualquier momento no se debe reescribir ni una línea de código, sólo es necesario modificar un parámetro en un archivo de configuración. En un principio, el framework utilizaba el ORM Propel, pero la evolución del mismo ha permitido alternar entre este y Doctrine, convirtiéndose este último en el ORM por defecto para las versiones más actuales (13).

La implementación que realiza Symfony simplifica el proceso de desarrollo de una aplicación; esto se debe a que el controlador frontal y el layout son comunes a todas las acciones de la aplicación aunque se pueden tener varios controladores y varios layout. La vista se puede transformar en un archivo sencillo sin necesidad de programarla (13).

## 3.3.1 Patrones de Diseño Symfony

Un patrón es una descripción de un problema y la solución, a la cual se le da un nombre y se puede aplicar a nuevos contextos. (15) Los patrones de diseño se dividen en dos grandes grupos los “General Responsibility Assignment Software Patterns” (GRASP) y los “Gang of Four” (GOF).

### Patrones GRASP

Los patrones GRASP establecen los principios fundamentales de la asignación de responsabilidades a objetos. En Symfony se aplican los cinco patrones de este grupo donde interactúan diseñadores y programadores. Seguidamente se describe la solución que brindan estos patrones, el problema que resuelve y en que parte del framework de Symfony se aplica.

- **Alta Cohesión:** Asignar una responsabilidad de manera que la cohesión permanezca alta. ¿Cómo mantener la complejidad manejable? (15) Permite una alta colaboración entre las clases de un sistema. Symfony permite asignar responsabilidades a una clase con una alta cohesión siempre que se realice un trabajo de acuerdo a la estructura planteada por el framework, en el cual debe existir una acción para cada una de las plantillas. Las acciones contenidas en cada una de las clases poseen una fuerte relación, teniendo un sentido común y donde su mayor responsabilidad radica en definir las acciones para cada una de las plantillas de un módulo. Permitiendo así que dichas clases sean flexibles y aporten un alto nivel de modularidad a los sistemas que se implementen con el framework en cuestión.

## Capítulo 3: Análisis y diseño del Sistema

---

- **Bajo Acoplamiento:** Asignar una responsabilidad de manera que el acoplamiento permanezca bajo. ¿Cómo soportar bajas dependencias, bajo impacto del cambio e incremento de la reutilización? (15) Permite mantener un bajo nivel de dependencias entre las clases que conforman un sistema y una alta probabilidad de que puedan ser reutilizadas. Symfony cumple con este patrón de diseño casi en su totalidad, debido que el nivel de dependencia entre las clases que contienen las acciones es muy bajo por no decir que ninguno. Esto es posible gracias a la sintaxis y la propia estructura del framework, dando al traste con el patrón anterior. Las acciones son métodos con el nombre `executeNombreAccion` de una clase llamada `nombreMóduloActions` que hereda de la clase `sfActions` y se encuentran agrupadas por módulos. Lo cual brinda gran nivel de reutilización a cada uno de los módulos que se implemente en una aplicación. Además si se quiere obtener la reutilización a nivel de acciones, se puede utilizar una sintaxis alternativa para distribuir las acciones en archivos separados. En este caso, cada clase acción extiende de `sfAction` y su nombre es `nombreAccionAction`. El nombre del método es simplemente `execute`. Mientras el nombre del archivo es el mismo que el de la clase a diferencia del anterior que es `action.class.php`.
- **Creador:** Se encarga de asignar a la clase B la responsabilidad de crear una instancia de clase A. ¿Quién debería ser el responsable de la creación de una nueva instancia de alguna clase? (15) Este patrón al igual que los anteriores se puede evidenciar en las clases que contienen las acciones, las cuales son encargadas de instanciar a las clases del modelo que representan las entidades. Su uso también se puede percibir en las clases del modelo, las cuales son generadas automáticamente por cualquiera de los ORM utilizado por el framework, y aunque el mismo es transparente al programador, se puede apreciar cuando se debe crear una clase B que agrega, contiene, registra instancias, o tiene los datos de inicialización que se pasarán a otro objeto A. El caso típico es cuando se desea guardar una tupla en una tabla de la base de datos que tiene relación de uno a muchos con otra tabla, la cual es mapeada como una relación de agregación, donde solo es necesario crear un solo objeto específicamente de la clase de uno y esta será la responsable de crear la de muchos.
- **Controlador:** Asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase. ¿Quién debe ser el responsable de gestionar un evento de entrada al sistema? (15) Este patrón facilita la centralización de actividades, no porque las realice sino porque las delega

## Capítulo 3: Análisis y diseño del Sistema

---

en otras clases con las que mantiene un modelo de alta cohesión. Todas las peticiones web en el framework son manejadas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación. El controlador frontal es imprescindible en Symfony, se encarga de numerosas tareas entre las que se encuentran: crear la configuración de la aplicación y el contexto; cargar e inicializar las clases del núcleo del framework; decodificar la URL de la petición para determinar la acción a ejecutar y los parámetros de la petición; ejecutar los filtros; ejecutar la acción y producir la vista. Las acciones también hacen uso de este patrón conteniendo la lógica de la aplicación; verifican además la integridad de las peticiones y preparan los datos requeridos por la vista utilizando el modelo y definiendo las variables para la vista, por lo que se puede decir que son el corazón de cada aplicación. Este patrón se encuentra ejemplificado en las clases `sfFrontController`, `sfActions`, así como todos los archivos `action.class.php` que son creados en cada uno de los módulos de la aplicación.

- **Experto:** Asignar una responsabilidad al experto en información -la clase que tiene la información necesaria para realizar la responsabilidad-. ¿Cuál es el principio general para asignar responsabilidades a los objetos? (15) Su uso se encuentra enmarcado fuertemente en el mapeo y abstracción de la base de datos, se puede reflejar siempre que se generan de forma automática ambas capas del modelo, es uno de los patrones más utilizados, debido a que tanto Propel como Doctrine, generan las clases de acceso a datos asignándoles las responsabilidades de ejecutar todas las funcionalidades comunes de las entidades que representan y de la cual poseen información. Es válido mencionar que este patrón suele ser usado por el programador, pero es el propio framework quien se encarga de implementarlo a través de los dos ORM de los que se puede hacer uso.

