

Títol: Sistema de recomendación personalizada de contenido video

Volum: 1

Alumne: Victor Codina Busquet

Director/Ponent: Luigi Ceccaroni

Departament: LSI

Data: 23 de gener del 2008

DADES DEL PROJECTE

Títol del Projecte: Sistema de recomendación personalizada de contenido vídeo

Nom de l'estudiant: Victor Codina Busquet

Titulació: Enginyeria Informàtica

Crèdits: 37,5

Director/Ponent: Luigi Ceccaroni

Departament: Llenguatges i Sistemes Informàtics (LSI)

MEMBRES DEL TRIBUNAL

President: Javier Béjar Alonso

Vocal: Angel Olivé Duran

Secretari: Luis José Talavera Méndez

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

Agradecimientos

En primer lugar agradezco a TMT Factory el haberme dado la posibilidad de realizar mi Proyecto Final de Carrera en su empresa, y ofrecerme el soporte necesario para llevarlo a cabo. En especial quiero agradecer a Luigi Ceccaroni, el director del proyecto y actualmente mí superior como trabajador de la empresa, la atención y dedicación con la que ha dirigido el proyecto, además de ser mi guía en los momentos que no sabía cómo avanzar. Tampoco puedo olvidarme de los compañeros del trabajo que me han ayudado siempre que lo he necesitado durante la realización del proyecto.

Tengo que agradecer muy especialmente a mis padres, el esfuerzo moral y económico que han llevado a cabo a lo largo de toda la carrera y el apoyo que me han dado siempre que lo he necesitado. A mi novia Gemma por haber estado a mi lado y darme ánimos en los momentos difíciles, así como a los amigos con los que he compartido buenos momentos durante todo este tiempo.

Índice

Resumen Ejecutivo	13
Capítulo 1	15
1 Introducción	15
1.1 La Empresa	16
1.2 Motivaciones	16
1.3 Objetivos del proyecto	17
1.4 Estado del arte	18
1.4.1 Clasificación	18
1.4.2 Ejemplos de sistemas de recomendación	21
1.5 Estructura de la memoria	22
Capítulo 2	23
2 Estado actual de IntegraTV Home	23
2.1 Componentes del sistema	23
2.1.1 FrontEnd	26
2.1.2 BackOffice	28
2.2 Distribución de contenidos	29
2.3 Tecnologías utilizadas	32
2.3.1 Almacenamiento	32
2.3.2 Comunicaciones	32
Capítulo 3	33
3 Sistema de Recomendación	33
3.1 Ontología utilizada	33
3.1.1 Perfil de Usuario	33
3.1.1.1 Información de preferencias	33
3.1.1.2 Información de usuario personal	35
3.1.2 Contenido de vídeo	35
3.2 Descripción del sistema de recomendación	36
3.2.1 Componente basado en el contenido	39
3.2.1.1 Aspectos teóricos	39
3.2.1.2 Caso 1	40
3.2.1.3 Caso 2	41
3.2.2 Componente basado en el historial	42

3.2.2.1	Caso 1	42
3.2.2.2	Caso 2	44
3.2.3	Componente basado en el modelo bayesiano	45
3.2.3.1	Aspectos teóricos	45
3.2.3.2	Caso 1	46
3.2.3.3	Caso 2	47
3.2.4	Componente de combinación	48
3.2.4.1	Aspectos teóricos	48
3.2.4.2	Caso 1	49
3.2.4.3	Caso 2	50
3.3	Evaluación del sistema de recomendación	51
3.3.1	Metodología	52
3.3.2	Conclusiones de la evaluación	53
Capítulo 4	56
4	Desarrollo del sistema de información	56
4.1	Metodología empleada: Métrica 3	56
4.2	Análisis del Sistema de Información	58
4.2.1	Definición del sistema	58
4.2.1.1	Alcance del sistema	58
4.2.1.2	Entorno tecnológico para el desarrollo	58
4.2.2	Modelo de Requisitos	59
4.2.2.1	Especificación de los requisitos	59
4.2.2.2	Identificación de actores	62
4.2.2.3	Identificación de casos de uso	63
4.2.2.4	Especificación de casos de uso	66
4.2.3	Modelo Conceptual	71
4.2.4	Análisis de los casos de uso	73
4.2.4.1	Caso de uso: Alta de contenido	74
4.2.4.2	Caso de uso: Activar perfil de usuario	75
4.2.4.2.1	Extensión: Crear perfil de usuario	75
4.2.4.3	Caso de uso: Pedir recomendación de usuario	77
4.2.5	Definición de interfaces de usuario	79
4.2.5.1	Identificación de perfiles	79
4.2.5.2	Especificación de la interfaz de usuario	79

4.2.6	Especificación del plan de pruebas.....	81
4.2.6.1	Alcance de pruebas	81
4.2.6.1.1	Pruebas unitarias	81
4.2.6.1.2	Pruebas de integración	82
4.2.6.2	Pruebas de aceptación del sistema	82
4.3	Diseño del Sistema de Información.....	83
4.3.1	Definición de la arquitectura del sistema	83
4.3.1.1	Requisitos de diseño.....	83
4.3.1.2	Definición de los niveles de la arquitectura	83
4.3.1.3	Identificación de subsistemas de diseño	85
4.3.1.4	Requisitos del entorno tecnológico	87
4.3.2	Diseño de la Arquitectura de Soporte.....	90
4.3.2.1	Tecnologías de soporte.....	90
4.3.2.1.1	Net Remoting.....	90
4.3.2.1.2	Soporte multihilo de .Net.....	93
4.3.2.2	Mecanismos genéricos de diseño	95
4.3.3	Diseño de la realización de los casos de uso	98
4.3.3.1	Caso de uso: Alta de contenido	99
4.3.3.2	Caso de uso: Activar perfil de usuario	102
4.3.3.2.1	Extensión: Crear perfil de usuario	105
4.3.3.3	Caso de uso: Pedir recomendación de usuario	108
4.3.4	Diseño de la interfaz de usuario	118
4.3.4.1	Subsistema IU Cliente.....	118
4.3.4.2	Subsistema IU Admin.Contenidos	125
4.3.5	Diseño de la capa de datos.....	128
4.3.5.1	Subsistema de persistencia.....	128
4.3.5.2	Modelo físico de datos	130
Capítulo 5	135
5	Planificación	135
5.1	Planificación temporal.....	135
5.2	Alcance económico	139
Capítulo 6	141
6	Conclusiones.....	141
6.1	Resultados obtenidos	142

6.2	Mejoras futuras	142
6.3	Limitaciones	143
6.4	Valoración personal.....	143
Bibliografía.....		145
ANEXOS		147
I.	Glosario de términos	147
II.	Lógica de los parámetros de configuración	149
III.	Documentación de la solicitud de patente	155

Índice de Ilustraciones

Ilustración 1:	Organigrama de la aplicación IntegraTV Home	24
Ilustración 2:	Menú principal del FrontEnd.	26
Ilustración 3:	Menú principal del servicio de videoclub	27
Ilustración 4:	Pantalla de identificación de usuario del BackOffice	28
Ilustración 5:	Pantalla de modificación de datos de un contenido de vídeo.....	29
Ilustración 6:	Arquitectura física del sistema de IntegraTV Home.....	31
Ilustración 7:	Diagrama de interacción de los componentes de recomendación en el caso 1.	37
Ilustración 8:	Diagrama de interacción de los componentes de recomendación en el caso 2.	38
Ilustración 9:	Diagrama de casos de uso del Servicio de Recomendación.....	64
Ilustración 10:	Modelo conceptual del Servicio de Recomendación.	72
Ilustración 11:	Diagrama de secuencia del flujo básico de caso de uso AltaContenido	74
Ilustración 12:	Diagrama de secuencia del flujo básico del caso de uso ActivarPerfil junto con la ejecución de la extensión del caso de uso CrearPerfil.....	76
Ilustración 13:	Diagrama de secuencia del flujo básico del caso de uso PedirRecomendación..	78
Ilustración 14:	Diagrama de despliegue de la arquitectura física del sistema.....	85
Ilustración 15:	Diagrama de paquetes con las dependencias entre subsistemas.....	87
Ilustración 16:	Equipo de televisión interactiva (Media Center HiGrade P60).....	88
Ilustración 17:	Mando a distancia de IntegraTV.....	88
Ilustración 18:	Arquitectura simplificada del proceso de interacción remota en Net Remoting.	91
Ilustración 19:	Esquema básico de una pool de hilos.....	94
Ilustración 20:	Aspecto estático del patrón de diseño Proxy.....	97
Ilustración 21:	Aspecto estático del patrón de diseño Command.....	98
Ilustración 22:	Aspecto estático del diseño del caso de uso AltaContenido.	100
Ilustración 23:	Aspecto dinámico del diseño del caso de uso AltaContenido.....	101
Ilustración 24:	Aspecto estático del diseño del caso de uso ActivarPerfil.	103
Ilustración 25:	Aspecto dinámico del diseño del caso de uso ActivaPerfil.....	104
Ilustración 26:	Aspecto estático del diseño del caso de uso CrearPerfil.	105

Ilustración 27: Aspecto dinámico del diseño del caso de uso CrearPerfil.	107
Ilustración 28: Aspecto estático del diseño del caso de uso Pedir recomendación de usuario (parte 1).	109
Ilustración 29: Aspecto estático del diseño del caso de uso Pedir recomendación de usuario (parte 2).	110
Ilustración 30: Aspecto dinámico del diseño del caso de uso Pedir recomendación de usuario (operación pedirRecomendación)	111
Ilustración 31: Aspecto dinámico del diseño del caso de uso Pedir recomendación de usuario (operación recomendarUsuario).....	112
Ilustración 32: Aspecto dinámico del diseño del caso de uso Pedir recomendación de usuario (operación asíncrona execute).....	114
Ilustración 33: Aspecto dinámico del diseño del caso de uso Pedir Recomendación de Usuario (operación asíncrona callBack).	116
Ilustración 34: Mapa de navegación de las interfaces de usuario del FrontEnd.....	119
Ilustración 35: Diseño de la interfaz de usuario: GestiónPerfil.....	121
Ilustración 36: Diseño de la interfaz de usuario: CrearPerfil.	121
Ilustración 37: Diseño de la interfaz de usuario: Ingredientes.	122
Ilustración 38: Diseño de la interfaz de usuario: Preferencias.	123
Ilustración 39: Diseño de la interfaz de usuario: InformaciónPersonal.	124
Ilustración 40: Diseño de la interfaz de usuario: PedirRecomendación.....	125
Ilustración 41: Mapa de navegación de las interfaces de usuario del BackOffice.	126
Ilustración 42: Diseño de la interfaz de usuario: DatosContenidoVideo (ficha Ingredientes)..	127
Ilustración 43: Diseño de la interfaz de usuario: DatosContenidoVideo (ficha descriptores). .	128
Ilustración 44: Arquitectura simplificada de Hibernate.	130
Ilustración 45: Esquema de la base de datos del sistema.	132
Ilustración 46: Fichero XML de mapeo a la base de datos de la clase Perfil.....	133
Ilustración 47: Diagrama de Gantt con la planificación inicial del proyecto.	137
Ilustración 48: Diagrama de Gantt con la duración real de las tareas realizadas.	138

Índice de Tablas

Tabla 1: Especificación del caso de uso: Alta de contenido.	67
Tabla 2: Especificación del caso de uso: Activar perfil.	68
Tabla 3: Especificación del caso de uso: Crear perfil.	69
Tabla 4: Especificación del caso de uso: Pedir recomendación de usuario.	71
Tabla 5: Coste del proyecto en cuanto a horas trabajadas por el estudiante.	139
Tabla 6: Coste del software utilizado para el desarrollo.	139
Tabla 7: Coste total del proyecto.....	140

Resumen Ejecutivo

Con este proyecto se quiere mejorar uno de los productos principales de la empresa, que consiste en un sistema de televisión interactiva que permite a los usuarios poder disfrutar de varios servicios tales como una aplicación ofimática, una guía local y la adquisición de contenidos de distintos tipos: películas, series, clips, música, información cultural y videojuegos entre otros.

Dado que no existe ningún servicio de personalización en el sistema y la relevancia que tiene para los usuarios el servicio de adquisición de contenidos de vídeo, se pretende mejorar este servicio añadiéndole personalización. De manera que se quiere desarrollar un sistema de recomendación de contenido de vídeo que funcione de forma complementaria al servicio de adquisición de contenidos. Para ello será necesario diseñar un sistema de recomendación de contenido de vídeo adecuado al entorno del sistema e integrarlo en forma de nuevas funcionalidades del sistema.

Actualmente existen varios sistemas de recomendación de contenido de vídeo que funcionan relativamente bien, pero únicamente en entornos con un gran número de usuarios que continuamente interactúan con el sistema. Esto implica, que los algoritmos de recomendación de contenido de vídeo actuales sólo tengan cabida en el entorno Web, dónde el número de usuarios puede llegar a ser considerable y se dispone de una alta *densidad de información*¹.

Se pretende diseñar un sistema de recomendación en el que la precisión de las recomendaciones no dependa tanto de la densidad de información disponible, sino más bien en la calidad de la información, dado que estas son las características del entorno donde se implantará el sistema.

El entorno se caracteriza por tener: **pocos usuarios**, el coste inicial del servicio de IntegraTV influirá en el número de usuarios ya que sólo estará al alcance de unos pocos; y un **número de contenidos de vídeo reducido**. Pero a diferencia de otros entornos, se dispone de una valiosa información descriptiva de cada uno de los contenidos, gracias a que la empresa dispone de personal experto en descripciones de contenidos.

Dado que el proyecto consiste en extender las funcionalidades de una aplicación existente, el desarrollo de dicha extensión vendrá marcado por el diseño del sistema actual. Se quiere conseguir que las nuevas funcionalidades propias del servicio de recomendación no afecten al funcionamiento actual del sistema, por lo tanto, se desarrollará en la medida de lo posible como un sistema independiente. Así, el servicio de recomendación desarrollado en este proyecto será fácilmente adaptable a otros productos de la empresa que requieran personalización de contenidos de vídeo.

¹ La densidad de información es la cantidad de contenidos consumidos y valorados por cada usuario, respecto al total de contenidos.

Capítulo 1

1 Introducción

La cantidad de información que debe manejarse actualmente para encontrar aquellos elementos que susciten nuestro interés provoca que acabemos necesitando ayuda para encontrar aquello que deseamos. La compra de un libro, cuando se nos ofrece una gran variedad, o la selección de noticias cuando tenemos a nuestro alcance miles de fuentes, se vuelven tareas arduas. Esto ha provocado que los sistemas de recomendación hayan pasado a ser una herramienta indispensable en muchas tareas de la vida cotidiana.

El primer sistema de recomendación conocido desde la existencia de Internet apareció en 1994 y se llamó Ringo. Su objetivo era ayudar a los usuarios a buscar música de su interés. Se ha de tener en cuenta, que esto es más difícil que localizar el *Ford Focus* más barato del mercado o que encontrar las gafas de sol de la marca *Porsche* más baratas. El coste de las cosas es fácilmente medible, pero música “interesante” es algo cualitativo y que depende mucho del oyente.

Lo que Ringo hacía era bastante simple. Consistía en dar una lista de veinte títulos de música y después preguntaba una por una: si era del gusto del usuario, si no lo era, o si no la había escuchado. Eso inicializaba al sistema con una pequeña muestra (llamada DNA) de las preferencias del usuario. Después, cuando el usuario pedía una recomendación, el sistema comparaba su DNA con las muestras de los otros usuarios del sistema. Cuando Ringo identificaba las más similares, buscaba en ellas música que el usuario no hubiera oído, y se la recomendaba.

Aunque la idea era simple, la existencia de Internet hizo que el sistema de recomendación tuviera un éxito abrumador. Gracias a los miles de usuarios que llegaron a usarlo, las recomendaciones tenían prácticamente el éxito asegurado.

1.1 La Empresa

Este proyecto es fruto del trabajo del estudiante, de Ingeniería en Informática de la Facultad de Informática de Barcelona (FIB), Victor Codina Busquet como trabajador de la empresa TMT Factory. El proyecto de fin de carrera pues se encuadra dentro de un proyecto de modalidad B según la guía docente vigente.



TMT Factory (razón social: BCN d'Infografia) es una empresa de ingeniería del software, con sede en Barcelona y presencia en el mercado desde 1997. La actividad principal de la empresa es el desarrollo de servicios integrales de comunicación, multimedia y tecnología.

Uno de sus productos de más éxito ha sido IntegraTV, que consiste en un sistema de televisión interactiva que permite a los usuarios cliente poder disfrutar de una amplia variedad de tipos de contenido: películas, series, música, información cultural, videojuegos, etc. Actualmente existen dos versiones de este producto:

- *IntegraTV Hotel.* Versión que ya está funcionando en varios hoteles de la geografía española, pensada para funcionar en hoteles de alta categoría como un servicio de ocio para los clientes del hotel.
- *IntegraTV Home.* Versión que está en una fase experimental y no disponible en el mercado, pensada para que los usuarios puedan disfrutar de los servicios que ofrece el sistema desde su hogar.

1.2 Motivaciones

A continuación se describen los motivos principales del por qué se ha decidido desarrollar un sistema de recomendación de contenido vídeo para la versión de IntegraTV Home.

- **El sistema de distribución de contenidos actual** limita notablemente al usuario en cuanto a los contenidos que puede elegir. Esto es así porque los usuarios sólo disponen de los contenidos que están almacenados de forma local en su equipo de televisión interactiva, al ser inviable por ahora ver los contenidos en tiempo real directamente desde el servidor. Por lo tanto, el hecho de que los usuarios sólo puedan escoger en un momento dado contenidos entre un número reducido de opciones implica que sea importante personalizar la lista de contenidos que se les ofrece.
- **Los contenidos de vídeo son los más apreciados por los usuarios** según unas estadísticas de uso del sistema. Lo que implica que mejorando este servicio, seguramente aumente en mayor medida la satisfacción de los usuarios y por lo tanto la calidad del producto.

- Todo servicio personalizado necesita **disponer de un perfil de usuario y obtener una retroalimentación** continuada por parte de éstos para poder ofrecer recomendaciones personalizadas a cada usuario. De manera que en estos momentos no es viable ofrecer un servicio personalizado a IntegraTV Hotel, donde los usuarios suelen ser mucho más esporádicos que los de un hogar.
- **Actualmente no existe personalización** alguna a la hora de asignar los contenidos a cada cliente. Esto implica en general, que los usuarios adquieran menos contenidos al no tener suficientes opciones afines a sus preferencias para elegir.

Por estos motivos, se vio la posibilidad de ofrecer un servicio de recomendación personalizada de contenidos de vídeo que permitiera mejorar la calidad del servicio actual. Para ofrecer este servicio también sería necesaria una gestión adecuada de los perfiles de usuario que hasta ahora no existía, así como extender la gestión de los datos de contenidos.

1.3 Objetivos del proyecto

Los objetivos principales del proyecto son:

1. Diseñar un sistema de recomendación de contenido vídeo adecuado al entorno del sistema, que se caracteriza por tener una baja densidad de información disponible.
2. Desarrollar todas las funcionalidades propias del servicio de recomendación, que permitan gestionar los datos de contenidos y de usuario así como la gestión de las recomendaciones hechas por el sistema de recomendación diseñado.

Para llevar a cabo el diseño del sistema de recomendación se han identificado las siguientes tareas:

- Realizar un estudio del estado del arte sobre el problema de la recomendación y los sistemas de recomendación.
- Seleccionar las alternativas más adecuadas al contexto del sistema y al dominio de contenidos de vídeo.
- Diseñar un sistema de recomendación a partir de alguna de las alternativas seleccionadas, en caso de existir, o crear un nuevo sistema en base a los conocimientos adquiridos durante el estudio del estado del arte.
- Construir un prototipo del sistema de recomendación y evaluar la precisión de sus recomendaciones.

Por lo que se refiere al desarrollo del sistema de recomendación como un nuevo servicio del producto de IntegraTV Home, seguirá una metodología propia de la Ingeniería del Software, en la que podemos identificar básicamente las siguientes etapas:

- Analizar y diseñar las nuevas funcionalidades propias del servicio de recomendación.
- Construir y testear las funcionalidades propias del servicio de recomendación.
- Implantar la nueva versión del IntegraTV Home en los equipos interactivos de los clientes actuales.

Durante el desarrollo del sistema se pretenden lograr los siguientes objetivos:

- Mejorar el diseño y rendimiento de la base de datos de la aplicación actual.
- Minimizar los cambios en los componentes de IntegraTV Home
- Procurar que el nuevo servicio sea fácilmente accesible y usable para los usuarios.

1.4 Estado del arte

Aunque el origen de los sistemas de recomendación puede remontarse principalmente al extenso trabajo en ciencia cognitiva, teoría de la aproximación y recuperación de la información, los sistemas de recomendación surgieron como un área de investigación independiente a mediados de la década de los noventa. Por esos años fue cuando los investigadores empezaron a centrarse en problemas de recomendación que dependían explícitamente de las valoraciones de usuario.

La forma más común de formular el problema de la recomendación es desde una perspectiva probabilística, en la que la tarea de recomendar es reducida al problema de calcular el valor esperado de un voto (valoración) de usuario sobre algún contenido, dado lo que el sistema conoce del usuario.

1.4.1 Clasificación

Una forma de clasificar los sistemas de recomendación es según la metodología utilizada para construir el perfil de usuario. Se suelen clasificar por si utilizan preferencias explícitas, implícitas o una combinación de ambas para calcular las recomendaciones.

Se considera que un sistema de recomendación utiliza **preferencias explícitas**, cuando el usuario expresa sus preferencias conscientemente.

Como por ejemplo en el caso de valorar un contenido, donde la valoración consiste en dar un valor numérico dentro de un rango definido.

Los sistemas que utilizan **preferencias implícitas** deducen las preferencias a partir del comportamiento del usuario y de su historial. Esto permite que en la mayoría de los casos no sea necesario pedir al usuario gran cantidad de información sobre sus preferencias para que pueda ser recomendado.

En general, los sistemas de recomendación utilizan tanto preferencias explícitas como implícitas para construir los perfiles de usuario. De manera que inicialmente suelen utilizar métodos explícitos para obtener la información básica del perfil, para más adelante, a medida que el usuario interactúa con el sistema de recomendación, utilizar preferencias implícitas con el fin de refinar el perfil inicial y mejorar sus predicciones.

La forma más conocida de clasificar a los sistemas de recomendación es según la metodología que utilizan para resolver el problema de la recomendación. En la literatura encontramos básicamente cuatro metodologías distintas: basada en el contenido, basada en casos, basada en filtro cooperativo y finalmente la metodología híbrida que se basa en la combinación de las anteriores.

Recomendación basada en el Contenido

Consiste en buscar objetos que se adapten lo mejor posible a las preferencias del usuario. El elemento clave de esta metodología de recomendación es la medida de similitud que nos indica cuánto de relacionado está un objeto con un determinado usuario.

Los principales inconvenientes son: la sobre-especialización, que consisten en que las recomendaciones resultantes suelen ser siempre muy similares dado que se basan en la misma información; y que dependiendo del dominio de los objetos a recomendar, es difícil extraer buena información de éstos para realizar un cálculo fiable de la similitud.

La ventaja respecto a los demás métodos es que se pueden realizar recomendaciones sin la necesidad de tener un historial previo, por lo que permite realizar predicciones independientemente del historial del usuario.

Recomendación basada en Casos

Consiste en recomendar objetos similares a los que el usuario ha valorado positivamente en el pasado. En esta metodología los elementos clave son la medida de similitud entre objetos, y la clasificación de los objetos del historial de usuario según una medida de la relevancia de cada objeto por el usuario.

Los principales inconvenientes son: la sobre-especialización, y que la calidad de las recomendaciones va estrictamente relacionada con la cantidad de valoraciones que ha hecho previamente el usuario.

La ventaja principal es que la recomendación es independiente de cada usuario, y por lo tanto soluciona el problema de la poca densidad de información. Un ejemplo lo encontramos en Montaner [16].

Recomendación Cooperativa

Consiste en recomendar objetos a un usuario basándose en las valoraciones de otros usuarios con un perfil similar. Las metodologías de filtro cooperativo se pueden dividir en dos grandes categorías [7].

- **Memory-based (user-based) collaborative filtering:** Son sistemas que utilizan heurísticos que realizan predicciones en base a toda la base de datos de usuario para generar una predicción.
- **Model-based (item-based) collaborative filtering:** Son sistemas que utilizan un modelo (creado a partir de la base de datos de valoraciones de usuario) para realizar las predicciones. Para construir este modelo se utilizan diversas técnicas de aprendizaje; las más usadas son
 - redes bayesianas [9];
 - técnicas de clasificación;
 - técnicas basadas en reglas.

Varios estudios (por ejemplo, Sarwar et al. [18]) demuestran que los algoritmos de filtro cooperativo *item-based* tienen mejor rendimiento en dominios con gran escasez de datos. Por otro lado, también se ha demostrado empíricamente en Pennock and Horvitz [17] que la combinación de ambas metodologías de filtro cooperativo (memory-based y model-based) puede ofrecer mejores resultados que utilizando únicamente una de las metodologías.

Los principales inconvenientes de estos métodos son: la baja densidad de información disponible, usuarios con gustos especiales son mal recomendados, un nuevo objeto no es recomendable hasta que no sea valorado por un número determinado de usuarios, y finalmente son poco escalables.

La ventaja principal es que solventan los inconvenientes de las metodologías basadas en contenido y en casos. Ejemplos de sistemas que utilizan este método son: *Amazon* en Linden et al. [14], y *Recommendation Explorer* en Efron and Geisler [12].

Recomendación Híbrida

Para poder resolver las limitaciones de cada tipo de recomendación, en muchas ocasiones, se utilizan sistemas híbridos. Algunos ejemplos son: *Fab* [6], un sistema de recomendación de documentos web; y *Racofi Music* [1], un sistema de recomendación de música. Dependiendo de la forma en que combinan las diversas aproximaciones los sistemas híbridos se pueden clasificar en [8]:

- **Weighted:** sistema donde, dado un objeto, se combinan las diferentes puntuaciones de cada componente mediante una fórmula lineal.
- **Switching:** sistema que escoge un único componente dependiendo de la situación de recomendación.
- **Cascade hybrids:** sistema jerárquico donde el componente secundario refina las decisiones del primario.
- **Feature augmentation:** sistema que consiste en utilizar un componente para generar una nueva característica para cada objeto, aumentando así la información disponible para el componente primario.

1.4.2 Ejemplos de sistemas de recomendación

Dado que el presente proyecto trata de un sistema de recomendación de contenido de vídeo, se mencionarán algunos ejemplos reales de sistemas de recomendación del dominio de vídeo. Cabe destacar que alguno de ellos ha servido de inspiración para el diseño del sistema de recomendación desarrollado en este proyecto.

Personal Programming Guide (PPG [2]). Sistema de recomendación de programas de TV que utiliza una metodología híbrida. En particular combina tres metodologías: la basada en el contenido (llamado el *Explicit User Model*); la de filtro cooperativo basado en técnicas de clasificación (el *Stereotypical User Model*), realiza las predicciones a partir de la información de las categorías de telespectador, y la basada en casos (llamado el *Dynamic User Model*) que realiza predicciones en base al análisis del comportamiento televisivo del usuario. Finalmente el sistema ordena las predicciones del usuario suministradas por los tres modelos en base a una medida de la confianza de sus predicciones.

Avatar [4],[5]. Sistema de recomendación híbrido de contenidos de vídeo, basado en agentes, que combina como en el caso anterior tres metodologías: dos de filtro cooperativo y una basada en el contenido. El sistema de recomendación se divide en tres módulos: el *Agente Bayesiano* (basado en filtro cooperativo) que utiliza la información referente a los programas vistos y las preferencias de los usuarios para realizar las recomendaciones, el *Agente Semántico* (basado en el contenido) que utiliza información actualizada sobre las preferencias de los usuarios y del conocimiento del dominio (ontología), y el *Agente de perfiles* (basado en filtro cooperativo) que compara distintos perfiles de usuario para realizar las predicciones. Las recomendaciones de los tres agentes son combinadas utilizando técnicas de redes neuronales.

Recommendation Explorer [12]. Consiste en un sistema de recomendación de películas basado en filtro cooperativo del tipo *item-based*. Para realizar las recomendaciones utiliza un modelo de relaciones de las películas disponibles. Para cada película se guardan dos conjuntos de información:

- Un conjunto almacena las recomendaciones cercanas, que consisten en referencias a otras películas que tienen aspectos en común con la película en cuestión. Por ejemplo, que sean del mismo director.
- El otro conjunto almacena referencias a otras películas con las que tiene una relación más débil.

La estructura utilizada para construir el modelo de relaciones de películas consiste en una matriz, transformada mediante la aplicación del *Singular Value Decomposition* (SVD), vista la gran ineficiencia en la generación de recomendaciones que suponía utilizar una matriz $N \times N$. Con esta técnica se consigue reducir la dimensionalidad de la matriz consiguiendo un modelo de similitud mucho más robusto.

1.5 Estructura de la memoria

La memoria está organizada como se describe a continuación. En el **capítulo 2**, se proporciona una descripción del estado actual de la aplicación a la que se pretende extender la funcionalidad, añadiéndole un servicio de personalización de contenidos de vídeo. En el **capítulo 3** se describe en detalle todo lo relacionado con el diseño del sistema de recomendación: la ontología utilizada, el modelo de usuario utilizado, las metodologías utilizadas y finalmente las conclusiones de una primera evaluación del sistema. El **capítulo 4** detalla el proceso de desarrollo del servicio de recomendación como una extensión del sistema actual, utilizando la metodología de desarrollo de software Métrica 3. Finalmente se muestra la planificación y el seguimiento del proyecto, así como las conclusiones extraídas durante la realización del mismo.

Capítulo 2

2 Estado actual de IntegraTV Home

Dado que los sistemas de recomendación necesitan un conocimiento previo de los usuarios para su correcto funcionamiento, como datos personales y un historial de visionado, y no usuarios esporádicos como suelen ser los clientes de un hotel, el servicio de recomendación a desarrollar se integrará en la versión de *IntegraTV Home*.

IntegraTV Home consiste en una aplicación con arquitectura cliente-servidor formada por varios componentes distribuidos, los cuales se comunican entre sí. En los siguientes apartados se describirán más a fondo los componentes que forman el sistema, el funcionamiento de la distribución de contenidos a los clientes y las tecnologías utilizadas para desarrollarlo.

2.1 Componentes del sistema

En la Ilustración 1 se puede observar un organigrama con la interacción entre los diversos componentes que conforman el sistema *IntegraTV Home*, tanto de la parte servidor como de la parte cliente.

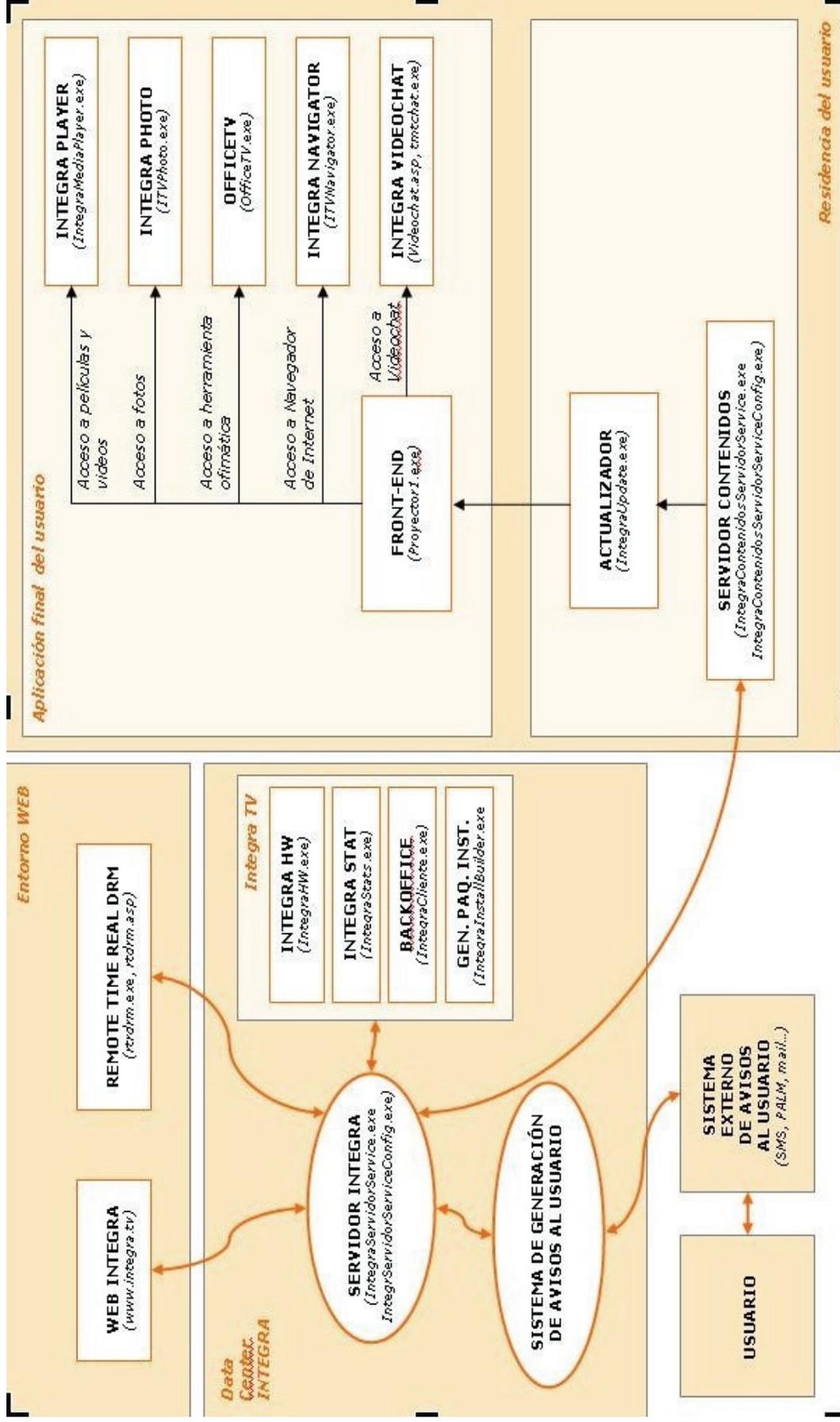


Ilustración 1: Organigrama de la aplicación IntegraTV Home

En el organigrama básicamente se pueden identificar: los **componentes de la parte del cliente**, que sirven de interfaz para poder interactuar con los diversos servicios que ofrece el sistema, y están instalados en el sistema de televisión interactiva del hogar; los **componentes de la parte del servidor**, que son los procesos que se encargan de la gestión y distribución de datos así como del control del sistema.

A continuación se hará una breve descripción de cada uno de los aplicativos que conforman el producto de *IntegraTV Home*.

De la parte del usuario final (cliente):

- **FrontEnd**. Aplicación de escritorio que se ejecuta en el equipo de televisión interactiva del usuario y que éste utiliza para interactuar con el sistema, recibir los contenidos e utilizar los servicios.
- **Integra Player**. Reproductor TV de contenidos audiovisuales.
- **Office TV**. Aplicación de tele-trabajo con herramientas ofimáticas desde el TV.
- **Integra Navigator**. Consiste en un navegador Web adaptado.
- **Integra Photo**. Aplicación que permite al usuario gestionar sus fotos.
- **Actualizador**. Aplicación encargada de las actualizaciones y gestión de comprobaciones del sistema.
- **Servidor Contenidos**. Aplicación encargada de recibir los contenidos del Data Center.

De la parte del servidor de Integra:

- **Servidor Integra**. Aplicación residente en el servidor central, encargada de recibir y distribuir contenidos a los diferentes clientes del servicio.

De la parte de los administradores:

- **BackOffice**. Aplicación con la que los administradores gestionan los contenidos y actualizaciones del sistema.
- **Integra HW**. Se encarga de la gestión del parque informático.
- **Integra Stat**. Aplicación encargada de la vigilancia y monitorización en tiempo real del HW/SW.
- **Remote Time Real DRM**. Monitorización en tiempo real vía Web de derechos de contenidos.

A continuación se pasará a explicar en más detalle el funcionamiento de los componentes del sistema que juegan un papel importante para llevar a cabo la integración del servicio de recomendación en el sistema actual.

2.1.1 FrontEnd

Como se ha comentado anteriormente, se trata de la aplicación de escritorio instalada en el equipo de televisión interactiva del cliente. Dado que se trata de una aplicación dependiente del equipo desde donde se ejecuta (en este caso un equipo de televisión interactiva), la identificación del cliente se asocia a la máquina que se le ha proporcionado, a la cual se le asigna un código secreto.

La aplicación está pensada para que la puedan utilizar distintos usuarios, de los cuales uno es el responsable de gestionar la aplicación, y tiene permisos para utilizar plenamente todos los servicios del sistema además de configurar las cuentas de los demás usuarios. Cada usuario, incluido el administrador del equipo, se identifica utilizando un nombre identificador (login) y una contraseña.

Todo usuario con los permisos suficientes, puede adquirir un contenido de los que le ofrece el sistema. Si el tipo de contenido que ha pedido, no está incluido en su pack de cliente, entonces el cliente ha de pagar por el servicio. Para completar este proceso, el usuario ha de introducir un código secreto que identifica el equipo del cliente.

Los servicios que ofrece el sistema van desde la adquisición de películas (servicio videoclub) hasta el visionado de canales de televisión digital específicos. En la Ilustración 2 se puede ver el menú principal de la aplicación.



Ilustración 2: Menú principal del FrontEnd.

El servicio de videoclub funciona como la bandeja de entrada de un gestor de correo tal como se observa en la Ilustración 3. Cada vez que un contenido es descargado en el equipo del cliente, se añade el contenido en la lista de contenidos disponibles dentro de su “género” correspondiente. En el momento que el usuario selecciona alguno de los contenidos recién descargados, se marca como “seleccionado” y se actualiza el número de contenidos por seleccionar del género correspondiente.



Ilustración 3: Menú principal del servicio de videoclub

La versión actual del sistema no permite al usuario descargar contenidos bajo demanda, únicamente se van asignando de forma automática los nuevos contenidos de los que dispone la empresa a cada uno de los clientes.

Actualmente las acciones que puede realizar un usuario con un contenido disponible en su equipo son:

- **adquirirlo** para tener permisos de reproducción durante un periodo de tiempo determinado;
- **bloquearlo/desbloquearlo** para evitar que se borre automáticamente del equipo o bien al contrario, para permitir que se borre de forma automática;
- **borrar** el contenido del equipo (directamente del disco duro) para liberar espacio y permitir la entrada de más contenidos.

2.1.2 BackOffice

Se trata de la herramienta de gestión de contenidos y de actualizaciones del sistema. Además actualmente se utiliza para gestionar la distribución de los contenidos tanto en hoteles como entre los diversos clientes de prueba de la versión de IntegraTV Home.

Únicamente los usuarios con permisos de administrador pueden utilizar esta aplicación. Para el control de acceso se utiliza la típica autenticación de login/password como se observa en la Ilustración 4.



Ilustración 4: Pantalla de identificación de usuario del BackOffice

En relación a la gestión de los contenidos, que es la parte de la aplicación más relacionada con el servicio de recomendación, el sistema permite dar de alta, modificar y borrar una ficha de contenido. Los datos de los contenidos de vídeo que actualmente se utilizan son de ámbito general, por lo que exista poca información para utilizarla en las recomendaciones. Además no existe una semántica asociada a los pocos datos disponibles, lo que implica que sean datos poco útiles para realizar recomendaciones en base a éstos. En la Ilustración 5 se observa la interfaz de usuario con la se modifican actualmente los datos del contenido de vídeo.

