



UNIVERSIDAD DE HUELVA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DIPLOMATURA DE INFORMÁTICA

Proyecto Fin de Carrera:

Especificación de un
Sistema de Recomendación
basado en Etiquetado Social

Alumno: D. Diego Montiel Araque.

Director: D. José Carpio Cañada.

Huelva a 28 de Junio de 2012

Dedicatoria.

A mí esposa Julia por haber estado a mi lado y darme ánimos en los momentos difíciles.

Agradecimientos.

Mi agradecimiento, a mis padres por su incansable esfuerzo y apoyo durante los años de carrera.

Agradecimiento también a, D. José Carpio Cañada por el constante ánimo y motivación que me ha brindado, así como la facilidad con la que me ha permitido realizar el proyecto.

A todos, muchas gracias.

Índice General

1.- Introducción.	1
1.1.- Motivaciones	2
1.2.- Metodología	4
1.3.- Objetivo.....	6
2.- Desarrollo del Sistema	7
2.1.- Herramienta.....	8
2.2.- Modelo de los Casos de Uso	9
2.3.- Diagramas de Secuencia	24
2.4.- Diagramas de Clase.....	36
3.- Conclusiones	39
4.- Bibliografía.....	41
5.- Anexos.....	42
5.1.- Código Java.....	42
5.2.- Código C++.....	47

1.- Introducción

Hoy en día disponemos de tal cantidad de información, que la habilidad para identificar cual sería la más útil, para cada una de nuestras necesidades, se ve mermada, convirtiéndose en un caos de información, sin que el usuario pueda encontrar a veces lo que realmente desea. No disponemos del tiempo suficiente para seleccionar lo más interesante de lo que esta disponible en Internet.

Los sistemas de recomendación (S.R.) surgen para solucionar este problema, pasando a ser una herramienta indispensable en muchas tareas de la vida cotidiana, ante el problema de filtrar contenido interminable.

De un modo resumido, un S.R. es un sistema que utiliza las opiniones y comportamientos de los usuarios de una comunidad para ayudarles a encontrar contenidos de su gusto, entre un conjunto sobrecargado de elecciones posibles.

Básicamente, los sistemas de recomendación reciben información del usuario acerca de productos o servicios en los que el usuario se encuentra interesado y le recomienda aquéllos que estén más cercanos a sus necesidades.

El objetivo de un buen sistema de recomendación es hacer llegar a cada usuario la información que necesita, evitando la que no le interesa. *“Sí le gusta este libro, tal vez le interesen también los siguientes”*. Las personas simplemente valoran lo que han visto y el sistema le recomienda otras fuentes afines a sus gustos.

En la actualidad, existe una gran cantidad de sitios especializados en Internet, ofreciendo millones de productos y servicios para su consumo, a continuación, citaremos algunos: Amazon (Página de compra por internet, incluye recomendaciones de productos), Netflix (Servicio de alquiler de DVD con recomendación según gustos), Reddit (Sistema de recomendación de noticias), StumbleUpon (Recomendación páginas web) Youtube o Last.fm (Sistema de música).

1.2.- Motivaciones

Tras cinco años de carrera, comencé a trabajar como informático, al principio como profesor de ofimática y a medida que la necesidad de servicios de la empresa crecía, mis atribuciones cambiaban, llegando a;

- Gestionar incidencias y problemas en aplicaciones corporativas y arquitectura de red, siendo la segunda línea de soporte en la atención a personas usuarios de los sistemas TIC (Tecnologías de la Información y la Comunicación) de la organización.
- Administrar las Aplicaciones y Servicios TIC de la Empresa.
- Aplicar y supervisar las políticas de seguridad definidas por la organización dentro del marco legislativo proporcionado por la Ley Orgánica de Protección de Datos de Carácter Personal.
- Comunicar de forma efectiva, conocimientos, procedimientos, resultados e ideas relacionadas con las TIC, sirviendo como asesoramiento a la organización.

Durante estos años, he podido observar que para las empresas, sobre todo en la pequeña y mediana, a la hora de realizar cambios funcionales o de mejora en sus sistemas de información, no se especifica. ¿Por qué? Esto normalmente ocurre por que los cambios son urgentes, hay que entregar el informe al Director hoy, y el especificar requiere tiempo, del cual no se dispone. Nos encontramos con la disyuntiva de especificar o programar.

Sí esta practica, no es muy habitual y el responsable no cambia, puede ser que no tenga una gran repercusión, sin embargo en un sistema de información donde se están realizando cambios funcionales o de mejora continua, sin recoger las especificaciones, con toda seguridad ese sistema con el paso del tiempo será inconsistente.

Después de algunos intentos fallidos por realizar mí proyecto fin de carrera, decido no retrasarlo más y me pongo en contacto con el Departamento para que me asigne un tutor.

En la conversación previa mantenida con mí tutor, hacemos hincapié en lo importante que es para un proyecto de software contar con sus requerimientos, sobre todo a la hora de aplicar, gestión de la mejora continua. Resultará más sencillo realizar la reforma de una vivienda si contamos con los planos, ya que conoceremos como esta distribuida y sus limitaciones.

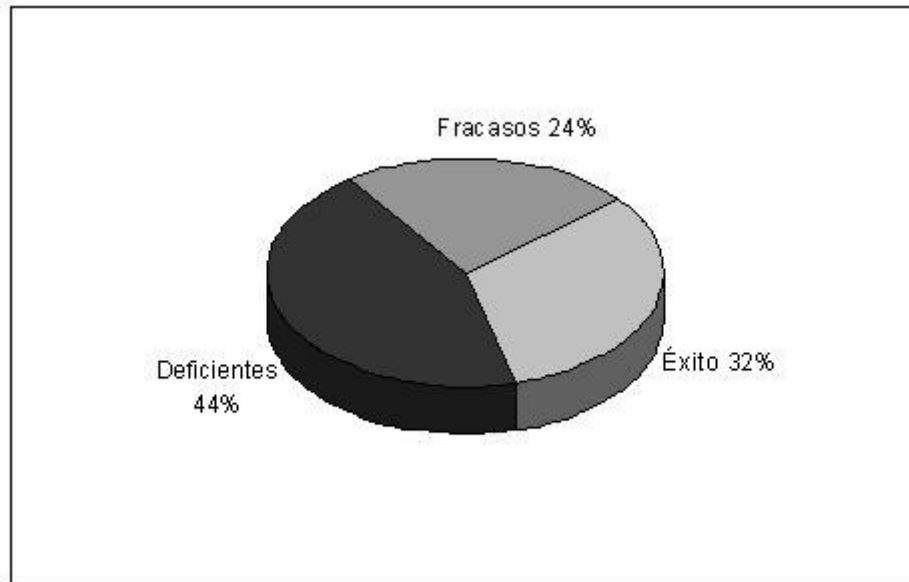
El motivo principal para elegir este tema, se justifica por la cantidad de proyectos de software, que no llegan a cumplir sus objetivos. Los sistemas de gestión existentes en el mercado actual, supuestamente no contemplan todas las necesidades de las empresas, por lo que estos sistemas permiten su personalización o adquisición de módulos específicos. Tal “personalización”, la mayoría de las veces, termina retrasando el proyecto en meses, o incluso en años.

El desarrollo de proyectos para el remplazo de plataformas y tecnologías obsoletas o modificaciones en los programas que forman el sistema, conllevan planificaciones de ejecución muy ajustadas y esto provoca que se omitan pasos importantes en la ingeniería del software, entre estos, la definición de las especificaciones.

Los estudios realizados por la consultora “Standish Group” (*The Chaos Report*) muestran que más del 44 % de los proyectos de software fracasan por no realizar un estudio previo de requisitos (desviación en plazo y coste).

[<http://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.html>]

En la grafica se detalla el resultado de dicho estudio:



Otras causas, como la falta de involucración (compromiso o interés) de la Dirección o las personas empleadas, expectativas poco realistas, comunicación deficiente y especificaciones cambiantes o incompletas, también son motivos de fracasos.

Por eso, como culminación de mi carrera, decidí embarcarme en este interesante proyecto, en el cual abarco la comprensión de un problema en términos de sus procesos y conceptos, el diseño de una solución utilizando objetos, y para aplicarlo a un producto concreto y efectivo para la vida real.

1.2.- Metodología

En todo proyecto de un sistemas informático, es de suma importancia el seguir alguna metodología de trabajo que permita a las personas implicadas el tener una disciplina que haga que todas las etapas del desarrollo del sistema, desde el inicio de la toma de requerimientos hasta las pruebas finales del sistema, sean no solo más coherentes sino también más formales.

“La Ingeniería de Requerimientos (IR) cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.” [Ingeniería del Software – Lizka Johany Herrera J.]

La meta de la IR es entregar una especificación de requisitos de software correcta y completa.

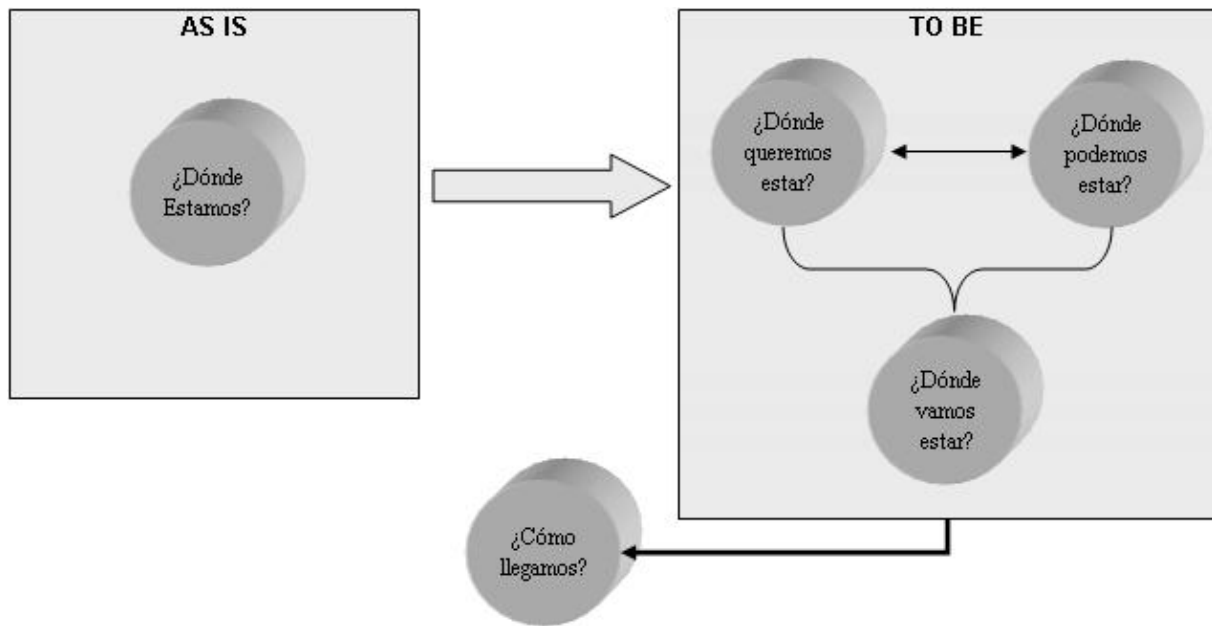
El modelo de IR establecido en este proyecto, es la metodología que permite la generación de modelos de los procesos de negocio. Se crean los modelos de los procesos de negocio actuales (As Is) y se establecen la generación de modelos mejorados (To Be), siendo la base de la metodología.

Se compone de una serie de fases que permiten la descripción, análisis, evaluación y modelado. Estas actividades ayudan a reconocer la importancia que tiene, para el desarrollo de un proyecto de software, realizar una especificación y administración adecuada de los requisitos de los clientes o usuarios.

- Descripción de los procesos de negocio: mediante la identificación de los actores (casos de uso), la información que fluye (diagramas de secuencia) y la información estática (diagramas de clase), especificamos lo que tenemos.
- Análisis y evaluación: se detectan los procesos susceptibles de ser automatizados y los puntos críticos. Se valoran cualitativa y cuantitativamente.
- Diseño: en base al punto anterior, decidimos que elementos del sistema se van a modificar, que elementos hardware y software se introducirán si fuese necesario.
- Implementación: se encarga de la organización y planificación de las acciones necesarias para llegar a la situación futura (To-Be)

De forma esquemática, en esta metodología, lo primero que hay que preguntarse es ¿Dónde estamos ahora? (As Is), las siguientes preguntas son ¿Dónde queremos estar? y ¿Dónde podemos estar?, ya que puede ocurrir que todo lo deseado no sea factible. A partir de aquí decidiremos donde vamos a estar, resultado de comparar la situación deseada y la situación real. Por ultimo, una vez especificado el entorno actual y futuro, solo queda acordar como lo voy hacer.