### Patrones GOF

Los patrones GOF se clasifican en tres grupos. Los estructurales tratan la combinación de clases su relación y la formación de estructura 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.

A continuación se describen algunos de los patrones GOF que están implementado en el framework de Symfony.

## Capítulo 3: Análisis y diseño del Sistema

---

- **Fábrica Abstracta (Abstract Factory):** Patrón creacional a nivel de objetos. Su propósito es proporcionar una interfaz para la creación de familias de objetos interdependientes o interrelacionados, sin especificar sus clases concretas. (17) Su aplicación se puede evidenciar en la clase `sfContext` utilizada cuando el framework necesita crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea, de esta forma se simplifica la decisión abstrayéndola dentro de una clase especializada.
- **Solitario (Singleton):** Patrón creacional a nivel de objetos. Su propósito es garantizar que una clase sólo tenga una única instancia, proporcionando un punto de acceso global a la misma. (17) Este patrón es utilizado en la clase `sfContext` la cual almacena una referencia a todos los objetos que forman el núcleo de Symfony. Debido que se encarga de enrutar todas las peticiones que se realizan a la aplicación.
- **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. (17) Se evidencia en la clase `sfConfig`, la cual crea un objeto que proporciona métodos estáticos para poder acceder a los parámetros de configuración desde cualquier punto de la aplicación.
- **Decorador (Decorator):** Patrón estructural a nivel de objetos. Su propósito es añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. (17) Es aplicado a las vistas donde el contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. De manera que el layout contiene el código HTML y los elementos comunes a todas las páginas, mientras las plantillas se encargan de mostrar el resultado de las acciones. Las clases `sfView`, `sfFilter`, y `sfWidgetFormSchemaDecorator` implementa dicho patrón.
- **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. (17) Su aplicación se evidencia en la clase

# Capítulo 3: Análisis y diseño del Sistema

---

sfFrontWebController esta clase es la encargada de determinar cual módulo y acción debe responder a las solicitudes de los usuarios. Encapsula las peticiones en forma de objetos permitiendo así parametrizar los clientes utilizando distintas peticiones, encolar las peticiones y ofrecer la posibilidad de deshacer las operaciones.

- **Cadena de Responsabilidades (Chain of Responsibility):** Patrón de comportamiento a nivel de objetos. Su propósito es proporcionar a más de un objeto la capacidad de atender una petición, para así evitar el acoplamiento con el objeto que hace la petición. Se forma con estos objetos una cadena, en la cual cada objeto satisface la petición o la pasa al siguiente. Su aplicación se encuentra reflejada en la clase sfEventDispatcher, la cual rige el funcionamiento casi completo del framework (17).

## 3.4 Modelo de Diseño

El modelo de diseño es una abstracción de la implementación del sistema. Es un producto de trabajo integral y compuesto que contiene los artefactos, clase de diseño, interfaz, paquete de diseño, subsistema de diseño, sucesos, señal, clase de comprobabilidad, ejecución de guiones de uso, operación, realización de una operación, y componente de servicio (16).

### 3.4.1 Diagrama de clase del Diseño

Después de realizar un minucioso estudio del framework de Symfony se decidió representar varios elementos que son de suma importancia ya que el programador se va relacionar con ellos para la implementación de las diferentes funcionalidades del sistema.

- **Modelo, Vista y Controlador:** representan paquetes lógicos en dependencia con la arquitectura del framework de desarrollo.
- **Propel:** es el ORM seleccionado para la implementación del sistema de recomendación.
- **[Nombre]Actions:** representa las clases que contienen las acciones de un modulo determinado las cuales controlan las funcionalidades del modulo.
- **[Nombre]Success:** es donde se presenta el contenido

## Capítulo 3: Análisis y diseño del Sistema

---

- **Layout:** decora las páginas de presentación de contenido, estas páginas deben ser decoradas por un único elemento de este tipo.
- **Index:** representa el control frontal, es creado de forma automática para toda la aplicación es de suma importancia para el framework ya que todo el flujo de trabajo pasa a través de él.
- **[Nombre]Peer, [Nombre], Base [Nombre]:** son las clases de acceso a datos, generadas automáticamente por el ORM Propel.

A continuación se describen los estereotipos en la Tabla 6 utilizados para las relaciones de clases y los paquetes anteriormente mencionados y luego se muestran los diagramas de la clase del diseño.

Estereotipo	Descripción
<<instantiate>>	Especifica que hay operaciones en la clase origen que crean instancias de la clase destino.
<<use>>	Especifica que la semántica del elemento origen depende de la semántica de la parte pública del destino.
<<build>>	Relaciona una página cliente con la del servidor, se expresa como que las páginas que se encuentran en el servidor construyen las páginas en el cliente.
<<submit>>	Relaciona un formulario y una página del servidor, donde el primero manda los valores de sus campos al servidor, para ser procesados por la página servidor.