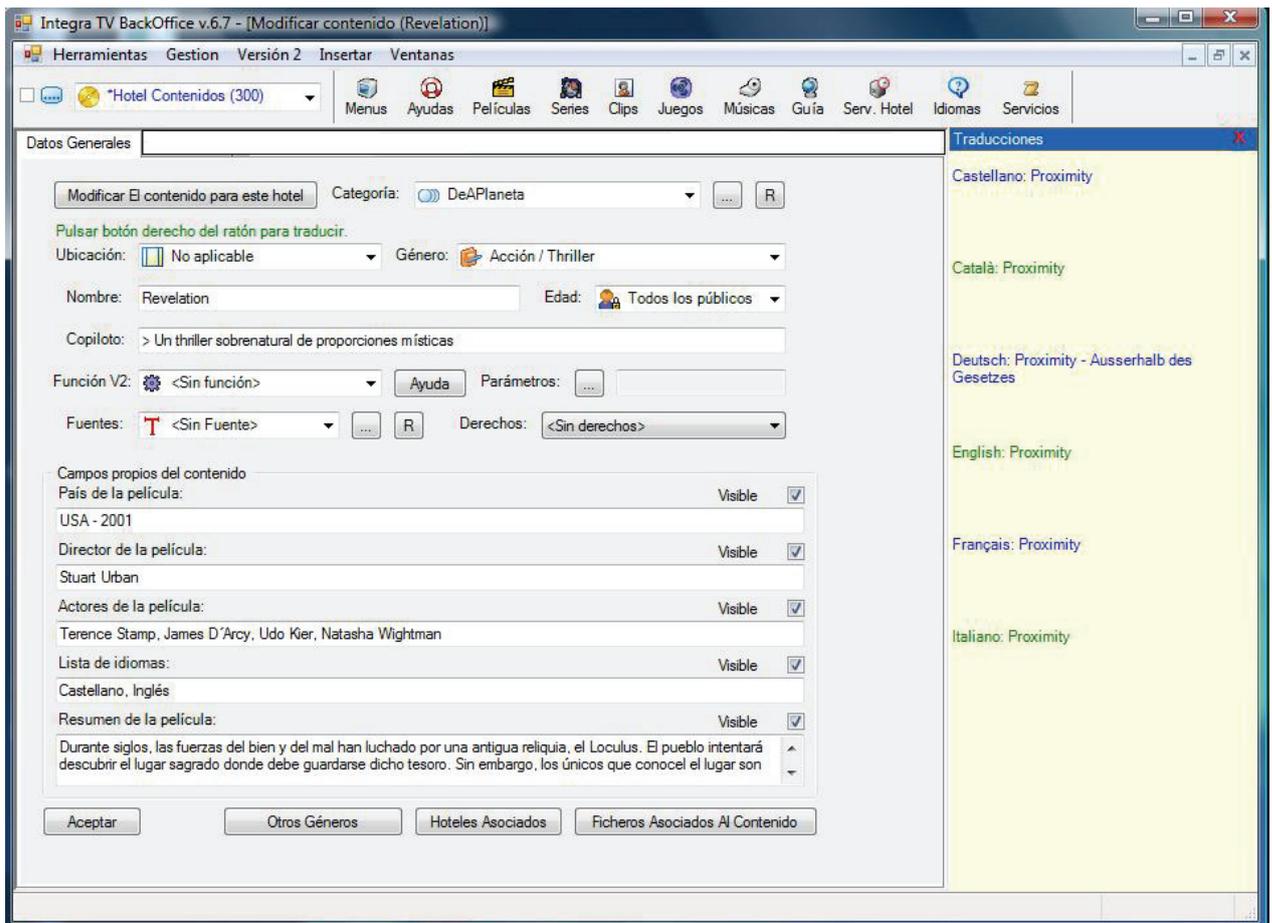


Ilustración 5: Pantalla de modificación de datos de un contenido de vídeo.

2.2 Distribución de contenidos

El sistema de distribución de contenidos está basado en una red Peer-to-Peer formada por todos los clientes de IntegraTV Home. Esto significa que cada cliente se conecta directamente a otro para la descarga de los contenidos sin la necesidad de una gestión central. Se basa en un funcionamiento similar al de la aplicación Kazaa², en el que el servidor central conoce qué archivos tiene cada uno de los clientes, de manera que cuando un cliente pide una determinado parte de un contenido, el servidor se encarga de indicarle desde qué clientes (máquinas) puede descargárselos.

Para hacer viable las descargas directas punto a punto de los contenidos, cada contenido es dividido en trozos del mismo tamaño. El servidor de Integra se encarga de mantener la lista de trozos de contenido que tiene cada cliente, que es almacenada utilizando distintos ficheros locales al servidor.

Cada cliente se encarga de mantener una lista de los contenidos totalmente descargados (los disponibles para el cliente), y otra lista con las películas

² Más información en la página oficial: <http://www.kazaa.com> (visitado el 10/01/08)

incompletas con los trozos actualmente descargados. Estas listas también son almacenadas en ficheros locales al equipo del cliente.

La Ilustración 6 muestra la arquitectura física de IntegraTV Home. El proceso que sigue la descarga de un contenido desde un determinado cliente del sistema es el siguiente:

1. El servidor asigna un nuevo contenido de forma automática a un determinado cliente. Esto implica que el servidor se comunica con el cliente para asignarle el contenido a descargar, que es añadido en la lista de películas incompletas.
2. El servidor comprueba si existen clientes que tengan el mismo contenido.
 - 2.1. En caso afirmativo, el servidor proporciona al cliente la lista de clientes con los trozos de contenido que posee cada uno. En ese momento el cliente es el que se encarga de descargar el contenido a partir de los trozos de cada uno de los clientes de la lista que le ha proporcionado el servidor.
 - 2.2. Si no ha sido distribuido el contenido a más clientes, entonces el servidor proporciona la dirección de uno de los servidores de descarga al cliente, a través del cual podrá descargar totalmente o parcialmente el contenido mediante descarga directa.
3. Una vez el cliente descarga todos los trozos que forman el contenido, pasa a formar parte de la lista de disponibles. De manera que la próxima vez que acceda al servicio de videoclub aparecerá la ficha del contenido descargado.

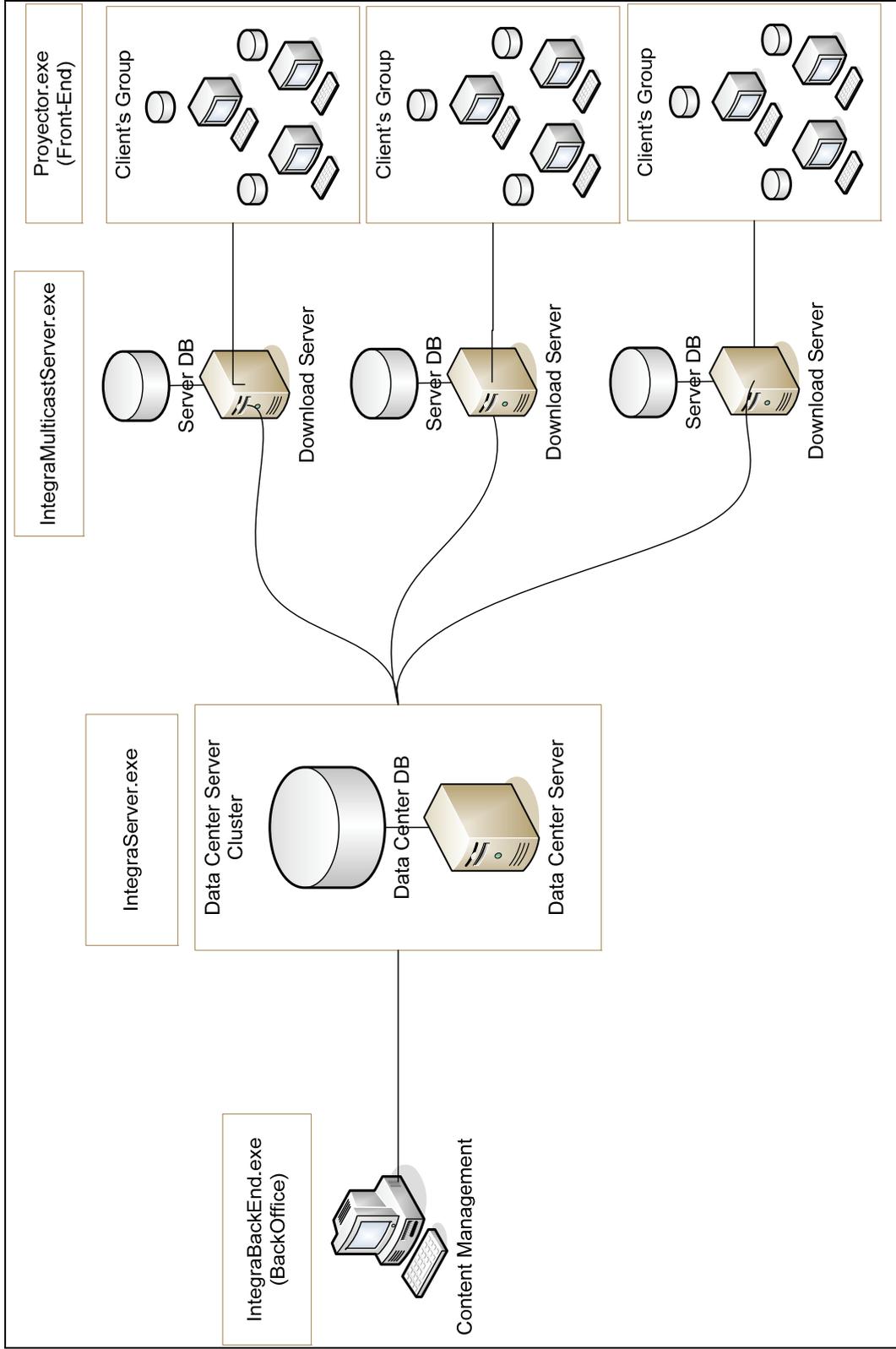


Ilustración 6: Arquitectura física del sistema de IntegraTV Home.

2.3 Tecnologías utilizadas

En este apartado se describirá brevemente las tecnologías utilizadas para el desarrollo de la aplicación de componentes distribuidos IntegraTV Home.

2.3.1 Almacenamiento

Actualmente la aplicación utiliza un almacenamiento de los datos distribuido entre los diferentes nodos que conforman la arquitectura del sistema. Los datos almacenados en cada uno de los nodos es el siguiente:

- **Servidor Central de Integra** (Datacenter Server). Utiliza el sistema gestor de base de datos relacional SQL Server y almacena toda la información relacionada con los contenidos, como son los hoteles y clientes a los que está asociado. Además se encarga de almacenar las listas que utiliza para la distribución de contenidos por medio de ficheros locales.
- **Servidor de descarga** (Download Server). Almacena las listas de descargas para la distribución de contenidos, del grupo de clientes que tiene asociados.
- **Equipos de cliente** (Client's Group). Utiliza el sistema gestor de base de datos relacional Microsoft Access y almacena todos los datos relativos a la aplicación local del cliente, como son datos del cliente y el historial de visionado. Además igual que lo servidores, almacena en ficheros locales las listas de los trozos de contenido que actualmente tiene descargados.

2.3.2 Comunicaciones

Para llevar a cabo las comunicaciones entre los diversos componentes que conforman la arquitectura del sistema que se ha observado en la Ilustración 6, se utiliza una librería de comunicaciones ad-hoc basada en algunos conceptos en la tecnología CORBA.

En las comunicaciones con el cliente, se utiliza encriptación para asegurar la privacidad de los datos durante las transmisiones a través de Internet.

Capítulo 3

3 Sistema de Recomendación

3.1 Ontología utilizada

El sistema de recomendación utiliza la información de las preferencias de usuario y de los contenidos de vídeo disponibles para calcular las recomendaciones. Dado que esta información juega un papel importante en el diseño del sistema se ha utilizado una ontología para representarla y eliminar posibles ambigüedades en el significado de los diferentes atributos utilizados.

Para realizar la ontología se ha utilizado la aplicación Protégé³, un editor de ontologías muy utilizado que ofrece una extensa variedad de herramientas para construir sistemas basados en el conocimiento.

En los siguientes apartados se describirán los distintos atributos que conforman el perfil de usuario y que describen los contenidos de vídeo, utilizando el formato:

- **Nombre atributo** [tipo] (valores posibles): breve descripción del atributo.

3.1.1 Perfil de Usuario

Los usuarios que harán uso del sistema de recomendación tendrán asociados un perfil específico según la información que hayan proporcionado. Esta información será tanto referente a sus preferencias como información personal.

3.1.1.1 Información de preferencias

Los diferentes atributos relativos a las preferencias de usuario son:

³ Más información en la Web oficial: <http://protege.stanford.edu/> (visitado el 10/01/08)

- **Ingredientes** [int] (valores discretos de 0 a 100): es el atributo más relevante de las preferencias de usuario, por el cual puede definir con precisión sus preferencias sobre los distintos ingredientes que definen a los contenidos de vídeo. Los ingredientes de contenido que se han utilizado son:
 - Acción
 - Diversión
 - Dramatismo
 - Erotismo
 - Humor
 - Sexo
 - Suspense
 - Violencia
 - Animación
 - Ciencia-Ficción
 - Historicismo
 - Música
 - Romanticismo
 - Terror

- **Actores preferidos** [actor] (lista de actores más *relevantes*⁴ de la base de datos)

- **Directores preferidos** [director] (lista de directores más relevantes de la base de datos)

- **Duración máxima preferente** [duración] (indiferente, 1h, 1.5h, 2h, 2.5h, 3h): indica cual es la duración máxima de contenidos que el usuario prefiere.

- **Géneros de contenido preferidos** [género] (lista de géneros de la base de datos)

- **Nacionalidades de contenido preferidas** [nacionalidad] (lista de nacionalidades más relevantes de la base de datos)

- **Peso preferencias** [int] (valores del 0 al 10): el usuario puede indicar para cada uno de los atributos comentados anteriormente la importancia que tienen para él.

Para los atributos *Actores*, *Directores*, *Géneros* y *Nacionalidades* se permite que el usuario escoja un máximo de 5 opciones para cada uno. Además en los campos formados por un lista de opciones muy amplia se dará la opción de escoger la opción “Otros”.

⁴ Se consideran actores/directores relevantes los que aparecen como mínimo un número determinado de veces en la base de datos de contenidos.

3.1.1.2 Información de usuario personal

En este apartado se describen los atributos relativos a la información personal del usuario.

- **Ocupaciones** [ocupación] (lista de ocupaciones predefinida): indica qué oficios posee el usuario.
- **Aficiones** [afición] (lista de aficiones predefinida): indica las aficiones del usuario
- **Estado Civil** [estado] (soltero/a, casado/a, divorciado/a, viudo/a).
- **Edad** [edad] (17 o menos, 18-24, 25-29, 30-34, 35-44, 45-54, 55-64, 65 o más).
- **Sexo** [sexo] (hombre, mujer).
- **Clase social** [clase] (baja, media, media-alta, alta).
- **Estudios** [estudio] (menos que bachillerato, bachillerato, 1-3 años de universidad, 4-5 años de universidad, estudios de tercer ciclo).
- **Nacionalidad** [bool] (española, no española).
- **Número de hijos** [int] (0, 1, 2, 3, 4 o más).

Para los campos *ocupaciones* y *aficiones* se permite que el usuario escoja un máximo de 5 opciones para cada uno. También existe la opción “Otros” en ambos casos.

3.1.2 Contenido de vídeo

La información almacenada para cada uno de los contenidos de vídeo consiste en los siguientes descriptores:

- **Ingredientes** [int] (valores discretos de 0 a 100): se utilizan para describir con precisión el tipo de contenido de vídeo. Los ingredientes de contenido utilizados se corresponden con los del atributo *ingredientes* del perfil de usuario.
- **Actores** [actor] (lista de actores principales del contenido)
- **Directores** [director] (lista de directores del contenido)
- **Duración** [int] (duración del contenido en minutos)
- **Géneros** [género] (lista de géneros a los que pertenece el contenido)

- **Nacionalidades** [nacionalidad] (lista de nacionalidades del contenido)
- **Novedad** [booleano] (cierto o falso): indica si el contenido es una novedad.
- **Clásico** [booleano] (cierto o falso): indica si el contenido es un clásico.

3.2 Descripción del sistema de recomendación

Después de analizar profundamente varias alternativas, y dado que el sistema trabajará en un entorno con una baja densidad de información disponible, se ha decidido utilizar la metodología híbrida para el diseño del sistema de recomendación. De esta manera, combinando varias metodologías, se reducirán las limitaciones que presenta cada una de ellas, consiguiendo una mayor personalización en las recomendaciones de los usuarios.

El sistema de recomendación está diseñado de forma modular, utilizando una arquitectura de componentes, para facilitar posibles extensiones y mejoras del sistema. En particular, está formado por tres componentes de recomendación:

- **componente basado en el contenido;**
- **componente basado en el historial;**
- **componente basado en un modelo Bayesiano.**

Y por un **componente de combinación** que utiliza un sistema de confianzas entre los usuarios y los componentes de recomendación para calcular la recomendación final.

Se propone que el sistema sea capaz de recomendar a los usuarios en dos escenarios claramente diferenciados. El primero se trata de la **recomendación a un usuario**, por petición propia o bien de forma automática por el sistema, y el segundo consiste en **recomendar un nuevo contenido** a todos los usuarios seleccionados por el sistema de recomendación. El sistema actúa de distinta forma según el escenario de recomendación:

- **Caso 1.** Cada vez que el usuario lo pida, o de forma automática por el sistema, se le recomienda una lista de n-contenidos (también llamada *top list*) construida a partir de las recomendaciones de los diversos componentes de recomendación, y de la combinación de éstas por el componente de combinación. En la Ilustración 7 se puede observar cómo interactúan los diversos componentes para calcular las recomendaciones en este escenario.
- **Caso 2.** Un contenido introducido recientemente en la base de datos es recomendado a los usuarios de forma automática. Para realizar la recomendación el sistema utilizará los distintos componentes para decidir a qué usuarios del sistema se les recomienda el contenido. En la Ilustración 8 se observa una visión global de la interacción de los componentes del sistema de recomendación.

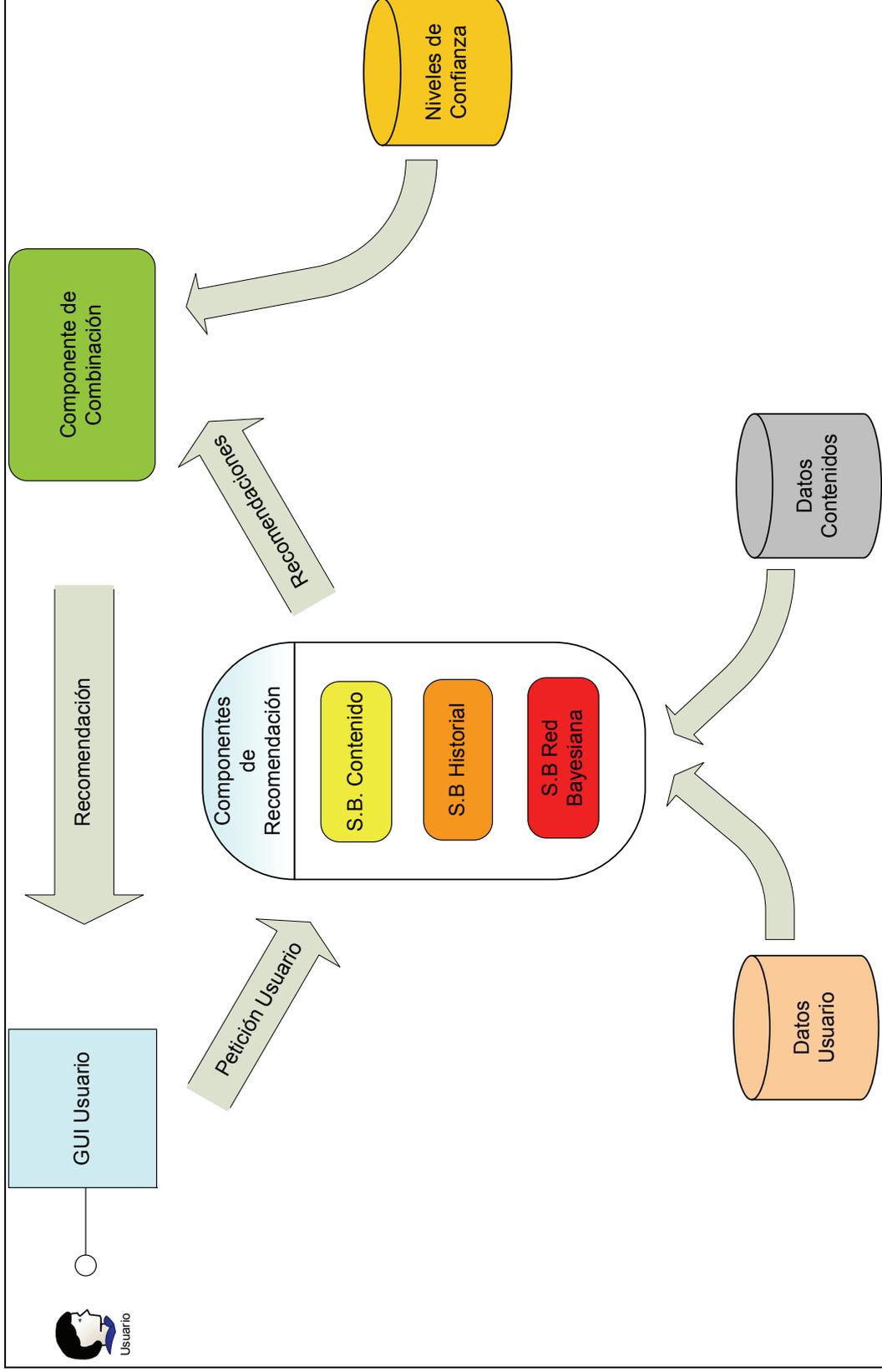


Ilustración 7: Diagrama de interacción de los componentes de recomendación en el caso 1.

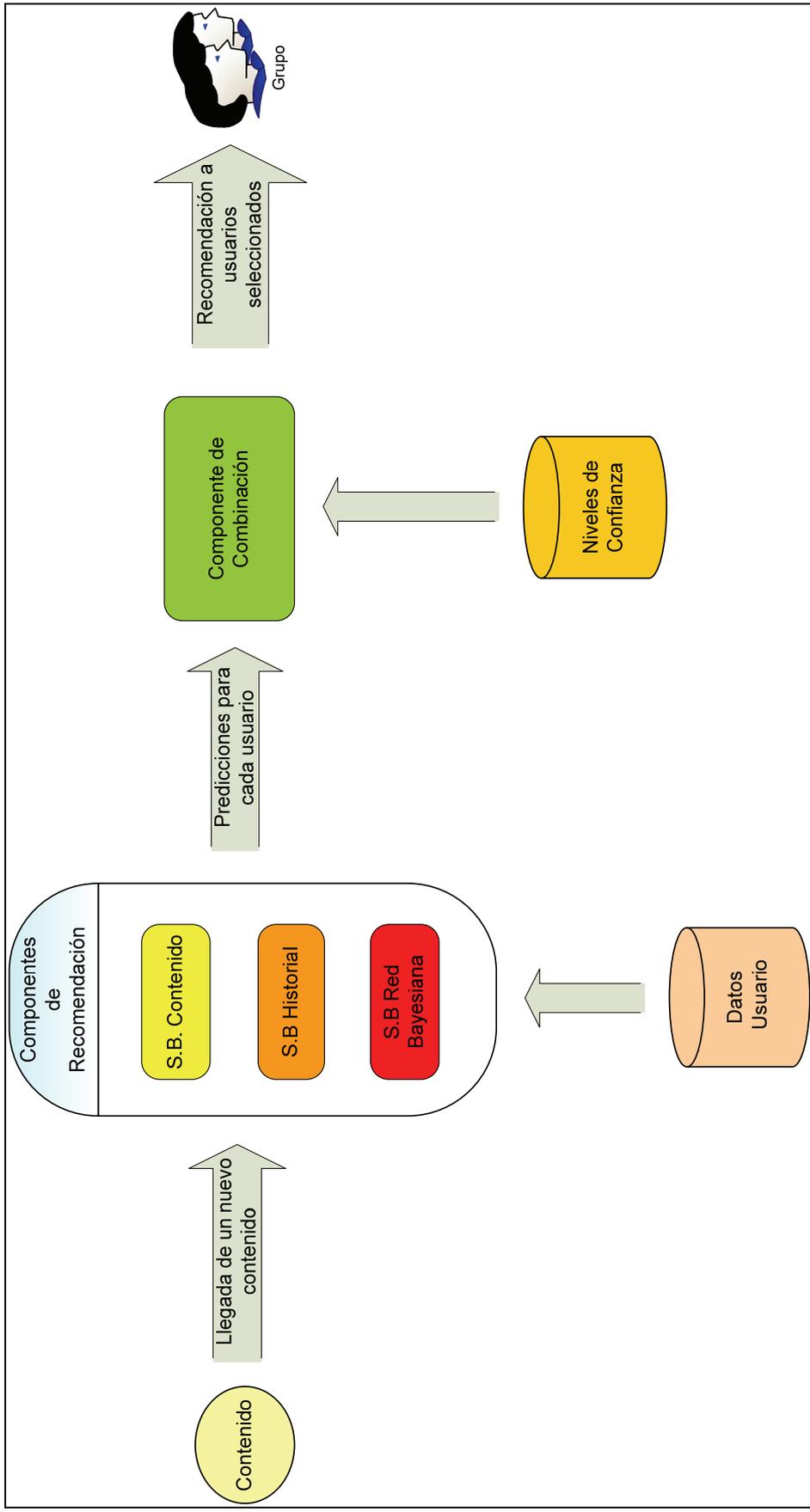


Ilustración 8: Diagrama de interacción de los componentes de recomendación en el caso 2.

En los siguientes capítulos se describirá con más detalle el funcionamiento de cada uno de los componentes aquí mencionados diferenciando entre los dos casos de recomendación.

3.2.1 Componente basado en el contenido

Componente que utilizará la información referente a las preferencias de usuario que éste mismo habrá proporcionado durante la creación de su perfil de usuario, para encontrar los contenidos que más se asemejen a sus preferencias.

El cálculo de la similitud es el resultado de: combinar el resultado obtenido a partir de la distancia Euclidiana entre los ingredientes de contenido del perfil de usuario y los del contenido; y calcular de coincidencias entre los descriptores del contenido y las preferencias del usuario, en el caso de disponer de información opcional⁵.

El espacio en el que se mide la distancia euclidiana consiste en el espacio Euclidiano, también llamado espacio Cartesiano, donde los ingredientes del perfil de usuario representan el punto X y los ingredientes del contenido el punto Y, con tantas dimensiones como ingredientes de contenido existan.

Dado que usando sólo esta función resultan recomendaciones poco variadas, utilizamos un algoritmo que combina la similitud con la diversidad del conjunto recomendado. De esta manera se valoran positivamente los contenidos más similares a los gustos del usuario y que a la vez sean diferentes del resto de los contenidos recomendados. Se puede encontrar más información sobre cómo combinar eficientemente la similitud con la diversidad en Smyth and Bradley [19].

3.2.1.1 Aspectos teóricos

Para calcular la similitud que existe entre los ingredientes del usuario y los ingredientes de un contenido se utiliza la siguiente variación de la distancia Euclidiana:

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

Donde a_i es un valor de un ingrediente del perfil de usuario, y c_i es un valor de un ingrediente del contenido que estamos comparando.

⁵ Se considera información opcional a todos los atributos del perfil de usuario menos el de los Ingredientes.

En el caso de que se disponga de información opcional sobre las preferencias del usuario, se calcula un valor de la similitud entre los descriptores de contenido y sus preferencias en base al número de coincidencias entre descriptores y a los pesos que el usuario ha asignado a cada descriptor según la importancia que tienen para él. En la siguiente fórmula se observa el cálculo de este valor:

$$\beta = \sum_{i=1}^n P_i \times N_i \quad (2)$$

Dónde P es el peso del descriptor "i" normalizado entre [0,1] en base a la información disponible sobre las preferencias del perfil, y N el número de coincidencias. El rango de valores de β es entre [0,1] por lo que si la suma supera ese valor se le asigna el valor máximo.

El resultado de la similitud es finalmente: $\gamma = \omega * \delta + (1 - \omega) * \beta$

Dónde γ, ω son valores dentro de rango [0,1]. Los valores de γ cercanos a 1 indican que el contenido es de interés para el usuario. El descriptor ω sirve para repartir el peso de la similitud entre el resultado a partir de la distancia euclidiana y el valor obtenido a partir del cálculo de las coincidencias.

Ejemplo simplificado del cálculo de la similitud:

Suponiendo que el número de coincidencias entre un perfil y contenido determinados es de: 2 actores, 1 director y 1 género, y que el peso asociado a cada descriptor es el mismo (0,2 por peso), tenemos que el valor de β es igual a 0,8 a partir de la ecuación (2). Con una ω de 0,5 (configuración en el que la similitud basada en la distancia Euclidiana tiene el mismo peso que el basado en coincidencias) y un valor de 0,7 en el cálculo de la ecuación (1), el resultado de la similitud sería de 0,75.

3.2.1.2 Caso 1

En este escenario, el componente basado en el contenido calcula la lista de contenidos más afines, utilizando la medida de similitud descrita anteriormente, al usuario a recomendar. A continuación se describe en pseudo-código el algoritmo para generar la lista de contenidos a recomendar.

Sean:

a: perfil de usuario (ingredientes + información opcional); **C**: conjunto total de contenidos de la base de datos que el usuario no ha visto; **k**: # contenidos que contendrá la lista de recomendación; **l**: límite máximo de contenidos a los que comparar (mejora la eficiencia del algoritmo).

función *CONTENIDO_1* (*a, C, k, l*) **devuelve** lista de *k*-contenidos
si *l*>0 entonces

```

        C':= k*l contenidos de C más similares al perfil, según la función
        SIM(a,c)
    sino
        C':= C
    fsi
    R := {}
    para i=1 hasta k hacer
        C':= ordenar cada c en C' según la función CALIDAD(a,c,R)
        R := R + primero(C')
        C':= C' - primero(C');
    fpara
    dev R
ffunción

```

Seguidamente se describe en detalle las funciones y variables más relevantes usadas en el algoritmo anterior:

- $SIM(a,c)$: calcula mediante la medida de similitud anteriormente descrita, la similitud que existe entre el perfil de usuario “a” y el contenido “c”.
- $CALIDAD(a,c,R)$: es el heurístico que nos mide la calidad de un contenido para ser recomendado, teniendo en cuenta el perfil del usuario (similitud) y los contenidos que ya están en la lista de recomendados (diversidad). Se define como:

$$Calidad(a,c,R) = (1 - \alpha) * Sim(a,c) + \alpha * Diversidad(c,R) \quad (3)$$

$$Diversidad(c,R) = 1, \text{ si } R = \{\} \\ = \sum_{i=0}^n (1 - Sim(c,r_i)) / n, \text{ en otro caso.} \quad (4)$$

El descriptor α sirve para modificar el peso que se le da a la función de similitud y a la diversidad. Para $\alpha < 0.5$ damos más peso a la similitud y para $\alpha > 0.5$ tendremos recomendaciones más diversas. Lo aconsejable es tener un α con valor próximo a 0.5. Podemos encontrar una evaluación de este heurístico y del algoritmo comentado con anterioridad en Smyth and Bradley [19].

3.2.1.3 Caso 2

En este caso, el componente calculará para todos los usuarios del sistema, la similitud entre el nuevo contenido y cada uno de los usuarios. Y lo hará mediante la función SIM definida en el apartado anterior. A continuación se describe en pseudo-código el algoritmo que genera la lista de similitudes.

Sean: U : conjunto de perfiles de todos los usuarios del sistema; c : contenido nuevo.

función CONTENIDO_2 (U, c) **devuelve** Lista de Real
S: Lista de Reales de tamaño $|U|$

para cada $u \in U$ hacer
 $S := \text{añadir}(S, u, \text{SIM}(u,c))$
fpara
dev S
ffunción

3.2.2 Componente basado en el historial

Este componente se basa en contenidos similares a los que el usuario ya ha visto y votado positivamente para realizar las recomendaciones.

3.2.2.1 Caso 1

El método para generar la lista de recomendación a un usuario concreto es el siguiente:

1. Se escogen un número determinado de contenidos del historial del usuario con la condición de que sean los mejor valorados por el usuario, y a la vez sean lo más variados posibles.
2. Se busca en la tabla de similitudes los contenidos más similares a cada uno de los escogidos en el paso 1.
3. Se genera la recomendación final a partir de las listas de contenidos similares obtenidas, dando más peso a las que tienen contenidos del género más visto por el usuario.

El sistema utilizará una matriz de tamaño $N \times k$, donde N es el número total de contenidos de vídeo de la base de datos, y k es el número de contenidos similares utilizado. De manera que habrá una lista de k -contenidos ordenados de forma decreciente según la función de similitud SIM para cada uno de los contenidos de la base de datos.

Para el cálculo de similitud entre contenidos se utilizará una versión diferente de la función SIM ya que en este caso se están comparando dos contenidos, y por lo tanto, no se tendrán en cuenta los mismos descriptores que los utilizados en el componente basado en el contenido.

La generación de la matriz de contenidos se hará al iniciar el sistema y se formará a partir de los contenidos disponibles en la base de datos. Una vez generada la matriz, el sistema se encargará de actualizarla con la llegada de nuevos contenidos a la base de datos. De igual forma, se actualizará la matriz cada vez que se dé de baja un contenido. A continuación se describe en pseudo-código el algoritmo para generar la lista de contenidos a recomendar.

Sean:

M: matriz de contenidos; **H**: historial televisivo del usuario (lista de contenidos); **n**: # contenidos que contendrá la lista de recomendación; **l**: # contenidos de muestra

Función HISTORIAL_1 (M, H, n, l) **devuelve** lista de n -contenidos (como máximo)

L, R, F: listas de contenidos

V: Lista de l enteros

L := seleccionar_contenidos (H, l)

Para *j=1 hasta l hacer*

c := obtener_contenido (L, l)

// Calculamos cuántos votos positivos existen en el historial de contenidos del mismo género que "c"

pr := contenidos_mismo_genero (H, c)

V := añadir(V, l, pr)

fPara

s := suma de los "l" enteros de V

Para cada *c ∈ L hacer*

R := obtener_lista (M, c)

R' := filtrar_contenidos (H, R, n, s, obtener_entero(V, l))

// Añade la lista R' a la lista F controlando que no haya repetidos.

F := añadir (F, R')

fPara

Dev *F*

fFunción

Seguidamente se describe en detalle las funciones y variables más relevantes usadas en el algoritmo anterior:

- **seleccionar_contenidos**:: esta función devuelve una lista con los contenidos del historial del usuario más relevantes, que luego se usarán para buscar contenidos similares a éstos.

Función *seleccionar_contenidos (H, l)* **devuelve** *Lista de l-contenidos*

R := {}

Para *i=1 hasta l hacer*

H := ordenar cada c en H según la función CALIDAD2(c,R)

R := R + primero(H)

H:= H - primero(H);

fPara

dev *R*

función

- **CALIDAD2 (c, R)**: es el heurístico que mide la calidad de un contenido *c*, teniendo en cuenta la importancia de este en el historial del usuario (relevancia), y el resto de contenidos seleccionados (diversidad).

$$CALIDAD2(c, R) = (1 - \alpha) * RELEVANCIA(c) + \alpha * DIVERSIDAD(c, R) \quad (5)$$

- **RELEVANCIA (c)**: es una medida de la relevancia de un contenido "c" para un usuario determinado. Se calcula como la valoración del usuario sobre el contenido "c", restándole un valor concreto que dependerá de la antigüedad del contenido. De esta manera, cada cierto tiempo el contenido irá perdiendo relevancia.

- **filtrar_contenidos**: esta función que filtra la lista de contenidos R, eliminando los contenidos que ya ha visto el usuario y también, los que tienen una similitud mayor que un determinado valor β , con algún contenido votado negativamente del historial de usuario.

Sean:

s: la suma de enteros de la lista V; **e**: el número de contenidos con voto positivo, del mismo género que el contenido “de muestra” de la lista L que se está procesando.

Función *filtrar_contenidos* (*H, R, n, s, e*) **devuelve** Lista de contenidos

Para cada $c \in R$ **hacer**

Si ($c \in H$) **entonces** $R := \text{quitar}(R, c)$ **fsi**

Para cada $r \in H \wedge \neg \text{eliminado}$ **hacer**

Si (*r* tiene un voto negativo) **entonces**

Si ($\text{SIM}(c, r) > \beta$) **entonces**

$R := \text{quitar}(R, c);$

eliminado := cierto

fsi

fsi

fpara

fPara

// Calculamos el número de contenidos de la lista R que habrá en la recomendación final, aproximando al entero más cercano.

$k := (e/s) * n$

si ($|R| > k$) **entonces** $R := \text{los } k \text{ primeros contenidos de } R$ **fsi**

dev R

función

3.2.2.2 Caso 2

En este caso se calcula para cada uno de los usuarios del sistema un promedio de las valoraciones en base a la similitud de cada contenido del historial con el contenido a recomendar.

A continuación se describe en pseudo-código el algoritmo que genera la lista de predicciones para cada usuario del sistema.

Sean:

H: conjunto de historiales televisivos de usuario de todo el sistema; *c*: contenido nuevo.

Función *HISTORIAL_2* (*H, c*) **devuelve** Lista de Real

S: Lista de Real de tamaño $|H|$

Max: Real

Para cada $h \in H$ hacer

//Calculamos el promedio de valoraciones de los contenidos del historial según la similitud con el contenido "c"

Si ("promedio valoraciones" > VAL_POSITIVA) **entonces**

$S := \text{añadir}(S, h, \text{"promedio de similitudes"})$

Sino

$S := \text{añadir}(S, h, '0')$

fsi

fPara

dev S

función

3.2.3 Componente basado en el modelo bayesiano

Este componente utiliza un clasificador Bayesiano para realizar las recomendaciones basándose en un cálculo de probabilidades. Concretamente se ha utilizado el llamado *Naive Bayesian Classifier*, por diversas razones: puede tratar la incertidumbre, trabaja con información incompleta, representa el conocimiento de una forma natural facilitando las recomendaciones cooperativas y son mucho más eficientes a la hora de realizar los cálculos que otros modelos Bayesianos.

Para el cálculo de probabilidades, el componente utiliza básicamente dos tipos de información:

- **Estática.** Se trata de información que en principio no cambiará demasiado a lo largo del tiempo. Está formada básicamente por la información personal proporcionada explícitamente por el usuario.

Dinámica. Se trata de información que va actualizándose de forma automática por el sistema a medida que el usuario va viendo contenidos. Está formada por: los géneros de contenido de vídeo más relevantes para el usuario según su historial, y por el género de contenido con la probabilidad más alta de ser del gusto del usuario; información que se actualiza después de calcular la recomendación ($\arg \max_{y_k \in Y}$).

3.2.3.1 Aspectos teóricos

Como se ha dicho, el clasificador utilizado pertenece al grupo de clasificadores llamados *Naive Bayesian* y consiste en un árbol de dos niveles, donde el nodo raíz representa el género de contenido a recomendar y los hijos están formados de la siguiente forma: habrá un nodo para cada descriptor referente a los datos personales (aficiones, estado civil, estudios, ocupaciones, edad, etc.),

y dos nodos para el tipo (género) de series y películas que al usuario le ha gustado más en el pasado (más relevantes según su historial).

Sea Y la variable aleatoria discreta que representa la clase de ingredientes de contenido, y las variables discretas $X_1 \dots X_n$ que forman los hijos del árbol, tal como se ha comentado anteriormente. De acuerdo con la regla de Bayes tenemos:

$$P(Y = y_k | X_1 \dots X_n) = P(Y = y_k) * P(X_1 \dots X_n | Y = y_k) \quad (6)$$

Dónde, y_k representa un determinado ingrediente de contenido.

Asumiendo que las variables $X_1 \dots X_n$ son condicionalmente independientes, utilizando este tipo de clasificador podemos simplificar la ecuación (6) de la siguiente manera:

$$P(Y = y_k | X_1 \dots X_n) = P(Y = y_k) \prod_i P(x_i | y_k) \quad (7)$$

$P(Y = y_k)$ es calculado mediante la frecuencia de aparición del valor y_k en la base datos utilizada. En cambio, para calcular el valor de $P(x_i | y_k)$ utilizamos un estimador que solventa el problema de la escasez de información, evitando así que las probabilidades tengan valor nulo. Se define como la siguiente ecuación:

$$P(x_i | y_k) = \frac{n' + mL}{n + m} \quad (8)$$

Dónde n es el número de usuarios de la base de datos con $Y = y_k$, n' es el número de usuarios con $Y = y_k$ y $X_i = x_i$, m es el número de nodos hijo considerados en la clasificación y L es la inversa de la cantidad de valores diferentes que puede tomar X_i .