Su representación gráfica sería:



La especificación de requisitos describe el comportamiento esperado del software una vez desarrollado. Gran parte del éxito de un proyecto de software radicará en la identificación de las necesidades del negocio, así como la interacción con los usuarios para la recolección, clasificación, identificación, priorización y especificación de los requisitos del software.

Lo que nos tiene que motivar para realizar un documento de especificaciones, es el hecho de crear algo que sea práctico y útil, con el suficiente detalle como para establecer un entorno de trabajo donde vayamos definiendo y cerrando casos de uso o escenarios, y el cliente vaya revisándolos y enviando tickets si encuentra algún tipo de bug (defecto) o error en el comportamiento del sistema.

Según la autora “Lizka Johany Herrera”, los principales beneficios que se obtienen de la Ingeniería de Requerimientos son:

- Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la IR consiste en una serie de pasos organizados y bien definidos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Disminuye los costos y retrasos del proyecto: Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la especificación de requisitos.
- Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).

- *Mejora la comunicación entre equipos: La especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.*
- *Evita rechazos de usuarios finales: La ingeniería de requerimientos obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.*

El documento de especificación se utiliza como guía para la realización, así como para el testeado del sistema. El sistema entregado cumplirá con las especificaciones de los casos de uso o escenarios definidos y será el cliente quien ejecute y compruebe que el sistema se comporta como se espera. Otra característica importante del documento, es que antes de iniciar los trabajos, este tiene que ser aprobado por el cliente, lo cual nos facilita una defensa valiosísima para evitar los temidos “poyaques”.

1.3.- Objetivo

En un proyecto comercial, para establecer el fin del mismo, planificaríamos una serie de actuaciones como: entrevista con los clientes, definición equipo de trabajo, encuestas con posibles usuarios, observaciones en vivo de la actividad diaria en la empresa. En nuestro caso, estamos ante un proyecto de ámbito académico, por lo que el propósito es conocido desde la concepción del mismo.

En este proyecto, se pretende establecer las especificaciones para un Sistema de Recomendación basado en Etiquetado Social, en el que a partir de la obtención y recopilación de datos procedentes de diversas fuentes web (redes sociales), por medio de robots o autómatas, las preferencias del usuario y sus datos personales, se le ofrezcan las propuestas que se ajusten más a sus preferencias.

“Las redes sociales se han convertido en bases de datos en la que se recogen informaciones de carácter personal y documentos sobre las actividades de la vida real de las personas que hacen uso de ellas.” [Privacidad y Seguridad Redes Sociales – Daniel Ortega 2011]

Los sitios web de redes sociales han contribuido positivamente en las comunicaciones entre las personas y han mejorado el uso de la versión anterior de Internet, mucho más estática, facilitando a la gente la posibilidad de expresar opiniones a través de plataformas digitales con normalidad y sobre cualquier tema. Por ejemplo, la función “me gusta” de Facebook, YouTube, blogs, Twitter, y otros. “Me gusta” permite a los usuarios de Internet indicar de forma pública o privada, que elementos son interesantes y posteriormente se puede utilizar esta información para, por ejemplo, realizar recomendaciones.

Su impacto se ha debido a su gratuidad y popularidad entre los adolescentes, principalmente, como un lugar donde hablar con los amigos y subir apuntes, fotos o videos. En definitiva, por la enorme cantidad de datos compartidos en estos sitios, hace que sean un lugar idóneo para obtener información y los sistemas de recomendación necesitan de muchos elementos etiquetados para hacer buenas recomendaciones, cuanta más información, más precisa será la recomendación.

Para su realización, nos apoyaremos en UML (Unified Modeling Language) lenguaje que permite modelar y en el programa Rational Rose para construir y documentar los elementos que forman un sistema software orientado a objetos. Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa: Booch, OMT (Object-Modeling Language) y OOSE (Object-oriented Software Engineering).

2.- Desarrollo del Sistema

Una vez presentado el proyecto, a grandes rasgos con sus propósitos y objetivos, y realizada la introducción a los sistemas de recomendación en general, así como de las redes sociales en Internet, llega el momento de pasar a detallar el proyecto que se ha realizado.

“La idea de utilizar los casos de uso para describir los requisitos funcionales fue introducida en 1986 por Ivar Jacobson [Jacobson92], uno de los contribuidores principales al UML y Procesos Unificados (UP), siendo sus principales virtudes la simplicidad y utilidad.”

En este trabajo, se exponen un conjunto de acciones planificadas de modo sistemático para lograr capturar los objetivos y requisitos del sistema, mediante la escritura de historias del uso del sistema.

Estas historias, cuya redacción sencilla y entendible componen los casos de uso, forman parte del análisis de requisitos, también conocidos como requerimientos, en el desarrollo de sistemas.

Conforme a la definición que aparece en el glosario estándar de términos en ingeniería del software del Instituto de Ingenieros Eléctricos y Electrónicos Std. 610.12-1990, los requisitos son *“(1) una condición o necesidad para resolver un problema o alcanzar un objetivo. (2) Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. (3) Una representación documentada de una condición o capacidad como (1) ó en (2).”*

Existen dos tipos bien diferenciados:

- Requisitos Funcionales: Aquellos que se refieren expresamente al funcionamiento del sistema.
- Requisitos No Funcionales: Son todos aquellos requerimientos no referidos al estricto funcionamiento del sistema, sino a otros factores externos.

En los dos epígrafes siguientes, definiremos estos requisitos (tanto funcionales como no funcionales), para el proyecto sobre el que versa la presente memoria.

Requisitos Funcionales

Como hemos dicho, los requisitos funcionales de un sistema software son aquellos que describen las funcionalidades que dicho sistema debe proporcionar a los usuarios del mismo, para cumplir su objetivo.

A grandes rasgos, las propiedades que el usuario espera de un Sistema de Recomendación como el nuestro, son las siguientes:

- Puntuar aquellos sitios web que ha visitado y desea evaluar.
- Recibir recomendaciones, que incluya objetos o ítems aún no conocidos por el usuario.
- Editar el perfil del usuario. Esta funcionalidad supone:
 - Poder modificar las puntuaciones ya realizadas.
 - Consultar los datos personales y preferencias del usuario.
 - Modificar los datos personales y las preferencias del usuario.
- Disponer de información clara acerca de todos los elementos (*items*) etiquetados existentes en el sistema, así como las características de estos.

Requisitos No Funcionales

Los requisitos no funcionales son las restricciones impuestas a los requisitos funcionales del sistema. Pese a que en principio puedan no parecer demasiado relevantes, son tan importantes como los requisitos funcionales y, en muchos casos, pueden incluso llegar a ser críticos para la aceptación del sistema.

Normalmente, estos requisitos especifican propiedades del sistema software en sí (como la velocidad o el rendimiento) y de la interfaz de usuario, además de todas las restricciones impuestas por la organización (por ejemplo: plazos, legalidad vigente o política empresarial).

Obviamente, al no estar ante un proyecto de tipo empresarial o comercial, no hay necesidad de someterse a restricciones por parte de una organización. Así, los requisitos no funcionales deben obtenerse y analizarse a partir de las necesidades hardware y software de los equipos informáticos, para dar al usuario la funcionalidad requerida de forma eficiente, y de la interfaz gráfica entre el usuario y la aplicación.

2.1.- Herramienta

Una obligación de la gran mayoría de empresas dentro de su Plan Informático estratégico, es que los sistemas de información se formalicen con un lenguaje estándar y unificado. Es decir, se exige que las partes que comprende el sistema de información, se visualice, especifique y documente con lenguaje común.

Se necesitaba un lenguaje de modelado que fuese gráfico, independiente del lenguaje de programación y de las características de los proyectos, a fin de modelar de un modo estándar los procesos de negocios y funciones del sistema.

¿Qué es UML?

“El Lenguaje Unificado de Modelado (UML) es un lenguaje para especificar, visualizar, construir, y documentar los artefactos de los sistemas software, así como para el modelado del negocio y otros sistemas no software” [Larman, 2003]

Estas son algunas de las razones por la cual es necesario adoptar UML como lenguaje de modelado, el cual cuenta con una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos.

- UML unificó la notación de Booch, Rumbaugh (OMT) y Jacobson para la OMG (Object Management Group), desarrollando una propuesta estándar, el cual permite unicidad y estandarización a la hora de hacer las notaciones, en cualquier parte del mundo tiene la misma interpretación
- Es independiente de la herramienta de desarrollo utilizada y de las características del proyecto tal como su funcionalidad y complejidad, entre otras.
- Generación de documentación y código automáticamente. De gran trascendencia, ya que mejora la comunicación entre el grupo de trabajo, la realización de modificaciones y ahorra tiempo en el desarrollo.

El Lenguaje de Modelado Unificado define tres elementos básicos, los bloques de construcción, las reglas y algunos mecanismos comunes que, relacionados entre sí, le dan al lenguaje su carácter de completo y preciso.

- Bloques de construcción: se compone de los elementos (estructurales, comportamiento, agrupación y anotación), las relaciones (une los elementos entre sí) y los diagramas (agrupaciones de elementos)
- Las reglas: establecen las pautas a la hora de realizar asociaciones entre objetos para poder obtener modelos bien formados, reglas semánticas.
- Mecanismos Comunes: elementos secundarios que proporcionan más detalle a los anteriores (explicaciones textuales, divisiones comunes o valores etiquetados)

En nuestro caso, nos centraremos en especificar:

1. Los Casos de Uso. La acciones.
2. Los Diagramas de Secuencia. Información que fluye.
3. El Modelamiento de Clases. Información estática.

2.2.- Modelo de los Casos de Uso

Los casos de uso describen las acciones del sistema, que producen un resultado determinado y que es de interés para alguien o algo. Cuando decimos “alguien o algo” hacemos referencia a que los sistemas son usados no sólo por personas, sino también por otros sistemas de hardware y software.

Un caso de uso representa una funcionalidad concreta dada por el sistema, lo que sucede y quien interviene. También se puede definir como la representación de una situación o proceso de interacción de un usuario con la aplicación o sistema.

Los casos de uso son tareas con significado, coherentes y con cierta independencia, que los actores realizan de forma cotidiana al utilizar el sistema. En un caso de uso pueden participar uno o varios actores.

En resumen, los casos de uso explican cómo se realiza una tarea de forma precisa, y constan de los siguientes elementos:

Nombre único e unívoco del caso de uso.

Descripción.

Pre-condiciones.

Poscondiciones.

Actores participantes.

Un actor modela una entidad que interacciona con el sistema, es decir, es un elemento del sistema. Al igual que ocurre con un caso de uso, un actor debe tener un nombre único y adecuado, que puede ir acompañado de una descripción del mismo.

Anteriormente hemos definido las propiedades que los usuarios esperarían de nuestro sistema, pasaremos a caracterizar de manera más específica y formal cómo nuestro sistema responderá a dichas funcionalidades:

- 1.- Adquirir información de las redes sociales.
El sistema debe actualizar la información de recomendación cada cierto tiempo, incorporando las nuevas peticiones que los usuarios hayan hecho en la red.
- 2.- Enviar recomendaciones nuevas para el usuario.
El sistema debe ofrecer al usuario una serie de direcciones que aún no ha visitado y evaluado y que se estima sean de su agrado.
- 3.- Preproceso de la Recomendación
 - a. Actualizar las recomendaciones del usuario. El sistema debe ser capaz de ofrecer nuevas recomendaciones a los usuarios, teniendo en cuenta la nueva información contenida en el modelo actualizado.
4. Configuración del Sistema para un usuario
 - a. Darse de alta en el sistema. El sistema debe proporcionar un mecanismo para que el usuario pueda registrarse en el sistema, de cara a ofrecerle un servicio personalizado.
 - b. Puntuar. El sistema ha de permitir al usuario evaluar numéricamente, ya sean nuevas para él, o bien ya evaluadas cuya puntuación desee modificar.
 - c. Visualizar datos personales. El sistema debe proporcionar la posibilidad de visualizar los datos personales del usuario.

- d. Modificar datos personales. El sistema debe brindar al usuario la capacidad de modificar sus datos personales cuando este lo desee.
 - e. Visualizar perfil del usuario. El sistema debe permitir al usuario visualizar la información de su perfil, incluyendo sus preferencias y la lista de puntuaciones realizadas previamente por él.
 - f. Modificar perfil del usuario. El sistema ha de proporcionar al usuario la capacidad de modificar los datos y preferencias que figuran en su perfil.
5. Generación de peticiones de información. Si se solicita una recomendación de un ítem, que aún no ha sido almacenado, se genera una petición para obtener datos sobre el ítem en las diferentes fuentes de información social.