Tabla 6 Estereotipos

# Capítulo 3: Análisis y diseño del Sistema

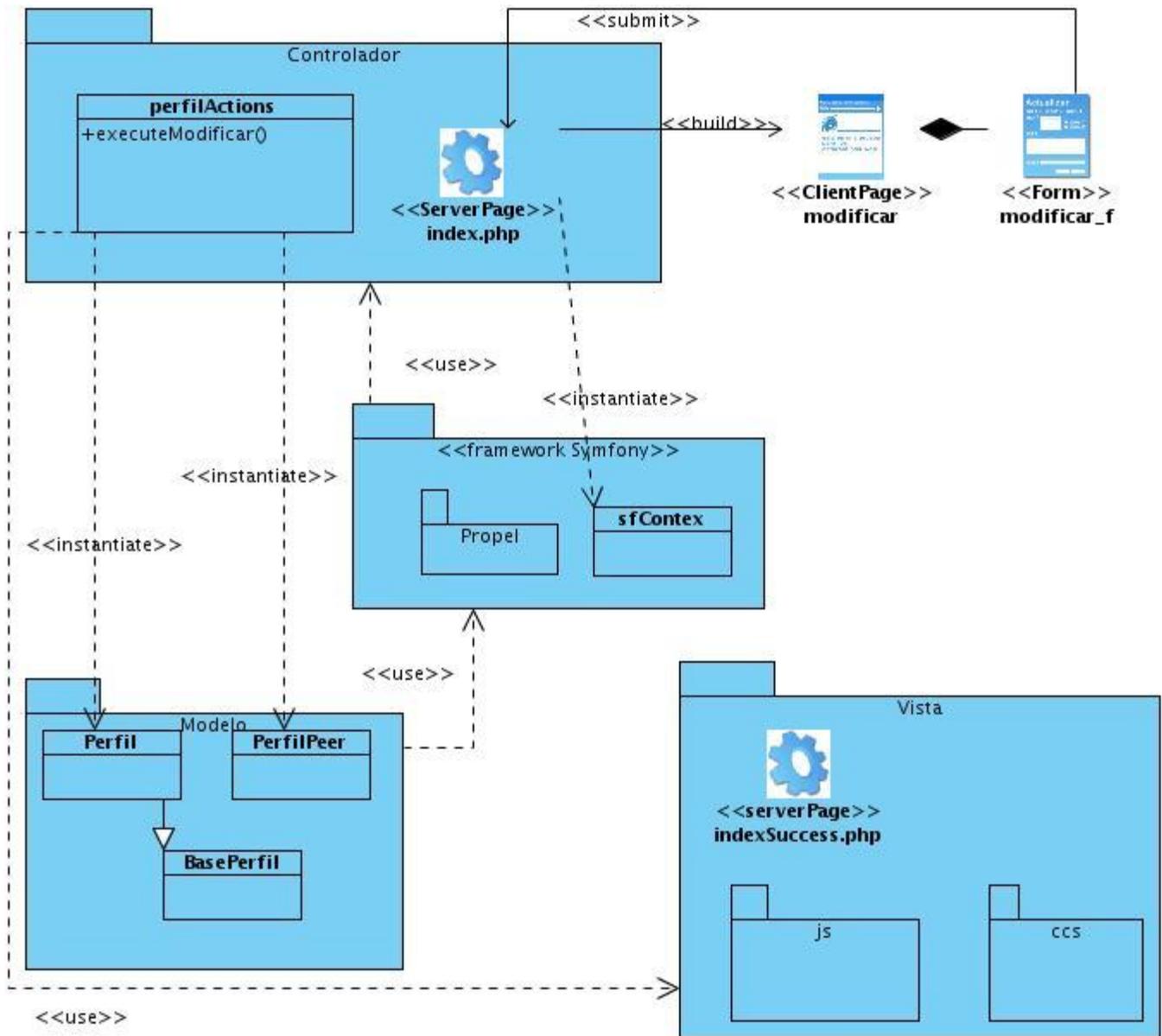


Figura 11 Diagrama de Clases de Diseño para el CU Modificar Perfil

# Capítulo 3: Análisis y diseño del Sistema

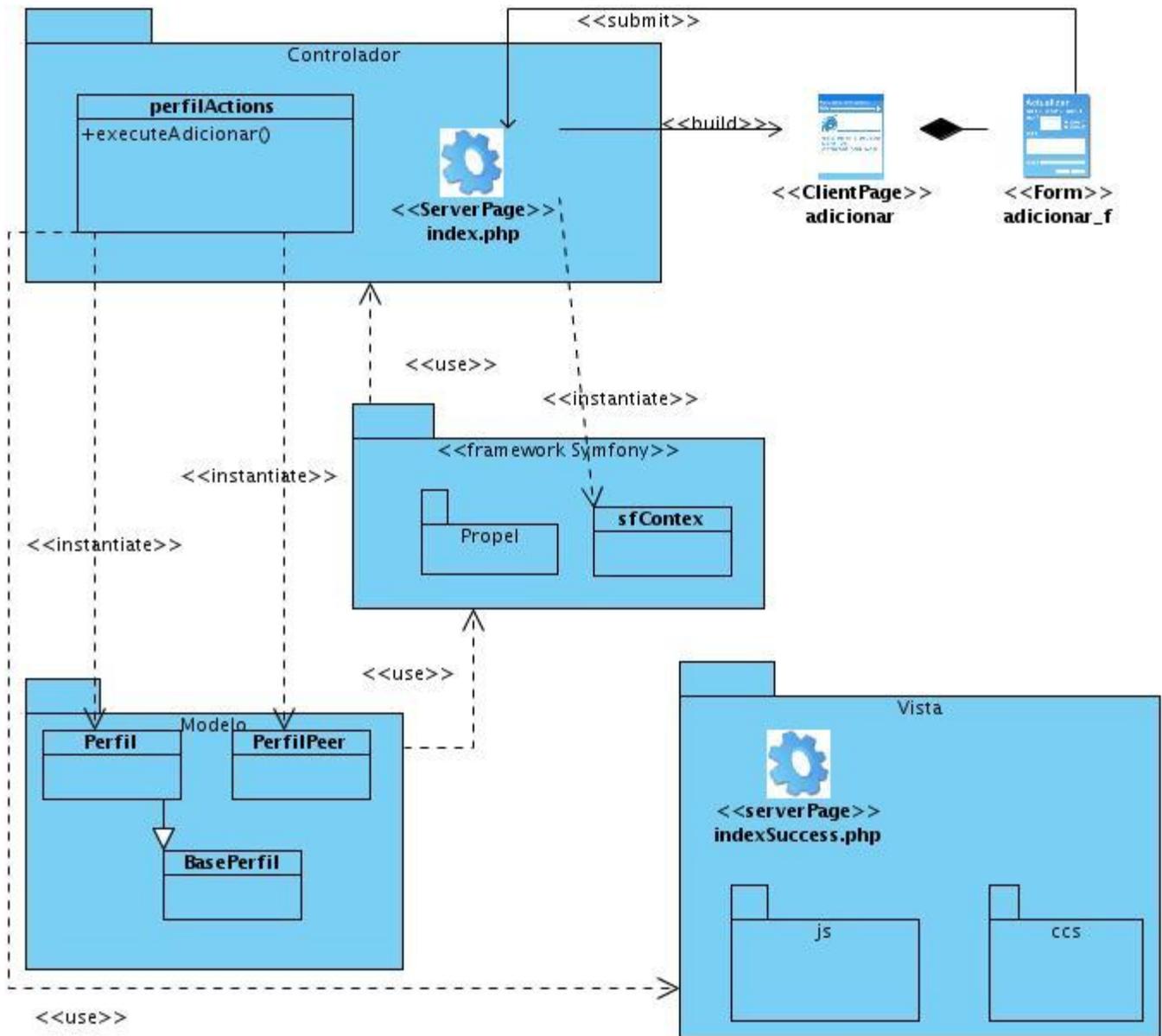


Figura 12 Diagrama Clases de Diseño para el CU Crear Perfil

# Capítulo 3: Análisis y diseño del Sistema

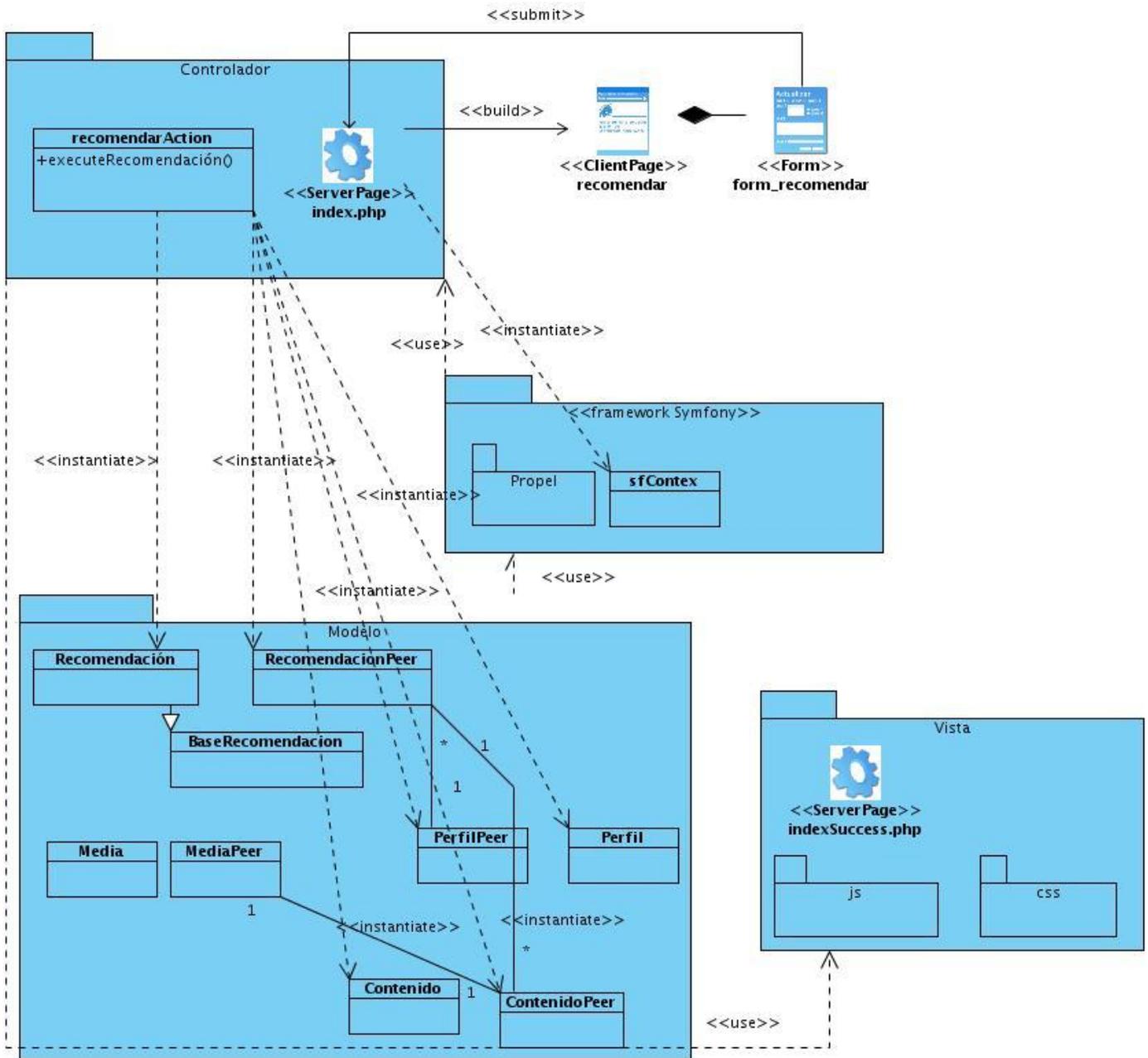


Figura 13 Diagrama de Clases del Diseño para el CU Pedir Recomendación

# Capítulo 3: Análisis y diseño del Sistema

---

## 3.6 Conclusiones

En este capítulo se generaron los artefactos modelo de análisis y de diseño los que dieron paso al modelo de implementación general basado en componentes del sistema de recomendación propuesto para el subsistema Web de PTARTV. También se mostraron las características principales de la arquitectura y los patrones de diseño framework de desarrollo Symfony.

# Capítulo 4: Validación de la Solución Propuesta

---

## Capítulo 4: Validación de la Solución Propuesta

### 4.1 Introducción

En el presente capítulo se describe el modelo de implementación, el cual está compuesto por el diagrama de componentes y de despliegue, para lograr esto se partió de los resultados obtenidos en el modelo de diseño del capítulo anterior, también se realizan las pruebas funcionales.

### 4.2 Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros. Seguidamente se muestra una vista general del modelo de implementación del sistema de recomendación propuesto para el subsistema Web de la plataforma PTARTV.