Más información sobre este tipo de clasificador en Mitchell, T.M [15] y en Chickering et al. ([10]).

3.2.3.2 Caso 1

Para generar la lista de recomendación, el sistema calcula para todos los ingredientes de contenido, las respectivas probabilidades de ser del gusto de un usuario en concreto. Luego, se recomendarán los contenidos que más se asimilen a las probabilidades obtenidas para cada ingrediente, utilizando un algoritmo similar al de la recomendación basada en contenido pero con algunas variaciones en la función de similitud. A continuación se describe en pseudo-código el algoritmo para generar la lista de contenidos a recomendar.

Sean:

B : base de datos sobre todos los usuarios; L : lista de $x_1 \dots x_n$ valores discretos de las variables $X_1 \dots X_n$ (nodos hijo) del usuario a recomendar; u : identificador de usuario; n : #contenidos que contendrá la lista de recomendación; l : límite máximo de contenidos a los que comparar (mejora eficiencia algoritmo)

Función BAYESIANO_1 (B, L, u, n, l) **devuelve** lista de n -contenidos

V : Lista de Real

p : Real

$p := 1$

Para $k = 1$ **hasta** #géneros considerados en Y **hacer**

Para cada $x_i \in L$ **hacer**

// Calcular la probabilidad en base a B con la ecuación 8

$p := p * P(x_i | y_k)$

fPara

$V := \text{añadir}(V, k, p)$

$p := 1$

fPara

// Normalizamos los valores de V para obtener las probabilidades de cada ingrediente de ser del gusto del usuario.

$V := \text{normalizar}(V)$

// Actualizamos la información referente a la última recomendación al usuario "u", con el ingrediente de contenido con más probabilidad de ser del gusto del usuario.

$B := \text{actualizar}(B, u, \max(V))$

// Para calcular la lista de recomendación utilizamos un algoritmo similar al Algoritmo1 con la diferencia de que ahora se buscarán contenidos similares a las probabilidades calculadas en V . Por lo tanto, la función SIM no hará uso del cálculo de coincidencias

Dev lista de n -contenidos

fFunción

3.2.3.3 Caso 2

En este caso el componente hará el mismo cálculo de probabilidades que se ha definido en el caso 1, con la diferencia de que ahora no se calculará una lista de recomendación. En vez de eso, se realiza un cálculo de la similitud entre los ingredientes del contenido a recomendar y las probabilidades del usuario.

Seguidamente se describe en pseudo-código el algoritmo para generar la lista de similitudes.

Sean:

B : base de datos sobre todos los usuarios; U : conjunto de identificadores de usuario; c : contenido nuevo.

Función BAYESIANO_2 (B, U, c) **devuelve** Lista de Real

V: Lista de Real

S: Lista de Real de tamaño $|U|$

p: Real

$p := 1$

Para cada $u \in U$ **hacer**

L := obtener nodos hijo de “*u*” a partir de “*B*”

V := inicializar(*V*)

Para $k = 1$ **hasta** #géneros considerados en *Y* **hacer**

Para cada $x_i \in L$ **hacer**

// Calcular la probabilidad en base a *B* con la ecuación 8.

$p := p * P(x_i | y_k)$

fPara

V := añadir (*V*, *k*, *p*)

$p := 1$

fPara

// Normalizamos los valores de *V* para obtener las probabilidades de cada ingrediente de ser del gusto del usuario.

V := normalizar(*V*)

// Actualizamos la información referente a la última recomendación al usuario “*u*”, con el ingrediente de contenido con más probabilidad de ser del gusto del usuario.

B := actualizar(*B*, *u*, max(*V*))

// Añadimos a la lista resultado la similitud entre las probabilidades y los ingredientes del nuevo contenido.

S := añadir(*S*, *u*, SIM(*V*, *c*))

fPara

dev *S*

fFunción

3.2.4 Componente de combinación

Se trata del componente del sistema de recomendación que combina los resultados de los tres componentes de recomendación descritos, en base a la confianza que cada usuario tiene respecto a las recomendaciones de los componentes para obtener la recomendación final.

3.2.4.1 Aspectos teóricos

El cálculo del nivel de confianza será independiente de cada usuario y se basará en dos descriptores: la cantidad de información disponible del usuario, y la información de retroalimentación. Este último se tratará de igual forma para los tres componentes, y se usará para conocer cuánto de exitosas son las recomendaciones de cada componente. Es decir, que cuantos más contenidos vea y vote positivamente el usuario, de los que ha recomendado uno de los componentes, más alto será su nivel de confianza para ese usuario. En

cambio, el parámetro de cantidad de información será calculada de distinta forma para cada componente:

- El basado en contenido dependerá del nivel de detalle con el que el usuario ha creado su perfil, es decir, de la cantidad de información opcional sobre sus preferencias que haya especificado.
- El basado en el historial dependerá del tamaño del historial del usuario.
- El Bayesiano dependerá de cuantos usuarios haya en el sistema, de la cantidad de información personal introducida por el usuario, y también del tamaño del historial del usuario.

Por lo tanto, se define el nivel de confianza como:

$$\text{Confianza}(s,u) = \alpha * \text{Éxito}(s,u) + (1 - \alpha) * \text{Disponibilidad}(s,u)$$

Éxito (s,u): Dado el usuario “u”, mide el éxito que ha obtenido el componente “s” teniendo en cuenta la proporción entre la puntuación total que ha dado el usuario (suma de valoraciones) y el número de recomendaciones de “s”.

Disponibilidad (s,u): Hay diferentes maneras de calcular esta medida dependiendo de cuál sea el componente “s”:

- Basado en el contenido: $\beta * \# \text{descriptores opcionales suministrados por } 'u' + \alpha$ (valor mínimo).
- Basado en el historial: hay una escala de valores según el $\#$ contenidos vistos por “u”.
(E.j. si $\#$ contenidos $\geq 5 \rightarrow 0.1$, y si $\#$ contenidos $\geq 50 \rightarrow 1$)
- Basado en el modelo Bayesiano: de igual forma que en el caso anterior, podemos hacer una escala de valores según el número de usuarios del sistema. Entonces tenemos que la disponibilidad es igual a:
 - $\delta * \text{“valor sobre } \# \text{ usuarios”}$
 - $+$
 - $\gamma * \# \text{descriptores personales de } 'u'$
 - $+$
 - $+ (1 - (\delta + \gamma)) * \text{“valor sobre } \# \text{ contenidos vistos por } 'u' \text{”}$

Se ha decidido que el rango de valores de la medida de confianza sea [0,1], por lo tanto, los valores devueltos por las funciones “Éxito” y “Disponibilidad”, incluyendo las constantes definidas, respetarán este rango de valores.

3.2.4.2 Caso 1

Dado un usuario, las listas de recomendación realizadas por los tres componentes de recomendación se combinarán basándose en una medida de

la confianza que habrá sido asignada a cada uno. Se dará más peso a las listas de los componentes con un nivel de confianza más alto. El resultado será una lista final de contenidos que formarán la recomendación para un determinado usuario.

Para que los niveles de confianza se vayan actualizando cada vez que el sistema realice una recomendación, se identificará cada contenido de la lista final con el componente que lo ha recomendado. Por lo tanto, a partir de las acciones del usuario sabremos el éxito de cada componente. A continuación se presenta el algoritmo para combinar las listas de recomendación.

Sean:

L_1 : lista de contenidos recomendados por el componente basado en contenido.

L_2 : lista de contenidos recomendados por el componente basado en el historial.

L_3 : lista de contenidos recomendados por el componente basado en el modelo Bayesiano.

n : #contenidos que contendrá la lista final

u : identificador del usuario al que hacemos la recomendación.

Función *GENERAR_LISTA_FINAL* (L_1, L_2, L_3, n, u) **devuelve** Lista de n -contenidos

*confianza*₁ = *CONFIANZA* ("contenido", u)

*confianza*₂ = *CONFIANZA* ("historial", u)

*confianza*₃ = *CONFIANZA* ("bayesiano", u)

// Comprobar que se cumple $\forall i, j \in \{1,3\} \wedge i \neq j, L_i \cap L_j = \emptyset$.

// En caso contrario, eliminamos los contenidos repetidos de la lista con menos confianza y lo marcamos como "contenido preferente".

// Calculamos el número de contenidos de la lista L_1 que habrá en la recomendación final, aproximando al entero más cercano.

$k := (\text{confianza}_1 / (\text{confianza}_1 + \text{confianza}_2 + \text{confianza}_3)) * n$

si ($|L_1| > k$) **entonces** $L_1' :=$ los k primeros contenidos de L_1 **fsi**

// Lo mismo para las listas L_2, L_3 .

dev $L_1' \cup L_2' \cup L_3'$

función

3.2.4.3 Caso 2

En este caso, el módulo de combinación es el encargado de decidir para cada usuario del sistema si le recomienda el contenido. Y esta decisión la realiza a partir de las similitudes proporcionadas por cada componente teniendo en cuenta los niveles de confianza.

El resultado será una lista de decisiones que determinará a qué usuarios se les recomienda el contenido. A medida que se va decidiendo para cada usuario si se le recomienda el nuevo contenido, se van guardando en los casos positivos los componentes que han decidido recomendarlo. De esta forma, dependiendo de la decisión del usuario, el nivel de confianza de los componentes se verá modificado.

Seguidamente se puede ver el algoritmo para decidir los usuarios a los que se les recomienda el contenido.

Sean:

L_1 : lista de medidas de similitud de todos los usuarios calculadas por el componente basado en contenido.

L_2 : lista de medidas de similitud de todos los usuarios calculadas por el componente basado en el historial.

L_3 : lista de medidas de similitud de todos los usuarios calculadas por el componente basado en el modelo Bayesiano.

U : conjunto de identificadores de todos los usuarios del sistema.

Función *GENERAR_LISTA_DECISIÓN* (L_1, L_2, L_3, U) **devuelve** Lista de Bool

B: Lista de Bool de tamaño $|U|$

*confianza*₁ = *CONFIANZA* ("contenido", u)

*confianza*₂ = *CONFIANZA* ("historial", u)

*confianza*₃ = *CONFIANZA* ("bayesiano", u)

Para cada $u \in U$ **hacer**

1. Convertimos las "similitudes" a predicciones (escala del 1-5)

2. Normalizamos las confianzas de cada componentes

3. Calculamos la predicción conjunta según la predicción de cada componente y el peso de su confianza.

Si (predicción = "positiva") entonces

$B := \text{añadir}(B, u, \text{true})$

Sino

$B := \text{añadir}(B, u, \text{false})$

fPara

dev B

función

3.3 Evaluación del sistema de recomendación

El hecho de que el sistema de IntegraTV Home está en estos momentos en fase de pruebas, implica que no haya suficientes usuarios para realizar una evaluación valiosa de la precisión del sistema de recomendación. Por este motivo, se ha intentado evaluar el sistema utilizando bases de datos públicas, que son normalmente la base de pruebas de la mayoría de sistemas de recomendación de contenido vídeo.

Dado que el sistema de recomendación se ha diseñado especialmente en base al entorno de aplicación de IntegraTV Home, dónde los datos disponibles tanto de usuarios como de contenidos están más elaborados, se hace difícil realizar una evaluación precisa utilizando bases de datos públicas, donde la información disponible es mucho más pobre a pesar de la gran cantidad de usuarios y contenidos de que disponen. Por este motivo, no ha sido posible realizar una comparativa real de la precisión del sistema diseñado con otros sistemas de recomendación actuales.

No obstante, se ha podido evaluar de forma interna, utilizando una base de datos externa tanto para los contenidos como para la información de usuario. Aunque con este método sólo se ha evaluado la precisión de las predicciones desde una perspectiva probabilística que, como se ha comentado anteriormente, consiste en predecir un valor esperado del voto del usuario sobre un contenido y calcular el error de desviación respecto a la valoración real. Este tipo de recomendación encaja en parte en la recomendación en el Caso 2, dónde dado un nuevo contenido se predice a qué usuarios les puede gustar más.

3.3.1 Metodología

La metodología seguida para realizar la evaluación ha sido la siguiente:

1. **Obtener la base de datos (*dataset*) desde una fuente externa.** Se ha utilizado el *dataset* de MovieLens⁶, un sistema de recomendación de películas desarrollado por el laboratorio de investigación GroupLens de la Universidad de Minnesota. Se escogió esta base de datos para la evaluación, porque posee unas características similares al del entorno para el cual se ha diseñado el sistema de recomendación. Las características del *dataset* son:
 - 100,000 valoraciones con valores dentro del rango [1,5] de 943 usuarios en 1682 contenidos de vídeo.
 - Cada usuario ha votado al menos 20 contenidos.
 - Información personal del usuario sobre edad, género, ocupación y código postal.
2. **Utilizar la base de datos IMDB para obtener más información sobre los contenidos** del *dataset* como géneros, actores, directores, duración y nacionalidades. Durante la recopilación de datos un 5% de contenidos no tenían información o era inválida, por lo que se procedió a eliminarlos de las pruebas ya que generarían ruido durante la evaluación.

⁶ Más información en la página oficial: www.movielens.org (visitado el 10/01/08)

3. **Dividir el *dataset* en dos partes: la de entrenamiento (base) y la de prueba (test).** Para la evaluación se han realizado dos pruebas diferentes: la primera dividiendo el 80% en base y el 20% en test (la más utilizada en pruebas de este tipo), donde el sistema tiene mayor información sobre los historiales de usuario; y la segunda prueba con una división del 50% en base y 50% en test, donde los usuarios son distintos en ambas partes, hecho que permite simular un entorno más real con usuarios que no poseen historial de visionado.
4. **Entrenar el sistema con el *dataset* base.** En las dos pruebas antes mencionadas el procedimiento para entrenar el sistema de recomendación sigue los mismos pasos. Básicamente consiste en utilizar una parte del *dataset* para crear los perfiles de usuario e inicializar la red Bayesiana, y con el resto, se predice para cada valoración usuario-contenido el valor esperado de la valoración que comparándolo con el valor real se obtiene el éxito de cada componente.
5. **Realizar las pruebas con el *dataset* test.** En ambos casos (50-50 / 80-20) se han ordenado las valoraciones por orden de fecha para realizar las predicciones en un orden determinado, como si de una simulación real se tratase. Para cada valoración de usuario se obtiene la predicción del valor esperado de cada componente que luego son combinadas según los niveles de confianza del usuario para obtener la predicción final.
6. **Calcular la precisión del sistema de recomendación** utilizando una métrica del error para cada una de las pruebas. La métrica utilizada ha sido el MAE (Mean Absolute Error), ya que es una de las métricas más utilizadas para el cálculo del error de las predicciones de los sistemas de recomendación. La fórmula matemática del MAE es:

$$|E_{MAE}| = \frac{1}{N} \sum_{i=1}^N |x_i - \hat{x}_i|$$

Donde x_i es el valor esperado y \hat{x}_i es el valor real.

3.3.2 Conclusiones de la evaluación

Como se ha comentado anteriormente, el tipo de metodología de evaluación que se ha seguido sólo ha permitido evaluar el sistema de recomendación desde un punto de vista interno. De manera que se han hecho varias versiones con configuraciones diferentes de los componentes del sistema de recomendación y se ha observado cuales son las que obtienen predicciones más precisas en base a la métrica MAE.

En las pruebas con el *dataset* dividido en 50-50, donde se presenta un entorno con más escasez de datos, se observan unos resultados que siguen el patrón esperado. Es decir, se observa como la utilización del componente de

combinación mejora la precisión de las predicciones respecto a las de los componentes de recomendación por separado. En cambio, con el *dataset* dividido en 80-20 el patrón que siguen los resultados es diferente, ya que se observa como las predicciones finales tienen igual o peor precisión que las predicciones del componente basado en el historial.

En resumen, la conclusión obtenida a partir de los resultados de las pruebas realizadas es que el sistema diseñado funciona como debería cuando el entorno presenta escasez de datos, que es cuando la mayoría de sistemas de recomendación de vídeo actuales obtienen peores resultados. Por lo contrario, se obtienen peores o los mismos resultados que si únicamente utilizáramos un sistema basado en casos en entornos donde la densidad de información disponible es más alta.

Dado que la metodología de evaluación utilizada no es la más adecuada para obtener unas conclusiones fiables sobre la precisión del sistema de recomendación en un entorno real, sería necesario evaluar el sistema de recomendación con usuarios reales para determinar si el sistema cumple con las expectativas esperadas.

Teniendo en cuenta que el sistema fue diseñado con el fin de mejorar la precisión de las recomendaciones en entornos con una baja densidad de información disponible, donde la mayoría de sistemas de recomendación actuales generan recomendaciones poco precisas, se podría decir a priori que el diseño del sistema de recomendación híbrido basado en confianzas cumple con los requisitos planteados en este proyecto.

Capítulo 4

4 Desarrollo del sistema de información

4.1 Metodología empleada: Métrica 3

Métrica 3 es una metodología que abarca el desarrollo de sistemas de información para las distintas tecnologías que conviven en la actualidad. Además, contempla también los aspectos de gestión que aseguran que un proyecto cumpla con sus objetivos en términos de calidad y coste.

Esta versión conserva la adaptabilidad, flexibilidad y sencillez de su antecesora: la versión 2.1., además de facilitar la automatización de las actividades propuestas en la estructura a través de herramientas, muchas de ellas disponibles en la página Web del Consejo Superior de Informática⁷. Sin embargo, su principal aportación consiste en su compatibilidad con los últimos estándares de Calidad e Ingeniería del Software, además de cubrir tanto desarrollos estructurados como orientados a objeto.

Métrica 3 tiene un enfoque orientado al proceso, descompone cada uno de los procesos en actividades, y éstas a su vez en tareas. Para cada tarea se escribe su contenido haciendo referencia a sus principales acciones, productos, técnicas, prácticas y participantes.

Los procesos de la estructura principal de la metodología son los siguientes:

- PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN
- DESARROLLO DE SISTEMAS DE INFORMACIÓN
 - Estudio de viabilidad del sistema (Proceso EVS)
 - Análisis del sistema de información (Proceso ASI)
 - Diseño del sistema de información (Proceso DSI)
 - Construcción del sistema de información (Proceso CSI)
 - Implantación y Aceptación del sistema (Proceso IAS)

⁷ Web oficial: www.csi.map.es (visitado el 10/01/08)

- MANTENIMIENTO DE SISTEMAS DE INFORMACIÓN

Se escogió esta metodología, porque su objetivo es ofrecer un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software para mejorar la productividad, para que se genere un producto bastante flexible, con capacidad de reutilización en la medida de lo posible, y que sea de fácil uso y mantenimiento.

La estructura de la metodología es adaptable, por lo que, no todos los procesos y actividades deben ejecutarse en su totalidad, sino que más bien, requiere una adaptación en función del proyecto y de la organización. Durante el desarrollo del software se ha seguido un ciclo de vida en cascada y retroalimentado, es decir, manteniendo el orden de ejecución de cada actividad y con varias iteraciones del ciclo de vida para realizar un desarrollo progresivo que asegure la robustez del sistema en todo momento.

En concreto al ser un proyecto académico y formar parte de una aplicación de la empresa debidamente analizada, se han seguido las siguientes actividades de la metodología:

- Proceso ASI.
 - Definición del sistema.
 - Modelo de Requisitos.
 - Análisis de los casos de uso.
 - Definición de interfaces de usuario.

- Proceso DSI
 - Definición de la arquitectura del sistema.
 - Diseño de la arquitectura de soporte.
 - Diseño de los casos de uso reales.
 - Diseño físico de los datos.
 - Verificación y aceptación de la arquitectura del sistema.

- Proceso CSI
 - Preparación del entorno de generación y construcción.
 - Generación del código.
 - Ejecución de las pruebas unitarias.
 - Ejecución de la pruebas de integración.
 - Ejecución de las pruebas del sistema.

4.2 Análisis del Sistema de Información

4.2.1 Definición del sistema

4.2.1.1 Alcance del sistema

El proyecto consiste en extender la funcionalidad del sistema IntegraTV Home para que permita la recomendación personalizada de contenidos de vídeo y la gestión de los datos necesarios para llevar a cabo las recomendaciones. Principalmente se gestionarán los perfiles de los usuarios a quienes irán destinadas las recomendaciones y la gestión de los datos de los contenidos de vídeo a recomendar.

Dado que el proyecto tiene como objetivo extender la funcionalidad de un sistema existente, la determinación del alcance del sistema viene definido por la aplicación a extender. Como se ha descrito anteriormente, IntegraTV Home es un sistema de componentes distribuidos basado en la arquitectura cliente – servidor. La integración del sistema de recomendación, implicará modificar algunos de los componentes que conforman el sistema. Concretamente serán modificados: la aplicación cliente (FrontEnd), la herramienta de gestión de contenidos (BackOffice) y finalmente el servicio del Servidor de Integra.

Como uno de los objetivos del proyecto es construir la extensión del sistema como si se tratara del desarrollo de un sistema independiente se analizará y diseñará como tal. Por esta razón se utilizará el término *Servicio de Recomendación* para referirse a la extensión del sistema que se ha desarrollado para este proyecto.

Respecto a los usuarios que interactuarán con el *Servicio de Recomendación* se pueden distinguir dos roles diferentes:

- **Cliente.** Representa al usuario que tiene contratado el servicio de IntegraTV Home y al cual irán destinadas las recomendaciones.
- **Administrador de Contenidos.** Representa al empleado de la empresa que se encarga de la gestión de los contenidos que se ofrecen a los clientes.

4.2.1.2 Entorno tecnológico para el desarrollo

Hardware necesario

Dado que el proyecto consiste en extender la funcionalidad de un aplicativo que ya está en funcionamiento real, inicialmente, no será necesaria la adquisición de nuevo hardware para el desarrollo del sistema.

Actualmente se utiliza un servidor para albergar los procesos de la parte del servidor y la gestión de la base de datos de contenidos. Por lo que se refiere a la parte del cliente, dispone en su hogar de un media center con las

características necesarias para: reproducir los contenidos, recibir y enviar datos del servidor (implica conexión a Internet) y dar soporte a los diferentes componentes de la parte cliente. Internamente a la empresa, se utilizan varios ordenadores para la administración del sistema de IntegraTV, que únicamente pueden ser usados por los usuarios con permisos de administrador.

Una vez se construya el servicio de recomendación y se integre completamente con la versión actual de *IntegraTV Home*, seguramente será necesario por cuestiones de eficiencia, añadir un servidor más al sistema para separar los componentes de la parte del servidor, donde se realizan los procesos de negocio, de la gestión de la base de datos.

Software necesario

Para facilitar la tarea de integración y minimizar los problemas de incompatibilidad con IntegraTV se ha utilizado el mismo entorno de programación que el utilizado para la construcción y mantenimiento del sistema actual.

Para la programación del sistema se utiliza el lenguaje de programación **C#** (*C sharp* o *C sostenido*), que es orientado a objetos y está desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

Como entorno de programación se ha utilizado Microsoft Visual Studio 2005 porque ofrece una gran variedad de herramientas que facilitan la programación de proyectos de un tamaño considerable.

Y para el almacenamiento de los datos se usará MS SQL Server 2005, que es el sistema gestor de base datos desarrollado por Microsoft, y con el que actualmente se gestiona el almacenamiento de los datos de los contenidos.

4.2.2 Modelo de Requisitos

4.2.2.1 Especificación de los requisitos

A continuación se detallan los **requisitos funcionales** que se deben cumplir para el desarrollo del servicio de recomendación agrupados por similitud de funcionalidad. (En cursiva están los términos descritos en el glosario que se puede ver en el Anexo I)

Gestión del perfil:

1. Respecto a los datos de usuario⁸.

⁸ Siempre que se utilice el término usuario de forma general, hará referencia al usuario con rol de cliente.

- No existe ningún dato que un usuario deba proporcionar al sistema obligatoriamente.
 - Para los *descriptores* con múltiples valores se permitirá un máximo de 5 valores.
 - *Información preferencias.*
 - El usuario podrá proporcionar la siguiente información sobre sus preferencias: géneros, directores, actores, nacionalidades y duración que más le guste; así como si prefiere las novedades o los clásicos.
 - El usuario también podrá proporcionar una valoración numérica con rango [0,1] de los *ingredientes de contenido*. En caso contrario se le asignará unos valores por defecto.
 - Cada usuario tendrá la opción de ordenar los descriptores de preferencias (géneros, directores, actores, nacionalidades y duración) por orden de importancia, asignándoles diferentes pesos a cada uno. Si no se ordenan por el usuario, se asignarán unos pesos por defecto.
 - *Información personal.*
 - El usuario podrá proporcionar la siguiente información personal: oficio/s, afición/es, estado civil, edad, sexo, clase social, estudios, nacionalidad, número de hijos, localidad y provincia.
2. En cuanto a los permisos de usuario y restricciones.
- Todo usuario podrá crear un nuevo perfil y así poder utilizar el servicio de recomendación.
 - Cada usuario tendrá asociado un único perfil.
 - Una vez creado un perfil, el usuario podrá modificar dicho perfil siempre que lo desee.
 - Un usuario puede indicar que no quiere utilizar el servicio de recomendación invalidando su perfil de usuario.

Gestión de contenidos:

1. En cuanto a los datos del *contenido*.
 - Los contenidos a recomendar deberán contener los siguientes datos: título, año, duración, géneros, actores más importantes, director/es,

nacionalidad/es, si es novedad o bien un clásico, y también una valoración de los *ingredientes de contenido* que lo describen.

2. Respecto a los permisos y restricciones.

- Un usuario del grupo de administradores del sistema podrá gestionar el alta, modificación y baja de los contenidos a recomendar.
- Las fichas de los contenidos dados de baja no se borrarán definitivamente del sistema sino que serán marcados como *contenidos no disponibles*.
- No se podrán crear dos contenidos con el mismo título y fecha de producción.
- El sistema tendrá acceso a la información de todos los contenidos, tanto los dados de baja como a los *contenidos disponibles*.

Gestión de la recomendación:

- El número de contenidos recomendados a un usuario, en la recomendación en el caso 1, estará limitado a número máximo que permita calcular la recomendación de forma eficiente. Por defecto el número de contenidos a recomendar será aquel que permita el cálculo óptimo de la recomendación en cada petición de usuario.
- Un usuario podrá pedir ser recomendado siempre y cuando posea un perfil asociado.
- Los usuarios podrán elegir el número máximo de contenidos que desean recibir por recomendación, dentro de un rango de valores cercanos al óptimo.
- Después de ver un contenido el usuario deberá realizar una *valoración de contenido*. Se ofrecerá la posibilidad de que sea una valoración nula por si al usuario no le apetece valorar el contenido en ese momento. Un ejemplo de valoración nula sería “no me interesa valorar”.
- Periódicamente, para cada usuario que no haya pedido recientemente una recomendación, se calcularán recomendaciones en el caso 1 de forma transparente al usuario.
- Periódicamente se calcularán recomendaciones en el caso 2 para cada contenido introducido recientemente por el administrador de contenidos. Cada contenido sólo será recomendado una vez.
- Periódicamente se aplicará un *factor de olvido* a cada uno de las valoraciones de contenido de los historiales de usuario. De modo que las valoraciones de los usuarios irán perdiendo relevancia con el paso del

tiempo. Este requisito viene dado por el diseño del sistema de recomendación descrito en el capítulo anterior, en particular, por el componente basado en el historial (ver capítulo 3.2.2).

- Un usuario podrá pedir ser recomendado siempre y cuando no exista ninguna petición suya en curso.
- Se requiere mantener un historial de las diferentes peticiones de recomendación realizadas, tanto las que provienen por petición del usuario como las recomendaciones automáticas que realiza el sistema periódicamente.
- Las peticiones de recomendación se podrán hacer en cualquier momento sin tener que esperar a los resultados de la recomendación para seguir utilizando el sistema. Es decir, las peticiones de recomendación serán asíncronas.
- El usuario podrá indicar el nivel de interés para cada uno de los contenidos recomendados antes de disponer de ellos. Además podrá indicar si ya ha visto alguno de los recomendados por algún otro medio. De esta manera se da la posibilidad de priorizar la descarga de los contenidos de más interés y evitar descargas innecesarias, lo que supondría una optimización en el sistema de distribución de contenidos actual (ver capítulo 2.2).

El nivel de interés y las valoraciones de contenidos se utilizarán como información de retroalimentación para el cálculo de las recomendaciones.

Entre los **requisitos no funcionales** cabe destacar los requisitos de integración que definen las restricciones a tener en cuenta durante el desarrollo de las nuevas funcionalidades propias del servicio de recomendación. Se ha definido el siguiente requisito:

- Se precisa minimizar los cambios en el sistema actual, sobretodo en cuanto a la parte de la lógica de negocio, de modo que el nuevo servicio de recomendación funcione de forma independiente en la medida de lo posible al funcionamiento global de la aplicación.

4.2.2.2 Identificación de actores

Como ya se ha descrito durante la descripción del alcance del sistema, existen dos perfiles de usuario claramente distintos. A continuación se describen el rol de cada uno de los actores identificados relacionados con las funcionalidades propias del servicio de recomendación.

- Administrador de contenidos (*AdminContenido*). Se encarga de la gestión de los contenidos utilizados para calcular las recomendaciones,

de manera que el sistema de recomendación siempre tenga la lista de contenidos disponibles y datos de baja actualizada, evitando que se recomienden contenidos no disponibles o con información errónea. Es un usuario con permisos para dar de alta y de baja contenidos, así como modificar los datos de los contenidos. Forma parte del personal interno de la empresa.

- *Cliente*. Representa al usuario del sistema que ha contratado los servicios de *IntegraTV Home*. Se trata del usuario que utilizará directamente el servicio de recomendación, y gestionará los datos de su perfil. Esto implica que cada usuario es responsable de la veracidad de los datos introducidos al sistema.

4.2.2.3 Identificación de casos de uso

En este apartado se van describir brevemente los casos de uso identificados durante el proceso de análisis del sistema de información. Los casos de uso se identifican a partir de las funcionalidades que ofrece un sistema definidas en el modelo de requisitos. Una buena guía para identificar los casos de uso de un sistema se encuentra en Ben-Menachem [3].

En la Ilustración 9 se observa el diagrama de casos de uso que representa la interacción de los actores del sistema con los casos de uso identificados.

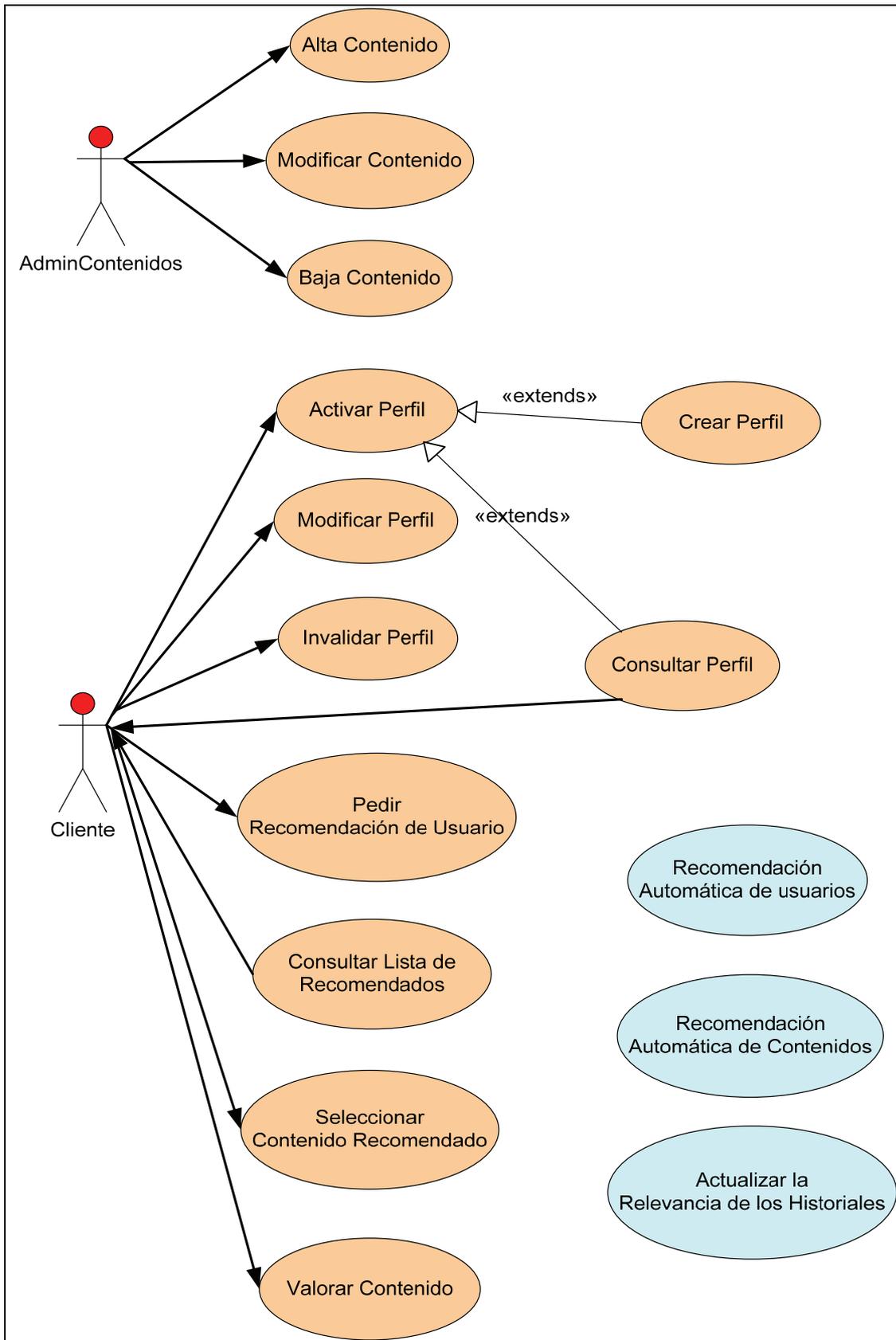


Ilustración 9: Diagrama de casos de uso del *Servicio de Recomendación*.

La utilización del color es para distinguir entre diferentes tipos de casos de uso. Los de color naranja son los más comunes, en los que los actores inician el caso de uso. Los de color azul son los llamados casos de uso temporales⁹, y en general no son iniciados por ningún actor del sistema.

La relación *extends* que se observa en el diagrama, indica que los casos de uso Crear y Consultar Perfil serán iniciados por el de Activar Perfil en determinados escenarios. El caso de Crear Perfil, se inicia cuando el usuario que quiere activar su perfil no tiene asociado un perfil. Esto sucederá normalmente la primera vez que utilice el servicio de recomendación. El otro caso se iniciará siempre que la activación del perfil sea satisfactoria.

A continuación se hace una breve descripción de los casos de uso identificados:

- **Alta / Modificación / Baja de un contenido.** Casos de uso iniciados por el actor *AdminContenido* con los que lleva a cabo todo lo relacionado con la gestión de contenidos de vídeo. Como es dar de alta un nuevo contenido, modificar los datos de uno existente y dar de baja un contenido para indicar que ya no está disponible para ofrecer a los usuarios.
- **Activar / Crear / Modificar / Invalidar / Consultar perfil.** Casos de uso iniciados por el actor *Cliente* con los que lleva a cabo las tareas de gestión de su perfil. Como es activar el perfil para indicar que quiere utilizar el servicio de recomendación, modificar y consultar los datos de su perfil e invalidar su perfil para indicar que quiere dejar de recibir recomendaciones.
- **Pedir recomendación de usuario.** Un usuario puede pedir que se le haga una recomendación en cualquier momento. La recomendación consiste en recibir una lista de contenidos que no supere el número máximo fijado por el usuario en el momento de hacer la petición. Este caso de uso se relaciona directamente con la *recomendación en el Caso 1* cuyo resultado es la *top list* (ver capítulo 3.2).
- **Consultar lista de recomendados.** Un usuario puede consultar la lista de contenidos recomendados cuando lo desee. Esta lista estará formada por los contenidos que se le han recomendado pero que aun no están disponibles para reproducir, y los contenidos recomendados que se han descargado completamente.
- **Seleccionar contenido recomendado.** Caso de uso en el que el usuario selecciona un contenido de la lista de recomendados por descargar, para ver información del contenido y mostrar su nivel de

⁹ Los casos de uso temporales son los que se inician periódicamente de forma predefinida por el sistema.

interés. Dependiendo del interés que indique el usuario, el contenido recomendado será descargado o no.

- **Valorar contenido.** Cada vez que un usuario reproduce completamente o parcialmente un contenido, se le pide una valoración que luego utiliza el sistema de recomendación para actualizar las confianzas de cada componente.
- **Recomendación automática de usuarios.** Periódicamente de forma transparente a los usuarios se les realizarán recomendaciones en el Caso 1 (ver capítulo 3.2).
- **Recomendación automática de contenidos.** Periódicamente, para los contenidos dados de alta recientemente se realizarán recomendaciones en el Caso 2 (ver capítulo 3.2).
- **Actualizar la relevancia de los historiales de usuario.** Periódicamente, se actualiza la relevancia de los contenidos de los historiales de usuario según un factor de olvido. Éste consiste en reducir la importancia de un contenido del historial para el usuario a medida que pasa el tiempo.

4.2.2.4 Especificación de casos de uso

A continuación se mostrará una especificación detallada de los casos de uso más representativos del Servicio de Recomendación.

Alta de contenido

En la Tabla 1 se muestra la especificación del caso de uso Alta de contenido.

Identificador:	AltaContenido	
Actores:	AdminContenido	
Precondiciones:		
Poscondiciones:	<ul style="list-style-type: none"> ▪ Se crea un nuevo contenido disponible para ser recomendado. 	
Descripción:	El administrador de contenidos da de alta un nuevo contenido con los datos introducidos.	
Flujo Básico:	AdminContenido	Sistema
	1. Indica al sistema que quiere dar de alta un nuevo contenido	
		2. Muestra un formulario para rellenar los datos del contenido

	3. Introduce los datos del contenido: título, año, duración, géneros, actores más importantes, director/es, nacionalidad/es y también una valoración de los ingredientes de contenido	
		3. Verifica que los datos introducidos sean correctos y que no existe un contenido con el mismo identificador
		4. Pide confirmación del usuario para crear el contenido
	5. Confirma la creación del contenido.	
		6. Finalmente crea el nuevo contenido marcándolo como “nueva entrada” y notifica al usuario que la creación ha sido satisfactoria
Flujo Alternativo:		
	5. Cancela la creación del contenido y termina el caso de uso	
Excepciones:	<ul style="list-style-type: none"> ▪ Existe algún error en los datos del contenido: el sistema notifica el error y vuelve al paso 2. ▪ Durante la creación del contenido ha ocurrido algún error: el sistema avisa al administrador y se termina el caso de uso. 	

Tabla 1: Especificación del caso de uso: Alta de contenido.

Activar perfil de usuario

En la Tabla 2 se muestra la especificación del caso de uso Activar perfil de usuario.

Identificador:	ActivarPerfil
-----------------------	---------------

Actores:	Cliente	
Precondiciones:	<ul style="list-style-type: none"> ▪ El usuario no tiene activado el perfil de usuario. 	
Poscondiciones:	<ul style="list-style-type: none"> ▪ Se activa el perfil del usuario con los datos introducidos por el cliente. 	
Descripción:	El cliente activa su perfil de usuario para poder utilizar los servicios del sistema de recomendación.	
Flujo Básico:	Usuario	Sistema
	1. Indica al sistema que quiere activar el perfil de usuario	
		2. Comprueba que no tiene asignado ningún perfil de usuario
		3. Notifica al usuario que es necesario crear un nuevo perfil
	4. Indica que acepta crear su perfil	
		5. Crea el nuevo perfil {caso de uso <i>CrearPerfil</i> }
Flujo Alternativo 1:		
		2. Comprueba que tiene asignado un perfil de usuario
		3. Activa el perfil existente y muestra al usuario el perfil que ha acaba de crear
	4. Confirma y termina el caso de uso	
Flujo Alternativo 2:		
	4. Rechaza la creación del perfil de usuario	
Excepciones:		

Tabla 2: Especificación del caso de uso: Activar perfil.

Crear perfil de usuario (extensión de Activar perfil de usuario)

En la Tabla 3 se muestra la especificación del caso de uso Crear perfil de usuario.