En la ilustración 1 se muestran los casos de uso identificados a partir de las características que ofrece un sistema de recomendación.

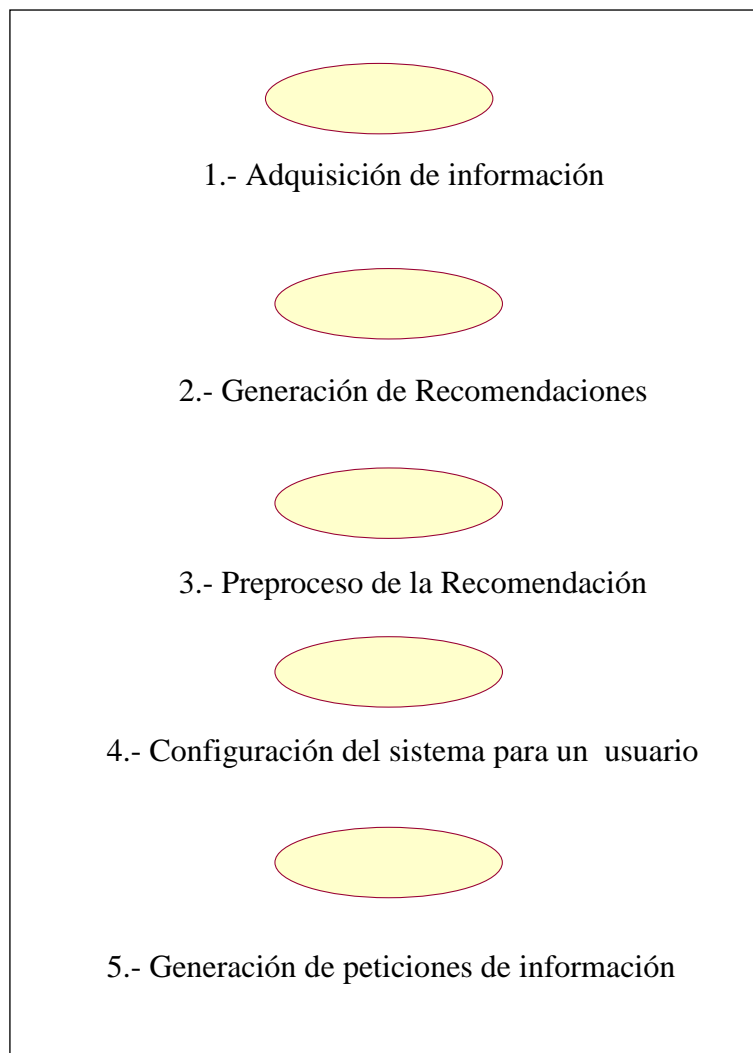


Ilustración 1: Diagrama General del Sistema

En el caso de nuestro sistema nos encontramos ante varios casos de uso que deben ser descritos con mayor profundidad.

A continuación, describimos detalladamente cada uno de los casos de uso mostrados en las figuras anteriores, y exponemos los diagramas de aquellos que requieren de mayor detalle en su descripción y, acto seguido, se describirán los principales flujos de eventos de dichos casos de uso.

Caso de Uso 1.- Adquisición de Información

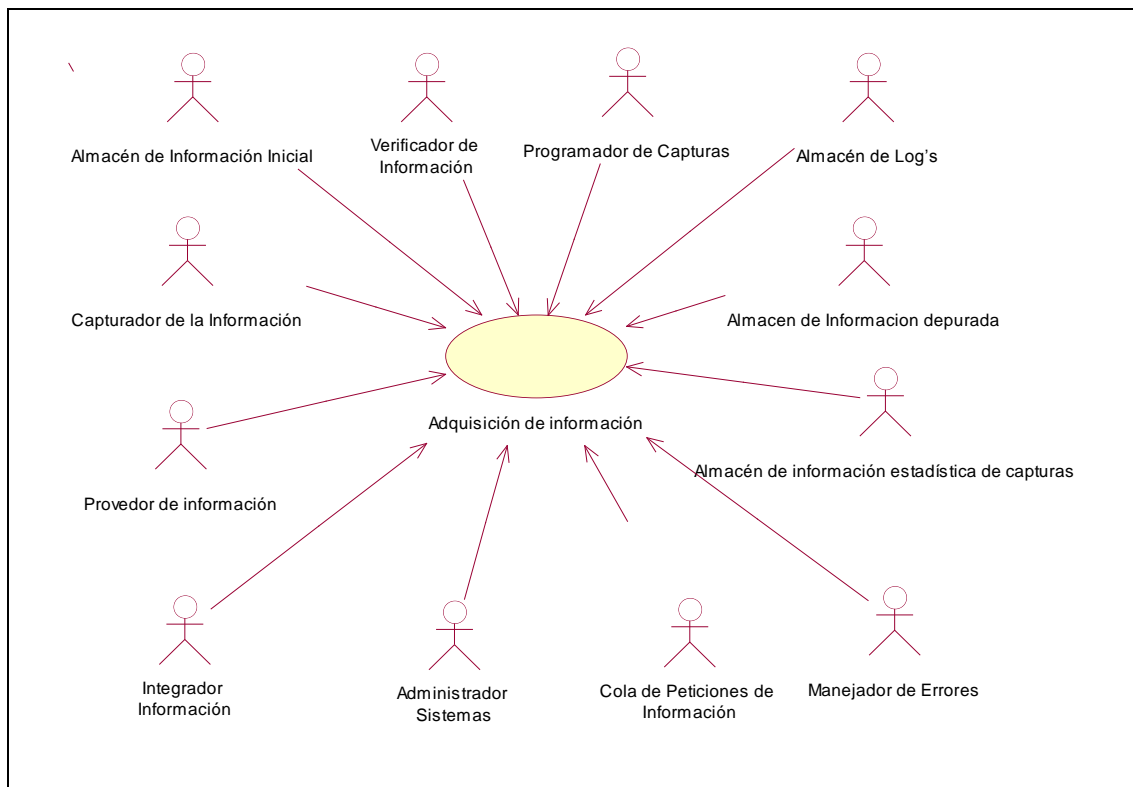


Ilustración 1: Caso de uso 1. Adquisición de Información

Caso de Uso 1.- Adquisición de Información	
Descripción:	<p>En este caso de uso, un capturador de contenido obtiene la información de los recursos etiquetados necesarios para realizar las recomendaciones. La información se obtiene de los diferentes proveedores de información previamente definidos, en nuestro caso, nos centraremos en los proveedores de redes sociales del tipo tuenti o facebook.</p> <p>Las redes sociales son estructuras sociales compuestas de grupos de personas, las cuales están conectadas por uno o varios tipos de relaciones, tales como amistad, parentesco, intereses comunes o que comparten conocimientos.</p> <p>La solicitud de información se hace de forma regular, una cola de peticiones indicará el orden en que deben realizarse y en el caso de que las fuentes de datos estén temporalmente inaccesibles, la petición permanecerá en la cola hasta que sea atendida.</p> <p>La obtención y recopilación de datos procedentes de diversas fuentes web, puede implicar la transformación y validación de datos en el caso de que los modelos de fuentes de datos no concuerden con el modelo de datos deseado.</p> <p>El verificador de información coteja que la información obtenida es correcta. Provee de un mecanismo para eliminar datos equivocados o redundantes, si fuese necesario.</p>

	<p>El resultado de la captura de contenidos puede estar disponible en distintos formatos, como norma general, los datos se almacenarán en una base de datos o se enviará a un adaptador de contenidos. El integrador de información analiza el formato de la información obtenida y la transforma, si es necesario, al formato establecido.</p> <p>La información adquirida se guardará en un almacén inicial, tras el proceso de validación y transformación (este puede tardar varios días) esta información se moverá al almacén de información depurada.</p>
Precondiciones:	<p>La cola de peticiones tiene que estar inicializada.</p> <p>El formato de la información y el método de adquisición deben estar definidos previamente.</p> <p>El programador de capturas debe estar configurado.</p> <p>Los almacenes de información tienen que estar disponibles.</p> <p>Los proveedores de información deben estar disponibles.</p> <p>Información estadística del verificador de información.</p>
Poscondiciones:	<p>El almacén inicial contiene datos extraídos de diferentes fuentes, sin duplicados, con datos conectados, url, usuarios, etc... útil para realizar recomendaciones.</p> <p>Los ítems recopilados han sido etiquetados.</p> <p>El verificador contendrá la información para realizar estadísticas sobre las capturas.</p> <p>Los errores conocidos han sido notificados.</p> <p>La actividad ha sido almacenada en el registro de logs.</p> <p>El manejador de errores ha informado al administrador del sistema.</p>

Actor	Descripción
Proveedor de Información	<p>Entidad que ofrece servicios tecnológicos, en nuestro caso, nos centraremos en los proveedores de redes sociales del tipo tuenti o facebook.</p> <p>Es un sitio web cuya característica fundamental es la de servir de Puerta de entrada (única) para ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios relacionados a un mismo tema. Incluye: enlaces, buscadores, foros, documentos, aplicaciones, compra electrónica, etc.</p>
Capturador de Información	Entidad que captura la información solicitada a los proveedores.
Almacén de Información Inicial	<p>Espacio físico donde se almacena y proporciona la información obtenida por el capturador en las redes sociales.</p> <p>Después de la verificación, la información pasará al almacén de información depurada.</p>

Integrador Información	Entidad que analiza el formato de la información obtenida y la convierte, si es necesario, al formato establecido.
Administrador Sistema	Hace referencia al responsable de la aplicación. Es el que se encarga de asegurar el correcto funcionamiento del sistema y su mejora.
Cola de Peticiones de Información	Registro de las llegadas de información solicitada a los proveedores y que aún no ha sido atendida.
Verificación de Información	Entidad que valida que la información es correcta, es decir, que la url obtenida existe.
Programador de Capturas	Entidad donde se definirán cuando se solicitará información a los proveedores.
Almacén de Log's	Espacio físico donde se graban todos los eventos o sucesos que se realizan en el sistema. Proporcionando la información necesaria para monitorizar el sistema. Estos, pueden implementarse con ficheros de texto plano que genera el servidor web, y en los que se registra cada una de las peticiones de páginas realizadas por los clientes al servidor.
Manejador de Errores	Servicio que informa de los errores al Administrador.
Almacén de Información Depurada	Espacio físico donde se almacena y proporciona los datos íntegros para realizar recomendaciones.
Almacén de información estadística de capturas	Espacio físico donde se almacena y proporciona los datos (número de capturas realizadas, su origen, link, tags..) para analizar y mejorar el capturador de información.

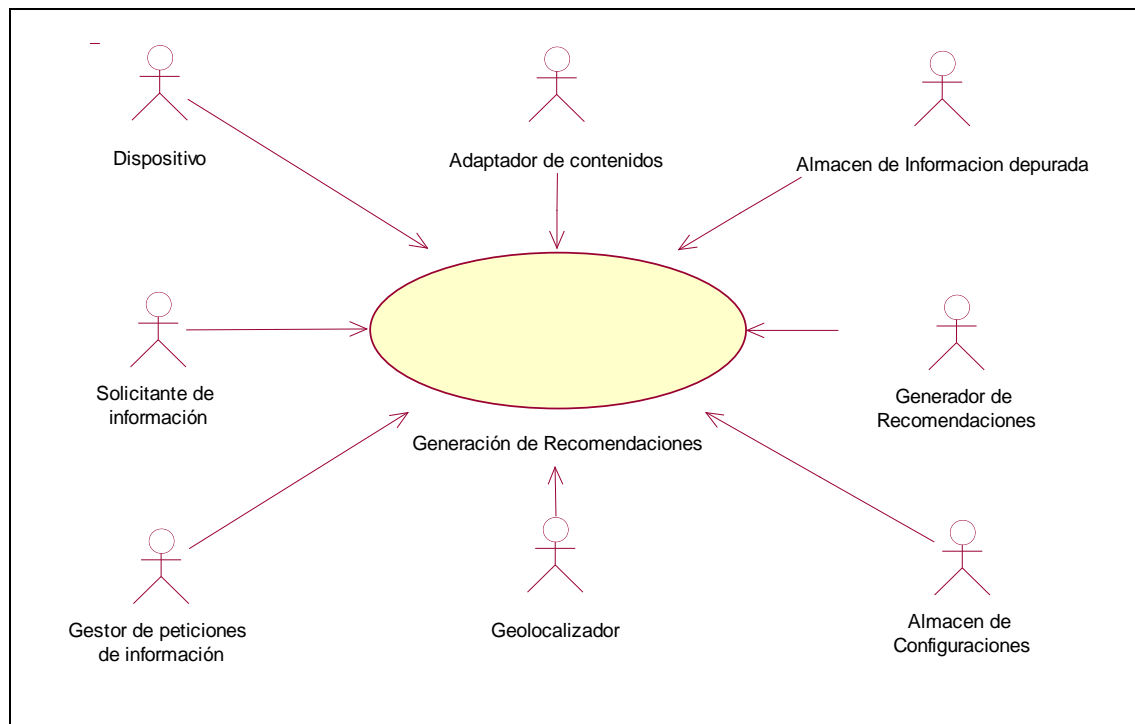
Caso de Uso 2.- Generación de Recomendaciones