# Capítulo 4: Validación de la Solución Propuesta

---

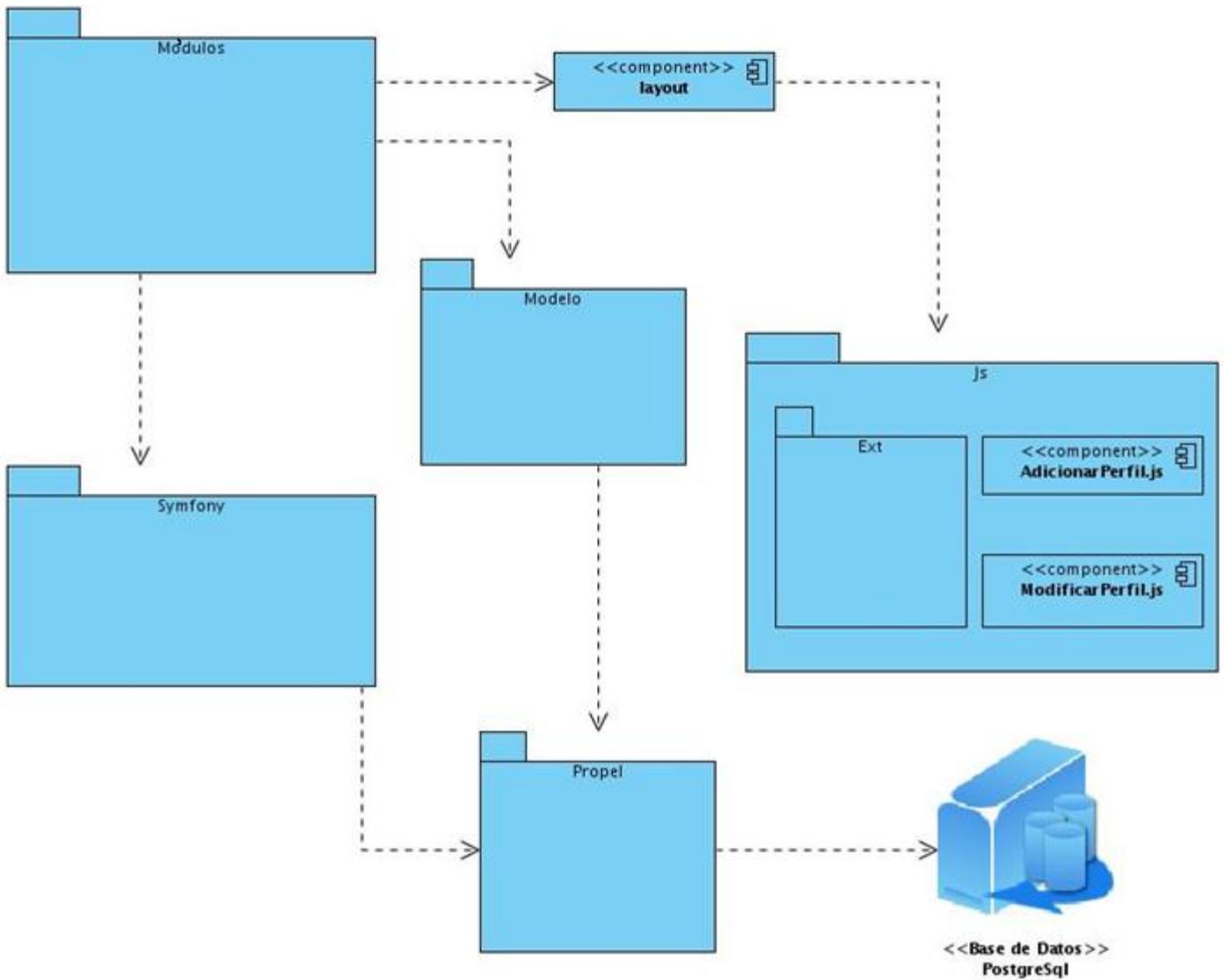


Figura 14 Diagrama de Componentes. Vista General

A continuación se muestra cada módulo del diagrama de implementación de una forma más detallada con el propósito de una mejor comprensión.