Identificador:	CrearPerfil	
Actores:	Cliente	
Precondiciones:	▪ El usuario no tiene asociado ningún perfil de usuario.	
Poscondiciones:	▪ Se crear un nuevo perfil de usuario con los datos introducidos por el cliente.	
Descripción:	El cliente crea su perfil de usuario para poder completar la activación de su perfil.	
Flujo Básico:	Usuario	Sistema
		1. Muestra el formulario al usuario para que complete su perfil.
	2. Introduce los datos relativos a sus preferencias e información personal	
		3. Valida los datos introducidos por el usuario
		4. Almacena el perfil de usuario y notifica al usuario que el perfil se ha creado correctamente
	5. Confirma y termina el caso de uso	
Excepciones :	▪ Algunos de los datos introducidos no es correcto: el sistema avisa al usuario de que existe algún error y vuelve al paso 2.	

Tabla 3: Especificación del caso de uso: Crear perfil.

Pedir recomendación de usuario

En la Tabla 4 se muestra la especificación del caso de uso Pedir recomendación de usuario.

Identificador:	PedirRecomendación
Actores:	Cliente

Precondiciones:	<ul style="list-style-type: none"> El cliente tiene activado el perfil de usuario. 	
Poscondiciones:	<ul style="list-style-type: none"> Se registra una petición de recomendación del usuario y se calcula una recomendación para dicha petición. 	
Descripción:	El cliente solicita que se le recomienden un número determinado de contenidos.	
Flujo Básico:	Usuario	Sistema
	1. Pide que se le haga una recomendación indicando cuantos contenidos como máximo quiere que se le recomienden	
		2. Valida que el número de contenidos máximo este dentro del rango acotado.
		3. Comprueba que no exista una petición pendiente del usuario en cuestión
		4. Pide confirmación al usuario para seguir con la petición de recomendación con el número de contenidos indicado
	5. Acepta seguir con la petición de recomendación	
		6. Registra la petición de recomendación de usuario y se notifica al usuario que la petición está en proceso
	7. Confirma y deja de participar en el caso de uso	
		8. Comprueba que no existan más peticiones de usuario pendientes o en curso
		9. Calcula la recomendación de la nueva petición de usuario
		10. Finaliza la petición de usuario y le asigna el resultado de la recomendación

Flujo Alternativo 1:		
		5. Comprueba que existen peticiones de usuario pendientes
		6. Encola la nueva petición de usuario
Flujo Alternativo 2:		
	4. Cancela la petición de recomendación y termina el caso de uso	
Excepciones :	<ul style="list-style-type: none"> ▪ El número de contenidos máximo indicado por el usuario no está dentro del rango definido: el sistema avisa al usuario que el número elegido no es válido y vuelve al paso 1. ▪ Existe una petición pendiente del usuario: el sistema avisa al usuario que no se le harán nuevas recomendaciones hasta que no se termine la petición que está pendiente o en curso. 	

Tabla 4: Especificación del caso de uso: Pedir recomendación de usuario.

4.2.3 Modelo Conceptual

En este apartado se describe el modelo conceptual del sistema de recomendación, donde están representados los conceptos y asociaciones más relevantes extraídos a partir del análisis del sistema. Pretende ser una visión general de los conceptos utilizados durante el análisis, por lo que no se ha definido en detalle cada uno de los conceptos.

En la Ilustración 10 se observa la representación del modelo conceptual. Dado que se ha utilizado la metodología Métrica 3 para el desarrollo del sistema, se ha utilizado un diagrama de clases en UML para su representación. No obstante, el modelo conceptual aquí presentado, vendría a ser como la ontología de un sistema basado en el conocimiento.

4.2.4 Análisis de los casos de uso

En este apartado se describirán las clases necesarias para la realización de los casos de uso especificados en detalle anteriormente. Las clases que se han identificado en la fase del análisis se clasifican según los siguientes tipos:

- **Clases Vista.** Representan las interfaces de usuario, por lo que, deben permitir que el usuario introduzca o seleccione toda la información necesaria para la realización de la tarea que el usuario desee hacer, además de realizar los primeros procesos de validación para que la información que se va a procesar tenga al menos una estructura válida.

Estas clases mantienen una estrecha relación con las clases *Controlador*, ya que a éstas es a quien se le envían los datos recogidos de las pantallas del usuario, y también se le indica la acción que se desea realizar. Por otra parte, las clases *Controlador*, devuelven a las interfaces de usuario la información necesaria para ser mostrada al usuario, como por ejemplo, los mensajes informativos, de error o de advertencia.

- **Clases Controlador.** Estas clases están estrechamente relacionadas, como se comentó anteriormente, con las clases *Vista*, ya que de éstas últimas, recibe la información emitida por el usuario necesario para el procesamiento de la acción que se pretende realizar. Se podría entonces expresar, que las clases *Controlador* son el medio por el cual, las interfaces comunican a la aplicación lo que desean hacer.

El objetivo de las clases controlador es asegurar que la realización de los casos de uso sea correcta mediante la interacción con las clases entidad.

- **Clases Entidad.** Son las clases que encapsulan información del dominio de la aplicación que en la mayoría de los casos necesitará ser persistente. Generalmente las clases entidad se relacionan directamente con clases del modelo conceptual, aunque no siempre conservan las mismas asociaciones.

Para distinguir los distintos tipos de clase se utiliza la siguiente nomenclatura:

- ***VistaNombre*** para identificar las Clases de Interfaz de Usuario;
- ***ControlNombre*** para identificar las Clases de Control;
- ***Nombre*** para referirse a las Clases Entidad.

4.2.4.1 Caso de uso: Alta de contenido

Las clases identificadas para este caso de uso son:

- *VistaAltaContenido*: Es el formulario donde el administrador de contenidos introducirá los datos del contenido.
- *ControlAltaContenido*: Se encarga de comprobar que no exista un contenido con el mismo título y fecha, y en el caso de no existir crea el contenido con los datos proporcionados por la vista.
- *Contenido*: Representa al nuevo contenido creado, y encapsula toda la información relativa a un contenido del sistema.

En la Ilustración 11 se puede observar el diagrama de secuencia del flujo básico del caso de uso aquí descrito.

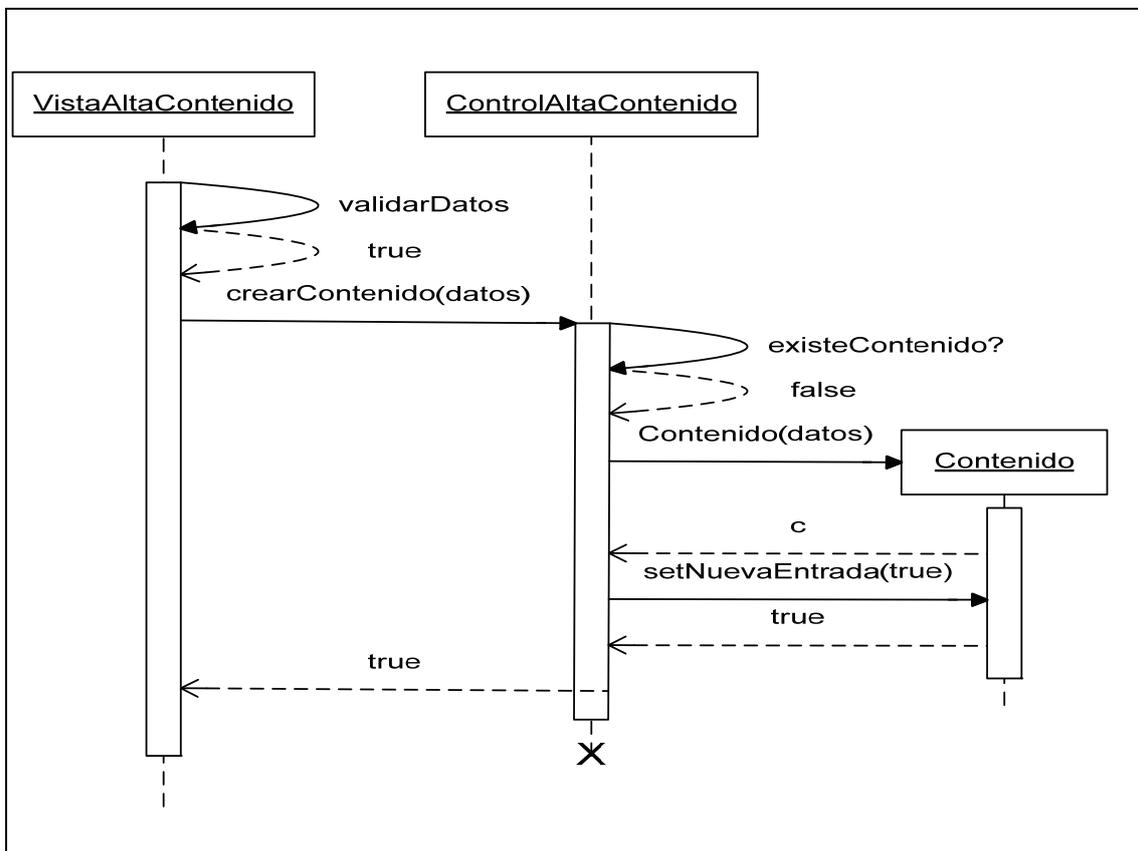


Ilustración 11: Diagrama de secuencia del flujo básico de caso de uso AltaContenido

4.2.4.2 Caso de uso: Activar perfil de usuario

Las clases identificadas para este caso de uso son:

- *VistaActivarPerfil*: Es el formulario que utiliza el cliente para indicar que quiere activar el servicio de recomendación
- *ControlaCrearPerfil*: Se encarga de gestionar el proceso de activación del perfil de usuario.
- *Cliente*: Representa el cliente que tiene desactivado el perfil de usuario, y encapsula la información relativa al usuario cliente de IntegraTV.

4.2.4.2.1 Extensión: Crear perfil de usuario

Las clases identificadas para este caso de uso son:

- *VistaCrearPerfil*: Es el formulario que utiliza el cliente para introducir la información personal y sus preferencias.
- *ControladorCrearPerfil*: Se encarga de gestionar el proceso de creación del perfil de usuario.
- *Perfil*: Representa el perfil del cliente, y encapsula los datos de las preferencias y información personal proporcionados por el cliente.
- *Cliente*: Representa el cliente con el que se asocia el perfil de usuario.

En la Ilustración 12 se puede observar el diagrama de secuencia de las distintas clases identificadas del flujo básico del caso de uso aquí descrito.

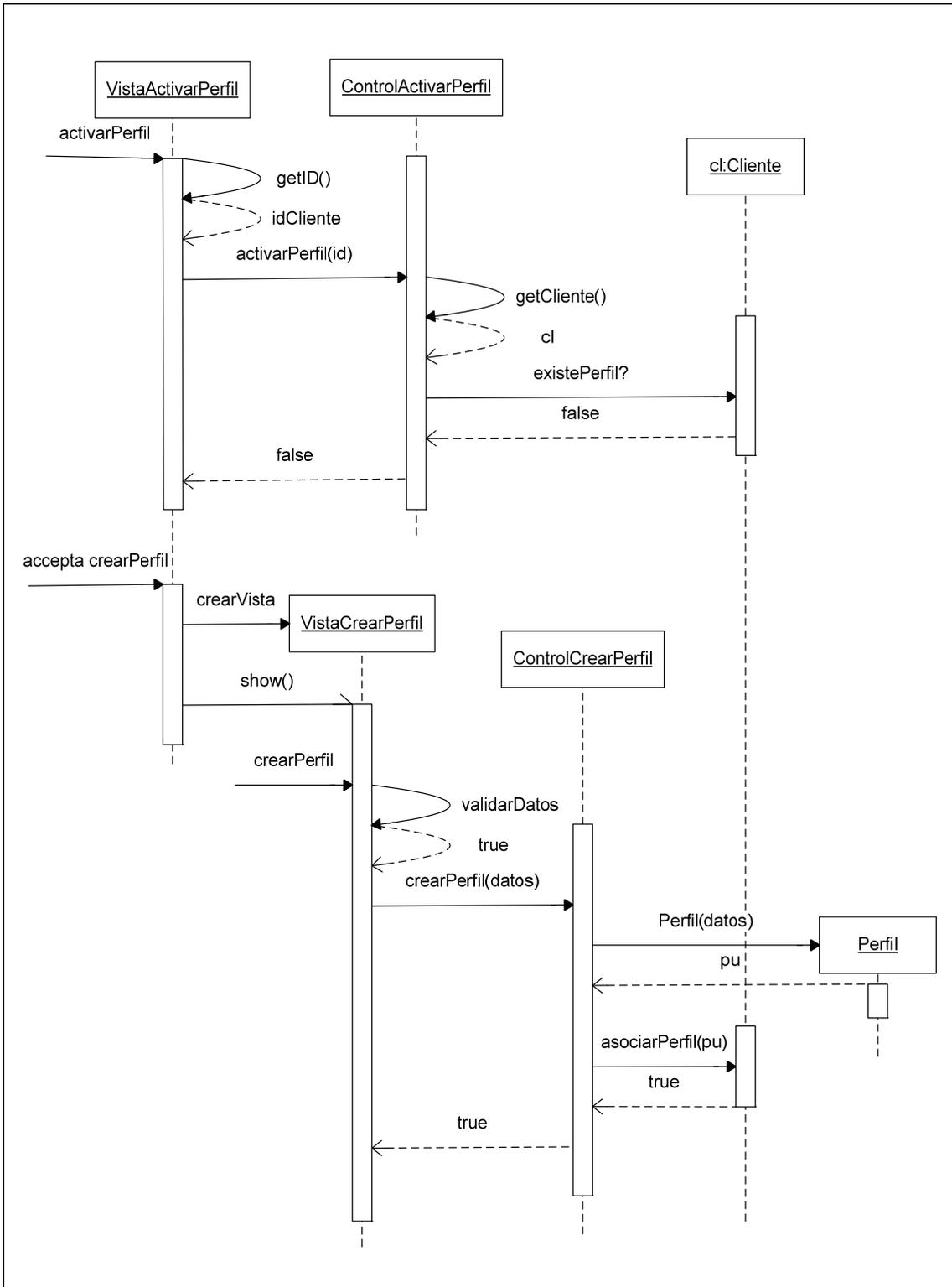


Ilustración 12: Diagrama de secuencia del flujo básico del caso de uso *ActivarPerfil* junto con la ejecución de la extensión del caso de uso *CrearPerfil*.

4.2.4.3 Caso de uso: Pedir recomendación de usuario

Las clases identificadas para este caso de uso son:

- *VistaRecomendaciónUsuario*: Interfaz a través de la cual el usuario realiza la petición de recomendación.
- *ControlRecomendaciónUsuario*: Se encarga de realizar el proceso de petición del usuario que la ha solicitado.
- *Perfil*: Representa el perfil del cliente que ha solicitado la recomendación.
- *PeticiónUsuario*: Representa la petición de recomendación por parte del cliente y encapsula toda la información relativa a la solicitud de recomendación.
- *ControlPeticiones*: Se encarga de gestionar las peticiones que llegan al sistema de recomendación.
- *ControlRecomendación*: Se encarga de gestionar el proceso de cálculo de la recomendación dada una petición.

En la Ilustración 13 se puede observar el diagrama de secuencia de las distintas clases identificadas del flujo básico del caso de uso aquí descrito.

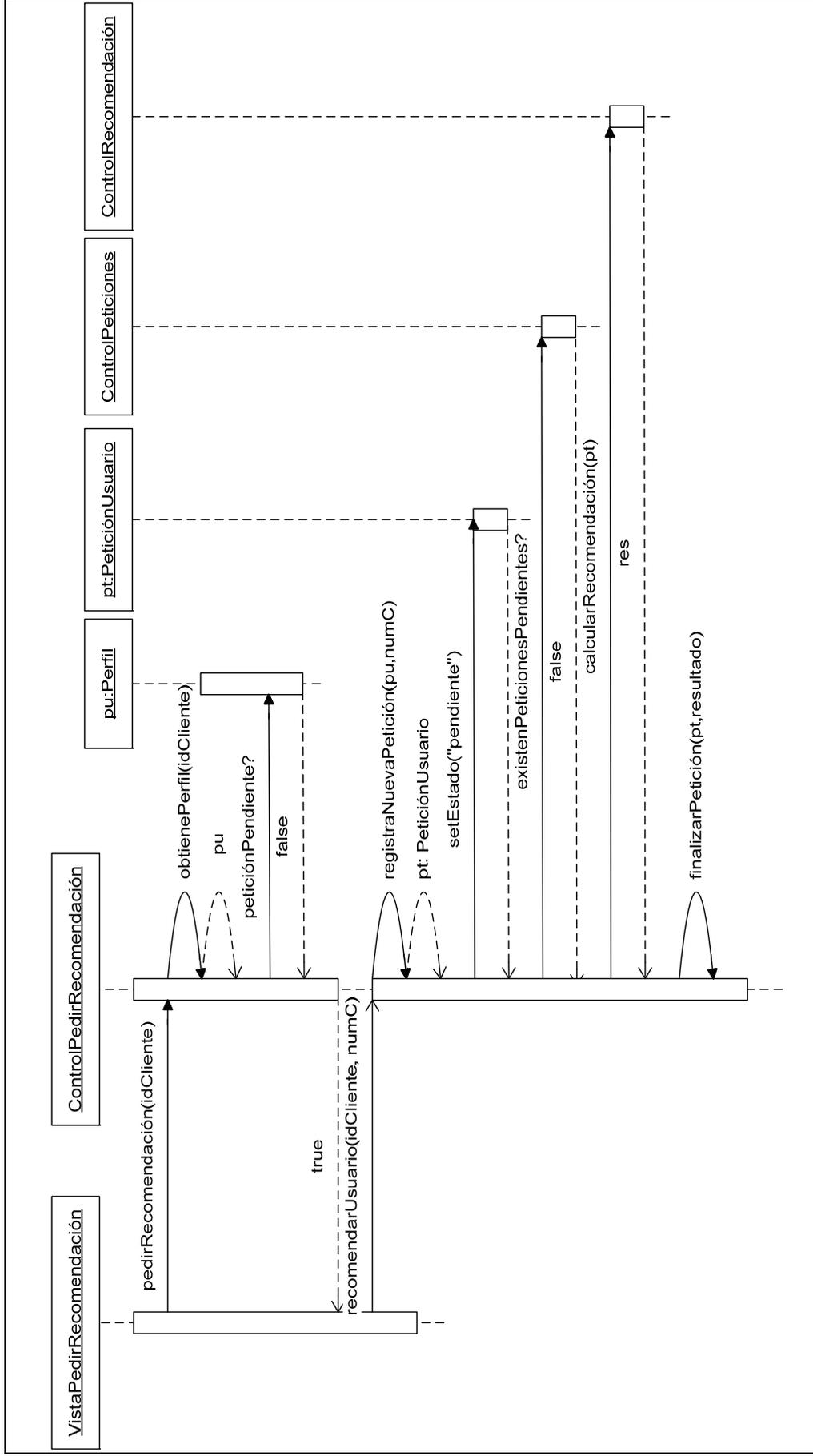


Ilustración 13: Diagrama de secuencia del flujo básico del caso de uso PedirRecomendación.

4.2.5 Definición de interfaces de usuario

Las interfaces de usuario que se utilizarán serán todas de tipo gráfico, siendo implementadas por medio del uso de menús, ventanas, diálogos, mensajes, y otros elementos comunes en este tipo de diseño. Asimismo, se emplearán estándares de navegabilidad y usabilidad en general, para hacer eficaz la comunicación persona-máquina.

Es importante comentar, que si durante cualquier operación se detecta algún error/excepción, se mostrará un mensaje de error al usuario. Por otra parte, y con respecto a los mensajes informativos, existe un elemento común, y es la aparición de uno de ellos, cuando se ha realizado una tarea de manera satisfactoria.

4.2.5.1 Identificación de perfiles

Ante la existencia de dos usuarios claramente distinguibles y definidos, como son el administrador de contenidos y el cliente, se analizarán por separado los perfiles de cada uno de ellos.

Administrador de contenidos. Se trata de un usuario especializado en la gestión de los contenidos que ofrece el sistema, con un nivel básico de informática. Dado que constantemente se están renovando los contenidos, hará un uso frecuente de la aplicación, por lo que la usabilidad de la interfaz jugará un papel importante en el rendimiento de sus tareas.

Cliente. En general se trata de un usuario con nivel básico de informática, cuyo objetivo es disfrutar del sistema de la forma más sencilla posible. Conociendo esto, la interfaz de usuario deberá de ser amigable y muy fácil de utilizar. También es muy importante el aspecto de la interfaz, de manera que se habrán de cuidar los detalles al hacer el diseño.

4.2.5.2 Especificación de la interfaz de usuario

A continuación, se detallan las interfaces de usuario, siendo descritas por medio de los formatos de las pantallas, los mensajes y los diálogos. Se señalarán para los casos de uso más relevantes, las interfaces que en ellos intervengan.

Alta de contenido

Las interfaces relacionadas son:

- *VistaAltaContenido.* Es el formulario donde el administrador de contenidos introducirá los datos necesarios para crear un nuevo contenido.

Los mensajes informativos relacionados son:

- Indicar que el contenido se ha creado satisfactoriamente.

Los mensajes de error relacionados son:

- Avisar que el contenido que se quiere crear ya existe en el sistema.
- Avisar que alguno de los datos del contenido no son válidos.

Activar perfil de usuario

Las interfaces relacionadas son:

- *VistaActivarPerfil*. Interfaz formada únicamente por el botón de activación del perfil.
- *VistaConsultarPerfil*. Interfaz con la que se muestra al usuario el perfil que tiene asociado.

Los mensajes informativos relacionados son:

- Indicar al usuario que es necesario crear un perfil ya que no tiene ninguno asociado

Crear perfil de usuario

Las interfaces relacionadas son:

- *VistaCrearPerfil*. Formulario a través del cual el cliente lleva a cabo la creación de su perfil.

Los mensajes informativos relacionados son:

- Indicar que el perfil se ha creado correctamente.

Los mensajes de error relacionados son:

- Avisar que alguno de los datos introducidos no son válidos.

Pedir una recomendación de usuario

Las interfaces relacionadas a este caso de uso son:

- *VistaPedirRecomendación*. Es el formulario donde el cliente podrá pedir una recomendación de forma explícita indicando cuántos contenidos como máximo quiere que se le recomienden.

Los mensajes informativos relacionados a este caso de uso son:

- Notificar al cliente que la petición se ha procesado correctamente, y en breve recibirá la lista de contenidos recomendados.
- Mostrar un diálogo de confirmación para seguir con el proceso de recomendación.

Los mensajes de error relacionados a este caso de uso son:

- Avisar al cliente que no puede pedir una nueva recomendación porque tiene una petición pendiente.

4.2.6 Especificación del plan de pruebas

Las pruebas a las cuales serán sometidos los distintos componentes del *Servicio de Recomendación* tendrán como objetivo verificar el correcto funcionamiento de los mismos, y determinar así, si cumplen con los requerimientos exigidos.

Para la realización de las pruebas se usará por una parte, un prototipo, que servirá inicialmente para probar las interfaces del usuario; y una vez avanzados en el desarrollo del sistema, permitirá por medio de datos simulados, comprobar el funcionamiento de las distintas clases, tanto de forma unitaria como de manera integrada de parte de ellas.

Finalmente, una vez obtenida una primera versión completa del producto, se deberá hacer todo un bloque de pruebas para determinar si se cumplen los objetivos marcados.

4.2.6.1 Alcance de pruebas

4.2.6.1.1 Pruebas unitarias

Comprenden las verificaciones asociadas a cada componente del sistema de información. Su realización tiene como objetivo verificar la funcionalidad y estructura de cada componente individual.

Clases Vista

Como se ha comentado anteriormente, estas clases representan las distintas interfaces que se le presentan a un usuario. Para comprobar que dichas clases funcionan correctamente, se debe verificar que proporcionan los elementos necesarios de recogida o selección de la información. Asimismo, se comprobará que contengan elementos que mejoren la comunicación persona-máquina. Por otra parte, impedirán la petición de realización de acciones sin haber sido recopilada toda la información que se requiera.

Clases Controlador

Para probar unitariamente cada una de las clases controlador que surjan, se verificará que todos los métodos que ofrezcan den respuesta a una acción que se desee realizar. Por otra parte, se comprobará también que ofrece un tratamiento de las excepciones que reciba.

4.2.6.1.2 Pruebas de integración

Tienen por objetivo verificar el correcto ensamblaje entre los distintos componentes.

Este tipo de prueba, consistirá en comprobar que la aplicación cumple con todos los requerimientos señalados, verificando que las clases funcionen correctamente de manera conjunta, para lograr la consecución de las acciones que un usuario pueda solicitar.

Se trata pues, de verificar, que se cumplen todas las especificaciones dadas en cada uno de los casos de uso, y que las respuestas obtenidas son las detalladas, tanto en la generación de excepciones como en la culminación de los procesos de manera satisfactoria; y que se muestran los mensajes oportunos en cada situación.

Por otra parte se comprobará, que la información que se muestre al usuario es la correcta y actual en todo momento, acreditándose así, la integridad de la información.

4.2.6.2 Pruebas de aceptación del sistema

Las pruebas de aceptación, van dirigidas a validar que el sistema cumple los requisitos de funcionamiento esperado, recogidos en el catálogo de requerimientos y en los criterios de aceptación del sistema de información, y conseguir la admisión final del sistema.

Se darán por aceptada la aplicación, cuando se haya demostrado que tanto las pruebas unitarias como las de integración obtienen los resultados esperados.

Resumiendo, cuando los requisitos de presentación, organización de la información, rendimiento, seguridad y cumplimiento de los objetivos sean completamente satisfactorios, entonces, se dará por admitida y válida al producto terminado.

4.3 Diseño del Sistema de Información

4.3.1 Definición de la arquitectura del sistema

Dado que el proyecto en si es una extensión de un sistema existente, la definición de la arquitectura del sistema viene notablemente marcada por la arquitectura actual del sistema que se pretende extender. Por este motivo la arquitectura utilizada es la cliente-servidor de componentes distribuidos.

4.3.1.1 *Requisitos de diseño*

A continuación se describen los requisitos funcionales que afectan al diseño del sistema.

- Se requiere que el sistema sea capaz de atender peticiones simultáneas de distintos clientes. De manera que un determinado cliente pueda pedir una recomendación independientemente de lo sobrecargado que esté en ese momento el servidor.
- Las peticiones simultáneas serán servidas manteniendo el orden de llegada al servidor, por lo que inicialmente el cálculo de la recomendación seguirá un proceso secuencial.
- Las peticiones de recomendación de usuario serán más prioritarias que las recomendaciones automáticas de nuevos contenidos.
- Se requiere construir un sistema fácilmente escalable para que en un futuro se pueda escalar a un gran número de usuarios. De modo que pueda existir distribución de carga entre diversos servidores en caso de necesidad.
- Dada la gran envergadura de la aplicación de IntegraTV se requiere que el diseño del sistema sea modular para facilitar luego su mantenimiento y construcción.
- Como en el Servicio de Recomendación no se manejan datos muy confidenciales, la seguridad no es un aspecto relevante en el diseño. Simplemente se requiere un nivel de seguridad básica en la que los datos personales de los usuarios no sean fácilmente accesibles por otros usuarios del sistema.

4.3.1.2 *Definición de los niveles de la arquitectura*

Teniendo en cuenta los requisitos de diseño especificados y el tipo de aplicación que se está desarrollando, se ha utilizado el patrón arquitectónico en tres capas junto con la tecnología orientada a objetos. Las ventajas de utilizar este patrón arquitectónico son principalmente: la **flexibilidad**, permite una **alta**

escalabilidad, y facilita la construcción y el mantenimiento del sistema. A continuación se hace una breve descripción de las funciones que se realizan en cada una de las capas lógicas de la arquitectura.

- **Capa de Presentación.** La capa de presentación contiene los componentes necesarios para habilitar la interacción del usuario con la aplicación. Básicamente está formada por los componentes de interfaz de usuario que permiten procesar y dar formato a los datos de usuario, así como adquirir y validar los datos entrantes procedentes de éstos. Esta capa se comunica únicamente con la capa de negocio.
- **Capa de Negocio.** En esta capa es donde se implementan las reglas de negocio y procesos relacionados con las funcionalidades que ofrece el sistema. Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos para almacenar o recuperar datos de la base de datos.
- **Capa de Servicios de Datos.** Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

En cuanto a la arquitectura física del sistema se divide en varios niveles físicos, asociados a los nodos¹⁰ que conforman el sistema, los cuales separan los distintos niveles lógicos. En la Ilustración 14 se observan los distintos nodos que conforman la arquitectura física junto con la capa lógica a la que están asociados.

¹⁰ Se entiende por nodo cada partición física o parte significativa del sistema de información, con características propias de ejecución o función.

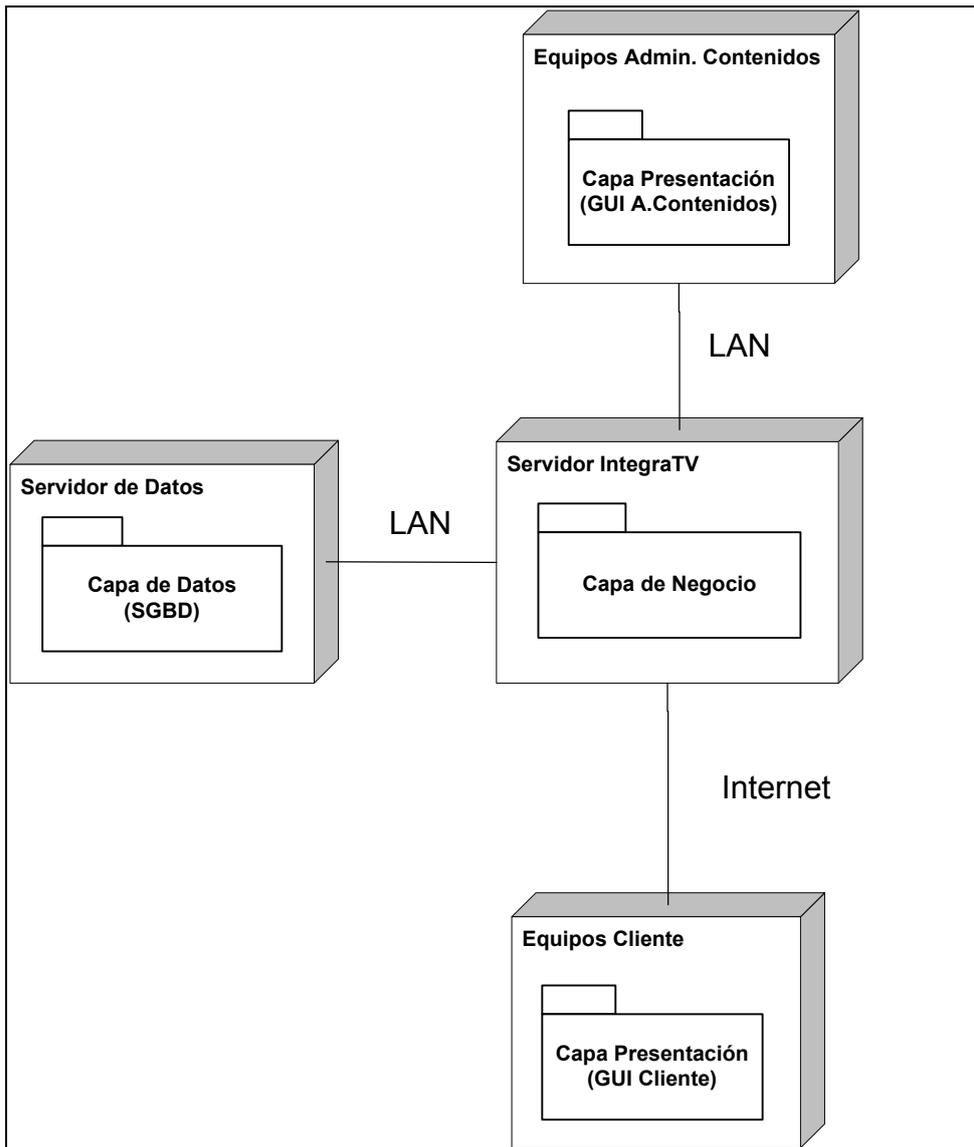


Ilustración 14: Diagrama de despliegue de la arquitectura física del sistema

Una observación es que aunque a priori se ha representado las capas lógicas de datos y de negocio en diferentes servidores por cuestiones de eficiencia, no significa que una vez en funcionamiento el sistema, se utilice el mismo servidor para albergar los procesos de ambas capas lógicas.

4.3.1.3 Identificación de subsistemas de diseño

Con el fin de facilitar el diseño del sistema y la posible reutilización de sus componentes, se ha dividido éste en varios subsistemas de diseño. A partir de la arquitectura del sistema identificada y de los componentes existentes con los que el sistema interactúa, se han identificado los siguientes subsistemas de diseño.

- **Interfaz de Usuario Cliente.** Conjunto de interfaces que permiten al usuario utilizar el servicio de recomendación. El diseño de este subsistema depende directamente del componente que el cliente utiliza actualmente (FrontEnd).
- **Interfaz de Usuario Administrador de Contenidos.** Conjunto de interfaces que permiten al administrador de contenidos gestionar los contenidos de vídeo que el sistema de recomendación utilizará para las recomendaciones. El diseño de este subsistema depende directamente de la aplicación de gestión de contenidos utilizada actualmente (BackOffice).
- **Servicio de Recomendación.** Subsistema que contiene la lógica de negocio de las funcionalidades propias del *Servicio de Recomendación* que se ha desarrollado en este proyecto. De este modo se consigue desacoplar el servicio de recomendación del sistema actual cumpliendo así con los requisitos de integración que se habían especificado durante el análisis. Este subsistema dependerá únicamente del componente *Servidor Integra* para llevar a cabo la distribución de los contenidos que sean recomendados.
- **Servicio de IntegraTV.** Subsistema que contendrá la interfaz de comunicación con el componente de la parte del servidor de IntegraTV Home, que permitirá actualizar las listas de contenidos asignados a cada usuario e identificar a los clientes. El diseño de la interfaz dependerá directamente del componente *Servidor Integra* del sistema.
- **Persistencia.** Subsistema formado por el conjunto de clases que se encargarán de la persistencia de las entidades persistentes del sistema utilizando un SGBD.

En la Ilustración 15 se muestra el diagrama de dependencias entre los subsistemas de diseño identificados.

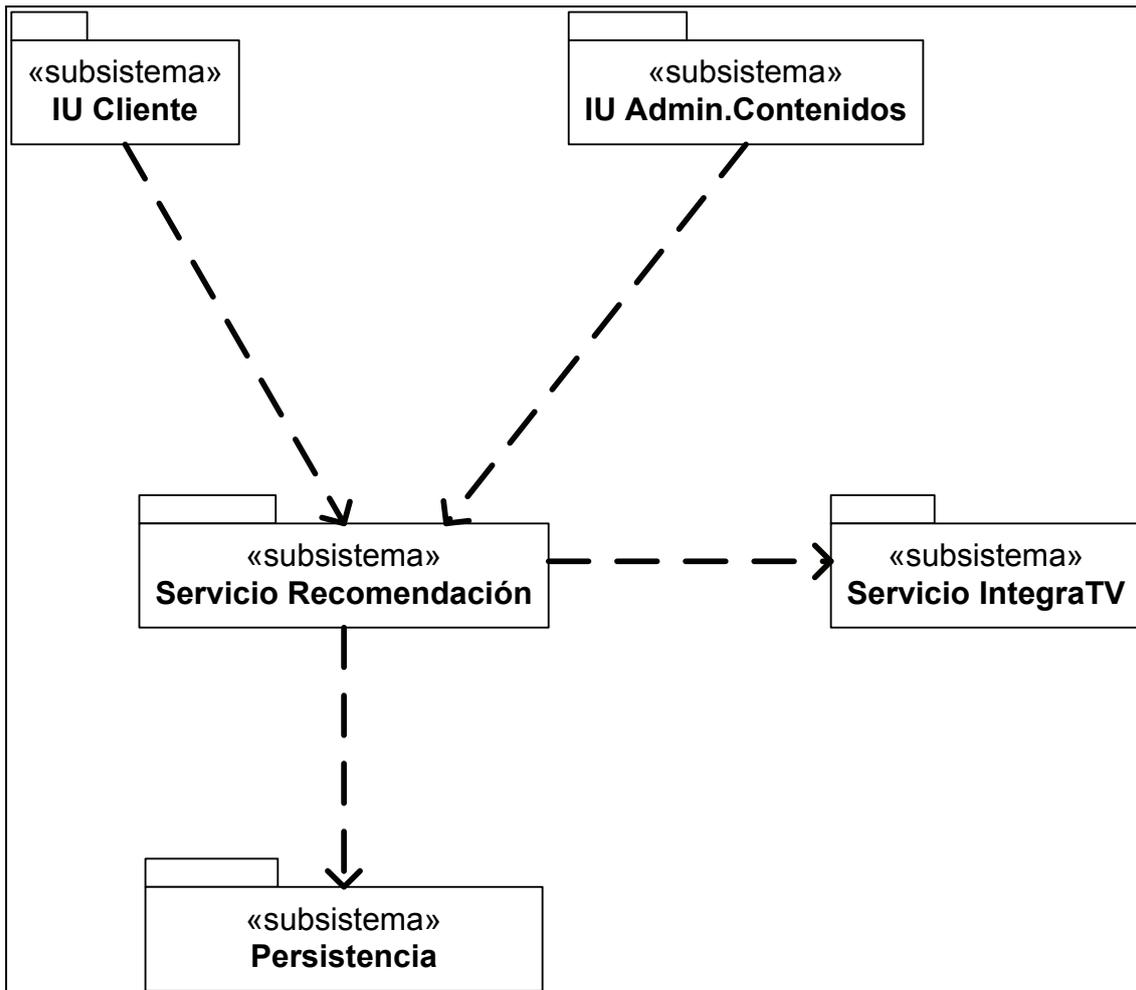


Ilustración 15: Diagrama de paquetes con las dependencias entre subsistemas.

4.3.1.4 Requisitos del entorno tecnológico

A continuación se define el entorno tecnológico para cada uno de los nodos que conforman la arquitectura del sistema.

Equipo Cliente

Consiste en un equipo de televisión interactiva formado por un equipo Media Center que permite conectar varios periféricos inalámbricos.

- Hardware:
 - Equipo Media Center HiGrade P60 (Ilustración 16) con un microprocesador Intel Core Duo a 1.67 GHz, un disco duro de 80 GB, 2Gb de RAM y salida DVI.



Ilustración 16: Equipo de televisión interactiva (Media Center HiGrade P60).

- Mando a distancia diseñado especialmente diseñado para IntegraTV (Ilustración 17) y un teclado inalámbrico con touchpad integrado.



Ilustración 17: Mando a distancia de IntegraTV.

- Software:
 - Sistema operativo Windows Media Center.
 - Componentes que forman parte del cliente del sistema de IntegraTV Home.
- Comunicaciones
 - Conexión BlueTooth.
 - Tarjeta de red Ethernet.
 - Red WiFi.
 - Conexión de internet ADSL.

Equipo Administradores

Se trata de los ordenadores que utilizan normalmente los trabajadores de la empresa. Estos pueden ser PC's de características distintas según su antigüedad. A continuación se describe la configuración más común.

- Hardware:
 - Procesador Intel Core Duo a 2 Ghz.
 - Memoria RAM de 2Gb.
 - Disco Duro de 150 Gb.

- Software:
 - Sistema operativo Windows XP Professional.
 - Componentes de la parte del administrador del sistema que forman parte de IntegraTV Home.

- Comunicaciones:
 - Tarjeta de Red Ethernet
 - Conexión de internet ADSL
 - Conexión LAN de 100 Mb/s

Servidores

Un servidor de aplicaciones es un ordenador que se utiliza para alojamiento web, bases de datos, correo electrónico, copias de seguridad de los datos de la empresa, etc. Ese equipo ha de estar conectado las 24 horas del día durante todo el año, tiene que soportar fuertes picos de actividad en función de las horas del día, que debe tener un gran rendimiento en ejecución de aplicaciones. Las características de los servidores utilizados para ofrecer el servicio de IntegraTV son:

- Hardware:
 - Procesador AMD Opteron 244 / 1.79GHz.
 - Memoria RAM 2Gb.
 - Configuración de discos duros Raid 5.
 - CDROM 48X Integrado.

- Software:
 - Sistema operativo Windows 2003 Server.
 - Bases de datos: MSDAC Y MSSQL Server.

- Comunicaciones:
 - IP pública estática.
 - 2 tarjetas de red NIC Gigabit integrada.

- Conexión LAN de 100 Mb/s.