Ilustración 2: Caso de uso 2. Generación de Recomendaciones

Caso de Uso 2.- Generación de Recomendaciones	
Descripción:	<p>Un solicitante realiza una petición de información desde su terminal de comunicación.</p> <p>El geolocalizador nos indicara la posición geográfica del cliente.</p> <p>Saber el tipo de dispositivo (ordenador o dispositivo móvil), nos sirve para prepara el formato de la respuesta.</p> <p>Se buscan las preferencias del solicitante, en el almacén de configuraciones, lo cual nos permite mejorar las recomendaciones realizadas.</p> <p>La petición es atendida por el gestor de peticiones de información, que recopilara los datos del dispositivo o terminal, su ubicación y preferencias del cliente, para pasar la petición al generador de recomendaciones.</p> <p>Con los datos obtenidos por el gestor de peticiones, el generador de recomendaciones, consultará en el almacén de información depurada y el resultado será enviado al adaptador de contenidos para que sea enviado al solicitante.</p>
Precondiciones:	<p>Los dispositivos permitidos deben estar configurados.</p> <p>El geolocalizador debe estar iniciado.</p> <p>Los almacenes de configuración e información depurada tienen que estar disponibles.</p>

	<p>El gestor de peticiones, se implementará mediante un catálogo de servicios, que es una lista estandarizada de las peticiones que se puede realizar.</p> <p>El generador de recomendaciones debe estar disponible, configurado y parametrizado.</p> <p>Los solicitantes tienen que estar registrados.</p> <p>El adaptador de contenidos contendrá los formatos estándar definidos.</p> <p>En la misma sesión puede realizarse una nueva petición o seleccionar la recomendación actual. En el caso de realizarse una nueva petición, se generaría una nueva adquisición de información.</p>
Poscondiciones:	<p>Sí no se ha producido ningún error el gestor de peticiones registrara en el archivo log que la petición ha sido atendida, por el contrario enviara un aviso al manejador de errores para que informe al Administrador.</p> <p>Los actores son inicializados a la espera de una nueva solicitud.</p>

Actor	Descripción
Almacén de Información Depurada	Espacio físico donde se almacenan y proporcionan los datos íntegros para realizar recomendaciones.
Generador de Recomendaciones	Entidad que aplica técnicas de filtrado al almacén de información depurada en base a los métodos empleados para realizar las recomendaciones.
Almacén de configuraciones	Espacio físico donde se almacenan y proporcionan los datos necesarios para el sistema sobre las preferencias o puntuaciones de los solicitantes o clientes.
Geolocalizador	Es el encargado de obtener la información de ubicación geográfica (latitud y longitud) a partir de la dirección IP del dispositivo de conexión.
Gestor de peticiones de información	Es el encargado de atender las peticiones de los usuarios proporcionándoles información y acceso rápido a los servicios estándar del sistema. Entidad que, con la información de la petición del dispositivo y preferencias del solicitante, compone los parámetros para el generador de recomendaciones.
Solicitante recomendación	Se corresponde con cada uno de los usuarios que se han registrado en el sistema para poder realizar tareas propias de ellos como pedir, buscar, evaluar, modificar sus datos personales y de perfil.
Dispositivo	Terminal para introducir o mostrar datos del sistema.
Adaptador de Contenidos	Entidad que adapta y convierte la recomendación realizada a un formato apropiado para el dispositivo utilizado, empleando los protocolos que mejor se adapten a los dispositivos y a las redes desde los que el usuario accede.

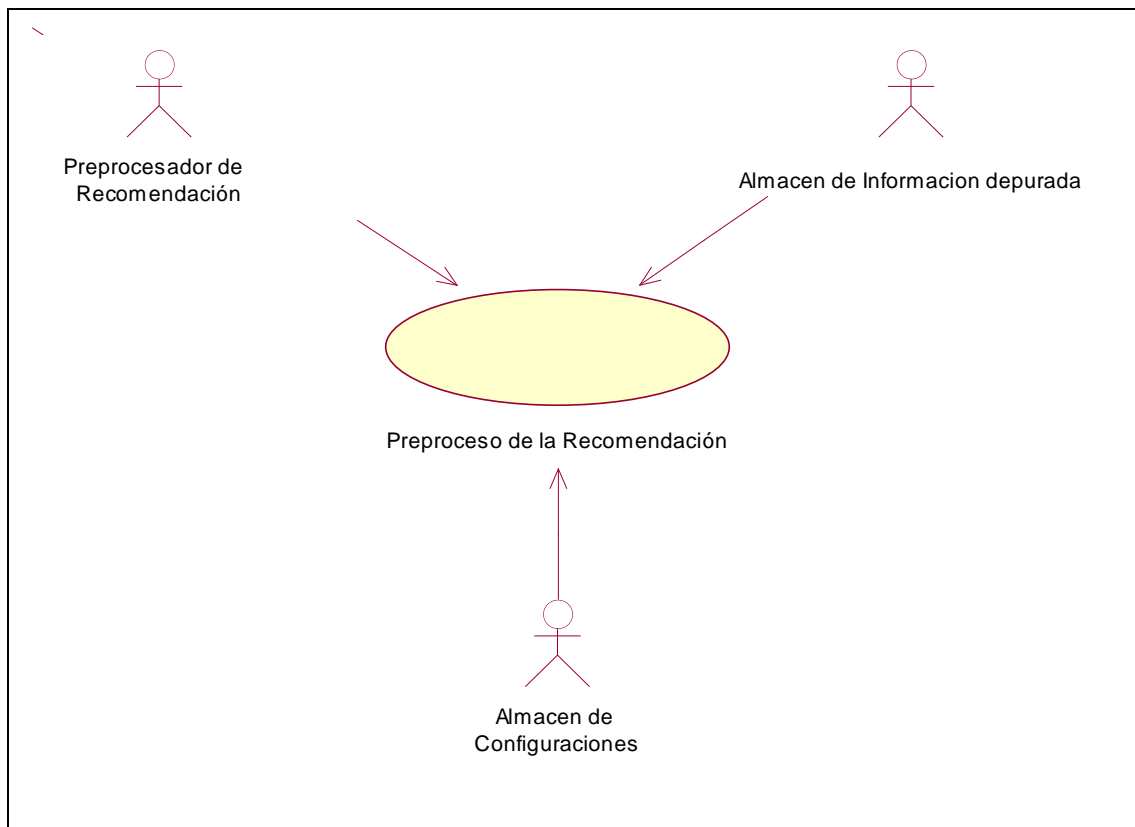
Caso de Uso 3.- Preproceso de la Recomendación

Ilustración 3: Caso de uso 3. Preproceso de la Recomendación

Caso de Uso 3: Preproceso de la Recomendación	
Descripción:	<p>En este caso, un organizador de información, tomará los recursos etiquetados del almacén de información depurada para clasificarlos según unos criterios establecidos. Por ejemplo: nivel de visitas, ubicación, origen, colectivo o dispositivo.</p> <p>Sí el solicitante de información está registrado, entonces el organizador consultará también, el almacén de configuración de perfiles, para obtener información de las preferencias de los clientes e introducir nuevos parámetros en la clasificación.</p> <p>Los criterios de clasificación se revisan continuamente para mejorar el sistema.</p> <p>La capacidad de los almacenes y los tiempos empleados por el preproceso se monitorizan regularmente.</p>
Precondiciones:	<p>Los almacenes de información depurada y configuración tienen que estar disponibles.</p> <p>El organizador de información debe estar configurado y tener acceso a los almacenes de información depurada y configuración.</p> <p>Los criterios para clasificar la información estarán definidos.</p> <p>La ejecución desentendida del preproceso debe estar definida.</p> <p>Antes de ejecutar el proceso de optimización, se realiza una copia de</p>

	respaldo
Poscondiciones:	El almacén depurado contiene datos clasificados e indexados para realizar recomendaciones de una forma ágil. El almacén de configuraciones ha sido reindexado. Las copias de respaldo del los almacenes se han realizado.

Actor	Descripción
Almacén de información depurada	Espacio físico donde se almacena y proporciona los datos íntegros para realizar recomendaciones.
Almacén de configuraciones	Espacio físico donde se almacena y proporciona los datos necesarios para el sistema sobre las preferencias o puntuaciones de los solicitantes o clientes
Preprocesador de recomendación	Entidad que organiza la información para las recomendaciones, a través de establecer clasificaciones, relaciones y jerarquías

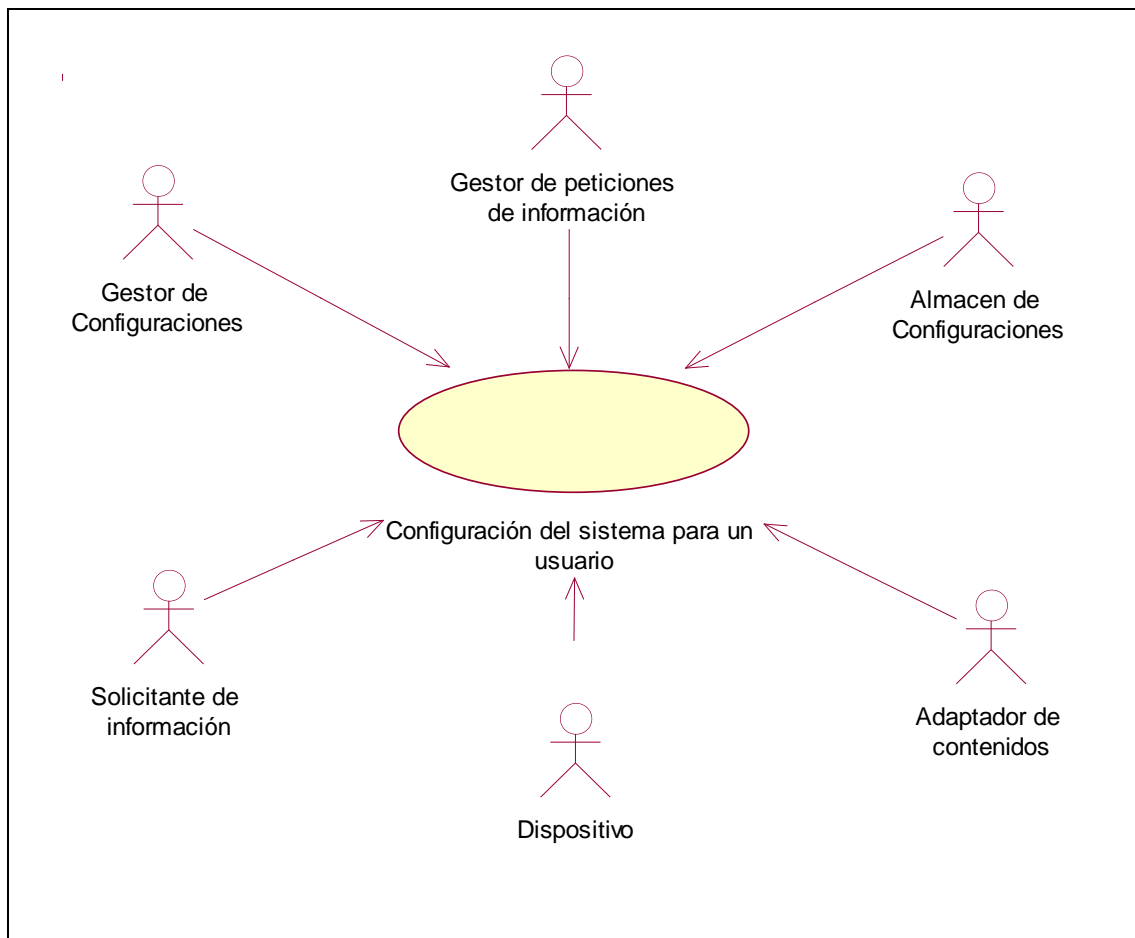
Caso de Uso 4.- Configuración del Sistema para un usuario (solicitantes registrados)