# Capítulo 4: Validación de la Solución Propuesta

---

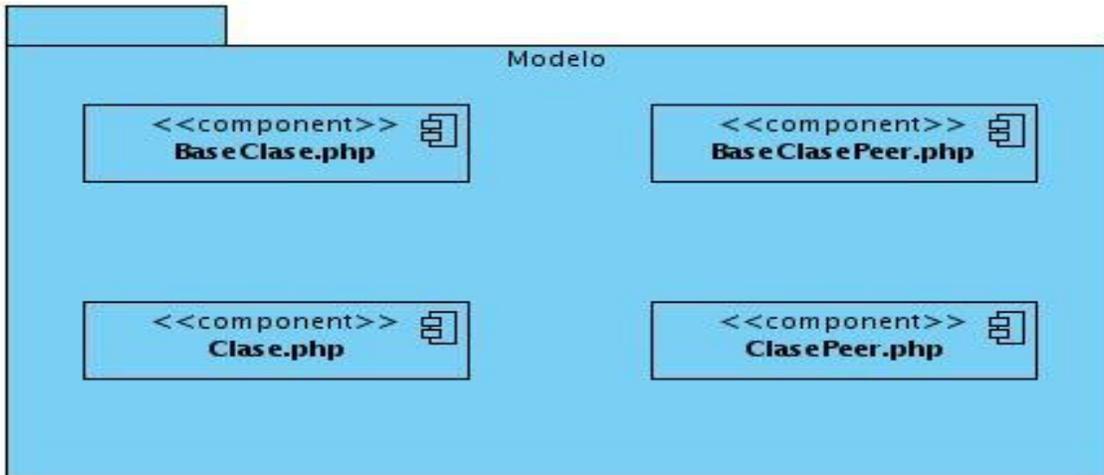


Figura 15 Diagrama de Componente. Paquete Modelo

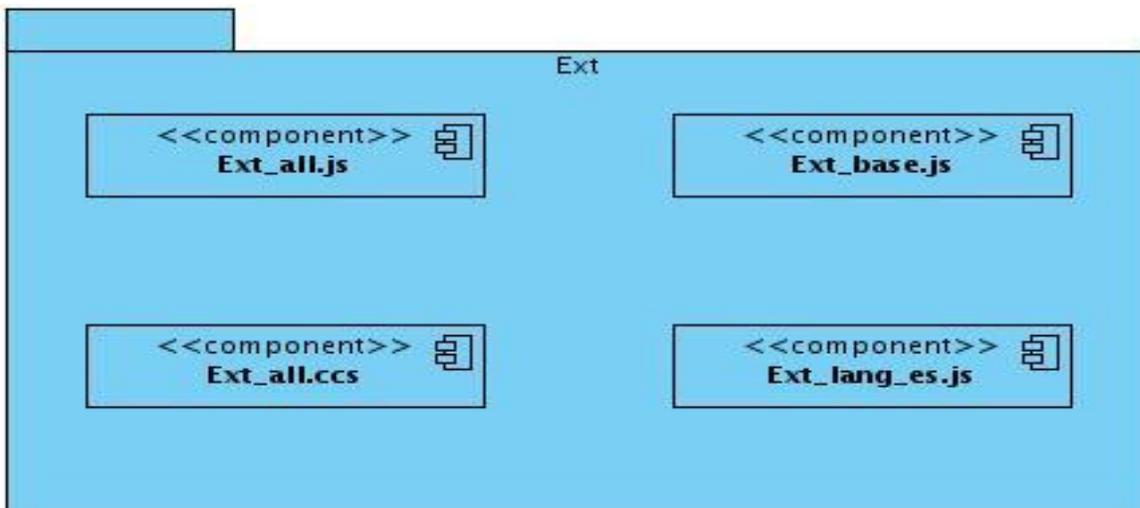


Figura 16 Diagrama de Componente. Paquete Ext.

# Capítulo 4: Validación de la Solución Propuesta

---

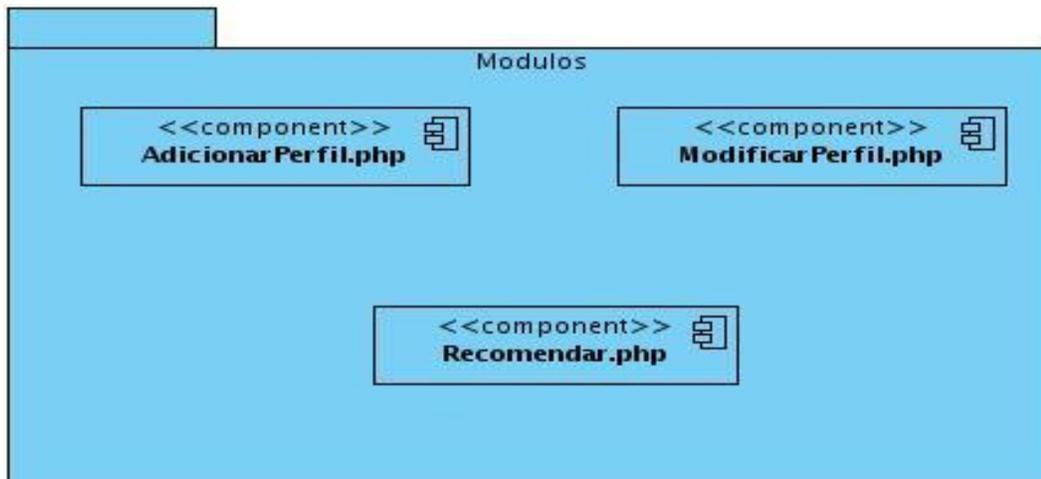


Figura 17 Diagrama de Componente. Paquete Módulos.

## 4.3 Modelo de Despliegue

El Modelo de Despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad en los nodos de cómputo. Cada nodo representa un recurso de cómputo y poseen relaciones que representan medios de comunicación entre ellos, además la funcionalidad de un nodo se definen por los componentes que se distribuyen en el mismo.

El Modelo de Despliegue del sistema de recomendación propuesto en la presente investigación se muestra a continuación, el cual consta de una PC cliente donde el usuario realiza las peticiones de recomendación a través del protocolo de comunicación HTTP (HyperText Transfer Protocol) y estas peticiones van a ser respondidas por el Servidor Web el cual se comunica a través de ADO (ActiveX Data Objects) con el Servidor de Base de Datos, encargado de manejar toda la información referente al usuario, los perfiles de usuarios y contenidos de medias.

# Capítulo 4: Validación de la Solución Propuesta

---

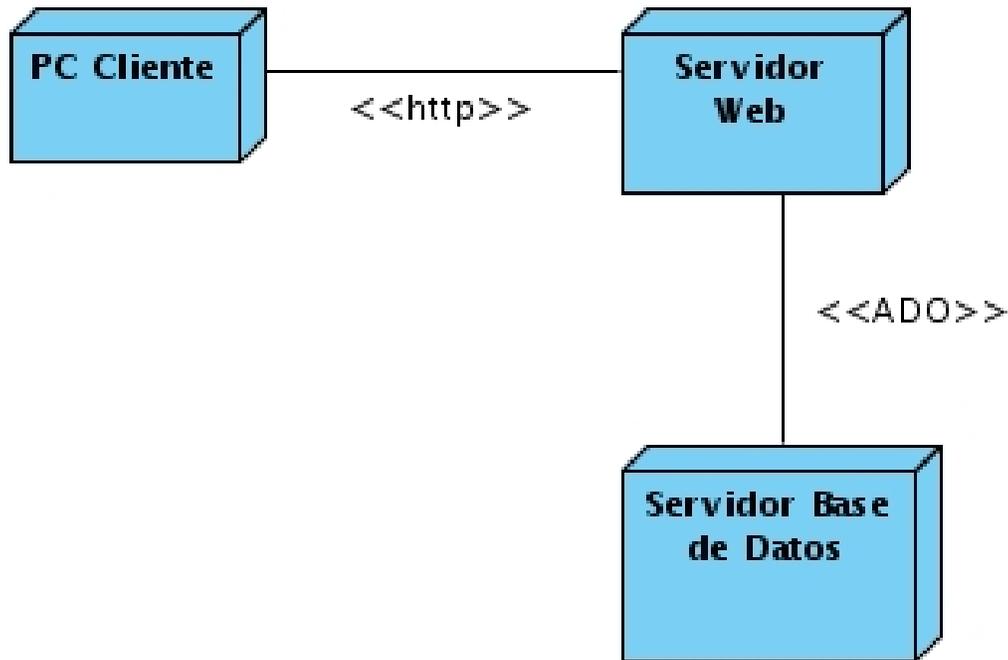


Figura 18 Modelo de Despliegue

## 4.4 Pruebas

Unos de los aspectos más importantes en el desarrollo de software son las pruebas ya que verifican el perfecto funcionamiento del mismo. En la presente investigación se seleccionaron las Pruebas Funcionales ya que estas pruebas se centran en las entradas, salidas de datos y funciones. Su objetivo principal es asegurar el correcto funcionamiento de los requisitos funcionales. Según Pressman estas pruebas pueden realizarse en base a dos enfoques fundamentales (Pressman 1998).