4.3.2 Diseño de la Arquitectura de Soporte

4.3.2.1 Tecnologías de soporte

Respecto a la arquitectura de soporte que se ha utilizado para el diseño del sistema cabe destacar que forma parte principalmente del framework de .Net. En concreto se han utilizado las tecnologías:

- *Net Remoting*, para la comunicación entre los subsistemas de los diferentes nodos que conforman la arquitectura del sistema.
- *Soporte multihilo de .Net*, para la gestión de las peticiones asíncronas del sistema. Permite resolver el requisito de atender peticiones simultáneas de distintos clientes de forma transparente al programador.

4.3.2.1.1 Net Remoting

Net Remoting es un modelo de programación distribuido basado en objetos remotos. A diferencia de otras tecnologías similares, Remoting forma parte del framework de .Net, lo que implica que los objetos remotos puedan utilizar todo el potencial que éste ofrece.

La verdadera ventaja del sistema de interacción remota es su capacidad para permitir la comunicación entre objetos pertenecientes a dominios de aplicación o a procesos distintos mediante diferentes protocolos de transporte, formatos de serialización, esquemas de duración de objetos y modos de creación de objetos. Entre las características del modelo de Net Remoting se destacan las siguientes:

- **Flexibilidad.** Ofrece un amplio rango de opciones de comunicación así como un control total sobre el ciclo de vida de los objetos remotos.
- **Modelo de programación sencillo y eficaz,** así como compatibilidad en tiempo de ejecución que permite interacciones transparentes. Una vez se ha establecido conexión con el objeto remoto, se puede acceder y crear objetos remotos de la misma manera que se haría de forma local.

En la Ilustración 18 se muestra el proceso general de interacción remota en Net Remoting.

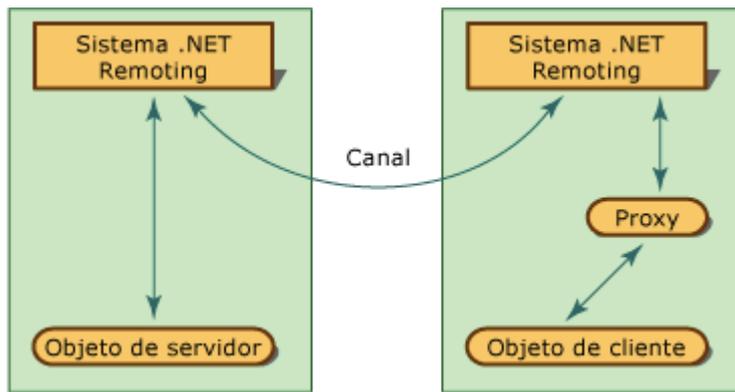


Ilustración 18: Arquitectura simplificada del proceso de interacción remota en Net Remoting.

El funcionamiento básico del proceso de interacción remota es el siguiente:

1. Si ambas partes de la relación están configurados correctamente, un cliente se limita a crear una nueva instancia de un objeto remoto que reside en el servidor.
2. El sistema de interacción remota crea un objeto proxy que representa a la objeto remoto y devuelve al cliente una referencia al objeto creado.
3. Cuando un cliente llama a un método, la infraestructura de interacción remota controla la llamada, comprueba el tipo de información y dirige la llamada por el canal hacia el proceso del servidor.
4. Un canal a la escucha detecta la solicitud y la reenvía al sistema de interacción remota del servidor, que a su vez busca (o crea, si es necesario) y llama al objeto solicitado.
5. A continuación el proceso se invierte: el sistema de interacción remota del servidor incluye la respuesta en un mensaje que el canal del servidor envía al canal del cliente.
6. Por último, el sistema de interacción remota del cliente devuelve el resultado de la llamada al objeto del cliente a través del objeto proxy.

Tipos de canales

Para que un cliente se pueda comunicar con un objeto remoto necesitamos un camino a través del que se pueda realizar la transferencia de datos. Este camino es lo que se llama canal. Existen tres tipos de canales en Net Remoting 2.0:

- **Canal TCP.** Como su nombre indica este tipo de canal utiliza el protocolo TCP para comunicarse. Los canales TCP transmiten datos en formato binario y son ideales cuando el rendimiento es relevante.

- **Canal IPC.** Se utiliza para la comunicación entre procesos o dominios de aplicación que residen físicamente en la misma máquina. Ofrece un rendimiento superior a los canales TCP.
- **Canal HTTP:** Estos canales utilizan el protocolo http para comunicarse. En general se utilizan para transportar mensajes de tipo SOAP. Son ideales cuando lo que prima es la interoperabilidad entre diferentes procesos de aplicación. Por el contrario el rendimiento de estos canales es mucho menor que el de los otros tipos de canal.

Tipos de objetos remotos

Existen varios tipos de objetos remotos según cuándo se crean las instancias de los objetos.

- Activados en el servidor (SAO: Server Activated Object):
 - Singleton. Para este tipo de objeto sólo existirá una instancia, independientemente de cuantos clientes estén dialogando con él.
 - SingleCall. Para este tipo de objeto se creará una nueva instancia para cada método que sea invocado por el cliente.
- Activados por el cliente (CAO: Client Activated Object). Para este tipo de objetos, una nueva instancia se creará cuando el cliente realice la primera llamada (utilizando la palabra clave new). De esta manera es posible mantener el estado entre llamadas de un mismo cliente.

Configuración de Net Remoting utilizada

Una vez se han descrito las diferentes configuraciones que permite utilizar el modelo de programación distribuida Net Remoting, se explicará qué configuración se ha elegido para dar soporte a las comunicaciones entre los distintos subsistemas de diseño identificados.

En cuanto al tipo de objetos utilizado, se ha elegido el tipo CAO, para la comunicación entre los subsistemas de interfaz de usuario (componentes de la parte cliente) y el subsistema Servicio de Recomendación. Básicamente, porque permite mantener el estado entre distintas llamadas de un cliente. Y esto implicará un mejor rendimiento en cuanto a velocidad de interacción en el componente del cliente, además de simplificar las operaciones de los objetos remotos, ya que se podrá almacenar información en el objeto evitando así tener que ser pasar la información por parámetro. Un ejemplo es la información de identificación de usuario, que se almacena en el objeto remoto correspondiente la primera vez que es llamado desde el componente cliente.

En cambio, para la comunicación entre los subsistemas del Servicio de Recomendación y del Servicio de Integra se utilizará el tipo de objetos SAO

Singleton. Se ha elegido este tipo de objetos ya que se pretende tener un control del acceso a la interfaz del subsistema de Integra por parte de los diferentes hilos de recomendación que forman parte del subsistema Servicio de Recomendación, con el fin de evitar inconsistencias en el sistema.

A lo que se refiere al tipo de canal, se ha optado por canales TCP entre las comunicaciones de los subsistemas ubicados físicamente en distintos nodos, y un canal IPC para las comunicaciones entre subsistemas del mismo nodo. Se ha elegido el canal TCP en vez de HTTP para las comunicaciones de los clientes (a través de Internet), dado que la aplicación cliente de IntegraTV Home posee un entorno que se caracteriza por ser bastante seguro y controlado, ya que tiene un control muy preciso de los clientes que acceden al sistema. En el caso que fuera una aplicación Web donde no se tuviera control de los clientes que acceden al servicio la opción correcta sería utilizar un canal HTTP.

4.3.2.1.2 Soporte multihilo de .Net

El framework de .Net ofrece el soporte necesario para trabajar con aplicaciones de múltiples hilos (threads), permitiendo que determinadas operaciones costosas del sistema puedan ser ejecutadas de forma transparente al usuario.

Método de invocación asíncrona de .Net

El proceso de invocación asíncrona utilizado está basado en el patrón de diseño Asíncrono de .Net. Una de las ventajas de utilizar este patrón, es que la clase que invoca las operaciones es la que decide si una particular operación debería ser asíncrona. Por lo tanto en la clase donde están las operaciones invocadas, no es necesario añadir código adicional para permitir un comportamiento asíncrono.

El patrón de diseño divide una operación síncrona en diversas partes: la operación de inicio, la operación de final y el objeto resultado. Para llevar a cabo esta partición, se utiliza el objeto delegado, que va asociado a una operación asíncrona. Un objeto delegado contiene los métodos que se encargan del proceso de invocación asíncrona. Éstos son:

- *BeginInvoke*, invoca la operación asíncrona con los parámetros de entrada de la operación a la que está asociado el delegado.
- *EndInvoke*, obtiene el resultado de la operación asíncrona y el valor de los parámetros de salida de la operación, o bien la excepción que se ha producido durante su ejecución en el caso que se haya producido algún error.

Existen básicamente dos métodos para saber cuándo una operación asíncrona ha terminado. El primero es el llamado método de la espera activa, y consiste en que el hilo principal se encarga de ir preguntando cada cierto tiempo si se ha completado la operación. El segundo método consiste en utilizar un

delegado asociado a una operación de retorno (Callback operation), que se ejecutará una vez la operación asíncrona haya terminado. De esta manera las clases que invocan la operación asíncrona se despreocupan completamente del estado de la ejecución, ya que es el mismo hilo secundario quien avisa de que ha finalizado la operación, invocando la operación que apunta el delegado de retorno.

El framework de .Net proporciona las herramientas necesarias para dar soporte al modelo de programación asíncrona. Estas herramientas son principalmente: primitivas de sincronización, tales como monitores y semáforos; además de una pool de hilos propia del framework que se encarga de gestionar los diferentes hilos de ejecución secundarios.

Net Thread Pool

Cada vez que se invoca una operación asíncrona, es necesario asignar un hilo secundario que se encargue de su ejecución. Dado que no puede ser el hilo en ejecución, ya que entonces dejaría de tratarse de una operación asíncrona, se encola la petición para que se asigne automáticamente un hilo secundario de la pool de hilos.

La pool de hilos consiste en un conjunto de hilos secundarios que son gestionados por el sistema de forma transparente al programador. En la Ilustración 19 se observa a modo de ejemplo una pool de hilos con dos hilos de ejecución.

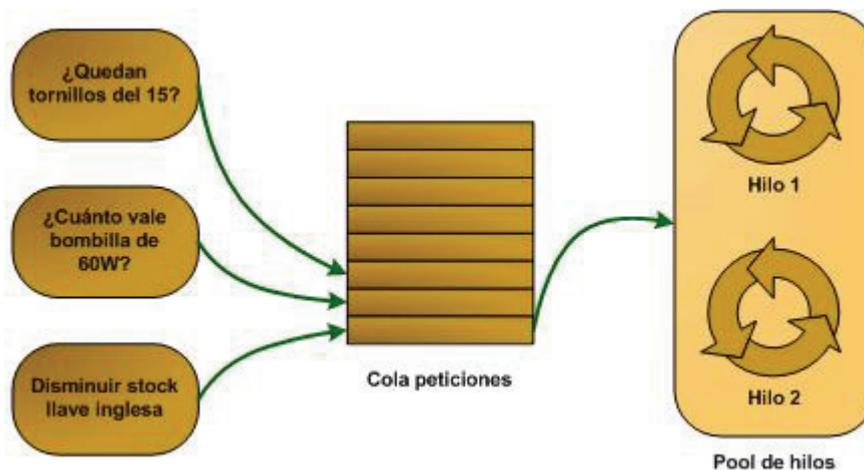


Ilustración 19: Esquema básico de una pool de hilos.

Una de las grandes ventajas de utilizar una pool de hilos, está en el hecho de que no es necesario crear y destruir hilos cada vez que se atiende una nueva petición. Esto es así porque se utiliza un número prefijado de hilos para atender a las peticiones que se van reutilizando a medida que van quedando libres.

La pool de hilos propia de .Net, no sólo es accesible a las aplicaciones que quieran hacer uso de él, sino que también está integrado con la mayoría de las clases incluidas en el entorno. Además, gran parte del framework de .Net utiliza directamente el mismo pool. Por ejemplo, Net Remoting hace uso de él para atender las peticiones sobre objetos remotos.

4.3.2.2 Mecanismos genéricos de diseño

Para resolver determinados problemas comunes y con el fin de conseguir un diseño reutilizable y fácil de entender se han utilizado algunos patrones de diseño durante el diseño de la realización de los casos de uso. Los patrones aquí descritos pertenecen a las listas: *Gang of Four* (GoF) y *General Responsibility Assignment Software Patterns* (GRASP). Se puede encontrar más información sobre patrones de diseño en Gamma et al. [13] y en Cooper, J. W [11].

A continuación se presenta para cada uno de los patrones de diseño utilizados, qué problema común resuelven y cómo lo resuelven. Y además para los patrones más representativos del diseño del sistema, se ha realizado una breve descripción de la utilización del patrón en algunos escenarios específicos.

Controlador

- Problema: hay que decidir a qué clase se le asigna la responsabilidad de controlar determinados eventos del sistema.
- Solución: se asigna la responsabilidad de recibir un determinado evento a un clase específica que se encargue de controlarlo. Esta clase puede representar a un caso de uso (*use-case controller*) o bien representar a todo un sistema o subsistema (*facade controller*).

En el diseño del presente sistema se han utilizado los dos tipos de controlador. Los *facade controller* se utilizan para representar las interfaces de comunicación del sistema que utilizarán remotamente tanto los clientes como los administradores. Y con el fin de lograr un diseño de clases con **bajo acoplamiento** y **alta cohesión** se utilizan los *use-case controller* para el control de los eventos del sistema de cada uno de los casos de uso.

Existe un *facade controller* para los clientes y otro para los administrados, los cuales agrupan los *use-case controller* que representan las funcionalidades disponibles a los usuarios.

Singleton

- Problema: es necesario controlar que sólo exista una instancia de una determinada clase.

- Solución: utilizar el patrón de diseño Singleton que proporciona la interfaz necesaria para gestionar el acceso a la única instancia de la clase.

El hecho de utilizar la programación multihilo implica que determinadas operaciones tengan de ser sincronizadas para evitar posibles inconsistencias con datos compartidos o bien para conservar un orden establecido de ejecución. Este es el caso de las operaciones que realizan el cálculo de la recomendación, ya que los requisitos de diseño de priorizar las peticiones simultáneas según el orden de llegada y el tipo de petición, implican la necesidad de controlar que el cálculo de la recomendación sea ejecutado únicamente por uno de los hilos secundarios en un momento dado.

Para llevar a cabo este control de concurrencia es necesario que todos los hilos accedan a la misma instancia de la clase *Recomendador* (que es la que contendrá las operaciones a sincronizar). De modo que utilizando el patrón Singleton es posible asegurar un acceso secuencial de las operaciones conflictivas mediante distintas primitivas de sincronización.

Diccionario e Iterador

- Problema: frecuentemente es necesario acceder a los elementos de un agregado, ya sea para buscar un elemento en concreto o bien para recorrer todos los elementos del agregado.
- Solución: asociar un clase Diccionario e Iterador a todos las clases que sean agregados y se quiera acceder a sus elementos. Estas clases proporcionan los métodos para acceder al agregado, de manera que las clases que acceden al agregado no se responsabilizan de la búsqueda o del recorrido de sus elementos.

Creador

- Problema: es necesario decidir qué clase es la responsable de crear instancias de un determinado objeto.
- Solución: siempre que sea posible, se asigna la responsabilidad de crear un determinado objeto a alguna de las clases que esté fuertemente asociada con dicho objeto. De esta manera se consigue un diseño con un bajo acoplamiento de las clases.

Experto

- Problema: es necesario decidir qué clase es la responsable de realizar una determinado operación.
- Solución: se asigna esa responsabilidad a la clase que dispone de la información para llevar a cabo la operación. De igual manera que en los patrones anteriores, con este patrón se consigue un diseño de clases con bajo acoplamiento y alta cohesión.

Proxy

- Problema: se necesita referenciar a un objeto de una forma más sofisticada que utilizando un simple puntero, ya que puede estar en espacios de memoria distintos.
- Solución: utilizar una clase que controle el acceso a dicho objeto.

El uso de este patrón de diseño viene dado como una consecuencia de utilizar el framework Net Remoting para comunicar los distintos nodos que conforman el sistema. Dado que existe la necesidad de conectar con objetos distribuidos que están en diferentes espacios de direcciones, se utiliza un proxy remoto para que proporcione una representación local del objeto que se pretende acceder.

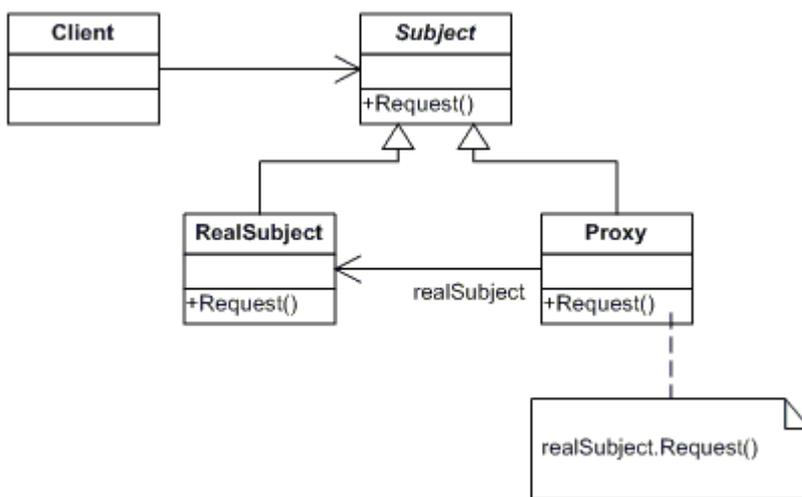


Ilustración 20: Aspecto estático del patrón de diseño Proxy.

En la Ilustración 20 se observa el aspecto estático de las clases que participan en el patrón de diseño. Client es la clase que quiere acceder al objeto remoto (el RealSubject) y lo hace a través de su representación local (Proxy) que es la clase que realmente accede al objeto remoto.

Command

- Problema: existen diferentes tipos de peticiones a objetos los cuales no tienen información sobre la operación que se está pidiendo, es necesario utilizar colas de peticiones o bien deshacer los cambios realizados por una petición.
- Solución: encapsular la información de la petición en un objeto que se encargue de su gestión.

Como se ha comentado anteriormente, para la invocación de las operaciones asíncronas del sistema, se ha utilizado el patrón de diseño Asíncrono que

facilita .Net para dar soporte a este tipo de invocaciones. Aunque utilizar el patrón de diseño facilita la tarea de gestión del proceso de invocación asíncrona, para cada operación asíncrona es necesario utilizar distintas operaciones que interactúan entre sí. Y el hecho de que existan dos operaciones asíncronas claramente diferenciadas implica que sea necesario agrupar de alguna forma estas operaciones para obtener un diseño más fácil de mantener y reutilizable. Para ello se ha utilizado el patrón Command ya que permite agrupar las operaciones que llevan a cabo la invocación asíncrona en una clase por tipo de petición.

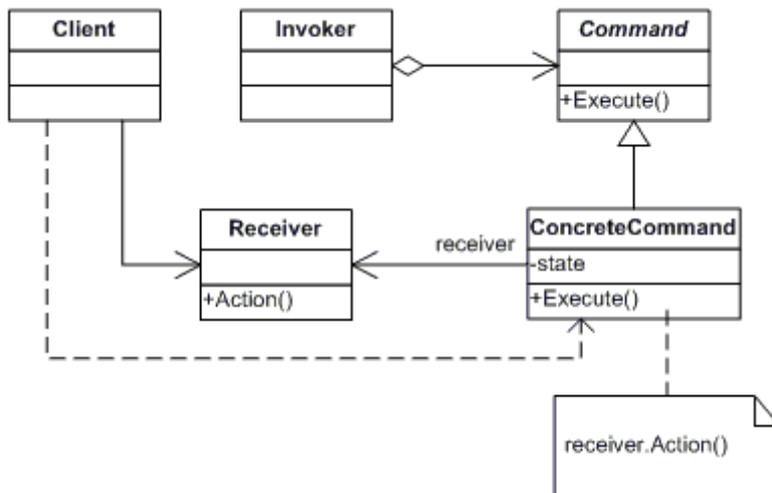


Ilustración 21: Aspecto estático del patrón de diseño Command.

En la Ilustración 21 se puede observar el diagrama estático de las clases que participan en el patrón de diseño. Client se encarga de crear la instancia concreta de Command (la operación a realizar) y asociarla con la instancia de Receiver que es quien realmente realiza la operación. Invoker es la clase que tiene acceso a la clase comando e invoca la operación.

4.3.3 Diseño de la realización de los casos de uso

En este apartado se describirán en detalle la interacción de las clases de diseño en cada uno de los escenarios de los casos de uso vistos en el capítulo de análisis. Los diferentes diagramas representados a lo largo de este capítulo están enfocados principalmente a las clases que pertenecen al subsistema de diseño Servicio de Recomendación, ya que es dónde se realizan los procesos de negocio más relevantes en cuanto al desarrollo del sistema. No obstante se mostrarán las clases más relevantes del resto de subsistemas de diseño con las que existe algún tipo de interacción.

En los siguientes capítulos se entrará en más detalle con el diseño de los subsistemas de la capa de presentación y del subsistema de persistencia, dónde se podrá ver cómo se integran con los distintos componentes de IntegraTV Home comentados anteriormente (ver capítulo 2.1).

Para cada caso de uso, se ha realizado el diseño de los eventos del sistema identificados durante el análisis haciendo uso de algunos de los patrones de diseño descritos en el capítulo anterior. La descripción de cada caso de uso se dividirá en:

- **aspecto estático**, donde se describirá mediante diagramas de clases, los datos y las relaciones de las clases de diseño que participan en el caso de uso;
- **aspecto dinámico**, donde se detallará la secuencia de operaciones de las clases participantes para la realización del caso de uso, en el escenario descrito en el capítulo de análisis.

4.3.3.1 Caso de uso: Alta de contenido

Aspecto estático

El diagrama de clases de la Ilustración 22 muestra las clases de diseño que participan en las operaciones diseñadas para este caso de uso. Un dato a tener en cuenta, es que el parámetro “datos” de tipo string se ha utilizado para simplificar la representación de la operación. En realidad la operación *crearContenido* tiene un parámetro para cada uno de los atributos descriptivos de un contenido. Estos atributos, ya descritos anteriormente en la fase de requisitos son:

- título
- año producción
- duración
- género/s,
- actores,
- director/es,
- nacionalidad/es,
- si es novedad o bien un clásico,
- valoración de los *ingredientes de contenido*.

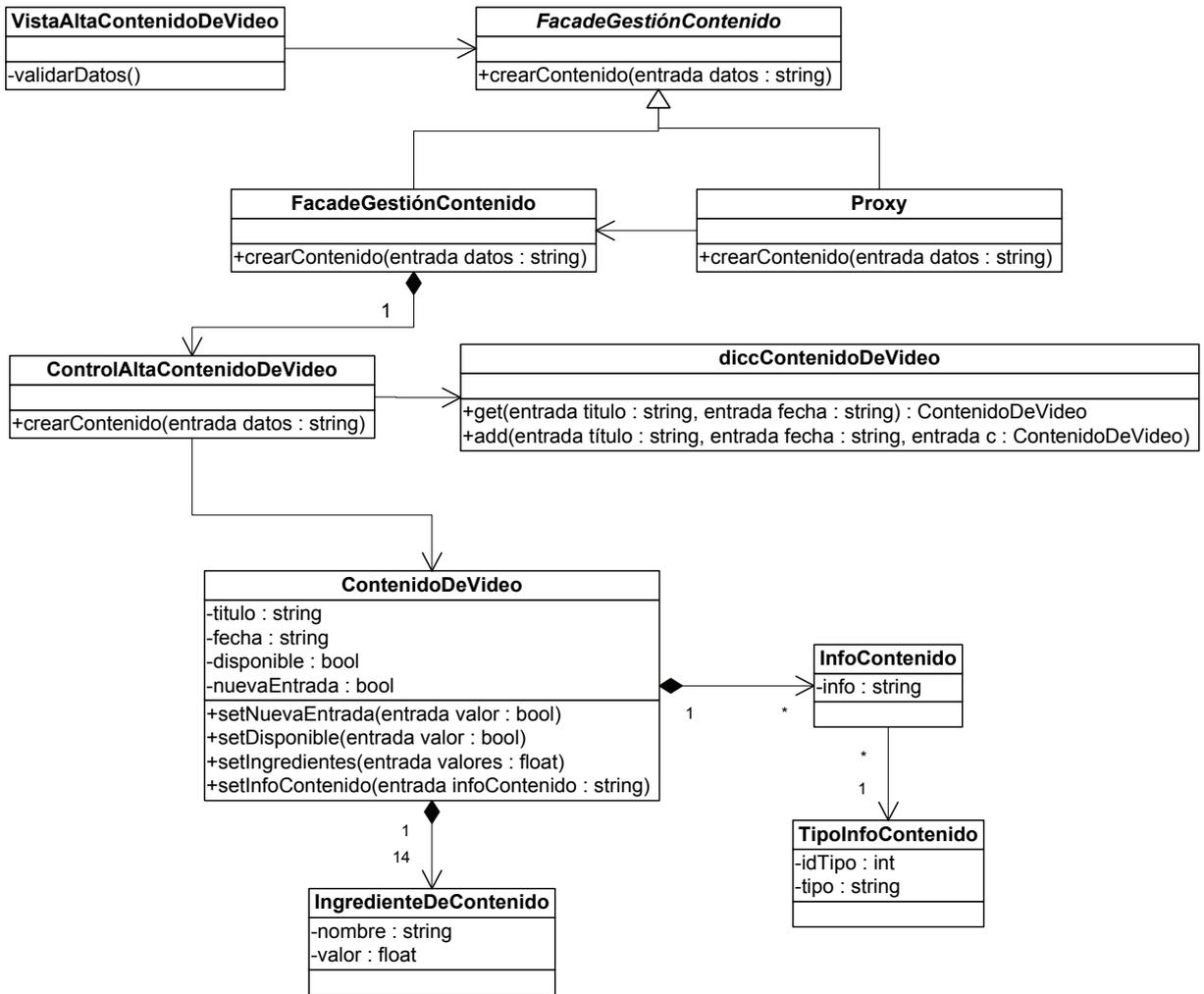


Ilustración 22: Aspecto estático del diseño del caso de uso AltaContenido.

Aspecto dinámico

Para este caso de uso se ha detectado únicamente la operación de sistema *altaContenido*, que se encarga de llevar a cabo la creación de un nuevo contenido con los datos introducidos por el usuario desde la interfaz.

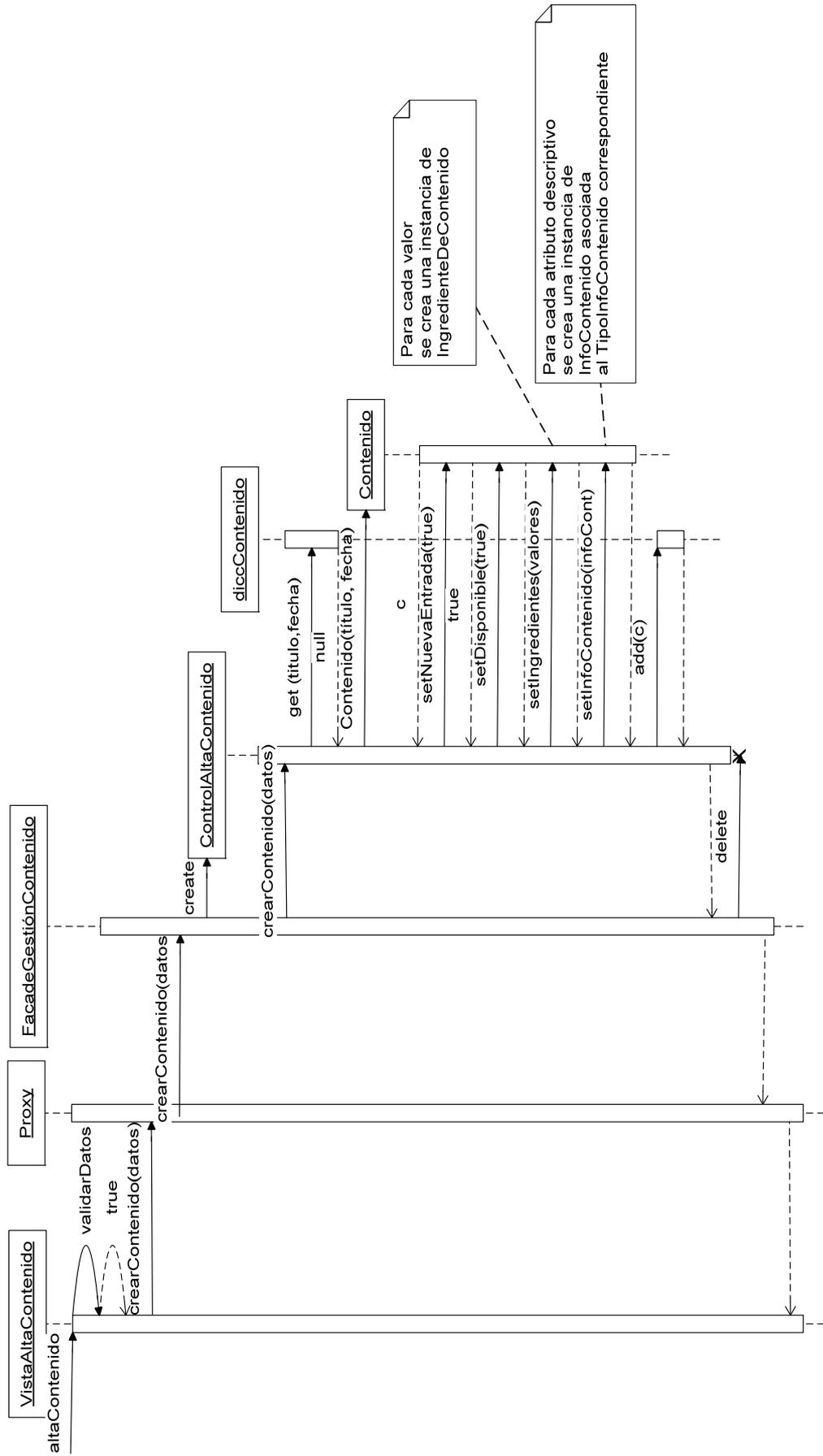


Ilustración 23: Aspecto dinámico del diseño del caso de uso AltaContenido.

En el diagrama de secuencia de la Ilustración 23 se observan las siguientes acciones por cada una de las clases de diseño que interactúan en el flujo básico del caso de uso:

1. *VistaAltaContenido* se asegura que los datos introducidos por el administrador de contenidos sean correctos y utiliza la clase *Proxy* para indicar al sistema que cree el nuevo contenido pasándole los datos validados.
2. *FacadeGestiónContenido* crea una instancia del use-case controller *ControlAltaContenido* que es la que se encarga de llevar a cabo la creación del nuevo contenido.
3. El *ControlAltaContenido* comprueba si existe un contenido con el mismo título y fecha utilizando el diccionario *diccContenido*.
4. Tras comprobar que no existe, el *ControlAltaContenido* crea el nuevo contenido con los datos proporcionados a través de la interfaz y marca el contenido como “nueva entrada”.
5. Finalmente el mismo controlador añade la instancia de contenido creada en el diccionario *diccContenido* y se devuelve el control a *VistaAltaContenido* la cual notifica al administrador que el proceso de creación ha sido satisfactorio.

Hay que tener en cuenta que en este caso el diccionario de la clase Contenido tiene como llave un valor multiatributo, el título y la fecha.

4.3.3.2 Caso de uso: Activar perfil de usuario

Aspecto estático

El diagrama de clases de la Ilustración 24 muestra las clases de diseño que participan en las operaciones diseñadas para este caso de uso.

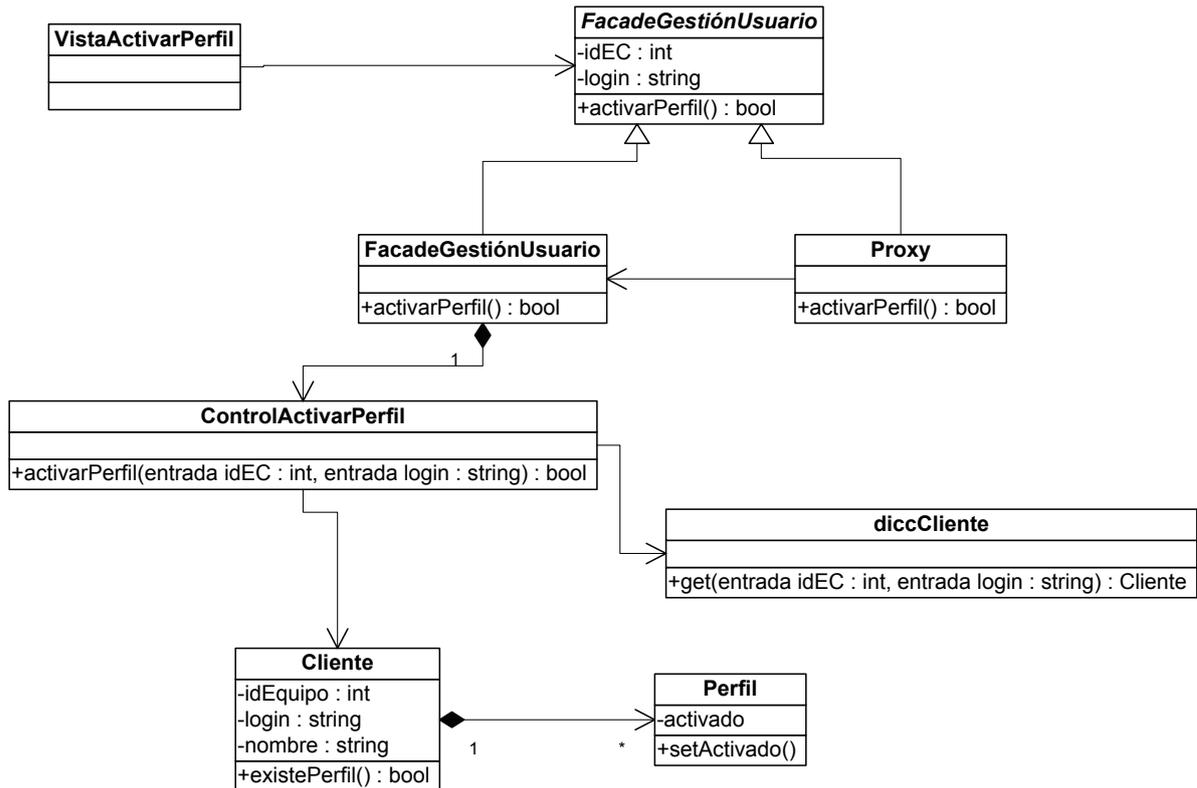


Ilustración 24: Aspecto estático del diseño del caso de uso ActivarPerfil.

Aspecto dinámico

Para este caso de uso se ha detectado únicamente la operación de sistema *activarPerfil*, que se encarga de llevar a cabo la activación del perfil en el caso que el usuario tenga uno asociado, o bien de pedir la confirmación al cliente para crear su perfil porque es la primera vez que utiliza el servicio.

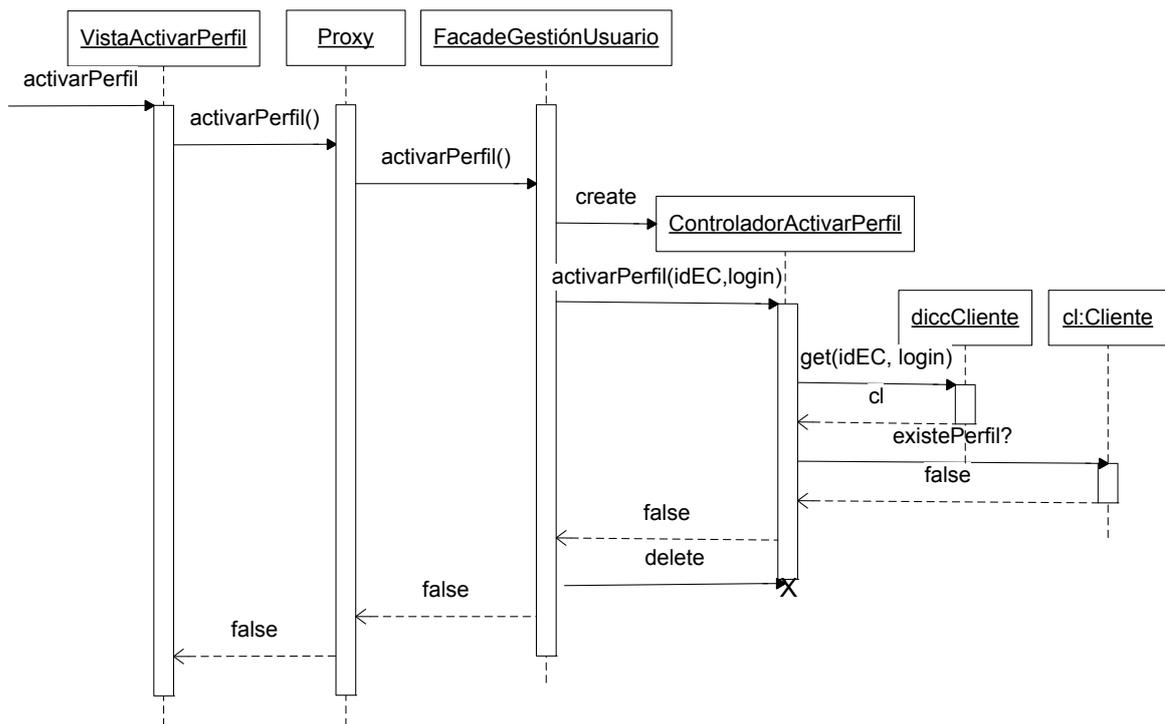


Ilustración 25: Aspecto dinámico del diseño del caso de uso ActivaPerfil.

En el diagrama de secuencia de la Ilustración 25 se observan las siguientes acciones por cada una de las clases de diseño que interactúan en el flujo básico del caso de uso:

1. *VistaActivarPerfil*. Indica al sistema que active el perfil del cliente mediante la clase Proxy.
2. *FacadeGestiónUsuario* crea una instancia del use-case controller *ControlActivarPerfil* que es la que se encarga de llevar a cabo la activación del perfil del cliente.
3. *ControlActivarPerfil* obtiene la instancia del cliente con el identificador proporcionado y comprueba si tiene un perfil asociado. Dado que no es ese el caso, devuelve el valor “false” para indicar a la *VistaActivarPerfil* que es necesario crear el perfil.
4. *VistaActivarPerfil* notifica al usuario que no tiene un perfil asociado por lo que es necesario crear el perfil.