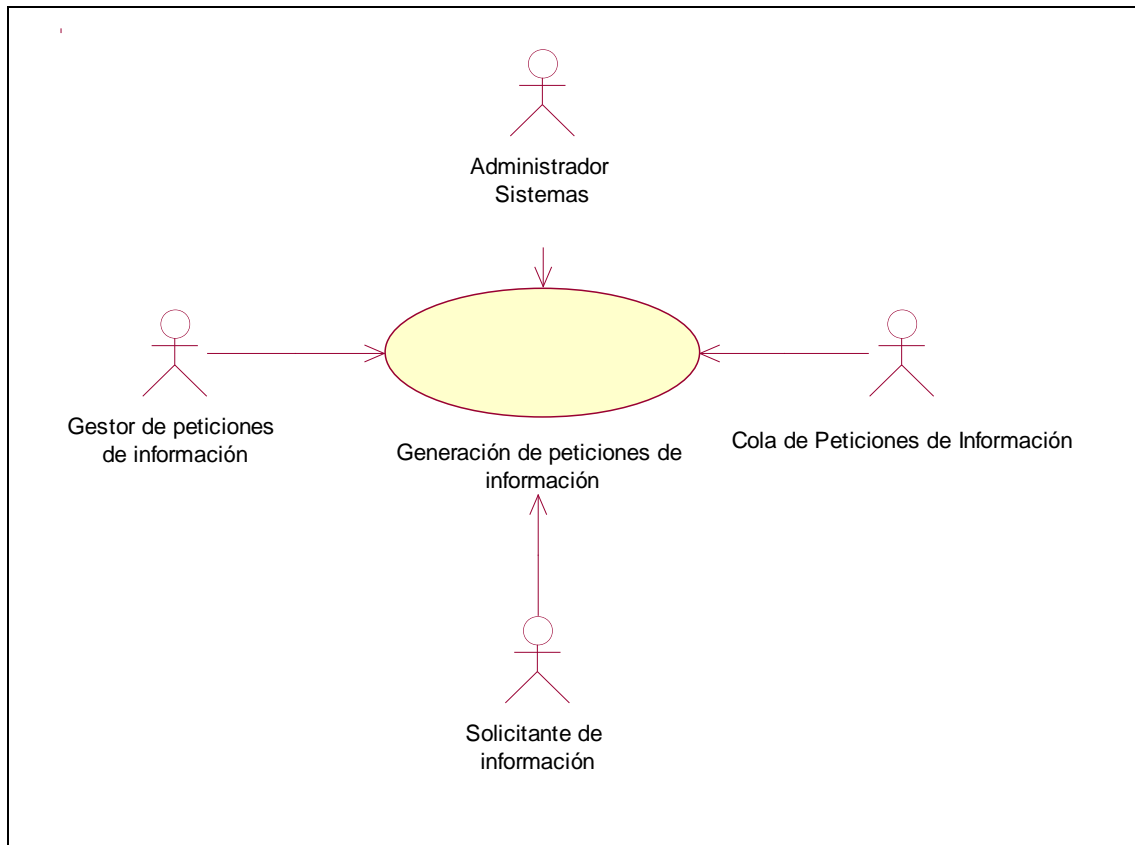
Ilustración 4: Caso de Uso 4. Configuración del Sistema para un usuario (solicitantes registrados)

Caso de Uso 4.- Configuración del Sistema para un usuario (solicitantes registrados)	
Descripción:	<p>El sistema de recomendación utiliza la información de las preferencias del usuario para hacer las recomendaciones.</p> <p>Los usuarios que hagan 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 y estará siempre accesible para su revisión por el propio personal usuario.</p> <p>Un Gestor de configuraciones clasificara a los usuarios según su perfil y se guardará en un almacén de configuraciones.</p> <p>Según el dispositivo o terminal desde el que se hace la petición, la respuesta tendrá que estar en el mismo formato, por lo que contaremos con un adaptador de contenidos.</p> <p>Los criterios de clasificación se revisan continuamente para mejorar el sistema.</p>

	Se realizan copias de respaldo
Precondiciones:	<p>El almacén de configuración tiene que estar disponible.</p> <p>El Gestor de Configuraciones debe estar parametrizado para obtener la información del usuario y crear los perfiles, además de tener acceso al almacén de configuración.</p> <p>Los dispositivos permitidos deben estar configurados.</p> <p>El personal usuario, tendrá acceso a sus datos con permisos de escritura y cambio.</p> <p>Adaptador de contenidos contendrá los formatos estándar definidos y al día.</p> <p>El Gestor de Peticiones tendrá acceso al almacén de configuraciones, para buscar información coincidente.</p> <p>Sí el usuario cambia sus datos, se reclasificara su perfil.</p>
Poscondiciones:	<p>Los datos proporcionados por el cliente, altas o modificaciones, quedan registrados en el almacén de configuraciones.</p> <p>El adaptador de contenidos, gestor de peticiones y el dispositivo son inicializados, para la próxima consulta.</p>

Actor	Descripción
Gestor de peticiones de información	<p>Es el encargado de atender las peticiones de los usuarios proporcionándoles información y acceso rápido a los servicios estándar del sistema.</p> <p>Algoritmo o función que con la información de la petición, del dispositivo y preferencias del solicitante, compone los parámetros para el generador de recomendaciones.</p>
Almacén de configuraciones	Espacio físico donde se almacena y proporciona los datos necesarios para el sistema sobre las preferencias o puntuaciones de los solicitantes o clientes
Adaptador de contenidos	Entidad que adapta y convierte la recomendación realizada a un formato apropiado para el dispositivo utilizado, empleando los protocolos que mejor se adapten a los dispositivos y redes, desde la que el usuario accede.
Dispositivo	Terminal, aparato para introducir datos o mostrar datos de un sistema de computación.
Gestor de configuraciones	Entidad que con la información proporcionada por el usuario, creara un perfil específico.
Solicitante de información	Se corresponde con cada uno de los usuarios que se han registrado en el sistema para poder realizar tareas propias de ellos como pedir, buscar, evaluar, modificar sus datos personales y de perfil.

Caso de uso 5.- Generación de peticiones de información



<p>Descripción:</p>	<p>En este caso de uso se anotarán las peticiones de información. Las peticiones se generarán de dos modos:</p> <ol style="list-style-type: none"> 1) Las que hace el administrador del sistema para mejorar y verificar el buen funcionamiento de los servicios prestados, por ejemplo al almacén de logs o al almacén de estadísticas. 2) Cuando un usuario solicita una recomendación y no hubiese información suficiente para atenderla, el sistema anotará dicha falta y se programará en la cola de peticiones de información. <p>Las peticiones son registradas por el gestor de peticiones y tras su validación, pasan a la cola de peticiones hasta que son atendidas.</p> <p>La cola de peticiones tiene que ser ágil, no puede tener un número elevado de peticiones sin responder.</p>
<p>Precondiciones:</p>	<p>El solicitante debe estar registrado. El rol de administrador tiene una cuenta con control total del sistema. Los servicios del gestor y la cola de peticiones están iniciados. Sí los ítems de la cola de peticiones no puedan ser remitidos al solicitante, tras x intentos, se eliminaran y se enviara un aviso al administrador que se registrara en el almacén de lóg.</p>

Poscondiciones:	Los ítems que no son contestados son notificados al administrador y eliminados.
-----------------	---

Actor	Descripción
Gestor de peticiones de información	Es el encargado de atender las peticiones de los usuarios proporcionándoles información y acceso rápido a los servicios estándar del sistema. Algoritmo o función que con la información de la petición, del dispositivo y preferencias del solicitante, compone los parámetros para el generador de recomendaciones.
Cola de peticiones de información	Registro de las llegadas del solicitante de información y que aún no ha sido atendida.
Administrador de Sistema	Hace referencia al responsable de la aplicación, es el que se encarga de asegurar el correcto funcionamiento del sistema y su mejora.
Solicitante de Información	Se corresponde con cada uno de los usuarios que se han registrado en el sistema para poder realizar tareas propias de ellos como pedir, buscar, evaluar, modificar sus datos personales y de perfil.

A continuación se describirán en detalle la interacción de las clases 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 Sistema de Recomendación, ya que es donde se realizan los procesos de negocio más relevantes en cuanto al desarrollo del sistema.

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 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. (Casos de Uso + Diagramas de Secuencia)
- **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.

2.3.- Diagramas de Secuencia.

“En cualquier sistema, los objetos interactúan entre sí intercambiando mensajes. Una interacción entre objetos puede necesitar de mensajes dentro de un contexto para lograr un propósito.” [UML – Raúl Alarcón]

UML provee un medio gráfico para representar la interacción entre los objetos a lo largo del tiempo en los diagramas de secuencia. *“Esta notación permite visualizar los mensajes de forma que podamos destacar sus elementos más importantes: nombre, parámetros (si los tiene) y secuencia. Gráficamente un mensaje se representa como una línea dirigida y casi siempre incluye el nombre de su operación.” [UML – Raúl Alarcón]*

Los Diagramas de Secuencia representan una interacción entre objetos de manera secuencial en el tiempo y muestra la participación de objetos en la interacción entre sus “líneas de vida” y los mensajes que ellos organizadamente intercambian en el tiempo.

Los componentes de un diagrama de secuencia son:

- Un Objeto o Actor.
- Mensaje de un objeto a otro objeto.
- Mensaje de un objeto a si mismo.

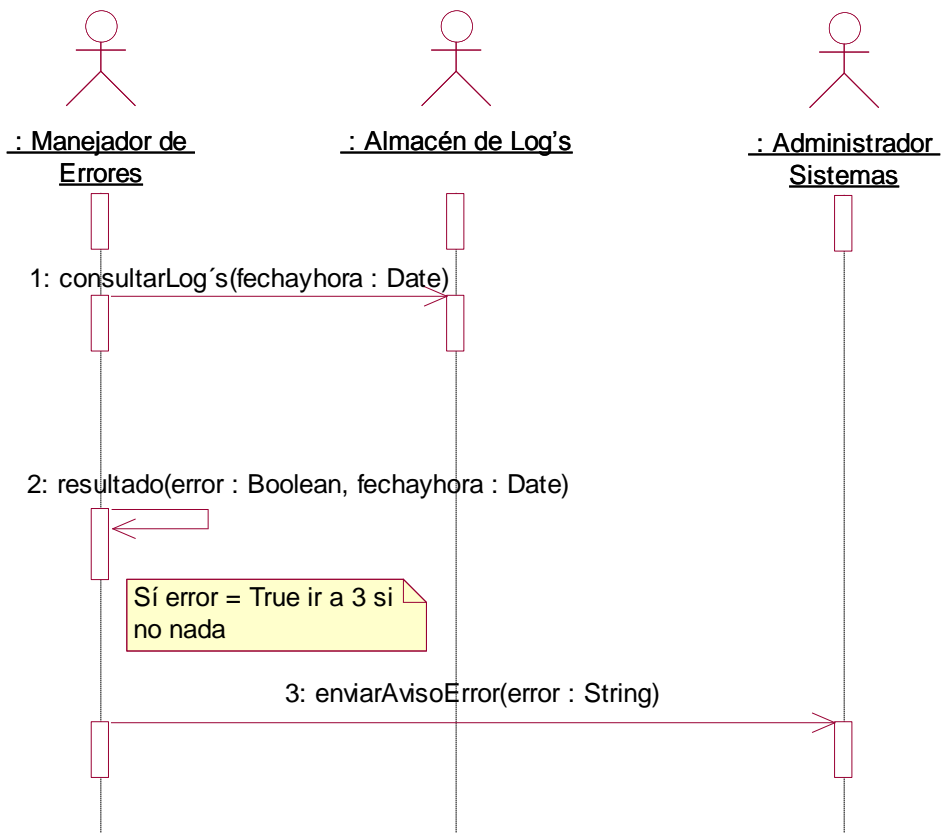
En las fases iniciales de análisis y diseño, realizaremos un diagrama de secuencia por cada escenario. Estos diagramas permitirán:

- Razonar más en detalle como es el comportamiento de un escenario.
- Obtener nuevas clases y objetos en el escenario.
- Detectar cuales son los métodos de las clases, al observar como se relacionan los objetos entre sí para llevar a cabo la tarea encomendada en el escenario.