- **Prueba de Caja Blanca:** En la prueba de caja blanca, se analiza la estructura lógica del programa y para cada alternativa que pueda presentarse, los datos de prueba ideados conducirán a ella.
- **Prueba de Caja Negra:** En la prueba de la caja negra, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta.

## Capítulo 4: Validación de la Solución Propuesta

---

Luego de haberse analizado el tipo de aplicación y la composición de la misma, se escogió de las pruebas antes enunciadas, la Prueba de Caja Negra ya que en la presente investigación el sistema de recomendación que se propone para el subsistema Web de la plataforma PTARTV deber de ser una aplicación web que estará estructurada por más de un módulo, por lo que no es conveniente aplicar la Prueba de Caja Blanca.

### Descripción general del caso de uso Crear Perfil de Usuario.

Comienza cuando el usuario accede al sistema para crearse un perfil.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC1: Crear Perfil de Usuario.	EC 1.1 Perfil creado con éxito	El usuario accede al sistema para crearse un perfil, el sistema muestra una interfaz donde el usuario llena los datos del perfil y luego presiona el botón crear perfil.
	EC 1.2 Perfil creado falla.	El sistema emite un mensaje de error cuando el usuario deja campos vacíos o introduce valores incorrectos.

Tabla 7 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Crear Perfil de Usuario.

### Descripción general del caso de uso Modificar Perfil de Usuario.

Comienza cuando el usuario accede al sistema para modificar su perfil.

## Capítulo 4: Validación de la Solución Propuesta

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC1: Modificar Perfil de Usuario.	EC 1.1 Perfil Modificado con éxito	El usuario accede al sistema para modificar su perfil, el sistema muestra una interfaz donde el usuario modifica los datos del perfil y luego presiona el botón modificar perfil.
	EC 1.2 Perfil Modificado falla.	El sistema emite un mensaje de error cuando el usuario deja campos vacíos o introduce valores incorrectos.

Tabla 8 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Modificar Perfil de Usuario.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	En este campo solo letras
2	Apellidos	Campo de texto	No	En este campo solo letras
3	Edad	Campo de texto	No	En este campo solo número
4	Gustos	Campo de texto	No	En este campo solo letras
5	Actores	Lista desplegable	Si	Permite seleccionar un actor
6	Directores	Lista desplegable	Si	Permite seleccionar un director
7	Ingredientes	Lista desplegable	No	Permite seleccionar un ingrediente y especificar un valor numérico.
8	Productores	Lista desplegable	Si	Permite seleccionar un productor

## Capítulo 4: Validación de la Solución Propuesta

9	Crear	Botón	No	Permite crear un perfil
10	Modificar	Botón	No	Permite modificar un perfil
11	Número de Recomendación	Campo de Número	No	Permite Especificar el número máximo de recomendaciones

**Tabla 9 Descripción de las Variables de los casos de usos Crear Perfil de Usuario, Modificar Perfil de Usuario y Pedir Recomendación.**

Escenario	Variable 1 Nombre	Variable2 Apellidos	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC1.1 :Crear Perfil de Usuario con éxito	(Elvis) V	(Ferrera)V	El sistema muestra una interfaz donde se introducen los datos del Perfil de Usuario.	Satisfactorio	1- El usuario accede al sistema para crease un perfil. 2- El sistema muestra una interfaz para que el usuario introduzca los datos necesarios. 3- Luego de haber introducidos los datos presiona el botón crear perfil.
EC1.2: Crear Perfil de Usuario falla	(11Evis) I	(Ferrera)V	El sistema muestra un mensaje de error.	Satisfactorio	
	(nada) I	V	El sistema muestra un mensaje de error.	Satisfactorio	

**Tabla 10 Matriz de datos Caso de Uso Crear Perfil de Usuario.**

## Capítulo 4: Validación de la Solución Propuesta

Escenario	Variable 10 Nombre	Variable2 Apellidos	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC1.1: Modificar Perfil de Usuario con éxito	(Elvis) V	(Ferrera)V	El sistema muestra una interfaz donde se introducen los datos del Perfil de Usuario.	Satisfactorio	1- El usuario accede al sistema para crease un perfil. 2- El sistema muestra una interfaz para que el usuario modifique los datos necesarios. 3- Luego de haber introducidos los datos presiona el botón modificar perfil.
EC1.2: Modificar Perfil de Usuario falla	(11Evis)I	(Ferrera)V	El sistema muestra un mensaje de error.	Satisfactorio	
	(nada)I	V	El sistema muestra un mensaje de error.	Satisfactorio	

Tabla 11 Matriz de datos Caso de Uso Modificar Perfil de Usuario.

### Descripción general del caso de uso Modificar Perfil de Usuario.

Comienza cuando el usuario accede al sistema para pedir una recomendación.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
----------------------	-------------------------	---------------------------------

## Capítulo 4: Validación de la Solución Propuesta

SC1: Pedir Recomendación	EC 1.1 Pedir Recomendación con éxito.	El usuario accede al sistema para ser recomendado. El sistema muestra una interfaz y el usuario introduce que cantidad de recomendaciones quiere, presiona el botón recomendar y luego el sistema muestra las recomendaciones.
	EC 1.2 Pedir Recomendación falla.	El sistema emite un mensaje de error cuando el usuario deja el campo vacío o introduce un número fuera del rango especificado.

Tabla 12 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Pedir Recomendación.

Escenario	Variable 10 Número de recomendación	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC1.1: Pedir Recomendación con éxito	(4) V	El sistema muestra una interfaz donde se introducen el número de recomendación.	Satisfactorio	1- El usuario accede al sistema para pedir una recomendación. 2- El sistema muestra una interfaz para que el usuario introduzca el número de recomendación que quiere recibir. 3- Luego de haber introducido el número presiona el botón recomendación.
EC1.2: Pedir recomendación falla	(100)I	El sistema muestra un mensaje de error indicando que el número está fuera de rango.	Satisfactorio	
	(nada)I	El sistema muestra un	Satisfactorio	

# Capítulo 4: Validación de la Solución Propuesta

---

		mensaje de error indicando que faltan datos.		
--	--	--	--	--

Tabla 13 Matriz de datos Caso de Uso Pedir Recomendación.