4.3.3.2.1 Extensión: Crear perfil de usuario

Aspecto estático

El diagrama de clases de la Ilustración 26 muestra las clases de diseño que participan en las operaciones diseñadas para este caso de uso. De la misma manera como se hizo en el caso de uso AltaContenido, el parámetro “datos” de tipo string se ha utilizado para simplificar la representación de la operación. En realidad la operación *crearPerfil* tiene un parámetro para cada uno de los atributos descriptivos que forman el perfil (ver capítulo 2.1)

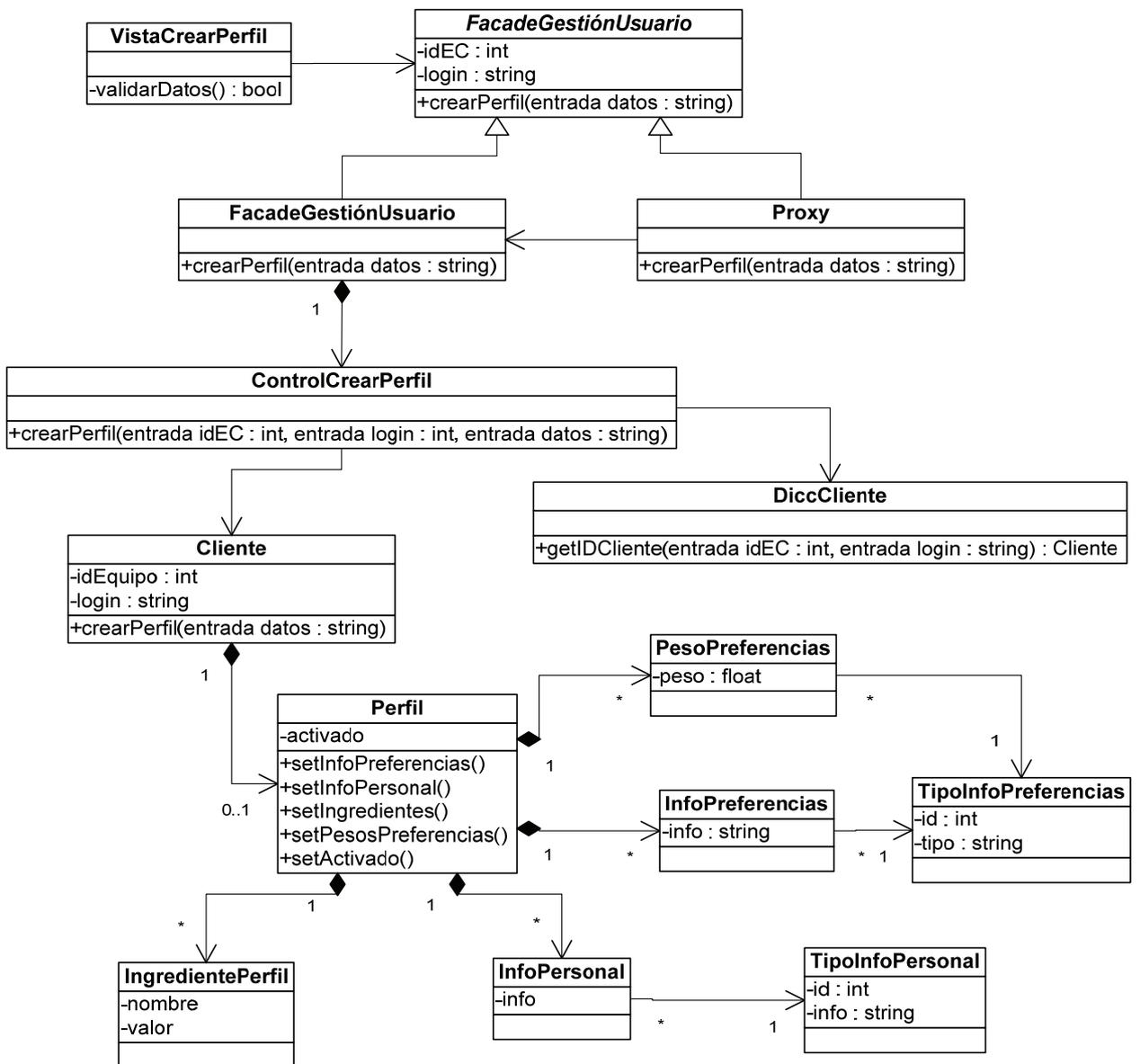


Ilustración 26: Aspecto estático del diseño del caso de uso CrearPerfil.

Aspecto dinámico

Para este caso de uso se ha detectado únicamente la operación de sistema *crearPerfil*, que se encarga de la creación del perfil de usuario.

En el diagrama de secuencia de la Ilustración 27 se observan las siguientes acciones por cada una de las clases de diseño que interactúan:

1. *VistaCrearPerfil* se asegura que los datos introducidos por el cliente estén en el formato correcto. Una vez validados, indica al sistema mediante el *Proxy* que cree un nuevo perfil para el cliente que ha iniciado el caso de uso.
2. *FacadeGestiónUsuario* crea una instancia del use-case controller *ControlCrearPerfil* que es la que se encarga de llevar a cabo la creación del perfil del cliente.
3. *ControlCrearPerfil* crea una nueva instancia de la clase *Perfil* con los datos proporcionados por *VistaCrearPerfil*.
4. El mismo controlador obtiene mediante *diccCliente* la instancia de *Cliente* que se corresponde con el identificador, y le asocia la instancia de *Perfil* que ha creado.
5. Finalmente *FacadeGestiónUsuario* devuelve el control a la vista indicando que la creación ha sido satisfactoria, momento en el cual *VistaCrearPerfil* muestra un diálogo avisando al cliente.

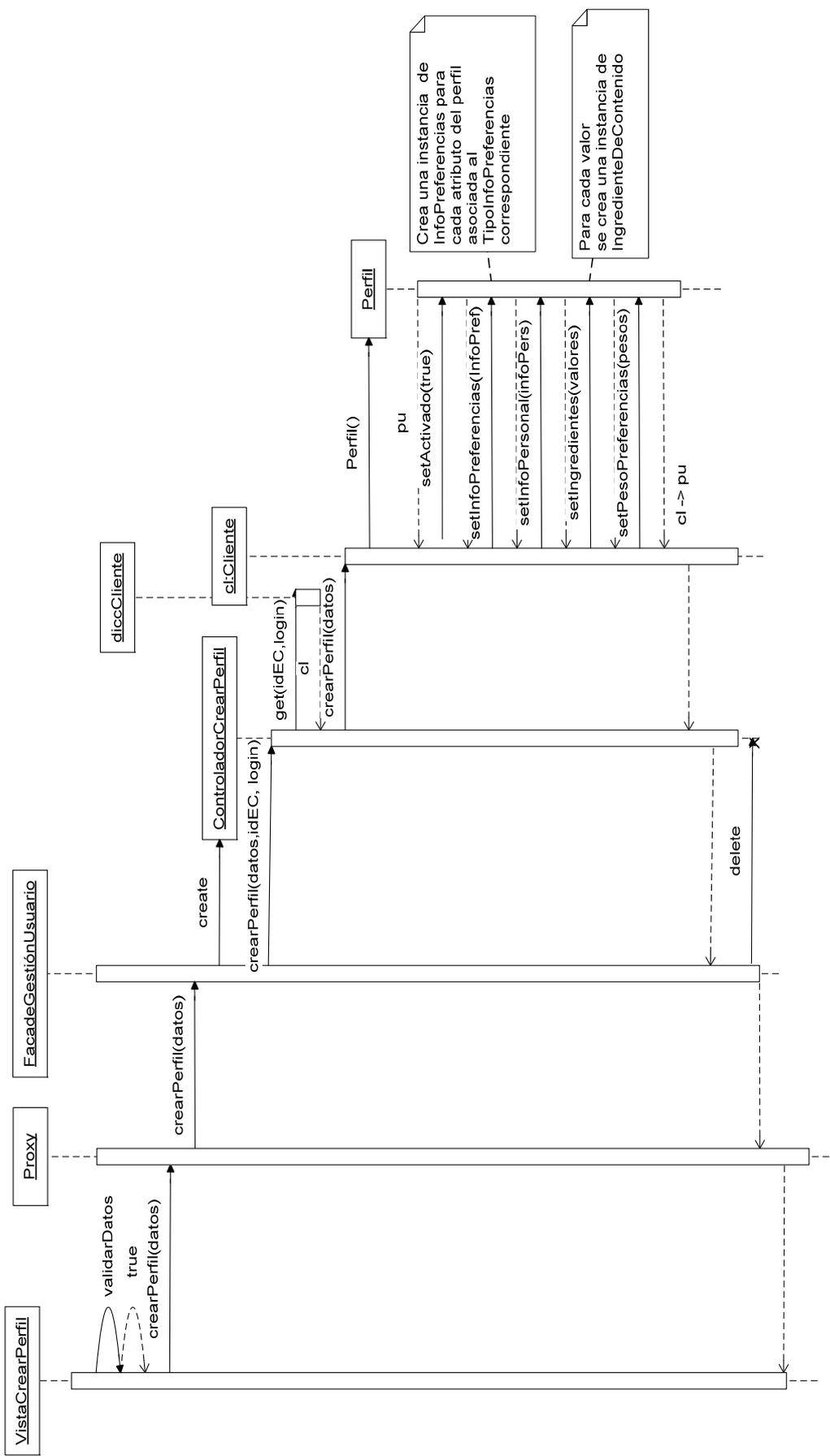


Ilustración 27: Aspecto dinámico del diseño del caso de uso CrearPerfil.

4.3.3.3 Caso de uso: Pedir recomendación de usuario

Aspecto estático:

El diagrama de clases de la Ilustración 28 muestra las clases de diseño que participan en las operaciones diseñadas para este caso de uso, sin contemplar algunas de las clases que interactúan en la comunicación con el subsistema de diseño Servicio de Integra.

Para simplificar la representación del diagrama, no se han representado los parámetros y valores de retorno de algunas operaciones. Concretamente estas operaciones son:

- *recomendaciónCaso1* cuyo parámetros son el Perfil y el número de contenidos máximo, y tiene como valor de retorno una lista de Contenido;
- *combinaciónCaso1* cuyo parámetros son las listas de Contenido de cada uno de los componentes de recomendación, y tiene como valor de retorno una lista de *ContenidoRecomendado*.

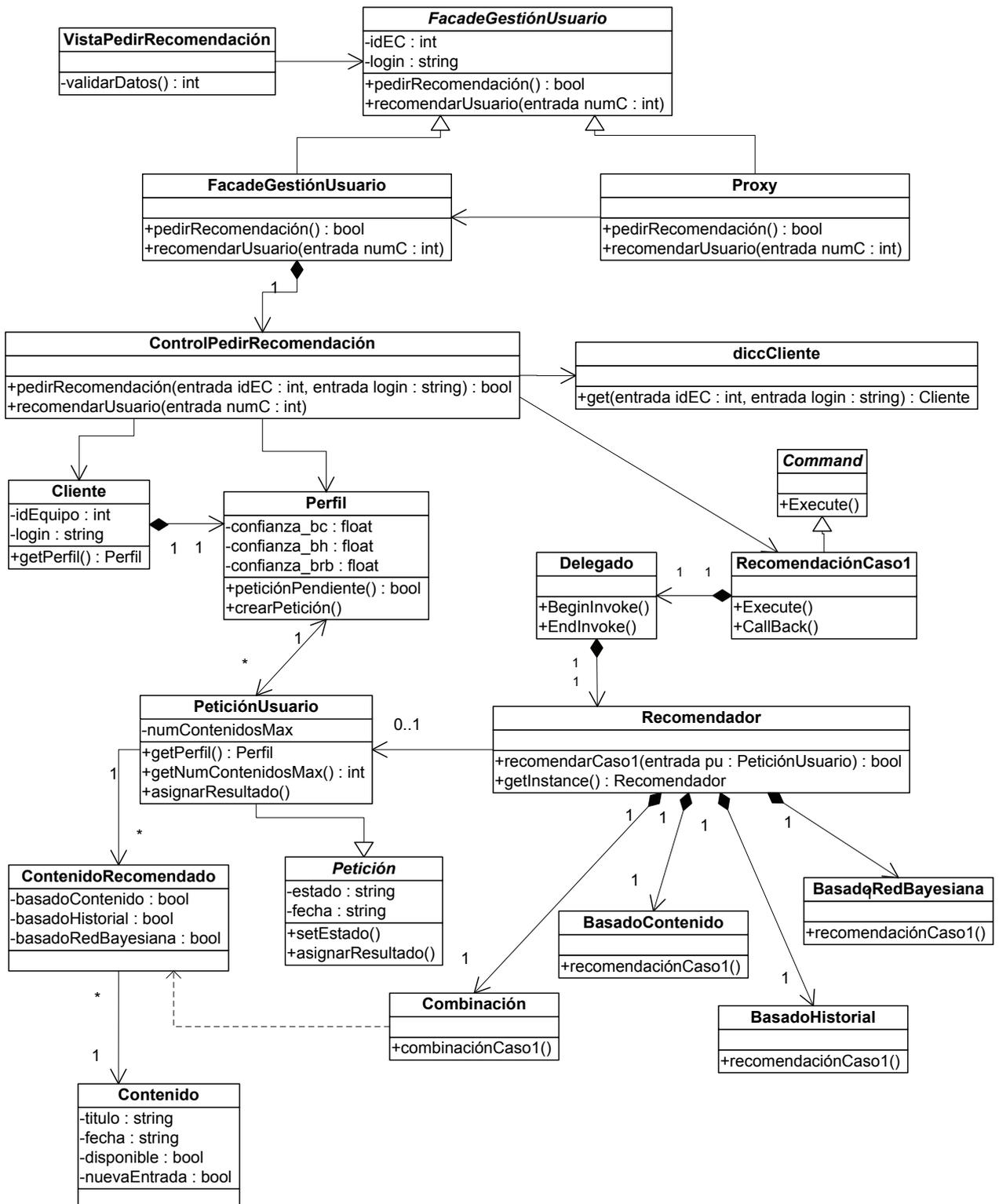


Ilustración 28: Aspecto estático del diseño del caso de uso Pedir recomendación de usuario (parte 1).

En el diagrama de clases de la Ilustración 29 se observan las clases que participan en el proceso de comunicación con el subsistema de diseño Servicio

de Integra. El parámetro de entrada *contenidos*, en la operación *actualizarListaCliente* de la interfaz del subsistema Servicio de Integra representa una lista de identificadores de contenido.

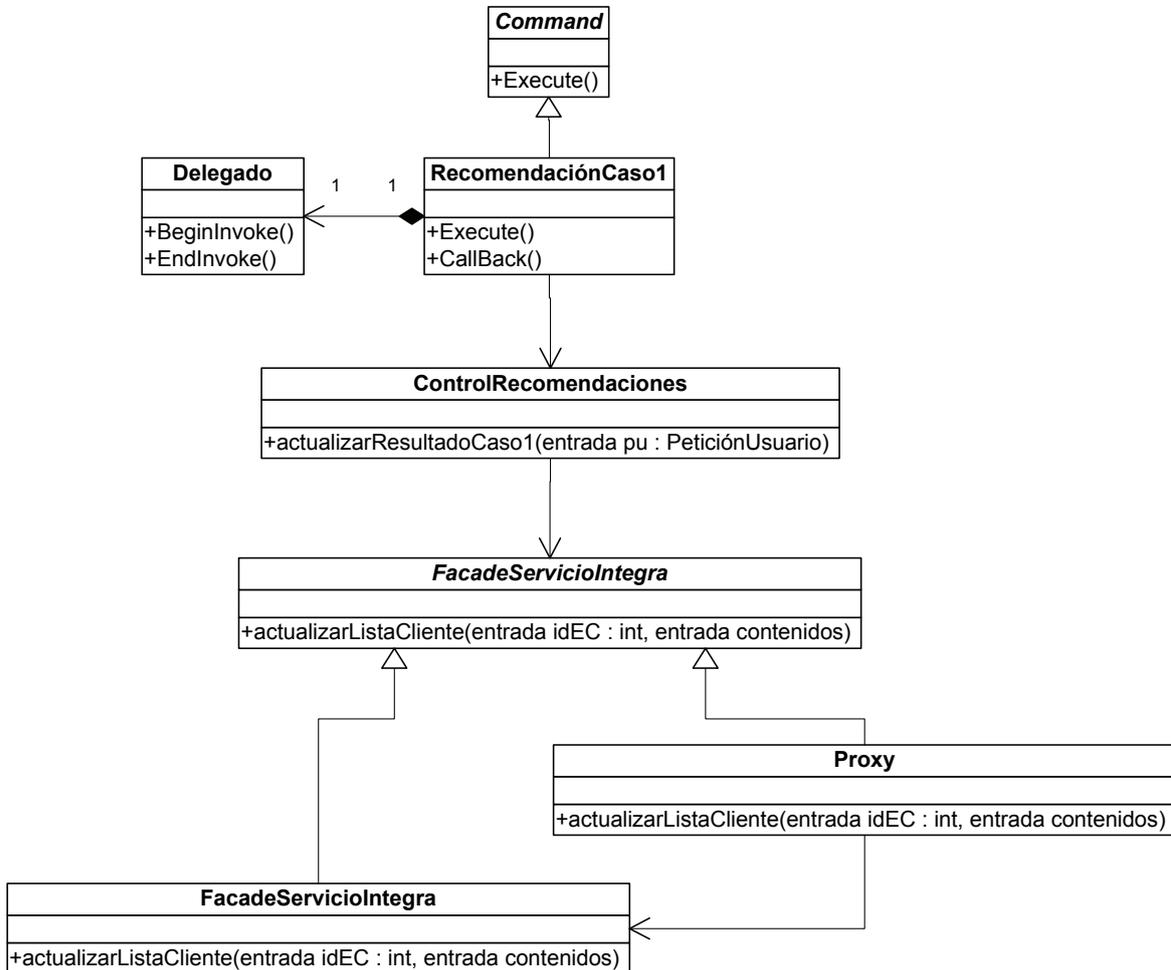


Ilustración 29: Aspecto estático del diseño del caso de uso Pedir recomendación de usuario (parte 2).

Aspecto dinámico:

Para este caso de uso se han detectado las operaciones de sistema *pedirRecomendación* y *recomendarUsuario*.

- *pedirRecomendación*: verifica que el cliente no tenga una petición pendiente para poder seguir con la recomendación, y en caso afirmativo pide confirmación al cliente para proseguir con la recomendación.

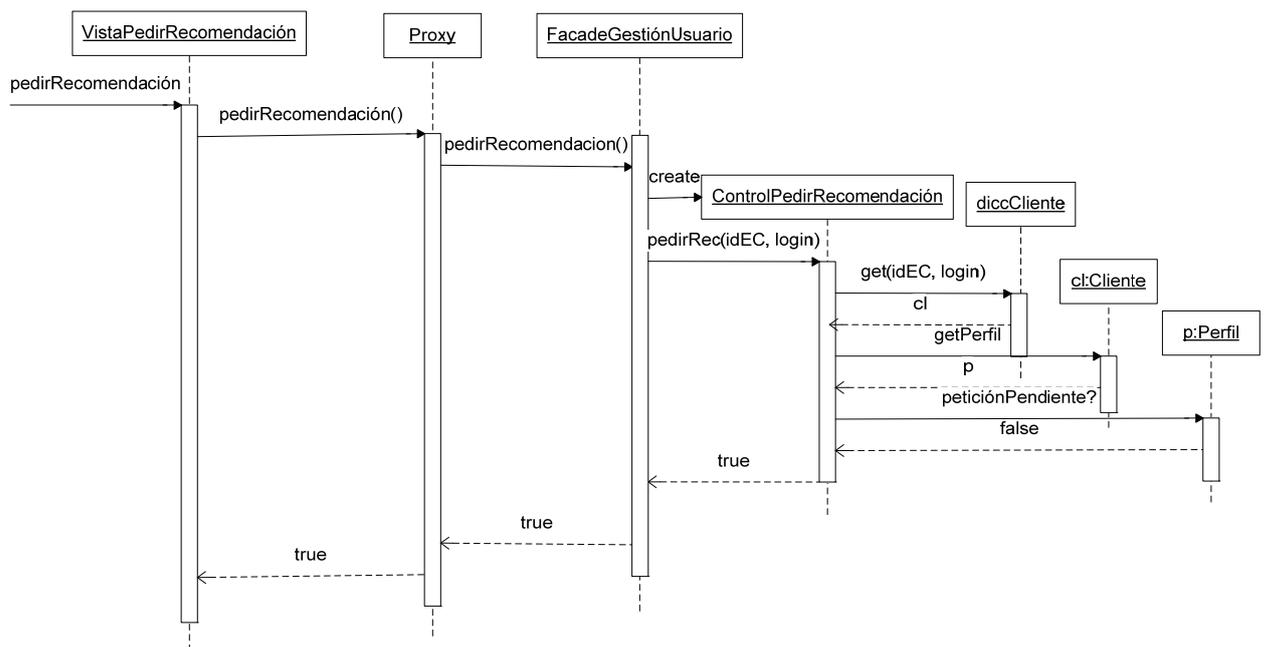


Ilustración 30: Aspecto dinámico del diseño del caso de uso Pedir recomendación de usuario (operación *pedirRecomendación*)

En el diagrama de secuencia de la Ilustración 30 se observan las siguientes acciones por cada una de las clases de diseño que interactúan en el flujo básico del caso de uso:

1. *VistaPedirRecomendación*. Indica al sistema que el cliente quiere una recomendación mediante la clase Proxy.
 2. *FacadeGestiónUsuario* crea una instancia del use-case controller *ControlPedirRecomendación* que es la que se encarga de llevar a cabo la petición de recomendación de usuario.
 3. *ControlPedirRecomendación* se encarga de obtener el perfil del cliente a recomendar con el identificador proporcionado y comprueba si la instancia de *Perfil* tiene asociado alguna petición pendiente.
 4. Una vez que el controlador comprueba que no tiene una petición pendiente devuelve el control a la vista indicándole que puede proseguir con la recomendación.
 5. *VistaPedirRecomendación* le muestra un diálogo al cliente pidiendo confirmación al cliente para proseguir.
- *recomendarUsuario*: registra la petición de usuario y utilizando el patrón de diseño Command, encapsula la operación de calcular la recomendación en una clase para que sea procesada por un hilo secundario, mientras se

notifica al cliente que la petición está en curso y se le devuelve el control a la interfaz.

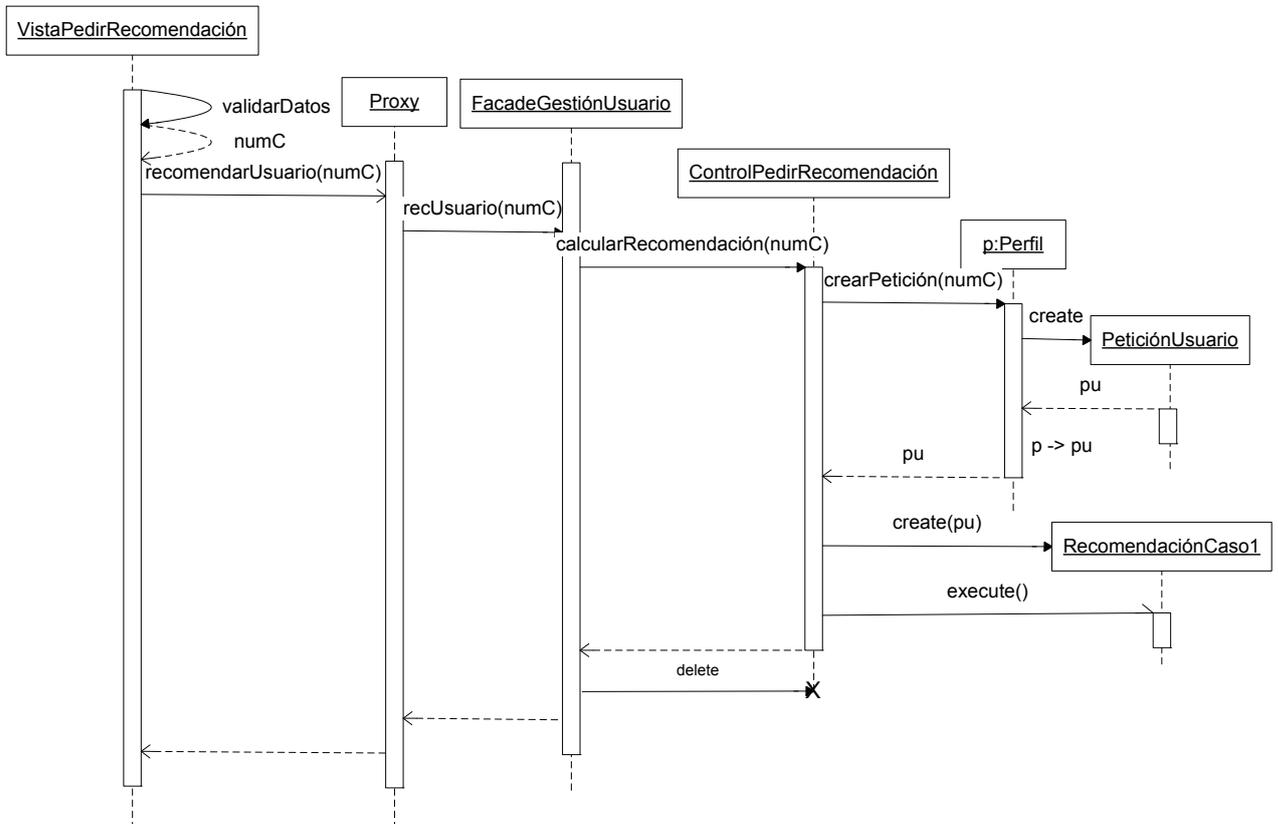


Ilustración 31: Aspecto dinámico del diseño del caso de uso Pedir recomendación de usuario (operación *recomendarUsuario*).

En el diagrama de secuencia de la Ilustración 31 se describe la secuencia de operaciones hasta el momento de pasar a ejecutar la operación asíncrona del cálculo de la recomendación en el caso 1.

1. *VistaPedirRecomendación* comprueba que el número de contenidos máximo indicado por el usuario esté dentro del rango aceptado, para luego indicar al sistema que calcule una recomendación para el cliente mediante la clase *Proxy*.
2. *FacadeGestiónUsuario* invoca a *ControlPedirRecomendación* que es el que lleva a cabo el proceso de registrar la petición de usuario y calcularle la recomendación.
3. *ControlPedirRecomendación* invoca la operación de crear petición de la clase *Perfil*, que le devuelve la instancia de *PeticiónUsuario*. Después obtiene la única instancia de la clase *Recomendador* (a través del patrón Singleton). A continuación el controlador crea una instancia de la clase *RecomendaciónCaso1*, le asocia la instancia del *Recomendador* y finalmente le invoca la operación asíncrona *execute*.

4. Después de que el use-case controller invoque la operación asíncrona, devuelve el control a `FacadeGestiónUsuario` que se encarga de destruirlo.
5. Cuando `VistaPedirRecomendación` recibe el control, muestra un diálogo al cliente notificándole que la petición está en curso.

En la Ilustración 32 se observa el diagrama de secuencia de la operación *execute*, que se encarga de calcular la lista de contenidos recomendados utilizando los algoritmos de los diferentes componentes de recomendación comentados en detalle en el capítulo 3.2. Por este motivo, no se mostrará el diseño de las operaciones que calculan la recomendación en cada uno de los componentes.

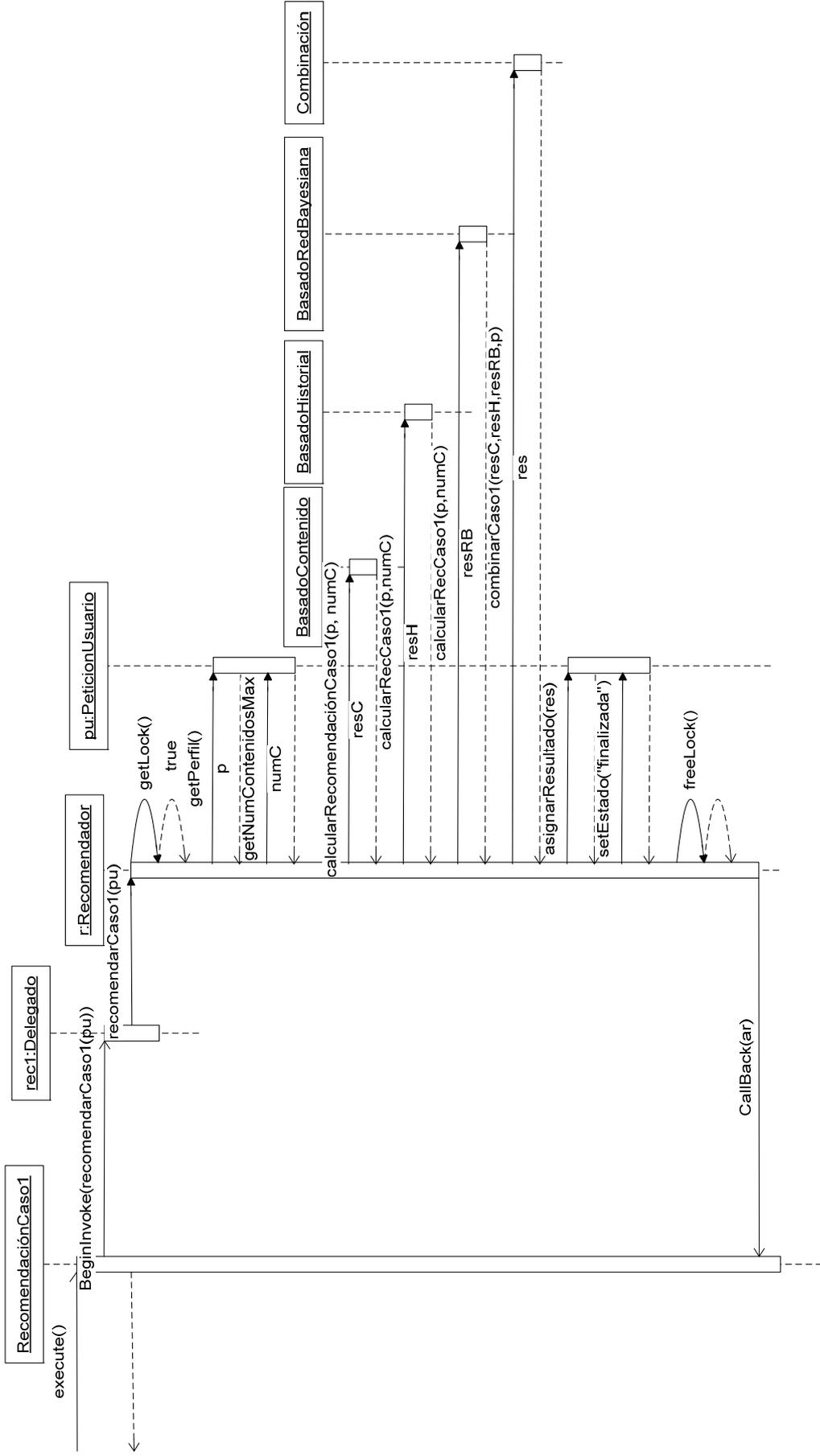


Ilustración 32: Aspecto dinámico del diseño del caso de uso Pedir recomendación de usuario (operación asincrónica *execute*).

A continuación se describe la secuencia de eventos que suceden en el flujo básico de la realización de la operación *execute*:

1. *RecomendaciónCaso1* invoca la operación asíncrona de la clase *Recomendador* que calcula la recomendación en el caso 1, siguiendo el patrón de diseño Asíncrono basado en delegados. De manera que después de llamar a *BeginInvoke* se asigna la ejecución de la operación a un hilo secundario de la pool de hilos.
2. Al ejecutarse la operación asíncrona, la primera acción llevada a cabo consiste en controlar que no exista ningún otro hilo ejecutando la operación. Este control se consigue invocando la operación *getLock()*. Una vez se tiene acceso al código sincronizado, *Recomendador* obtiene la instancia del perfil de la petición de usuario a resolver, así como el número de contenidos máximo que ha de contener la lista de recomendados de esta petición.
3. Luego se invocan las operaciones correspondientes al cálculo de la recomendación en el caso 1 de los tres componentes de recomendación, pasándoles los parámetros obtenidos en el paso 2. Una vez *Recomendador* tiene las listas de recomendación de cada componente invoca la operación correspondiente de la clase *Combinación* que formará la lista de recomendados final basándose en las confianzas asociadas al perfil.
4. Una vez el *Recomendador* posee el resultado de la recomendación, invoca la operación de asignar resultado de *PeticiónUsuario* que es quien se encarga de crear las asociaciones con las instancias de *ContenidoRecomendado* que conforman el resultado, y cambia el estado de la petición a “finalizada”.
5. Después de finalizar la petición, mediante la operación *freeLock()* se indica se ha terminado de ejecutar las acciones sincronizadas, por lo que si existen más hilos esperando, dejará pasar al primero que había invocado la operación.
6. Finalmente cuando se termina la ejecución de la operación asíncrona, el hilo secundario pasa a ejecutar la operación de retorno *callBack(ar)*. El parámetro de esta operación es del tipo *IAsyncResult*: una clase del framework de .Net donde se almacenan todos los datos referentes a la ejecución de la operación asíncrona, como por ejemplo el valor de retorno.

En la Ilustración 33 se observa el diagrama de secuencia de la operación de retorno *callBack*, que se encarga de obtener el resultado de la operación asíncrona y actualizar la lista de contenidos asignados al cliente recomendado a través de la interfaz del Servicio de Integra.

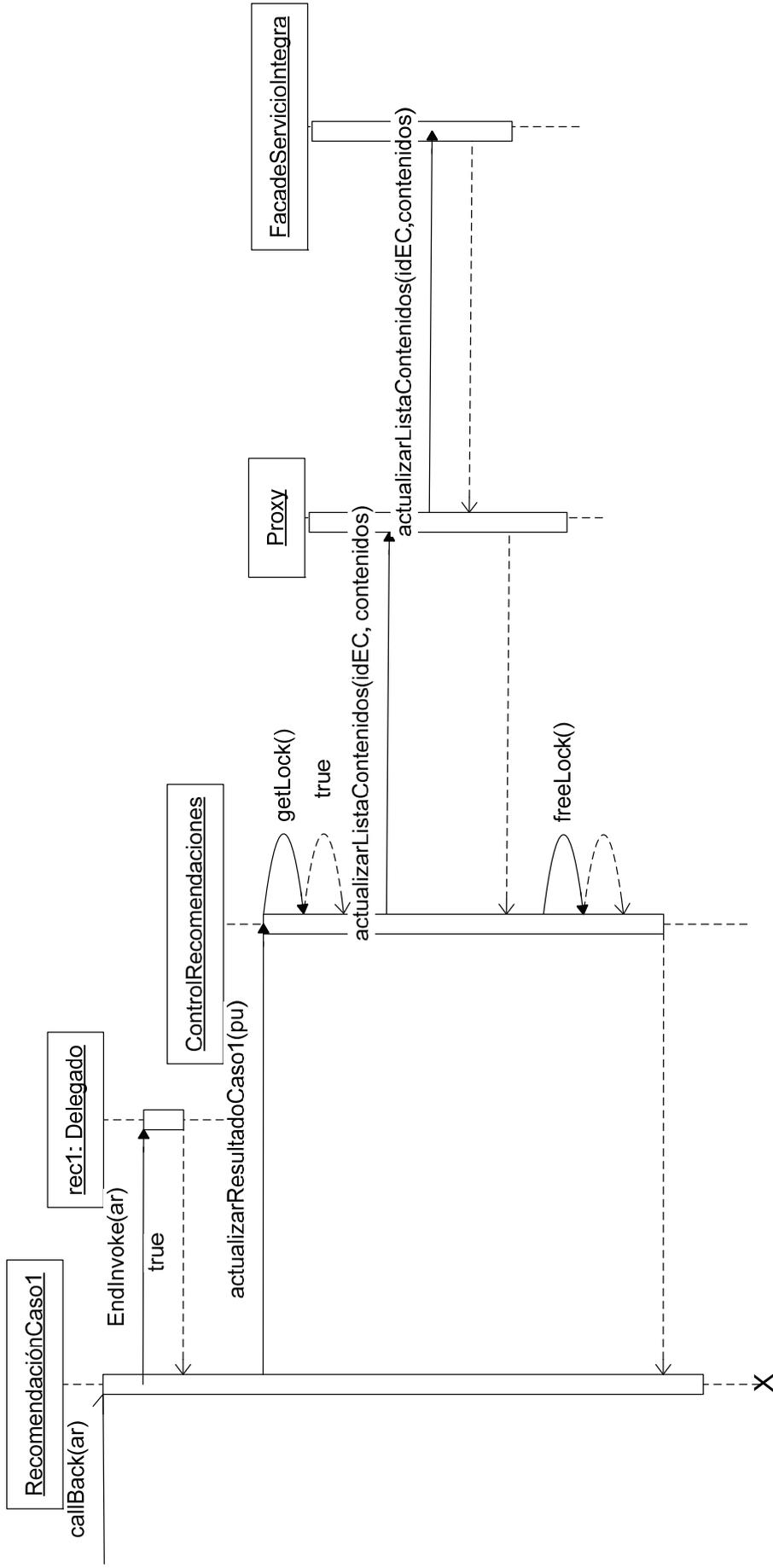


Ilustración 33: Aspecto dinámico del diseño del caso de uso de Pedir Recomendación de Usuario (operación asíncrona *callBack*).

A continuación se describe la secuencia de eventos que tienen lugar en el flujo básico de la realización de la operación *callBack(ar)*:

1. Dado que cuando se ejecuta la operación de *callBack* significa que la operación asíncrona ha finalizado, la primera acción llevada a cabo consiste en obtener el resultado de la operación invocando *EndInvoke(ar)* de la instancia de *Delegado* asociado a la operación.
2. En el caso de que el resultado sea positivo (valor = "true"), *RecomendaciónCaso1* se comunica con la clase *ControlRecomendaciones* para actualizar la lista de distribución de contenidos del usuario recomendado. Para evitar inconsistencias en las actualizaciones de las listas, es necesario sincronizar el acceso a la operación que realiza la actualización. Por este motivo, se ha utilizado el patrón Singleton para el diseño de la clase *ControlRecomendaciones*, de manera que sólo exista una instancia en el sistema, y se controle así el acceso de los diferentes hilos.
3. De igual manera que en la operación *execute*, se controla que sólo exista un hilo ejecutando la operación de actualizar la lista de contenidos mientras los otros esperan a que termine.
4. Una vez se obtiene acceso al código sincronizado, se invoca la operación de actualización de la lista de contenidos de la interfaz al Servicio de Integra *FacadeServicioIntegra*. Si la actualización es correcta se libera la barrera para permitir la ejecución de otro hilo en espera y se termina la ejecución del hilo actual.

4.3.4 Diseño de la interfaz de usuario

Como se ha podido ver en capítulos anteriores, el diseño de la interfaz de usuario depende directamente de los componentes de la parte cliente de la aplicación de IntegraTV. En concreto del FrontEnd (la aplicación del cliente) y BackOffice (la aplicación que utilizan los administradores). Por este motivo en los siguientes apartados se describirán cómo se han integrado las interfaces de usuario en los distintos componentes, además de mostrar en detalle el diseño de las interfaces que participan en los casos de uso diseñados en el capítulo anterior.

4.3.4.1 Subsistema IU Cliente

El diseño del subsistema IU Cliente tiene como objetivo integrar las nuevas interfaces de usuario así como modificar las existentes, siguiendo en la medida de lo posible el diseño actual de la interfaz del FrontEnd. Este requisito de mantener la interfaz viene dado por el hecho de que es una interfaz plenamente validada por expertos en usabilidad. Hay que tener en cuenta que el dispositivo de interacción básica del cliente es el mando a distancia especialmente creado para el producto de IntegraTV, y esto limita notablemente la accesibilidad a las distintas funcionalidades que ofrece la aplicación.

El Subsistema IU Cliente está formado básicamente por las interfaces de usuario que permiten al cliente utilizar las funcionalidades propias del Servicio de Recomendación. En el mapa de navegación de la Ilustración 34 se observan el conjunto de pantallas que se han añadido o modificado como resultado de la extensión del servicio de recomendación (con título de color negro), junto a las pantallas más representativas del diseño actual (con título de color blanco).

En el mapa de navegación se observa que el diseño de la interfaz se divide en dos grupos bien diferenciados: las relacionadas con la gestión de la recomendación que forman parte del servicio del videoclub (parte izquierda del mapa), y las que pertenecen a la gestión del perfil de usuario dentro del menú de usuarios (parte derecha del mapa). Para facilitar la representación del mapa de navegación, las relaciones de doble dirección se han representado mediante el uso de la doble flecha, indicando como nombre de la relación la acción principal.

Aunque no se aprecia en el mapa de navegación, una característica del diseño de la interfaz actual, es que la mayoría de las pantallas tienen accesos directos a distintas pantallas principales de la aplicación. Concretamente enlazan con la pantalla del MenuPrincipal, la de configuración general y la de GestiónUsuarios. Además se ofrece un enlace con ayuda sobre la pantalla que el cliente está usando en ese mismo instante.