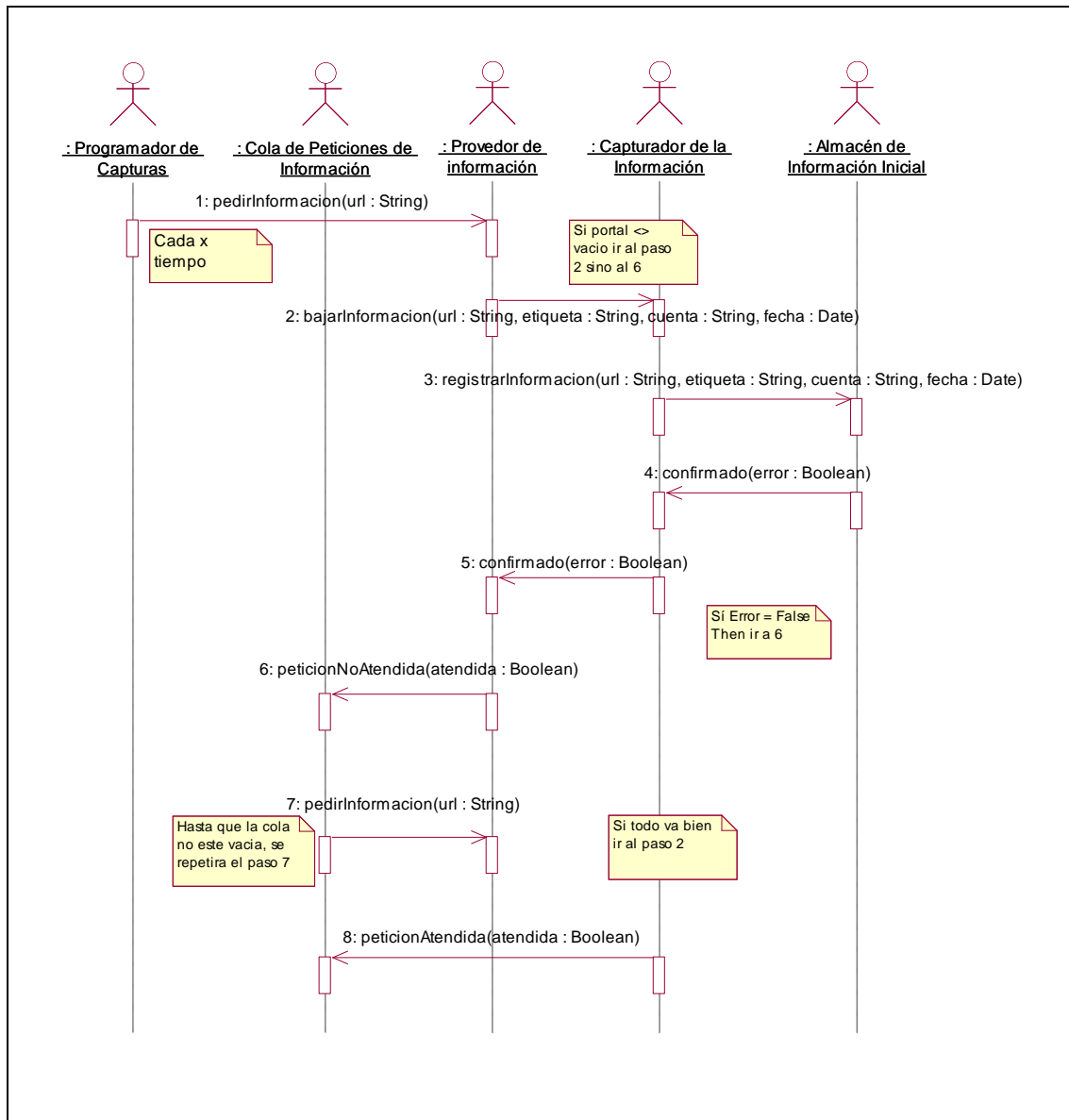
Estos diagramas también, tienen una función en las fases de prueba, con objeto de validar el código.

1.- Caso de Uso: Adquisición de Información

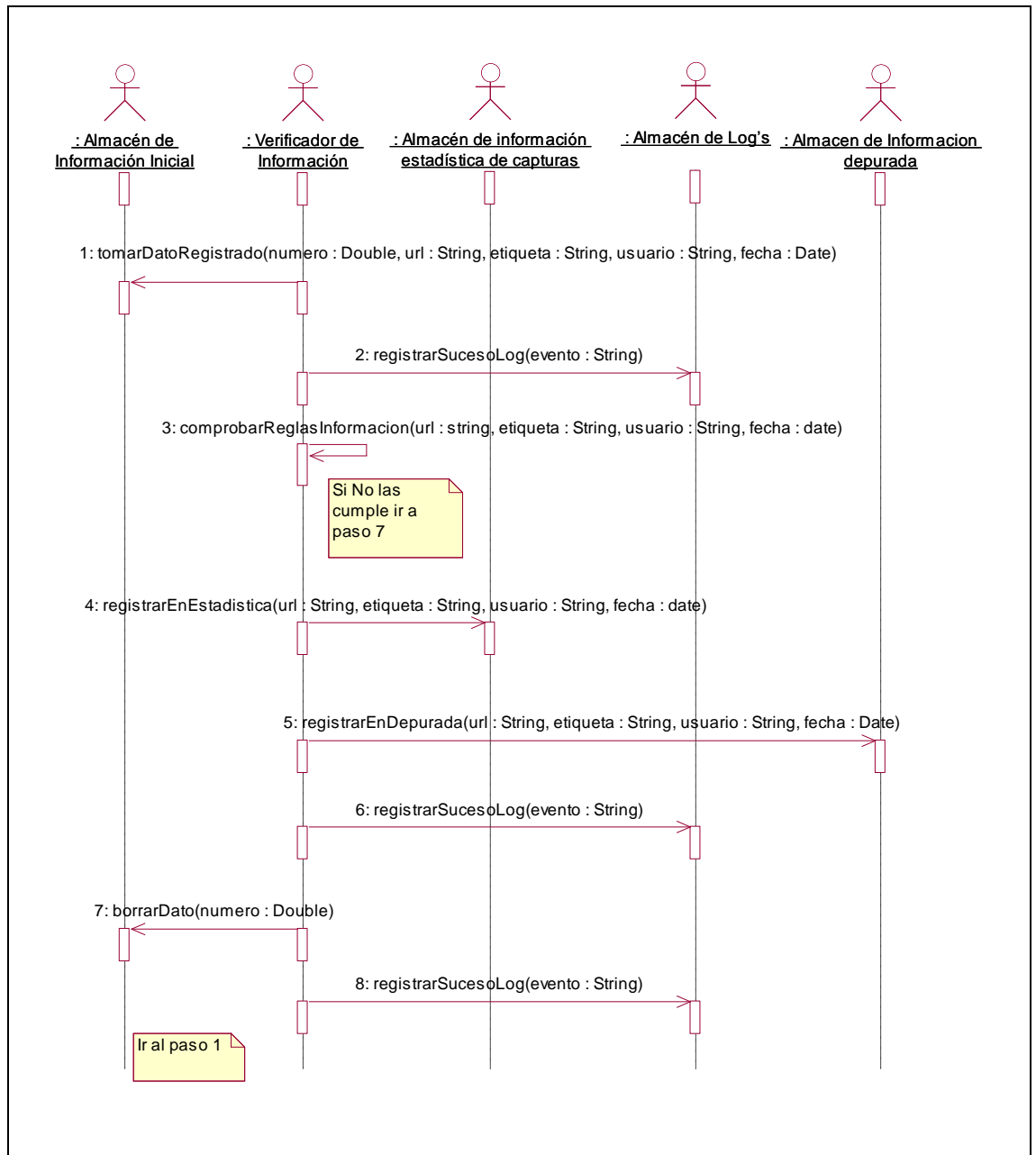
1.A.- Adquisición de Información: Advertencias