## 4.5 Resultados de las pruebas

Después de realizadas las pruebas al sistema, se obtuvieron resultados satisfactorios ya que se comprobó que las respuestas del mismo fueron la esperadas y cumplían con las especificaciones planteadas, coincidiendo con las descripciones de los casos de usos anteriormente expuestas. Con las pruebas aplicadas se comprobó también que el sistema de recomendación cumple con los objetivos propuestos en la presente investigación.

## 4.6 Conclusiones

En el presente capítulo se realizó la descripción de la construcción de la solución propuesta del sistema de recomendación para el subsistema Web de la plataforma PTARTV, donde se desarrollaron los principales artefactos a llevar a cabo en el proceso de implementación del sistema. También se mostró el Diagrama de Despliegue el cual describe la distribución física y lógica de la arquitectura del sistema y sus conexiones. La elaboración del Modelo de componentes permitió determinar los componentes de la aplicación y la relación entre ellos. Por último mediante los casos de prueba se validó el software verificando su perfecto funcionamiento comprobándose que no existían errores.

# Conclusiones Generales

---

## Conclusiones Generales

Los sistemas de recomendación han avanzado significativamente durante los últimos años, en los que se han podido observar varios sistemas que utilizan diferentes metodologías. No obstante, la actual generación de sistemas de recomendación necesita mejorar varios aspectos para lograr recomendaciones efectivas enfocadas a los usuarios.

Durante el desarrollo de esta investigación se logró el objetivo planteado en el cual se proponía desarrollar un sistema de recomendación enfocado a las preferencias de los usuarios para la plataforma PTARTV. Con el mismo se logró que dicha plataforma adquiriera nuevas funcionalidades enfocada a los gustos de los usuarios.

A pesar de que la aplicación fue desarrollada con herramientas totalmente libres, no se puede integrar a otras plataformas ya que fue concebida específicamente para la Plataforma de Transmisión Abierta para Radio y Televisión, no obstante su método de construcción es aplicable a cualquier sistema de recomendación que se desee desarrollar e inclusive los algoritmos utilizados.

Se propuso una medida de similitud llamada Coeficiente de Beltin, la cual mejora considerablemente los resultados de las recomendaciones dadas al usuario implicado en dicho proceso. Estos resultados son de vital importancia para los usuarios porque se deben adaptar con gran precisión a sus preferencias y gustos.

En Cuba la existencia de los sistemas de recomendación es casi nula, por lo que el presente trabajo científico aporta grandemente en el ámbito social ya que en estos tiempos la cantidad de información es extremadamente excesiva.

# Recomendaciones

---

## Recomendaciones

Luego de haberse terminado el desarrollo del sistema de recomendación propuesto y cumplido cabalmente con los objetivos trazados se exponen las siguientes recomendaciones:

- Continuar la implementación de nuevos algoritmos con el objetivo de obtener recomendaciones más precisas para los usuarios.
- Someter el sistema a pruebas de calidad.
- Pedir opinión de los usuarios acerca de sus recomendaciones.

# Bibliografía

---

1. click2check.com. *click2check.com*. [En línea] 4 de Junio de 2007. [Citado el: 18 de Junio de 2011.] [click2check.wordpress.com/.../¿que-cantidad-de-informacion-tiene-internet/](http://click2check.wordpress.com/.../¿que-cantidad-de-informacion-tiene-internet/) .
2. **Font, Miguel Seguido y Farre, Carles**. Upcommons. *Upcommons*. [En línea] 2002-2004. [Citado el: 19 de Junio de 2011.] [upcommons.upc.edu/pfc/...1/.../Master%20Thesis%20Seguido.pdf](http://upcommons.upc.edu/pfc/...1/.../Master%20Thesis%20Seguido.pdf).
3. **Codina, Victor**. Upcommons. *Upcommons*. [En línea] 2002-2004. [Citado el: 19 de Junio de 2011.] [upcommons.upc.edu/pfc/...1/.../Memoria%20PFC%20Victor%20Codina.pdf](http://upcommons.upc.edu/pfc/...1/.../Memoria%20PFC%20Victor%20Codina.pdf).
4. **Salazar, Maria Elena Rodríguez, Álvarez Hernández, Sergio y Bravo Núñez, Ernesto**. *Coeficientes de Asociacion*. Mexico : Plaza y Valdez, 2001.
5. Interactive and Cooperative Technologies Lab . *Interactive and Cooperative Technologies Lab* . [En línea] 1997. [Citado el: 19 de Junio de 2011.] [ict.udlap.mx/people/lulu/.../capitulo4.html](http://ict.udlap.mx/people/lulu/.../capitulo4.html) .
6. **Sanchez, María A. Mendoza**. *Metodologías de Desarrollo de Software*. Perú : S.A.C , 2004.
7. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*. 2002.
8. rolandojaldin. *rolandojaldin*. [En línea] [rolandojaldin.blogspot.com](http://rolandojaldin.blogspot.com).
9. **Schmuller, Joseph**. *Aprendiendo UML en 24 Horas* . Mexico : Pearson Education, 2000.
10. otroblogmas. *otroblogmas*. [En línea] [otroblogmas.com](http://otroblogmas.com).
11. **Paradigm, Visual**. *Visual Paradigm for the Unified Modeling Language: VP-UML 6.0 User's Guide*. 2007.
12. downloadsource. *downloadsource*. [En línea] [downloadsource.net](http://downloadsource.net).
13. **Fabien, Potencier**. *Symfony, la Guía Definitiva* . 2007.
14. **Lobo, Armando Robert**. *Tutorial Básico de ExtJS*. Ciudad de La Habana : s.n., 2006.
15. **Larman, Craig**. *UML Y PATRONES*. Mexico : s.n., 1999. s.n.
16. Ayuda del Rational(Español). *Ayuda del Rational(Español)*. [En línea] IBM Corp, 2006.
17. **Madrid, Facultad de informática - Universidad Politecnica de Madrid**. *Patrones "Gang of Four"* . s.l. :Unidad Docente de Ingeniería del Software.