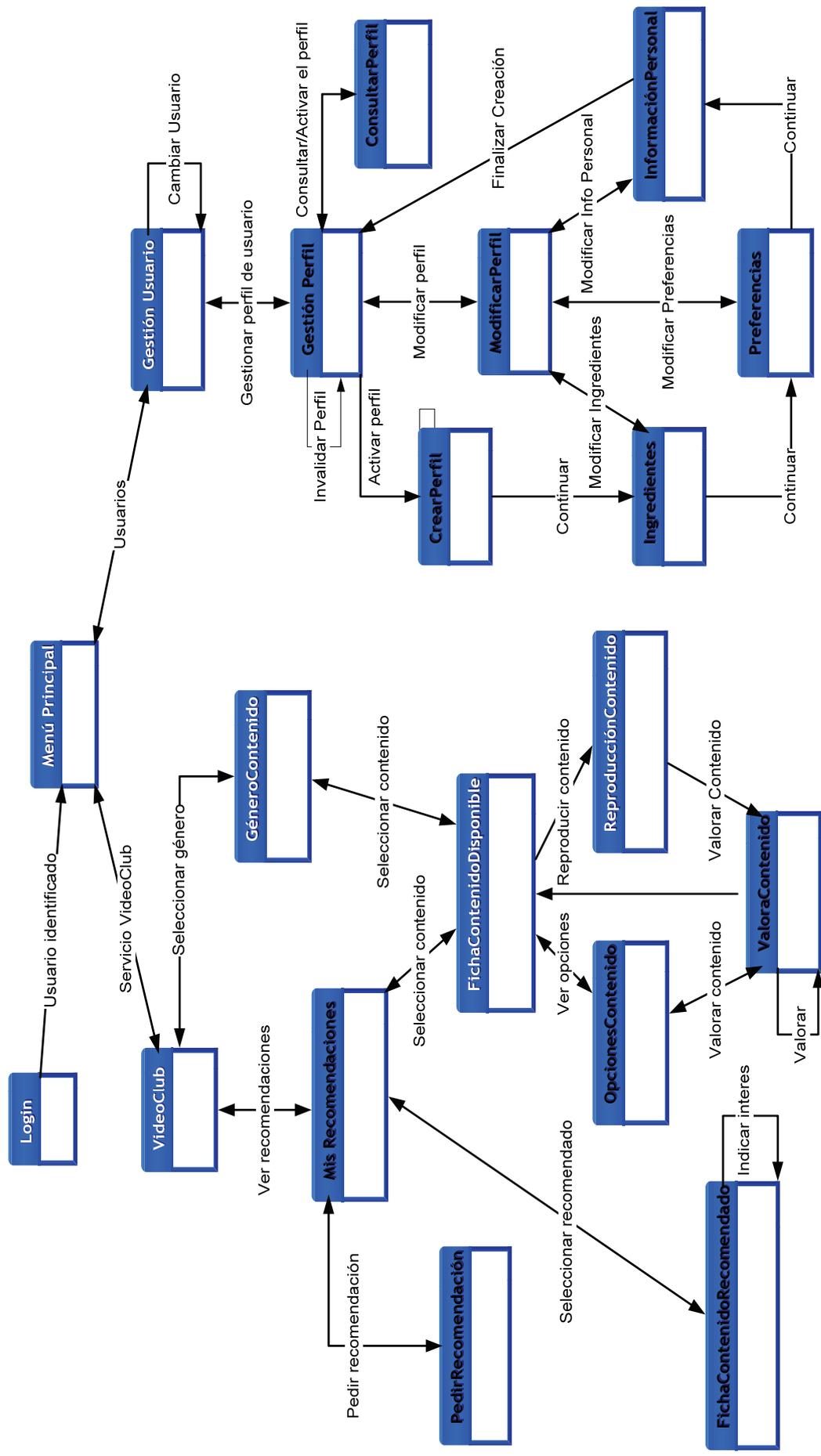


Ilustración 34: Mapa de navegación de las interfaces de usuario del FrontEnd

A continuación se comentarán brevemente las relaciones entre las interfaces de usuario más representativas del mapa de navegación.

Sobre las interfaces involucradas de la gestión de las recomendaciones destacar que:

- Desde la pantalla de MisRecomendaciones, el cliente puede pedir una recomendación, seleccionar un contenido recomendado señalado como disponible, y por lo tanto con la posibilidad de poderlo reproducir entre otras opciones. Además podrá seleccionar los contenidos recomendados señalados como no disponibles para indicar al sistema mediante un nivel de interés si el contenido ha de ser descargado o no al equipo del cliente.
- La funcionalidad de valorar un contenido puede darse desde dos puntos diferentes de la aplicación. Un caso es cuando termina la reproducción de un contenido, momento en el que aparece la pantalla de valoración. El otro punto es desde la pantalla de opciones de un contenido disponible, por lo que se permite hacer valoraciones de contenidos independientemente de si lo ha visto por medio de IntegraTV.

Sobre las interfaces involucradas en la gestión del perfil de usuario comentar que:

- Desde la pantalla GestionPerfil, un nuevo usuario dispone de la funcionalidad de activar el perfil, lo que le llevará al proceso de creación de dicho perfil.
- Desde la misma pantalla, un usuario con perfil creado tendrá las opciones de modificar, consultar e invalidar dicho perfil.

Una vez se ha visto una visión global del diseño de la interfaz de usuario del aplicativo FrontEnd, se mostrarán el diseño individual de las pantallas que se han visto durante el análisis y que interactúan en el diseño de la realización de los casos de uso vistos en el capítulo anterior.

Activar/Crear Perfil

Aquí se presenta el diseño de las interfaces que participan en el flujo básico del caso de uso ActivarPerfil, escenario en el cual el usuario no tiene un perfil asociado (ya que es la primera vez que utiliza el servicio de recomendación), y en consecuencia deberá crearlo. La pantalla desde donde se inicia el caso de uso es la que se observa en la Ilustración 35. Se puede ver que la única opción permitida a través de la interfaz para gestionar el perfil es la de activarlo, ya que las otras funcionalidades no se pueden iniciar sin un perfil creado.

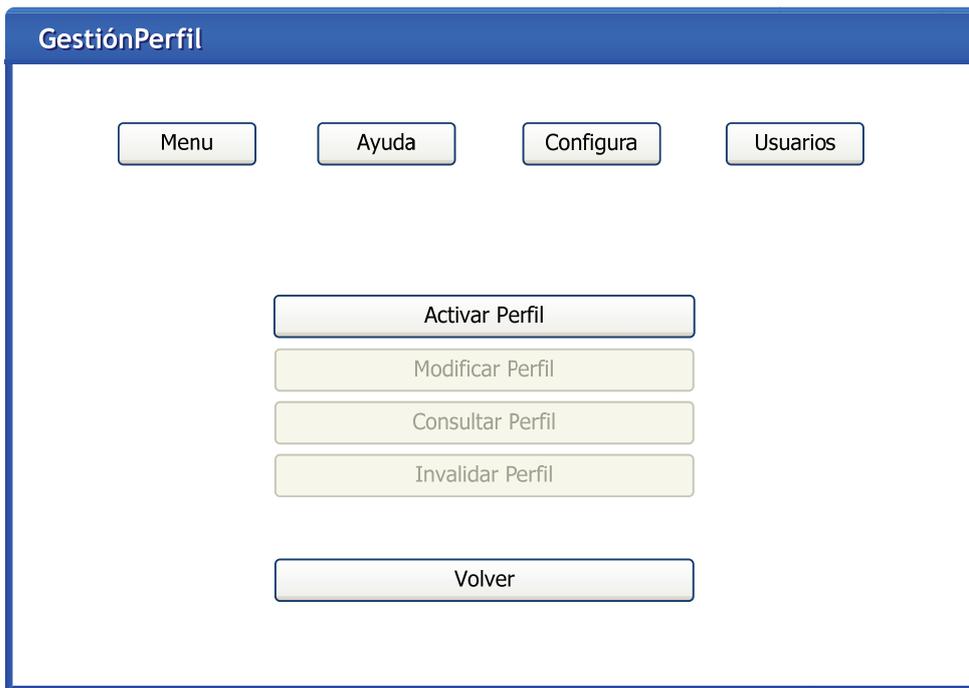


Ilustración 35: Diseño de la interfaz de usuario: GestiónPerfil.

Una vez el usuario pulse sobre el botón de Activar Perfil y dado que no tiene ningún perfil asociado, le aparecerá la pantalla de CrearPerfil (Ilustración 36) para indicarle que es necesario crear un perfil de usuario.



Ilustración 36: Diseño de la interfaz de usuario: CrearPerfil.

Si continua con el proceso de creación de usuario le aparecerán un seguido de pantallas dónde el usuario podrá indicar sus preferencias e información personal. No existe la obligación por parte del usuario, de proporcionar ningún dato de los distintos campos mostrados en las siguientes pantallas, de manera que el sistema se encarga de asignar un valor por defecto en los casos que es necesario indicar algún valor.

Del resto de interfaces involucradas en el caso de uso Crear perfil existen dos diseños de interfaz notablemente distintos según el tipo de dispositivo que utiliza el usuario para interactuar con el sistema: el mando a distancia o el teclado inalámbrico con touchpad integrado. Dado que el teclado permite al usuario una mayor interacción, las interfaces pueden ser más complejas que si se utiliza únicamente el mando a distancia, lo que permite reducir el número de pantallas necesarias.

A continuación se muestran las interfaces diseñadas para utilizar con el teclado inalámbrico. La primera pantalla que aparece es la de Ingredientes (Ilustración 37), donde el usuario indica el tipo de contenido que le gusta más, valorando con la barra deslizante cada uno de los ingredientes que definen un contenido.



Ilustración 37: Diseño de la interfaz de usuario: Ingredientes.

En el caso de que el usuario decida continuar sin proporcionar una valoración de los distintos ingredientes, se le asignará una valoración por defecto. La siguiente pantalla que aparece es la de Preferencias (Ilustración 38), dónde por medio de los diferentes *controles de interfaz* podrá configurar sus preferencias. Como se puede observar, el usuario dispone de distintos desplegables para

seleccionar los descriptores de contenido que prefiera y añadirlos a la lista de seleccionados que formarán parte de su perfil. Además el usuario podrá indicar por medio de un valor numérico, el peso que da a cada uno de los descriptores.

Preferencias

Peso preferencia

Actores preferentes

Añadir Quitar

Lista de actores seleccionados

Directores preferentes

Añadir Quitar

Lista de directores seleccionados

Géneros preferentes

Añadir Quitar

Lista de géneros seleccionados

Nacionalidades preferentes

Añadir Quitar

Lista de nacionalidades seleccionadas

Duración preferente

Preferencia en Novedades

Preferencia en Clásicos

Continuar

Volver

Ayuda

Ilustración 38: Diseño de la interfaz de usuario: Preferencias.

Para finalizar el proceso de creación del perfil, el usuario podrá proporcionar algo de información personal por medio de la pantalla Información Personal (Ilustración 39). De igual manera que en la pantalla anterior, se utilizan desplegados para seleccionar entre diferentes opciones, así como listas para los campos multivalor.

Información Personal

Ocupaciones

Aficiones

Número de Hijos

Estado Civil

Edad

Sexo

Lista de ocupaciones seleccionados

Lista de aficiones seleccionadas

Clase Social

Estudios

Nacionalidad

Finalizar

Volver

Ayuda

Ilustración 39: Diseño de la interfaz de usuario: Información Personal.

Pedir Recomendación

Aquí se presenta el diseño de las interfaces que participan en el flujo básico del caso de uso PedirRecomendación. Dado los pocos datos introducidos por parte del usuario y la reducida interacción que implica la realización de este caso de uso, sólo ha sido necesario el diseño de una pantalla (Ilustración 40). A través de esta pantalla el usuario podrá indicar cuantos contenidos como máximo quiere recibir en la recomendación. Como se ha comentado anteriormente durante el análisis de este caso de uso, se asignará por defecto un valor máximo que optimice el cálculo de las recomendaciones. Los distintos valores permitidos serán algunos próximos al valor por defecto.



Ilustración 40: Diseño de la interfaz de usuario: PedirRecomendación.

4.3.4.2 Subsistema IU Admin.Contenidos

El diseño del subsistema IU Admin. Contenidos tiene como objetivo integrar las nuevas interfaces de usuario que se encargan de la gestión de los contenidos de vídeo a recomendar, modificando lo menos posible el diseño actual de la interfaz de la aplicación BackOffice.

En el mapa de navegación de la Ilustración 41 se observan el conjunto de pantallas que se han añadido o modificado como resultado de la extensión del servicio de recomendación (con título de color negro), junto a las pantallas más representativas del diseño actual. Como se puede ver, únicamente ha sido necesario modificar la interfaz utilizada para crear y modificar datos de un contenido de vídeo.

A continuación se comentarán brevemente las relaciones entre las interfaces de usuario más representativas del mapa de navegación. Desde el menú principal es posible acceder a los menús de los distintos tipos de contenido que gestiona la aplicación. En el mapa se han representado los menús de los contenidos que son de vídeo: las películas, las series y los videoclips. Desde cada uno de estos menús es posible insertar un nuevo contenido correspondiente al menú, dar de baja uno existente, así como modificar los datos de los contenidos. Las funcionalidades dar de alta y modificar un contenido se realizan desde la interfaz *DatosContenidoVídeo*.

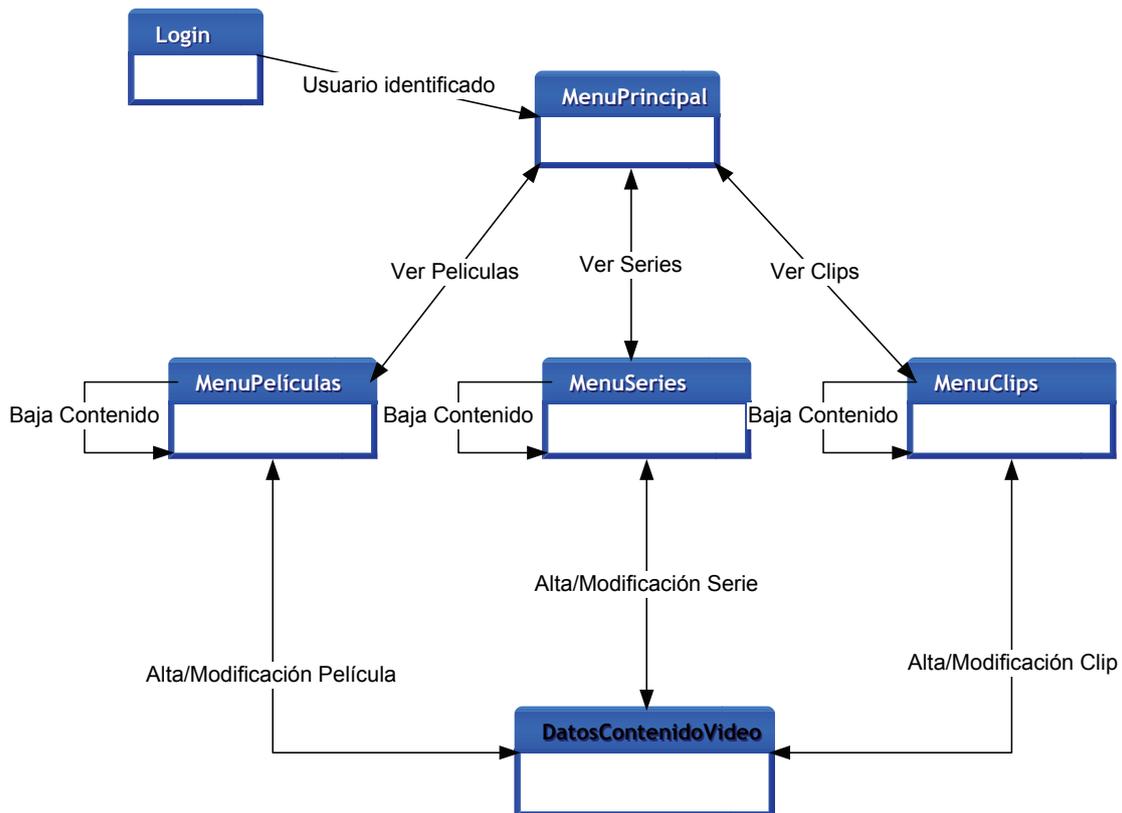


Ilustración 41: Mapa de navegación de las interfaces de usuario del BackOffice.

Una vez se ha visto una visión global del diseño de la interfaz de usuario del aplicativo BackOffice, se mostrará el diseño individual de las pantallas que se han visto durante el análisis y que interactúan en el diseño de la realización de algunos de los casos de uso vistos en el capítulo anterior.

Alta Contenido

Para minimizar los cambios en la interfaz actual, y como los cambios a realizar de la interfaz consisten en añadir nuevos campos que permitan al administrador introducir toda la información necesaria para poder recomendar los contenidos, se ha aprovechado la interfaz de usuario vista en la Ilustración 5. De manera que se han añadido dos fichas a la interfaz actual: una para los ingredientes del contenido y otra para los descriptores de contenido. Como ya se ha visto anteriormente, desde la ficha Datos Generales se puede finalizar la creación del nuevo contenido o aceptar los cambios realizados.

La ficha de Ingredientes Contenido (Ilustración 42) consiste en tantas barras de desplazamiento como ingredientes de contenido se utilizan para describir el tipo de contenido de vídeo.

NuevoContenidoVideo

Datos Generales | **Ingredientes Contenido** | Descriptores de Contenido Video

Acción	<input type="range"/>
Diversión	<input type="range"/>
Dramatismo	<input type="range"/>
Erotismo	<input type="range"/>
Humor	<input type="range"/>
Sexo	<input type="range"/>
Suspense	<input type="range"/>
Violencia	<input type="range"/>
Animación	<input type="range"/>
Ciencia-Ficción	<input type="range"/>
Historicismo	<input type="range"/>
Música	<input type="range"/>
Romanticismo	<input type="range"/>
Terror	<input type="range"/>

Ilustración 42: Diseño de la interfaz de usuario: DatosContenidoVideo (ficha Ingredientes).

La ficha Descriptores de Contenido Video (Ilustración 43) está formada por distintos controles de interfaz que permiten al administrador indicar los descriptores del contenido. Cabe destacar que para los descriptores con múltiples valores se utilizan desplegaables con los que se seleccionan y añaden a la lista correspondiente del descriptor.

The image shows a web application window titled 'NuevoContenidoVideo'. It has three tabs: 'Datos Generales', 'Ingredientes Contenido', and 'Descriptores de Contenido Video'. The 'Descriptores de Contenido Video' tab is active and contains the following elements:

- Actores:** A search dropdown, an 'Insertar' button, a 'Lista Actores' list with navigation arrows, and a 'Borrar' button.
- Directores:** A search dropdown, an 'Insertar' button, a 'Lista Directores' list with navigation arrows, and a 'Borrar' button.
- Géneros:** A search dropdown, an 'Insertar' button, a 'Lista Géneros' list with navigation arrows, and a 'Borrar' button.
- Nacionalidades:** A search dropdown, an 'Insertar' button, a 'Lista Nacionalidades' list with navigation arrows, and a 'Borrar' button.
- Duración:** A text input field followed by a spinner control.
- Radio Buttons:** Two radio buttons labeled 'Novedad' and 'Clásico'.

Ilustración 43: Diseño de la interfaz de usuario: DatosContenidoVideo (ficha descriptores).

4.3.5 Diseño de la capa de datos

4.3.5.1 Subsistema de persistencia

Como se ha visto en el apartado de la arquitectura del sistema, se ha separado la gestión de la persistencia de los datos de la lógica del dominio del sistema. De esta manera, el diseño del subsistema de persistencia será independiente del resto del sistema, lo que permitirá una mayor flexibilidad en el momento que se quiera cambiar la tecnología de gestión de la base de datos.

Dado que actualmente la persistencia de la aplicación de IntegraTV está gestionado por un **gestor de bases de datos relacional**, concretamente MS

SQL Server, y que el modelo relacional es uno de los más utilizados, se ha utilizado este modelo para la gestión de la persistencia. Entre las ventajas que aporta el modelo relacional cabe destacar:

1. Garantiza herramientas para evitar la duplicidad de registros, a través de campos clave o llaves.
2. Garantiza la integridad referencial: así al eliminar un registro elimina todos los registros relacionados dependientes.
3. Favorece la normalización por ser más comprensible y aplicable.

Teniendo en cuenta los requisitos de diseño especificados, en los que la eficiencia en las transacciones a la base de datos no es muy relevante, se ha decidido utilizar un motor de persistencia que realice el mapeo de las clases persistentes del dominio a la base de datos relacional, también llamado ORM¹¹. Así conseguiremos una gestión transparente de los datos persistente y centrar nuestros esfuerzos en la lógica de negocio que es la capa más relevante del desarrollo.

Dado que el framework de desarrollo del sistema utilizado es .Net, se ha elegido utilizar **NHibernate** que es una pasarela de Java a .Net del conocido Hibernate. Las principales características de este motor de persistencia son:

- Ofrece un modelo de programación natural.
- Utiliza los convenios nativos del framework de .Net.
- Soporta el mapeo de objetos con atributos complejos y relaciones.
- Se integra completamente con el resto de la aplicación .Net.
- Es código abierto bajo licencia LGPL (Lesser GNU Public License).

En la Ilustración 44 se observa desde un punto de vista a alto nivel la arquitectura de Hibernate.

¹¹ ORM proviene de las siglas de Object and Relational Mapping.

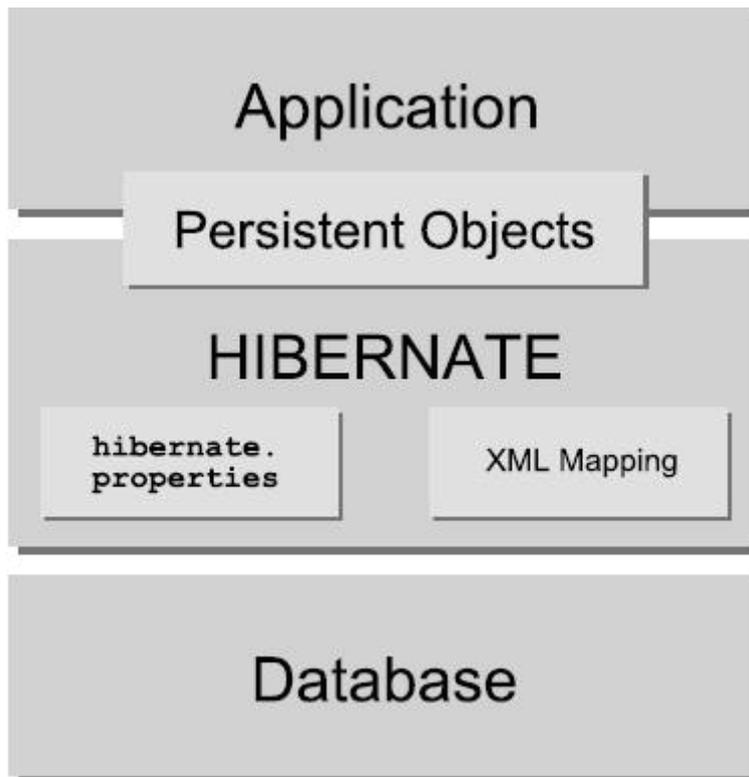


Ilustración 44: Arquitectura simplificada de Hibernate.

El diagrama muestra como Hibernate, a través de las definiciones de mapeo XML y los archivos de configuración, ofrece una gestión transparente de la persistencia de las entidades persistentes de la aplicación, encargándose de la comunicación con la base de datos.

Dado que se quiere tratar la persistencia como un subsistema aparte con el fin de que pueda ser separado físicamente del resto de la aplicación, se hará uso del ya conocido patrón Facade. De esta manera se conseguirá desacoplar notablemente la capa de negocio con la de datos, ya que únicamente existirá una interfaz visible con la que se podrá comunicar. Esto favorecerá que en un futuro se pueda cambiar el motor de la persistencia o bien el SGBD sin que eso afecte en absoluto a la capa de la lógica de negocio.

4.3.5.2 Modelo físico de datos

Como se ha comentado en el apartado anterior, el modelo de base de datos que se ha utilizado para la persistencia de las entidades del sistema, ha sido el modelo Relacional.

En la Ilustración 45 se observa el modelo físico de datos formado por las tablas relacionales que conforman las entidades persistentes del sistema. El diagrama es una representación de las clases de diseño persistentes obtenidas durante el diseño de la realización de los casos de uso. Para conseguir un diseño de la base de datos óptimo se ha seguido el proceso de normalización de bases de

datos relacionales. Este proceso consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo orientado a objetos al modelo relacional, con el fin de:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

Las tablas del modelo que ya existían en la base de datos del servidor de Integra eran la del EquipoCliente y la del Contenido, de las cuales se han obviado los atributos irrelevantes para el desarrollo de este proyecto. Dado que uno de los requisitos es desarrollar el sistema como una extensión de la actual aplicación acoplada lo mínimo posible, inicialmente se utilizará una **base de datos distribuida**. De esta manera se mantendrá la base de datos actual en el servidor de Integra y la base de datos desarrollada para el Servicio de Recomendación en el servidor dedicado a la capa de datos.

Las tablas de color que se observan en el diagrama del modelo de datos, son las tablas que forman parte de la base de datos actual de IntegraTV. Como se puede ver, se ha diseñado la distribución de manera que no existan dependencias de datos entre las tablas que utiliza el sistema actualmente y las tablas propias del Servicio de Recomendación. Lo que facilita la consistencia de la base de datos distribuida y permite optimizar el rendimiento en los accesos.

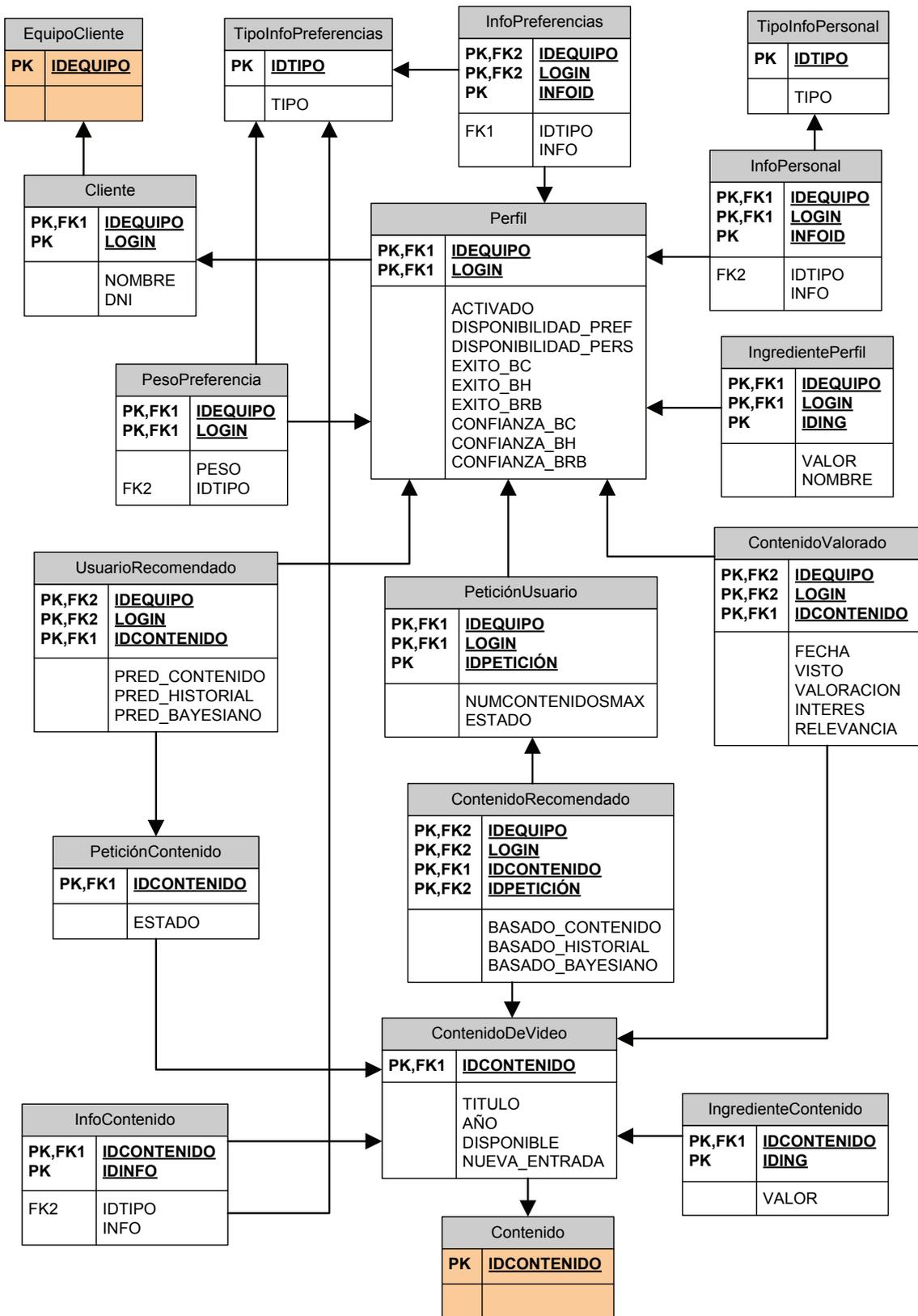


Ilustración 45: Esquema de la base de datos del sistema.

Para que el motor de persistencia NHibernate pueda realizar el mapeo de las clases persistentes a las tablas relacionales, ha sido necesario crear para cada una de las tablas su correspondiente fichero XML Mapping. A modo de ejemplo de la estructura de uno de estos ficheros xml, en la Ilustración 46 se muestra el fichero de mapeo de la clase Perfil.

```
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.0">

  <class name="ServicioRecomendacion.Perfil, ServicioRecomendacion"
    table="perfil">
    <composite-id name="id">
      <key-property name="idEquipo"/>
      <key-property name="login"/>
    </composite-id>
    <property name="Activado" column="activado" type="Boolean"/>
    <property name="Disponibilidad_Pref" column="disp_pref" type="Int16"/>
    <property name="Disponibilidad_Pers" column="disp_pers" type="Int16"/>
    <property name="Exito_BC" column="exito_bc" type="Float"/>
    <property name="Exito_BH" column="exito_bh" type="Float"/>
    <property name="Exito_BRB" column="exito_brb" type="Float"/>
    <property name="Confianza_BC" column="confianza_bc" type="Float"/>
    <property name="Confianza_BH" column="confianza_bh" type="Float"/>
    <property name="Confianza_BRB" column="confianza_brb" type="Float"/>
    <set name="ingredientes">
      <key column="idEquipo"/>
      <key column="login"/>
      <key column="idIng"/>
      <one-to-many class="ServicioRecomendacion.IngredientePerfil,ServicioRecomendacion"/>
    </set>
    <set name="infoPreferencias">
      <key column="idEquipo"/>
      <key column="login"/>
      <key column="idInfo"/>
      <one-to-many class="ServicioRecomendacion.InfoPreferencias,ServicioRecomendacion"/>
    </set>
    <set name="infoPersonal">
      <key column="idEquipo"/>
      <key column="login"/>
      <key column="idInfo"/>
      <one-to-many class="ServicioRecomendacion.InfoPersonal,ServicioRecomendacion"/>
    </set>
    <set name="pesoPreferencias">
      <key column="idEquipo"/>
      <key column="login"/>
      <key column="idTipo"/>
      <one-to-many class="ServicioRecomendacion.PesoPreferencia,ServicioRecomendacion"/>
    </set>
    <one-to-one name="cliente" class="Cliente">
      <column name="idEquipo"/>
      <column name="login"/>
    </one-to-one>
  </class>
</hibernate-mapping>
```

Ilustración 46: Fichero XML de mapeo a la base de datos de la clase Perfil.

Capítulo 5

5 Planificación

5.1 Planificación temporal

La planificación inicial del proyecto se realizó a finales del mes de marzo del 2007. Para conseguir una planificación más precisa, en su realización participó el jefe del proyecto, también director del departamento de investigación de la empresa, ya que había dirigido proyectos de investigación y desarrollo similares.

Dado que el objetivo principal del proyecto es extender la funcionalidad de un producto de empresa que aun está en fase de pruebas para ofrecer un mejor servicio, no han existido unos límites de entrega muy rígidos durante el desarrollo del mismo. De todas maneras, se ha procurado seguir la planificación establecida en la medida de lo posible. En la Ilustración 47 se observa el diagrama de Gantt con la planificación inicial del proyecto.

Durante la realización del proyecto, han surgido imprevistos que han hecho que determinadas tareas se alargasen más de lo esperado, y otras que no se pudieran terminar. Para representar estas desviaciones temporales sobre la planificación inicial, en el diagrama de Gantt de la Ilustración 48 se muestra la duración real de las tareas llevadas a cabo en este proyecto.

En cuanto al conjunto de tareas que conforman el objetivo de diseñar el sistema de recomendación, las tareas que se han desviado notoriamente de lo previsto han sido:

- Tareas 2 y 3: el estudio del estado del arte junto la selección de las alternativas adecuadas.
- Tareas 5 y 6: el desarrollo del prototipo del sistema de recomendación y su posterior evaluación.

A lo que se refiere al conjunto de tareas del desarrollo del sistema de información, cabe destacar que durante la fase del diseño fue complicado elegir la arquitectura de soporte (tarea 15) para que fuera compatible con la

arquitectura de comunicaciones existente. Esto implicó que el diseño de los casos de uso reales (tarea 16) se alargase bastante más de lo previsto.

Finalmente, durante la fase de construcción del sistema a finales de octubre, los máximos dirigentes de la empresa tomaron la decisión de no seguir adelante con el desarrollo del proyecto, ya que según un estudio de mercado que realizó la empresa, actualmente era inviable sacar el producto de IntegraTV Home al mercado. Por ese motivo, se decidió que sólo se construyera una parte del proyecto, en particular el subsistema de diseño Servicio de Recomendación. Las líneas discontinuas que se aprecian en las tareas de la fase de construcción del sistema en la Ilustración 48 indican que dichas tareas se quedaron por terminar.

Aunque no se ve reflejado en el diagrama de Gantt, las fases del ciclo de vida del desarrollo del sistema de información: análisis, diseño y construcción; siguen un proceso iterativo. De manera que el proceso de desarrollo del sistema no es estrictamente como se observa en el diagrama (cuando se termina una fase empieza la otra), sino que en algunas ocasiones es necesario volver a las fases anteriores para resolver un determinado problema de diseño o de construcción.

Si más adelante se decide reiniciar el proyecto de extensión del producto de IntegraTV Home, será fácil seguir con la construcción del sistema desde el punto donde se dejó, gracias al análisis y diseño del sistema que se ha realizado utilizando la metodología de Métrica 3. Y si eso no sucede finalmente, pero se quiere ofrecer un servicio de personalización de contenidos de vídeo en algún otro producto de la empresa, se podrá reutilizar fácilmente el sistema de recomendación desarrollado en este proyecto, gracias al diseño basado en patrones que se ha utilizado.

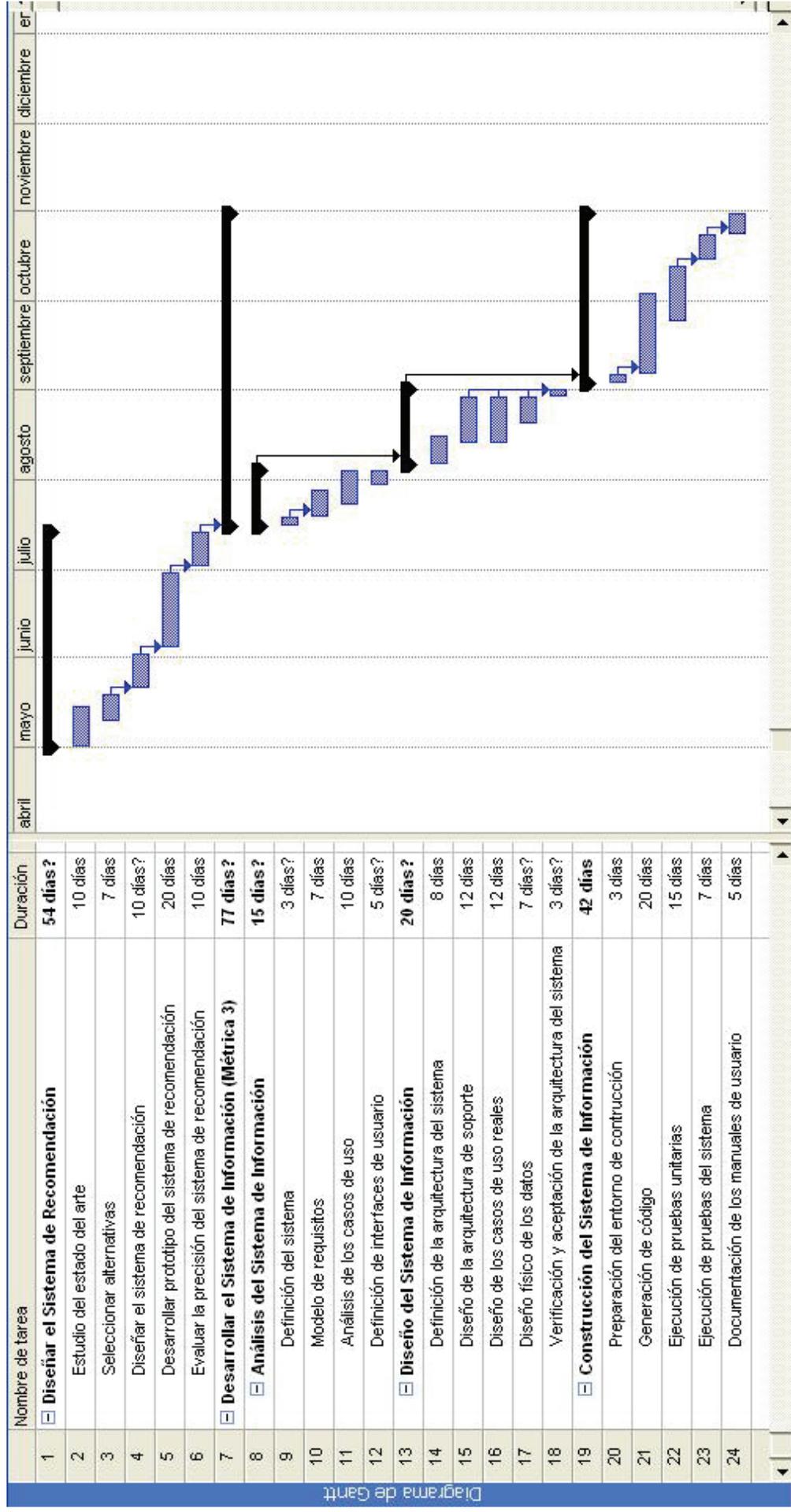


Ilustración 47: Diagrama de Gantt con la planificación inicial del proyecto.

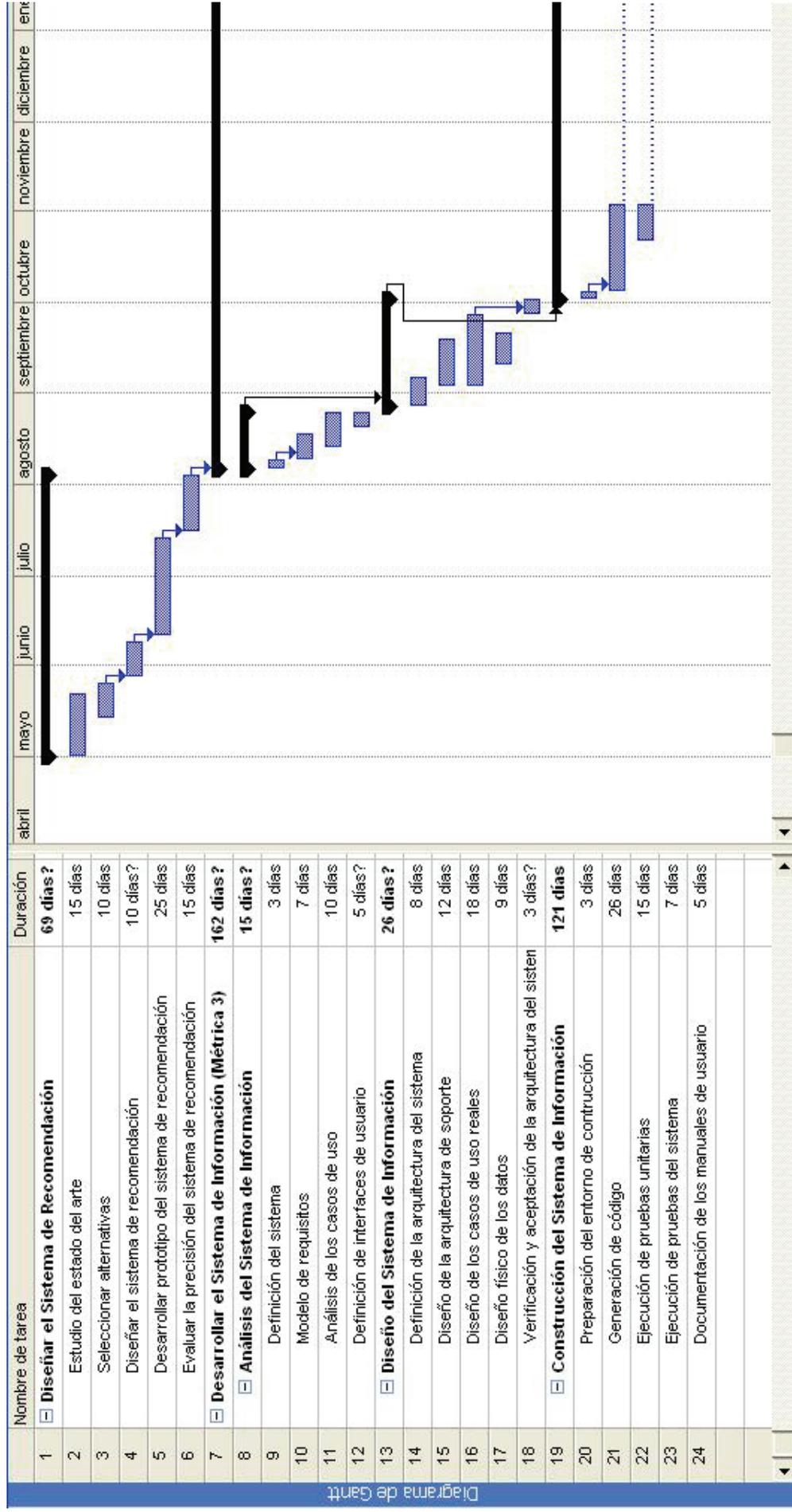


Ilustración 48: Diagrama de Gantt con la duración real de las tareas realizadas.