1.B.- Adquisición de Información: Captación

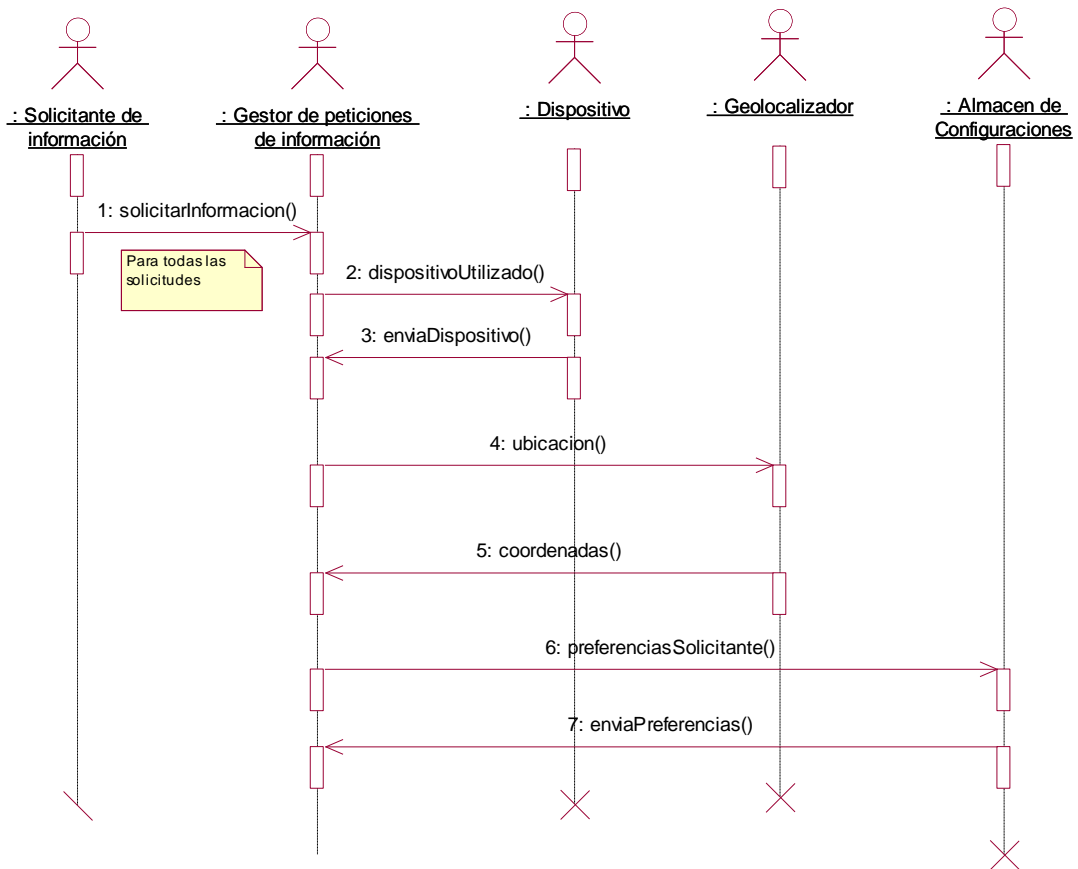


1.C.- Adquisición de Información: Verificación

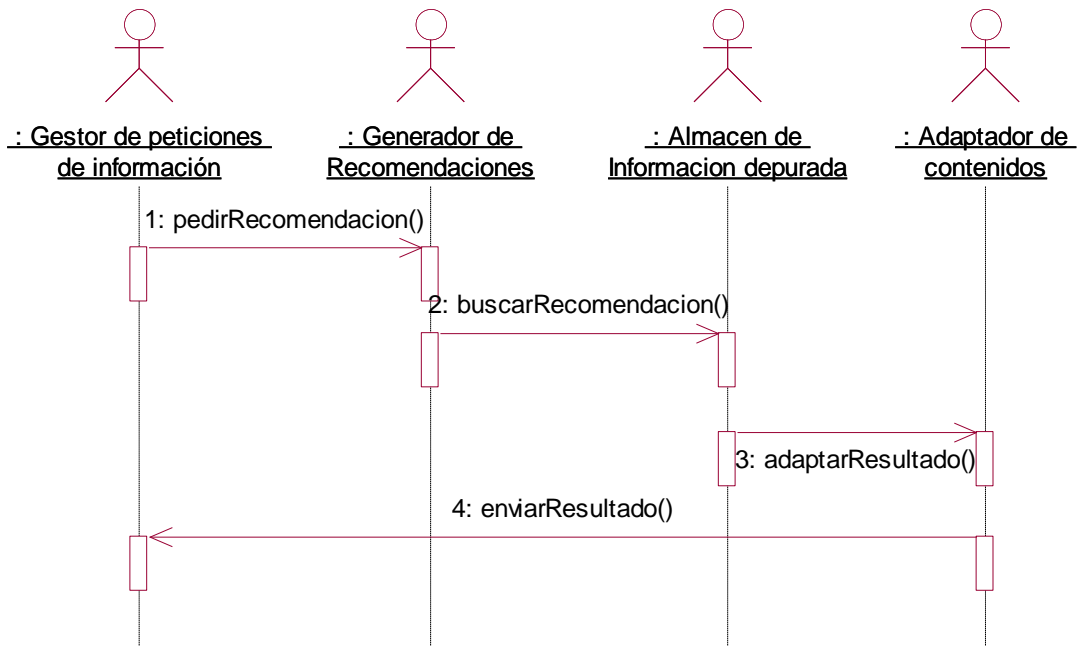


2.- Caso de Uso: Generación de Recomendaciones

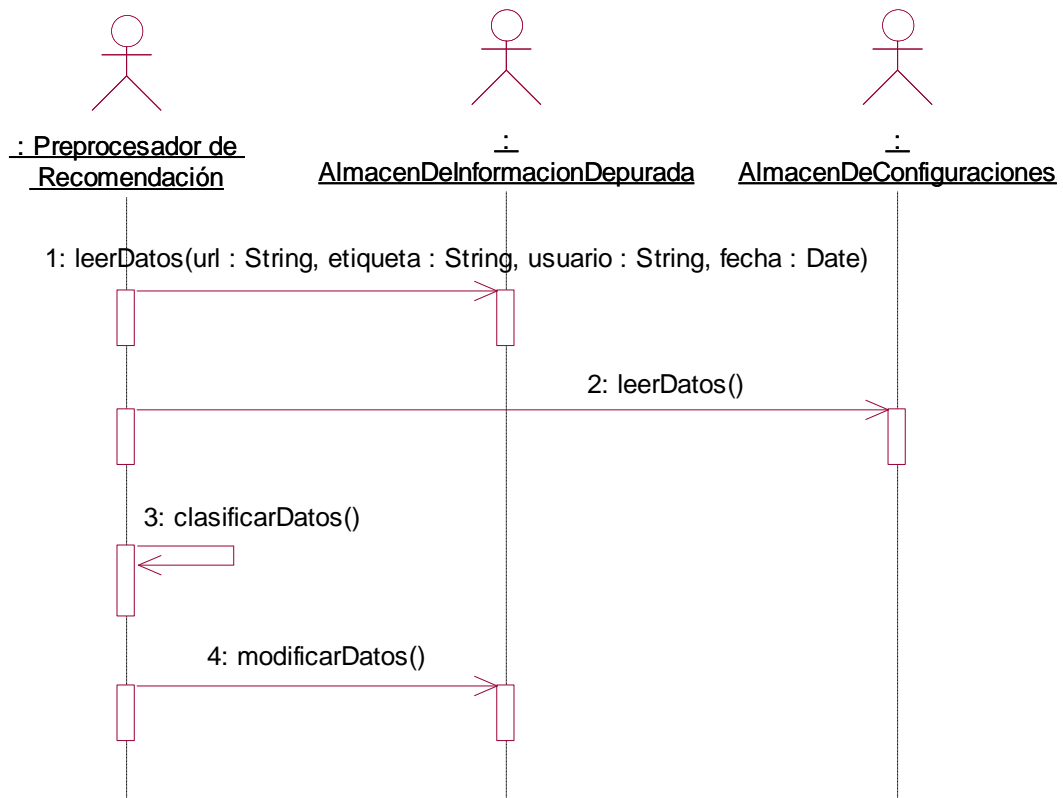
2.A.- Generación de Recomendaciones: Petición



2.B.- Generación de Recomendaciones: Resultado

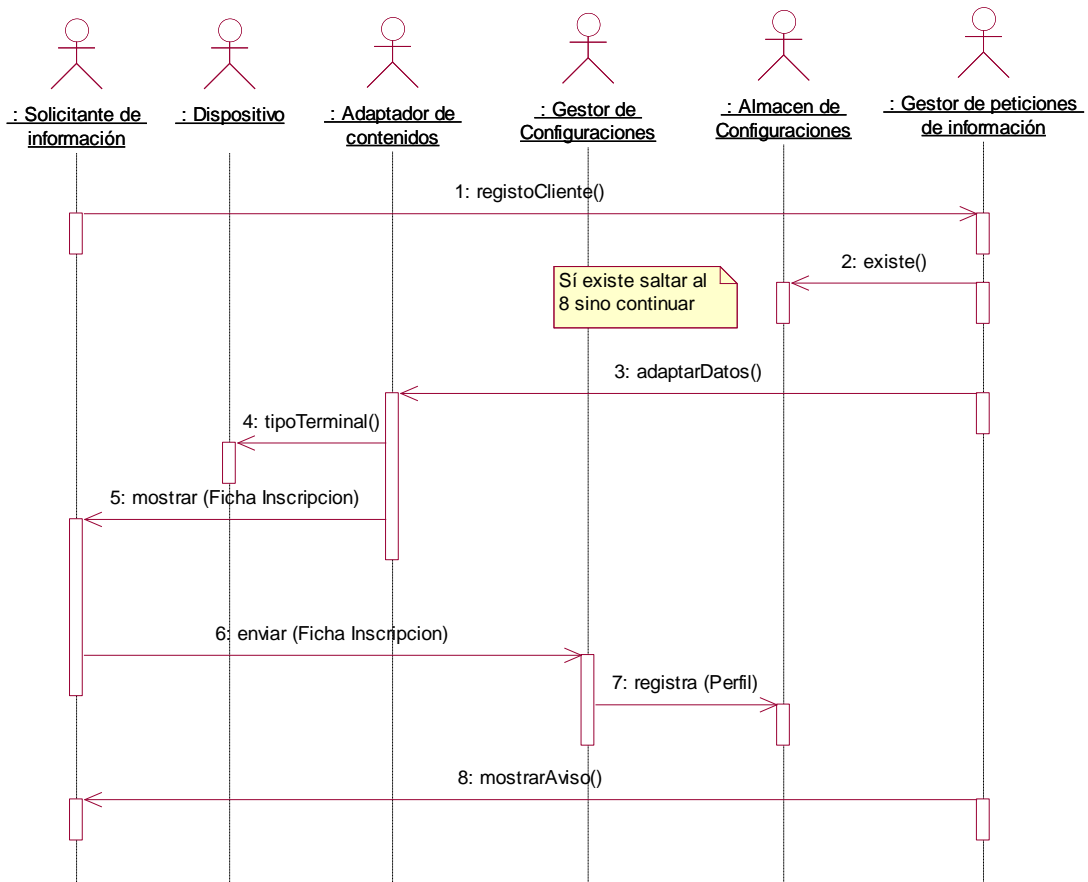


3.- Caso de Uso: Preproceso de Recomendación

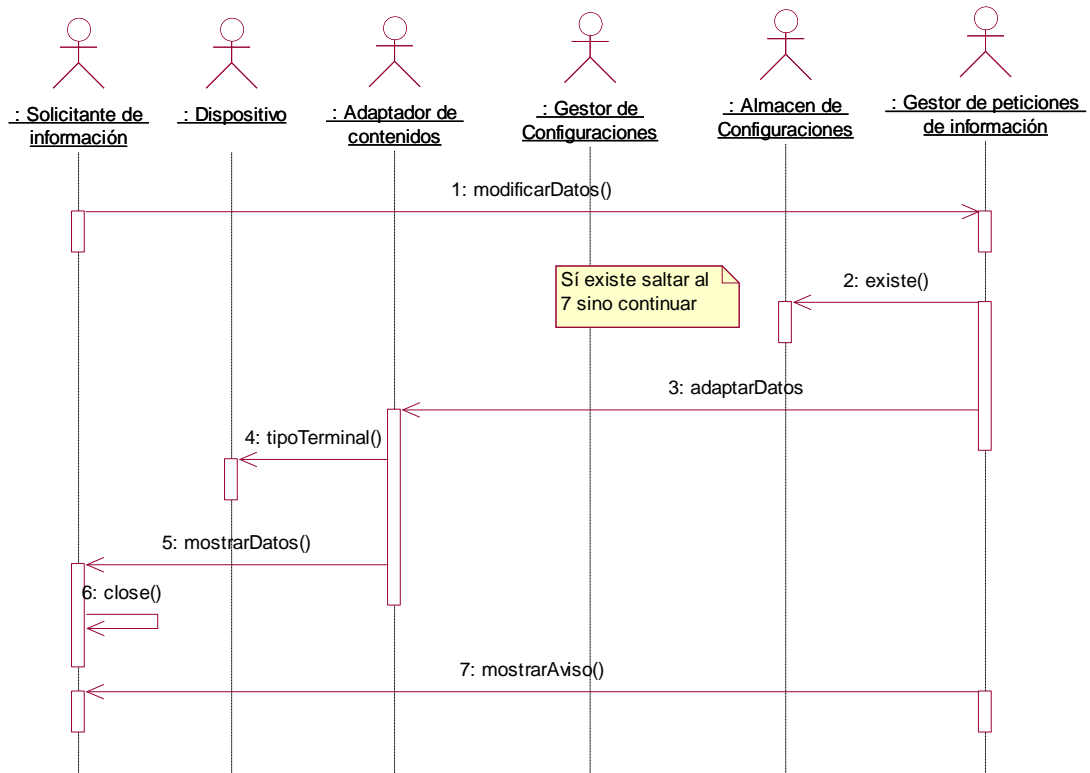


4.- Caso de Uso: Configuración del sistema para un usuario

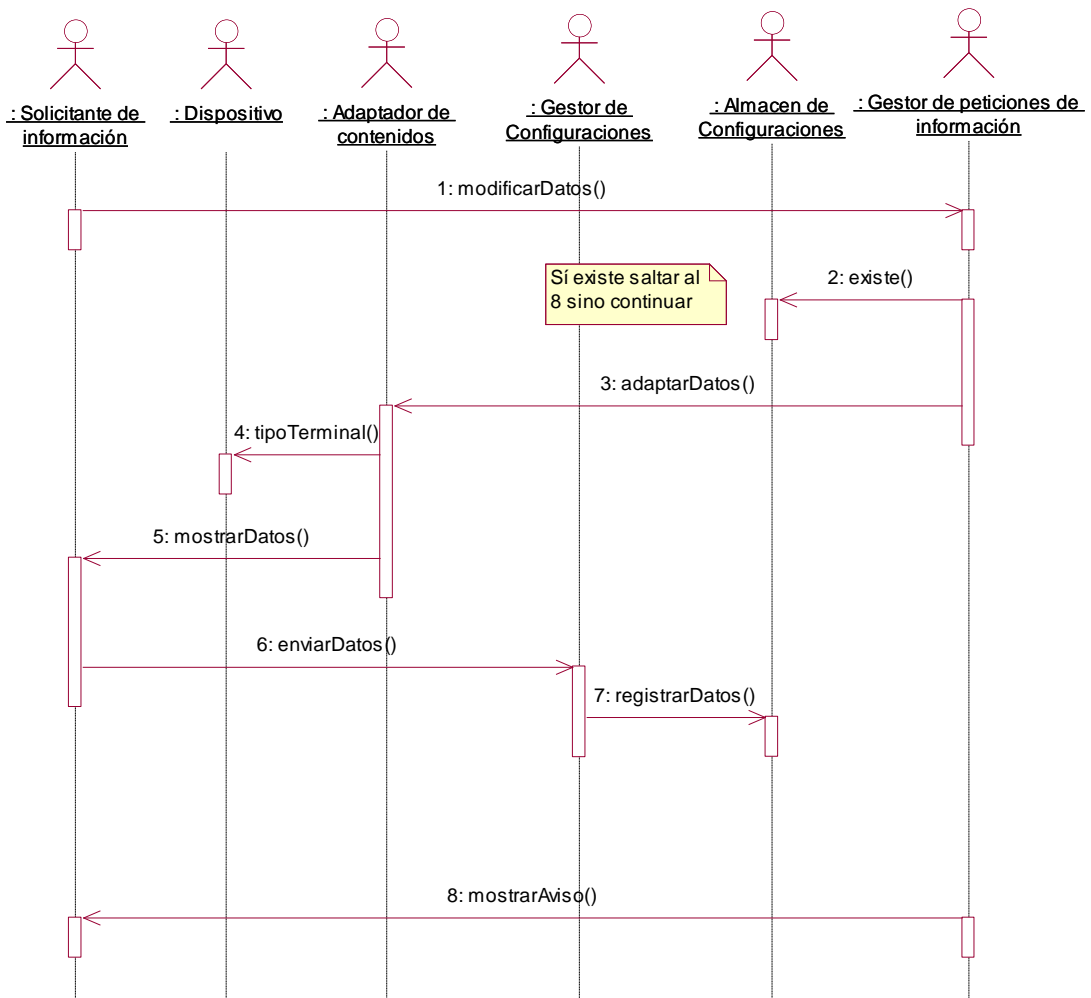
4.A.- Configuración del sistema para un usuario: Registro