5.2 Alcance económico

Se ha calculado el coste del proyecto en base a las horas dedicadas por parte del estudiante en concepto de Ingeniero Investigador de la empresa y de los costes relativos al software y hardware utilizado. En la Tabla 5 se observa el coste de las horas trabajadas por parte del estudiante en cada una de las tareas principales del proyecto (ver Ilustración 48).

Teniendo en cuenta que el estudiante ha trabajado el 75% de la jornada completa, es decir aproximadamente unas 6 horas al día, el coste día es igual a: **15 € hora x 6 horas día = 90 € día.**

TAREA	DURACIÓN (Días)	COSTE DÍA	COSTE TOTAL
Diseñar el sistema de recomendación	69	90 €	6210 €
Análisis del Sistema de Información	15	90 €	1350 €
Diseño del Sistema de Información	26	90 €	2340 €
Construcción del Sistema de Información	35	90 €	3150 €
			13050 €

Tabla 5: Coste del proyecto en cuanto a horas trabajadas por el estudiante.

En ningún caso se trata del coste real de las horas dedica al proyecto, ya que sería necesario contemplar además, las horas dedicadas por el resto del personal involucrado en el proyecto. Principalmente, habría que tener en cuenta las horas del: Director de Investigación, quién se ha encargado del seguimiento del proyecto y ha participado en varias reuniones; y del Diseñador, que se ha encargado del diseño de algunas de las interfaces.

En la Tabla 6 se observa el coste del software utilizado para el desarrollo.

Herramienta	Tipo de licencia	Coste
Windows XP Professional Edition	Licencia Microsoft	120 €
Microsoft Office 2003	Licencia Microsoft	280 €
Visual C# 2005 Express	Licencia Gratuita	0 €
SQL Server 2005 Express	Licencia Gratuita	0 €
Protégé 3.3	Open Source	0 €

Tabla 6: Coste del software utilizado para el desarrollo.

En cuanto al coste del hardware utilizado, sólo haría falta añadir el coste del ordenador utilizado para el desarrollo del proyecto que equivale a unos 1100€.

De manera que a partir de los costes descritos anteriormente se deduce de forma aproximada el coste total del proyecto, como se observa en la Tabla 7.

Coste de las horas trabajadas	13050 €
Coste del software utilizado	400 €
Coste del hardware utilizado	1100 €
COSTE TOTAL	14550 €

Tabla 7: Coste total del proyecto.

Finalmente comentar que, al ser un proyecto en el que el cliente es la misma empresa que lo ha desarrollado, se ha mostrado una aproximación del coste real del proyecto en vez de una previsión del mismo como se hubiera hecho en el caso que el destinatario fuera un cliente externo. En ese caso, se hubiera calculado el coste del proyecto en base a la planificación teórica y con los precios establecidos de cara a los clientes.

Capítulo 6

6 Conclusiones

Los sistemas de recomendación han progresado significativamente durante la última década, en la que se han podido ver numerosos sistemas que utilizan diferentes metodologías. No obstante, la actual generación de sistemas de recomendación necesita mejorar varios aspectos para lograr recomendaciones efectivas en entornos de aplicación con las características que se detallan a continuación.

En este proyecto se ha presentado un sistema de recomendación de contenido de vídeo especialmente diseñado para un entorno específico, con una baja densidad de información, y su desarrollo como parte de un sistema de televisión interactiva llamado IntegraTV Home, de la empresa TMT Factory.

El sistema de recomendación diseñado utiliza una metodología híbrida, que consiste en combinar distintas metodologías de manera que se reduzcan las limitaciones que presenta cada una de ellas. En particular se han utilizado tres metodologías: la basada en el contenido, la basada en casos y la de filtro cooperativo.

El **diseño del sistema de recomendación** sigue una arquitectura por componentes, lo que permite perfeccionar el sistema con suma facilidad, extendiéndolo con nuevos componentes o bien mejorando los existentes (ver capítulo 3). El diseño actual está formado por tres componentes, cada uno de los cuales corresponde a una de las metodologías comentadas. Las recomendaciones de cada componente son combinadas en base a un sistema de confianza personalizado para cada usuario, según la retroalimentación obtenida a partir de su interacción con el sistema.

Para el desarrollo del sistema de recomendación como parte de IntegraTV Home, se ha seguido la metodología de desarrollo de software Métrica 3 de forma simplificada, para adaptarla al desarrollo de este proyecto. Con el fin de conseguir en la medida de lo posible un sistema reutilizable en otras aplicaciones e independiente de la aplicación a extender, se ha realizado un diseño altamente modular separado por distintos **subsistemas de diseño** (ver capítulo 4). De esta manera, el sistema de recomendación construido en este proyecto podrá integrarse fácilmente en cualquier aplicación donde sea interesante añadir un servicio de personalización de contenido de vídeo.

6.1 Resultados obtenidos

Los resultados obtenidos como fruto del trabajo realizado en este proyecto han sido los siguientes:

- **El diseño y desarrollo de un prototipo de sistema de recomendación** de contenido de vídeo que utiliza tres metodologías: la basada en el contenido, la basada en casos y la de filtro cooperativo.
- **La integración del sistema de recomendación en una de las aplicaciones de la empresa TMT Factory**, en fase de pruebas en enero de 2008. Gracias al análisis y diseño completo del sistema que se ha realizado, sería relativamente fácil reemprender la construcción desde el punto donde se dejó, en particular la construcción de algunos de los subsistemas de diseño, como los que implican cambios en los componentes de la aplicación a mejorar.
- **La solicitud de una patente en Estados Unidos.** Se ha realizado una solicitud para patentar el sistema de recomendación desarrollado en este proyecto. Se adjunta como anexo de la memoria (ver anexo 4.III) la documentación de la solicitud.

6.2 Mejoras futuras

El sistema de recomendación presentado en este proyecto es mejorable en varios aspectos. Hay que tener en cuenta que el esfuerzo realizado durante el diseño del sistema de recomendación se centró más en las metodologías a utilizar y cómo combinarlas, que en el perfeccionamiento de cada uno de los componentes de recomendación. La asignación de ulteriores recursos permitirá perfeccionar cada uno de los componentes de recomendación de forma independiente y con más detalle.

A parte de poder perfeccionar cada uno de los componentes de recomendación, se podrían aplicar las siguientes mejoras al sistema:

- Enriquecer el perfil de usuario dinámicamente utilizando técnicas de aprendizaje para obtener preferencias implícitas a partir de la interacción del usuario con el sistema.
- Obtener más información sobre las preferencias de un nuevo usuario, utilizando un cuestionario sobre un número determinado de contenidos de géneros variados, dónde el usuario pueda realizar una valoración del contenido o bien indicar un nivel de interés de los contenidos que no haya visto. De este modo el componente basado en historial podría realizar recomendaciones a un nuevo usuario desde la primera vez que usara el sistema.
- Permitir que el sistema de recomendación pueda ajustar de forma automática algunos parámetros de configuración utilizados en la lógica

del cálculo de la recomendación. Actualmente es necesario un administrador del sistema para asignar el valor de los parámetros de configuración en base a las estadísticas de uso del sistema.

En cuanto al desarrollo del sistema cabe destacar que el diseño actual del proceso del cálculo de la recomendación está más pensado en priorizar las distintas peticiones según el orden de llegada o el tipo de petición, que en la eficiencia del cálculo. Es por eso que, aunque cada petición es atendida por hilos independientes, el cálculo de la recomendación se realiza de forma secuencial. Por lo tanto, en un entorno donde el número de peticiones simultáneas fuera considerable, podría ser adecuado estudiar la manera de realizar de forma paralela esa tarea y reducir así el tiempo de cálculo por petición.

6.3 Limitaciones

La principal limitación del sistema de recomendación es la escalabilidad. Dado que se ha diseñado especialmente para un entorno con un número de usuarios reducido, el coste en tiempo de cálculo de los algoritmos diseñados en cada uno de los componentes de recomendación depende fuertemente del número de usuarios utilizado; sobre todo, el componente basado en redes Bayesianas, que utiliza los datos de todos los perfiles de usuario en el cálculo de las recomendaciones y implica un gran consumo de la memoria del sistema para poder realizar el cálculo en tiempos aceptables.

6.4 Valoración personal

Por mi parte la valoración del proyecto ha sido muy positiva, debido a que se han abarcado en el desarrollo del proyecto varios ámbitos de la Ingeniería Informática. Sólo el hecho de realizar un proyecto de Investigación y Desarrollo me ha motivado personalmente por tener que encontrar la mejor solución a un problema totalmente desconocido antes de iniciar el proyecto (el de la recomendación de contenidos de vídeo) en un entorno específico, y finalmente aplicar dicha solución en una aplicación real de la empresa.

Concretamente, para diseñar un nuevo sistema de recomendación ha sido necesario estudiar profundamente el área de los Sistemas de Recomendación. Una de las áreas de investigación de la Inteligencia Artificial que ha ido ganando relevancia por parte de los investigadores desde mediados de la década de los noventa. El estudio realizado, me ha permitido adquirir nuevos conocimientos, además de motivarme a investigar más profundamente sobre esta área u otras parecidas de la Inteligencia Artificial en un futuro próximo.

También cabe destacar que desarrollar el sistema de recomendación como una extensión de una aplicación existente, con las limitaciones y restricciones que eso ha conllevado, ha implicado una dificultad añadida en el desarrollo del sistema. Aunque gracias a estas dificultades, he aprendido una amplia variedad de tecnologías, sobretodo del entorno .Net.

En cuanto a la utilización de la metodología de desarrollo de software Métrica 3, comentar que al tratarse de una metodología más enfocada a proyectos informáticos de gran envergadura ha implicado tener que simplificar varias de las actividades de la metodología para el desarrollo de este proyecto. De todas formas, el hecho de utilizar esta metodología me ha servido para aprender un método de desarrollo de software diferente de los métodos que he utilizado durante la carrera.

Bibliografía

- [1] Anderson M., Ball M., Boley H., Greene S., Howse N., Lemire D., McGrath S., *RACOFI: A Rule-Aplying Collaborative Filtering System*. In Proc. IEEE/WIC COLA'03, Halifax, Canada, October 2003.
- [2] Ardissono L., Gena C., Torasso P., Bellifemine F., Chiarotto A., Difino A., Negro B. *Personalized Recommendation of TV Programs*. Lecture Notes in Artificial Intelligence n. 2829. AI*IA: Advances in Artificial Intelligence, Pisa, Italy, pp. 474-486, © Springer Verla, 2003.
- [3] Ben-Menachem, M. *Writing effective use cases*. SIGSOFT Softw. Eng. Notes 26, 2001.
- [4] Blanco Fernández Y., Pazos Arias J. J., Gil Solla A., Ramos Cabrer M., Barragáns Martínez B. and López Nores M. *A Multi-Agent Open Architecture for a TV Recommender System: A Case Study using a Bayesian Strategy*. In Proc. of the IEEE Sixth International Symposium on Multimedia Software Engineering, 2004.
- [5] Blanco Fernández Y., Pazos Arias J. J., Gil Solla A., Ramos Cabrer M., López Nores M. and Barragáns Martínez B. *AVATAR: Modeling Users by Dynamic Ontologies in a TV Recommender System based on Semantic Reasoning*. In Proc. of the 3rd European Conference on Interactive Television: User Centred ITV Systems, Programmes and Applications (EuroITV-05), 2005.
- [6] Balabanovic M. and Shoham Y. *Combining Content-Based and Collaborative Recommendation*. In Communications of the ACM, 1997.
- [7] Breese, J.S., Heckerman, D., and Kadie, C. *Empirical analysis of predictive algorithms for collaborative filtering*. In Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence, 43-52. Morgan Kaufmann, 1998.
- [8] Burke R. *Hybrid Recommender Systems with Case-Based Components*. ECCBR: 91-105, 2004.
- [9] Cheesman, P., and Stutz, J. *Bayesian classification (AUTOCLASS): theory and results*. In Advances in Knowledge Discovery". AAAI Press, 1995.
- [10] Chickering, D., Heckerman, D. and Meek, C. *A Bayesian approach to learning Bayesian networks with local structure*. In UAI 97, Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence, 80–89. 1997.
- [11] Cooper, J. W. *C# Design Patterns: a Tutorial*. Addison-Wesley Longman Publishing Co., Inc., 2002.

- [12] Efron M. and Geisler, G. *Using Dimensionality Reduction to Improve Similarity Judgements for Recommendation*. Paper presented at the Joint DELOS/NSF Workshop on Personalization and Recommender Systems in Digital Libraries, (June 18-20, 2001, Dublin, Ireland). 2001.
- [13] Gamma, Erich; Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*, hardcover, 395 pages, Addison-Wesley, 1995.
- [14] Linden G., Smith B., York J. *Amazon.com Recommendations: Item-to-Item Collaborative Filtering*. IEEE Internet Computing, vol. 07, no. 1, pp. 76-80, Jan/Feb, 2003.
- [15] Mitchell, T.M.: *Machine Learning*. McGraw Hill, New York, 1997.
- [16] Montaner M. *Collaborative recommender agents based on case-based reasoning and trust*. Ph.D. Thesis, 2003.
- [17] Pennock D.M and Horvitz E. *Collaborative Filtering by Personality Diagnosis: A Hybrid Memory And Model-Based Approach*. Proc. Int'l Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering, Aug. 1999.
- [18] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. *Item-Based Collaborative Filtering Recommendation Algorithms*. In Proceedings of the Tenth International World Wide Web Conference, 285-295. ACM Press, 2001.
- [19] Smyth B. and Bradley K. *Improving Recommendation Diversity*. In D. O'Donoghue, editor, Proceedings of the Twelfth National Conference in Artificial Intelligence and Cognitive Science (AICS-01), pages 75-84, Maynooth, Ireland, 2001.

ANEXOS

I. Glosario de términos

Contenido. Término que con el objetivo de abreviar la escritura durante el desarrollo del proyecto se utiliza para referirse a un contenido de vídeo.

Descriptor. Término utilizado para referirse a parámetros y atributos de cualquier tipo.

Información preferencias. Consiste en los datos de usuario relativos a sus preferencias en cuanto a películas se refiere.

Información personal. Hace referencia a los datos de usuario de carácter personal.

Ingredientes de contenido. Atributos que se caracterizan por ser más concretos que los géneros de contenido y que definen de una forma precisa el tipo del contenido de vídeo.

Contenido no disponible. Consiste en un contenido de la base de datos que ya no está disponible para recomendar a los usuarios, y por lo tanto, que se ha dado de baja.

Contenidos disponible. Consiste en un contenido de la base de datos disponible para ser recomendado a los usuarios.

Valoración de contenido. Consiste en una votación del contenido por parte del usuario, con un rango numérico entre 1 y 5.

Factor de olvido. Representa al valor que se utiliza para reducir la relevancia de los contenidos del historial de usuario cada cierto tiempo.

II. Lógica de los parámetros de configuración

- **ALFA:** Este parámetro indica el peso que se le da a la información disponible y al éxito de las recomendaciones. Con valores mayores a 0.5, la confianza se verá más influenciada por el éxito de las recomendaciones. Por otro lado, con valores menores a 0.5 damos más peso a la información disponible.
- **ALFA1:** Con este parámetro controlamos que en las recomendaciones basadas en el contenido haya más o menos variedad. Con valores mayores a 0.5 significa que el sistema de recomendación prioriza la diversidad de las recomendaciones, arriesgándose a que alguno de los contenidos recomendados sea muy diferente a los “gustos” del usuario. O al contrario, podemos establecer que lo prioritario sea recomendar contenidos lo más similares posibles a las preferencias del usuario.
- **ALFA2:** Parámetro muy similar al anterior, pero con la diferencia que se controla la parte de la selección de los contenidos en el sistema basado en el historial. En concreto, se controla el balanceo de peso entre la relevancia de los contenidos del historial del usuario (su valoración) y la diversidad de la lista de seleccionados (conjunto de muestra). Podemos tener una recomendación donde la muestra seleccionada tenga un cierto grado de diversidad, y así evitar que siempre recomendemos contenidos muy similares, o bien, priorizar la relevancia sin tener en cuenta que la muestra sea muy similar.
- **BETA:** Parámetro que influye en el cálculo de la disponibilidad de información referente al componente Bayesiano, de manera que aumentamos o disminuimos la importancia del número de usuarios disponibles en el sistema. Como mayor sea el valor más necesario será una mayor cantidad de usuarios para que este componente (por lo que hace a la disponibilidad) sea de confianza.
- **DELTA_PROBABILIDAD:** Influye en el cálculo de la distancia Euclidiana (similitud) entre las probabilidades normalizadas calculadas por el componente Bayesiano y los ingredientes de los contenidos. De manera que con un valor mayor a 0.5 implica que el valor de la similitud dependerá más de los ingredientes con probabilidades altas.
- **DELTA_USUARIO:** Exactamente tiene el mismo objetivo que el anterior parámetro, con la diferencia que influye en el cálculo de la similitud entre las preferencias de los usuarios y los ingredientes de los contenidos.
- **DISTANCIA_EUCLIDIANA:** Parámetro que balancea el peso de los componentes del cálculo de la similitud. Por lo tanto, valores mayores a 0.5 implican más peso a la distancia Euclidiana, y menos al cálculo de las coincidencias.

- **GAMMA:** Parámetro que influye en el cálculo de la disponibilidad de información referente al componente Bayesiano, de manera que aumentamos o disminuimos la importancia de la información personal suministrada por el usuario.
 - **NdescriptoresMax:** De igual modo que en el caso anterior, el número de descriptores opcionales que un usuario puede suministrar es muy variable. Con este parámetro se determina a partir de cuántos descriptores se considera que la disponibilidad de información para el componente basado en contenido es máxima.
 - **NpersonalesMax:** Lo mismo que el anterior, pero referente a la disponibilidad de información personal.
 - **Nmuestras:** Indica cuantos contenidos de muestra serán seleccionados como máximo del historial de los usuarios. El hecho de usar un mayor número de muestras, puede contribuir a que la lista de recomendación resultante sea más diversa, ya que hay más posibilidades que los contenidos de muestra sean más diversos.
 - **Nrelaciones:** Indica el tamaño de la lista de los contenidos más similares que habrá para cada contenido de la base de datos (Número de columnas de la matriz de contenidos).
 - **SIMILITUD_ALTA:** Valor a partir del cual se considera que la similitud es alta.
 - **SIMILITUD_MEDIA:** Valor a partir del cual se considera que la similitud es notable. El sentido de este parámetro es tener un valor menos rígido para considerar que dos contenidos son similares (usado en el filtrado de contenidos cuando un usuario ha valorado muy negativamente un contenido)
- SIMILITUD_BAYESIANA:** Valor a partir del cual se considera que la similitud es alta para el modelo Bayesiano. En este caso la similitud es calculada totalmente a partir de la distancia Euclidiana entre las probabilidades y los ingredientes.
- **VALOR_MINIMO:** Valor que se da a un usuario el cual sólo tenemos como información disponible de sus preferencias los ingredientes de contenidos.

A continuación se muestran posibles configuraciones del sistema de recomendación.

CONFIGURACIÓN 1

- confianza **normal** => ALFA = 50

- distancia euclidiana **alta** => DISTANCIA_EUCLIDIANA=90
- grado de diversidad **50 %** (SBC y SBH) => ALFA1=ALFA2=50
- disponibilidadH (BETA= 85 y GAMMA=10)
- peso positivo SIM-PC **alto** => DELTA_PROB=90
- peso positivo SIM-UC **medio-alto** => DELTA_USUARIO=75
- número de muestras **normal** => Nmuestras=3
- número de relaciones **normal** => Nrelaciones=7
- valor mínimo de confianza **normal** => VALOR_MINIMO =40

CONFIGURACIÓN 2

- confianza **normal** => ALFA = 50
- distancia euclidiana **alta** => DISTANCIA_EUCLIDIANA=90
- grado de diversidad **25%** SBC y **50%** SBH) => ALFA1=75 y ALFA2=50
- disponibilidadH (BETA= 85 y GAMMA=10)
- peso positivo SIM-PC **alto** => DELTA_PROB=90
- peso positivo SIM-UC **medio-alto** => DELTA_USUARIO=75
- número de muestras **normal** => Nmuestras=3
- número de relaciones **normal** => Nrelaciones=7
- valor mínimo de confianza **normal** => VALOR_MINIMO =40

CONFIGURACIÓN 3

- confianza **normal** => ALFA = 50
- distancia euclidiana **alta** => DISTANCIA_EUCLIDIANA=90
- grado de diversidad **25%** SBC y **25%** SBH) => ALFA1=75 y ALFA2=75
- disponibilidadH (BETA= 85 y GAMMA=10)
- peso positivo SIM-PC **alto** => DELTA_PROB=90
- peso positivo SIM-UC **medio-alto** => DELTA_USUARIO=75
- número de muestras **normal** => Nmuestras=3
- número de relaciones **normal** => Nrelaciones=7
- valor mínimo de confianza **normal** => VALOR_MINIMO =40

CONFIGURACIÓN 4

- confianza **normal** => ALFA = 50
- distancia euclidiana **alta** => DISTANCIA_EUCLIDIANA=90
- grado de diversidad **50%** SBC y **50%** SBH) => ALFA1=50 y ALFA2=50
- disponibilidadH (BETA= 85 y GAMMA=10)
- peso positivo SIM-PC **alto** => DELTA_PROB=90
- peso positivo SIM-UC **alto** => DELTA_USUARIO=90
- distancia euclidiana **alta** => DISTANCIA_EUCLIDIANA=90
- número de muestras **normal** => Nmuestras=3
- número de relaciones **normal** => Nrelaciones=7
- valor mínimo de confianza **normal** => VALOR_MINIMO =40

CONFIGURACIÓN 5

- confianza **normal** => ALFA = 50

- distancia euclidiana **alta** => DISTANCIA_EUCLIDIANA=90
- grado de diversidad **50% SBC y 50% SBH**) => ALFA1=50 y ALFA2=50
- disponibilidadH (BETA= 85 y GAMMA=10)
- peso positivo SIM-PC **medio**=> DELTA_PROB=50
- peso positivo SIM-UC **medio**=> DELTA_USUARIO=50
- número de muestras **normal** => Nmuestras=3
- número de relaciones **normal** => Nrelaciones=7
- valor mínimo de confianza **normal** => VALOR_MINIMO =40

CONFIGURACIÓN 6

- confianza **normal** => ALFA = 50
- distancia euclidiana **alta** => DISTANCIA_EUCLIDIANA=90
- grado de diversidad **50% SBC y 50% SBH**) => ALFA1=50 y ALFA2=50
- disponibilidadH (BETA= 85 y GAMMA=10)
- peso positivo SIM-PC **alto**=> DELTA_PROB=90
- peso positivo SIM-UC **medio-alto**=> DELTA_USUARIO=75
- número de muestras **alta** => Nmuestras=5
- número de relaciones **normal** => Nrelaciones=7
- valor mínimo de confianza **normal** => VALOR_MINIMO =40

CONFIGURACIÓN 7

- confianza **normal** => ALFA = 50
- distancia euclidiana **alta** => DISTANCIA_EUCLIDIANA=90
- grado de diversidad **50% SBC y 50% SBH**) => ALFA1=50 y ALFA2=50
- disponibilidadH (BETA= 85 y GAMMA=10)
- peso positivo SIM-PC **alto**=> DELTA_PROB=90
- peso positivo SIM-UC **medio-alto**=> DELTA_USUARIO=75
- número de muestras **baja** => Nmuestras=2
- número de relaciones **normal** => Nrelaciones=7
- valor mínimo de confianza **normal** => VALOR_MINIMO =40

CONFIGURACIÓN 8

- confianza **normal** => ALFA = 50
- distancia euclidiana **alta** => DISTANCIA_EUCLIDIANA=90
- grado de diversidad **50% SBC y 50% SBH**) => ALFA1=50 y ALFA2=50
- disponibilidadH (BETA= 85 y GAMMA=10)
- peso positivo SIM-PC **alto**=> DELTA_PROB=90
- peso positivo SIM-UC **medio-alto**=> DELTA_USUARIO=75
- número de muestras **normal** => Nmuestras=3
- número de relaciones **alta** => Nrelaciones=12
- valor mínimo de confianza **normal** => VALOR_MINIMO=40

CONFIGURACIÓN 9

- confianza **normal** => ALFA = 50
- distancia euclidiana **normal** => DISTANCIA_EUCLIDIANA=75
- grado de diversidad **30% SBC y 50% SBH**) => ALFA1=50 y ALFA2=50

- disponibilidadH (BETA= 75 y GAMMA=15)
- peso positivo SIM-PC **alto**=> DELTA_PROB=90
- peso positivo SIM-UC **medio-alto**=> DELTA_USUARIO=75
- número de muestras **normal** => Nmuestras=3
- número de relaciones **normal** => Nrelaciones=7
- valor mínimo de confianza **normal** => VALOR_MINIMO =40

CONFIGURACIÓN 10

- confianza **normal** => ALFA = 50
- distancia euclidiana **normal** => DISTANCIA_EUCLIDIANA=75
- grado de diversidad **30% SBC y 70% SBH** => ALFA1=50 y ALFA2=50
- disponibilidadH (BETA= 75 y GAMMA=15)
- peso positivo SIM-PC **alto**=> DELTA_PROB=90
- peso positivo SIM-UC **medio-alto**=> DELTA_USUARIO=75
- número de muestras **normal** => Nmuestras=3
- número de relaciones **normal** => Nrelaciones=7
- valor mínimo de confianza **normal** => VALOR_MINIMO =50

III. Documentación de la solicitud de patente

RECOMMENDATION SYSTEM AND METHOD FOR MULTIMEDIA

CONTENT

FIELD OF THE INVENTION

The present invention relates to a recommendation system and method for providing a recommendation of multimedia content to a user. Particularly, the object of the present invention is choosing from a multimedia content database a list of titles the user is most likely to enjoy. Another object of the present invention is obtaining a list of users for recommending a new title arriving at the database.

BACKGROUND OF THE INVENTION

In an era of increased availability of multimedia content, recommendation technologies are a necessary tool to help people select what they consume. Nowadays, when a user wishes to watch a movie, he may have to choose from a database formed of hundreds or thousands of titles, as is the case of, for example, subscription TV channels, hotel video services or internet databases. These technologies help people manage multimedia content overload and discover multimedia content they would never have found on their own. An additional advantage of the present invention is saving the users the time otherwise employed in tedious searches.

The challenge in the field of recommendation technologies is matching multimedia content to people's preferences. There are basically three approaches for this task.

Content-based recommendation consists of recommending multimedia content that matches preferences explicitly expressed by the user, usually by means of filling a questionnaire. The key element of this method is the similarity measure that indicates how related is some multimedia content to a certain user. The main disadvantage is that recommendations are usually very similar to each other (overspecialization). In addition,

many times the user does not provide the system with enough information relating to his multimedia content preferences. On the other hand, the advantage with respect to other methods is that recommendations can be provided without using a record of previous user-behavior (user's history).

1. the user and on the success of the previously recommended titles by each approach.

2. Computer program comprising code adapted to perform the method of any of the previous claims when executed on a data-processing system.

3. Recommendation system (1) for multimedia content, comprising:

at least two recommendation components (2, 3, 4) for providing recommendation of content to users according to at least two different approaches;

a combination module (7) for combining the recommendation of multimedia content to users based on confidence levels and for providing a final recommendation to said users;

a confidence level database (8) for obtaining the confidence level of each recommendation of multimedia content to users based on the availability of information related to each user and on the success of the previously recommended titles by each recommendation component (2, 3, 4).

BCN D'INFOGRAFIA, S.L.
Att. Xavier Verdaguer
C/ Josep Pla, 19 - 7º - 5º
08019 - BARCELONA

Madrid, 19 de noviembre de 2007

Ref.: US1585.1

Asunto: Solicitud de patente en Estados Unidos referente a «*RECOMMEN-
DATION SYSTEM AND METHOD FOR MULTIMEDIA CONTENT*» a
nombre de BCN D'Infografía, S.L.

Estimados señores:

Les comunicamos que la oficina de Patentes en Estados Unidos a procedido a
asignar un número de registro a la solicitud de patente de referencia. El número
es el 11/974,561.

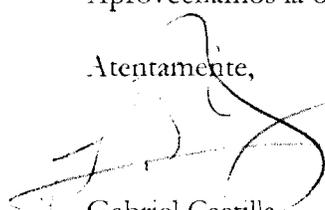
Aprovechamos la presente carta para enviarles la declaración de derechos de los
inventores, así como la autorización – poder para presentar ante la Oficina de
Patentes en Estados Unidos. Les agradecemos nos devuelvan ambos
documentos originales firmados para poderlos enviar y presentarlos con carácter
de urgencia ante la Oficina de Patentes.

Por favor, tengan en cuenta que la fecha límite para recibir estos documentos es
el **15 de diciembre de 2007**. En consecuencia, les agradecemos nos envíen
ambos documentos con la máxima prioridad.

Para cualquier cuestión o consulta relacionada con el presente expediente
pueden ponerse en contacto con Ester San Martín (esanmartin@pons.es).

Aprovechamos la ocasión para enviarle un cordial saludo.

Atentamente,


Gabriel Castilla
Departamento de Patentes
Pons Patentes y Marcas Internacional, S.L.

OFICINA CENTRAL

Gta. de Rubén Darío, 4
28010 Madrid
Tel. 91 700 76 00
902 280 480
Fax 91 308 61 03
902 44 11 33

DELEGACIONES

ALICANTE Tel.: 96 522 77 92 Fax: 96 522 74 55
BARCELONA Tel.: 93 214 23 00 Fax: 93 496 13 14
MADRID Tel.: 91 700 76 00 Fax: 91 308 61 03
SEVILLA Tel.: 95 492 03 53 Fax: 95 492 04 49
ZARAGOZA Tel.: 97 648 25 47 Fax: 97 648 25 62

Oscar Esplá, 24. 03003 Alicante
Consell de Cent, 367. 08009 Barcelona
Gta. Rubén Darío, 4. 28010 Madrid
Balbino Marrón, 3 (Edificio Viapol). 41018 Sevilla
Avda. César Augusto, 16, ppal. dcha. 50004 Zaragoza

inventor has sold and assigned, and by these presents hereby sells and assigns, unto

BCN D' INFOGRAFIA, S.L.

C/ Marina 16-18
Torre Mafre Planta 28
08005 Barcelona, Spain

Name and
address of
assignee

(hereinafter ASSIGNEE) all right, title and interest for the United States, its territories and possessions in and to his invention relating to

Title of
invention

RECOMMENDATION SYSTEM AND METHOD FOR MULTIMEDIA CONTENT

as set forth in his United States Patent Application

executed concurrently herewith

Check one

executed on _____

Serial No. 11/974,561 filed October 15, 2007

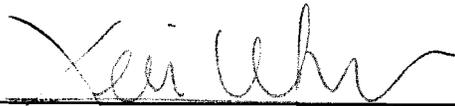
in and to said United States Patent Application including any and all divisions or continuations thereof and in and to any and all Letters Patent of the United States which may issue on any such application or for said invention, including any and all reissues or extensions thereof, to be held and enjoyed by said ASSIGNEE, its successors, legal representatives and assigns to the full end of the term or terms for which any and all such Letters Patent may be granted as fully and entirely as would have been held and enjoyed by the undersigned had this Assignment not been made;

Each of the undersigned hereby authorizes and requests the Commissioner of Patents and Trademarks to issue any and all such Letters Patent to said ASSIGNEE, its successors or assigns in accordance herewith;

Each of the undersigned warrants and covenants that he has the full and unencumbered right to sell and assign the interests herein sold and assigned and that he has not executed and will not execute any document or instrument in conflict herewith;

Each of the undersigned further covenants and agrees that at any time upon request of said ASSIGNEE, its successors, legal representatives or assigns he will communicate to said ASSIGNEE, its successors, legal representatives or assigns all information known to him relating to said invention or patent application and that he will execute and deliver any papers, make all rightful oaths, testify in any legal proceedings and perform all other lawful acts deemed necessary or desirable by said ASSIGNEE, its successors, legal representatives or assigns to perfect title to said invention, to said application including divisions and continuations thereof and to any and all Letters Patent which may be granted therefor or thereon, including reissues or extensions, in said ASSIGNEE, its successors, or assigns or to assist said ASSIGNEE, its successors, legal representatives or assigns in obtaining, reissuing or enforcing Letters Patent of the United States for said invention;

Each of the undersigned hereby grants the firm of COLLARD & ROE, P.C., 1077 Northern Boulevard, Roslyn, New York 11576, U.S.A., the power to insert in this Assignment any further identification which may be necessary or desirable to comply with the rules of the U.S. Patent and Trademark Office for recordation of this Assignment.



Name: **Xavier VERDAGUER**

Date: 29/11/2007

Names and
dates of
inventors



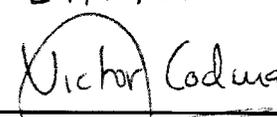
Name: **Luigi CECCARONI**

Date: 29/11/07



Name: **Jordi CASAS**

Date: 29/11/2007



Name: **Victor CODINA**

Date: 29/11/07

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

RECOMMENDATION SYSTEM AND METHOD FOR MULTIMEDIA CONTENT

the specification of which (check only one item below):

- is attached hereto.
- was filed as United States application
 Serial No. 11/974,561
 on October 15, 2007
 and was amended
 on _____ (if applicable).
- was filed as PCT international application
 Number _____
 on _____
 and was amended under PCT Article 19
 on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to **patentability** of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 or 365 of any foreign application(s) for patent or inventor's certificate or of any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed:

PRIOR FOREIGN/PCT APPLICATION(S) AND ANY PRIORITY CLAIMS UNDER 35 U.S.C. 119 OR 365:

COUNTRY (if PCT, indicate "PCT")	APPLICATION NUMBER	DATE OF FILING (day, month, year)	PRIORITY CLAIMED
			<input type="checkbox"/> YES <input type="checkbox"/> NO
			<input type="checkbox"/> YES <input type="checkbox"/> NO
			<input type="checkbox"/> YES <input type="checkbox"/> NO
			<input type="checkbox"/> YES <input type="checkbox"/> NO

below.

(Application Number)

(Filing Date)

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) or PCT international application(s) designating the United States of America that is/are listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in that/those prior application(s) in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to patentability as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application(s) and the national or PCT international filing date of this application:

PRIOR U.S. APPLICATIONS OR PCT INTERNATIONAL APPLICATIONS DESIGNATING THE U.S. FOR BENEFIT UNDER 35 U.S.C. 120:

U.S. APPLICATIONS		STATUS (Check One)		
U.S. APPLICATION NUMBER	U.S. FILING DATE	PATENTED	PENDING	ABANDONED
PCT APPLICATIONS DESIGNATING THE U.S.				
PCT APPLICATION NO.	PCT FILING DATE	U.S. SERIAL NUMBERS ASSIGNED (if any)		

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (List name and registration numbers):
KURT KELMAN, Registration No. 18,628
ALLISON C. COLLARD, Registration No. 22,532; **WILLIAM C. COLLARD**, Registration No. 38,411
EDWARD R. FREEDMAN, Registration No. 26,048; **FREDERICK J. DORCHAK**, Registration No. 29,298
ELIZABETH COLLARD RICHTER, Registration No. 35,103 **EDWARD J. CALLAGHAN**, Registration No. 46,594

Send Correspondence to: **COLLARD & ROE, P.C.** Customer No. **25889** Direct Telephone Calls to:
 1077 Northern Boulevard (name and telephone number)
 Roslyn, New York 11576 (516) 365-9802

2 0 1	FULL NAME OF INVENTOR	FAMILY NAME	FIRST GIVEN NAME	SECOND GIVEN NAME
	RESIDENCE & CITIZENSHIP	CITY	STATE OR FOREIGN COUNTRY	COUNTRY OF CITIZENSHIP
	POST OFFICE ADDRESS	POST OFFICE ADDRESS	CITY	STATE & ZIP CODE/COUNTRY
2 0 2	FULL NAME OF INVENTOR	FAMILY NAME	FIRST GIVEN NAME	SECOND GIVEN NAME
RESIDENCE & CITIZENSHIP	CITY	STATE OR FOREIGN COUNTRY	COUNTRY OF CITIZENSHIP	
POST OFFICE ADDRESS	POST OFFICE ADDRESS	CITY	STATE & ZIP CODE/COUNTRY	

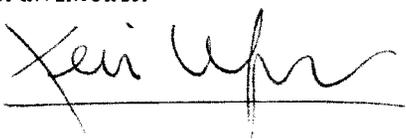
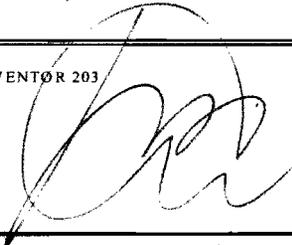
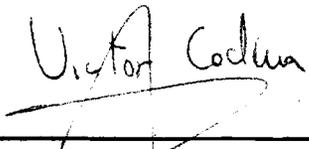
2	FULL NAME OF INVENTOR	FAMILY NAME CASAS	FIRST GIVEN NAME JORDI	SECOND GIVEN NAME
0	RESIDENCE & CITIZENSHIP	CITY SABADELL, BARCELONA	STATE OR FOREIGN COUNTRY SPAIN	COUNTRY OF CITIZENSHIP SPAIN
3	POST OFFICE ADDRESS	POST OFFICE ADDRESS C/ WALETERBENJAMIN 24, 2º - 4ª	CITY 08206 SABADELL, BARCELONA	STATE & ZIP CODE/COUNTRY SPAIN

2	FULL NAME OF INVENTOR	FAMILY NAME CODINA	FIRST GIVEN NAME VICTOR	SECOND GIVEN NAME
0	RESIDENCE & CITIZENSHIP	CITY LLORET DE MAR, GIRONA	STATE OR FOREIGN COUNTRY SPAIN	COUNTRY OF CITIZENSHIP SPAIN
4	POST OFFICE ADDRESS	POST OFFICE ADDRESS C/ PERE CODINA I MONT 7	CITY 17310 LLORET DE MAR, GIRONA	STATE & ZIP CODE/COUNTRY SPAIN

2	FULL NAME OF INVENTOR	FAMILY NAME	FIRST GIVEN NAME	SECOND GIVEN NAME
0	RESIDENCE & CITIZENSHIP	CITY	STATE OR FOREIGN COUNTRY	COUNTRY OF CITIZENSHIP
5	POST OFFICE ADDRESS	POST OFFICE ADDRESS	CITY	STATE & ZIP CODE/COUNTRY

2	FULL NAME OF INVENTOR	FAMILY NAME	FIRST GIVEN NAME	SECOND GIVEN NAME
0	RESIDENCE & CITIZENSHIP	CITY	STATE OR FOREIGN COUNTRY	COUNTRY OF CITIZENSHIP
6	POST OFFICE ADDRESS	POST OFFICE ADDRESS	CITY	STATE & ZIP CODE/COUNTRY

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

SIGNATURE OF INVENTOR 201 	SIGNATURE OF INVENTOR 202 
DATE 29/11/2007	DATE 29/11/2007
SIGNATURE OF INVENTOR 203 	SIGNATURE OF INVENTOR 204 
DATE 29/11/2007	DATE 29/11/2007