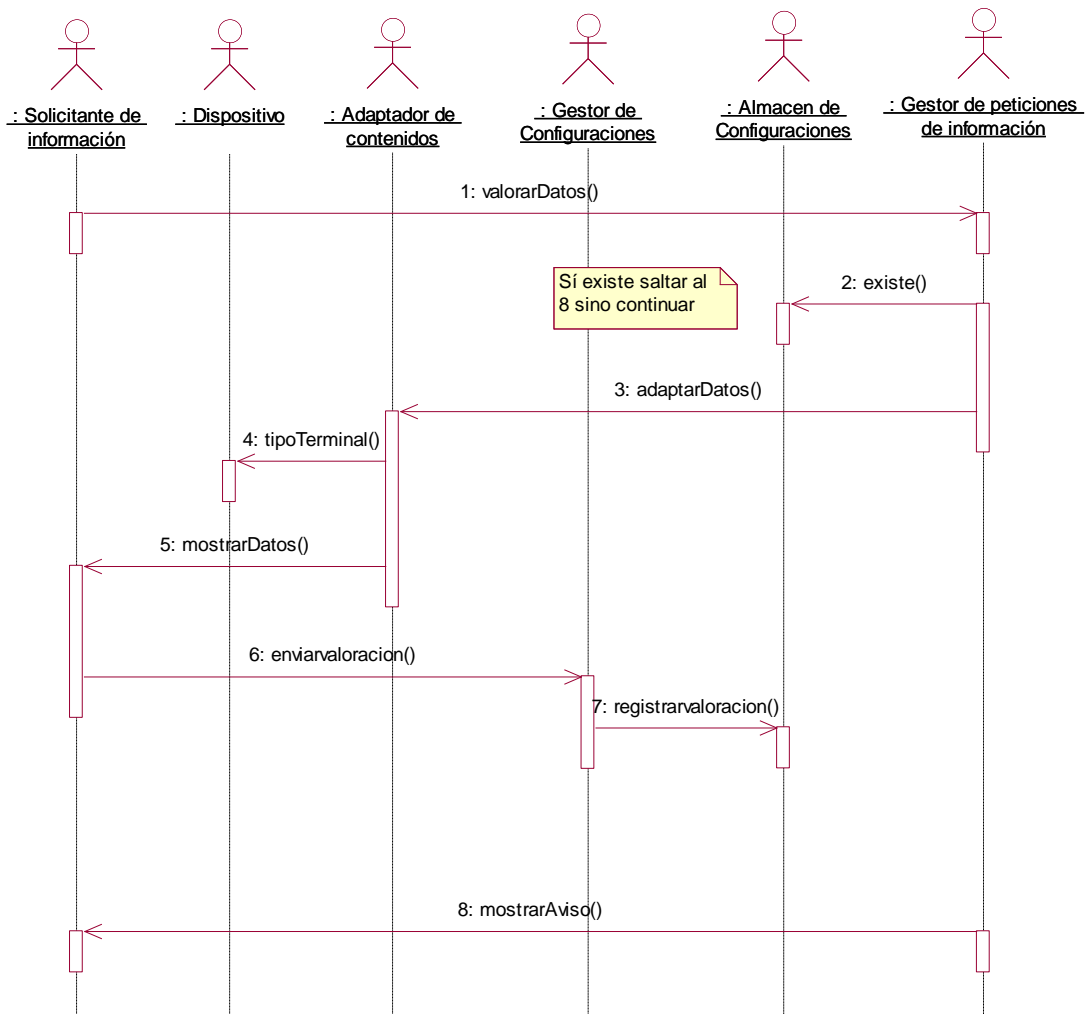
4.B.- Configuración del sistema para un usuario: Visualización



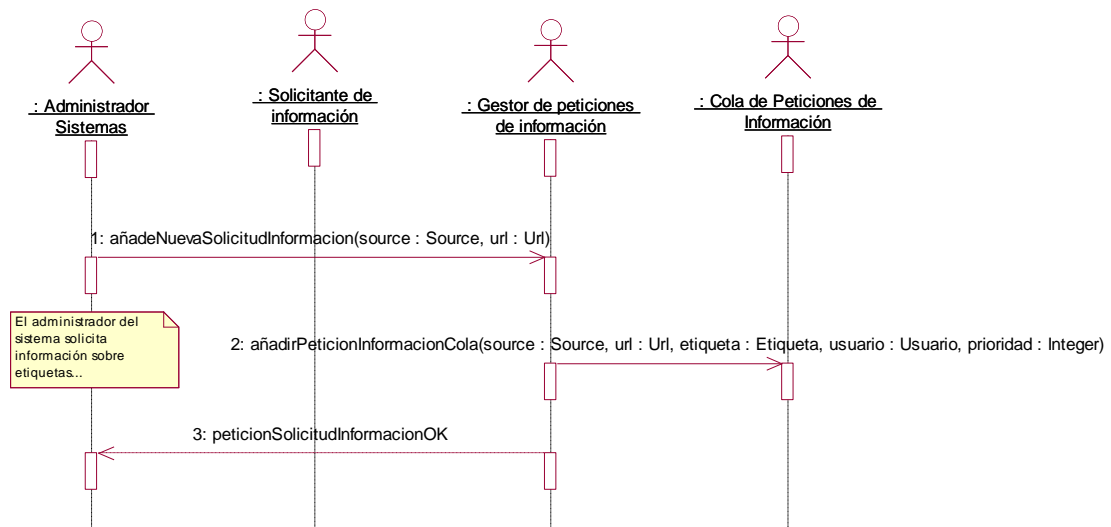
4.C.- Configuración del sistema para un usuario: Modificación



4.D.- Configuración del sistema para un usuario: Puntuación



4.- Caso de Uso: Generación de peticiones de información

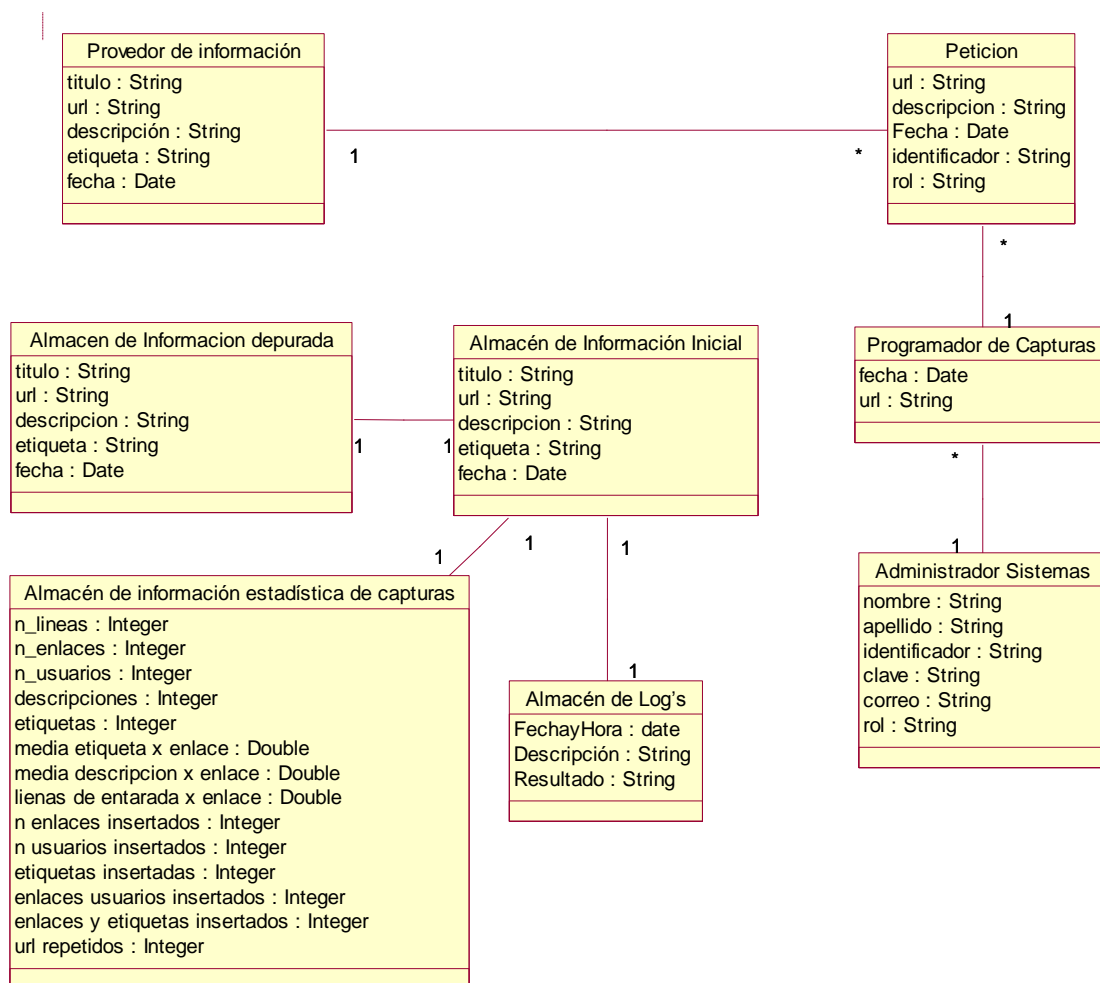


2.4.- Diagrama de Clases

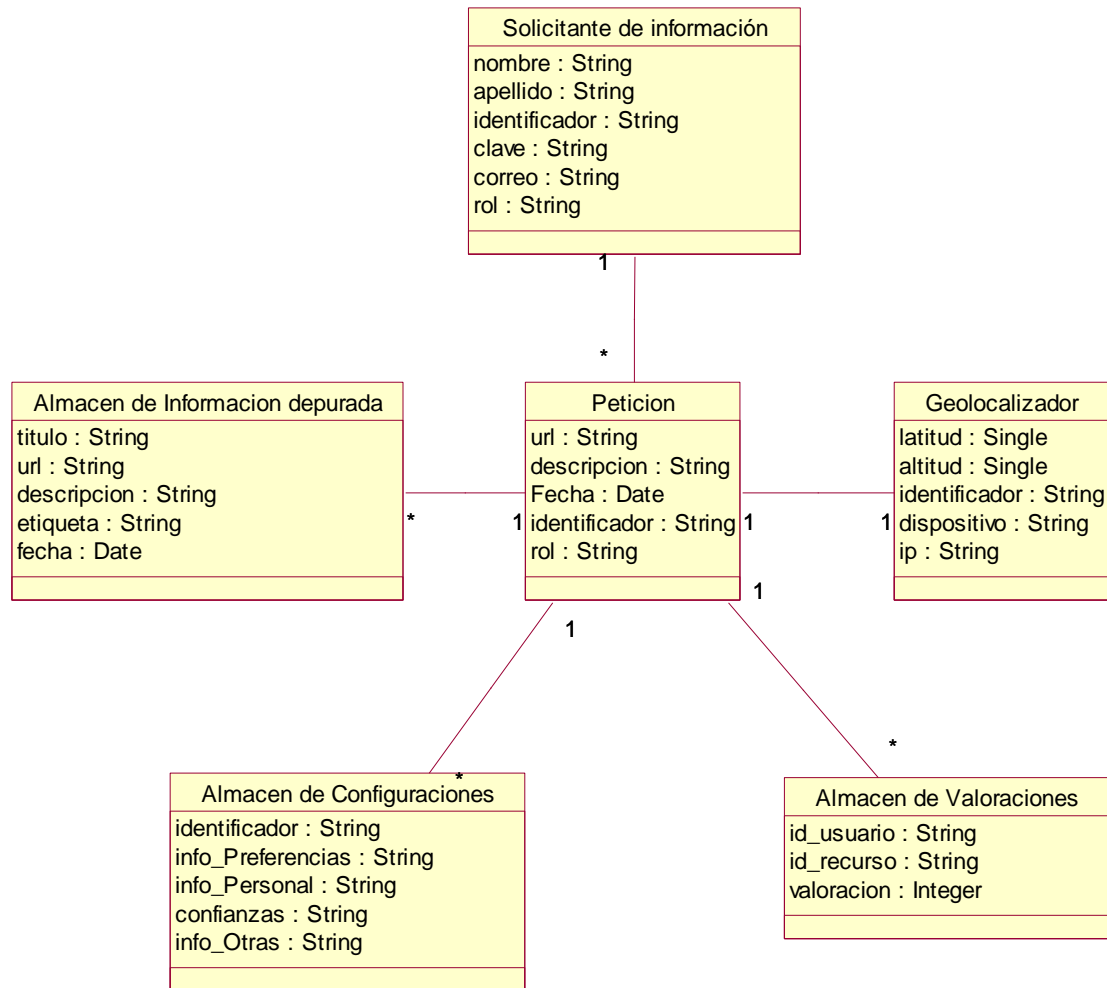
“Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones.” [UML – Raúl Alarcón]

“Los diagramas de clases se utilizan para modelar la información estática de un sistema. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino que también para construir sistemas ejecutables aplicando ingeniería directa.” [UML – Raúl Alarcón]

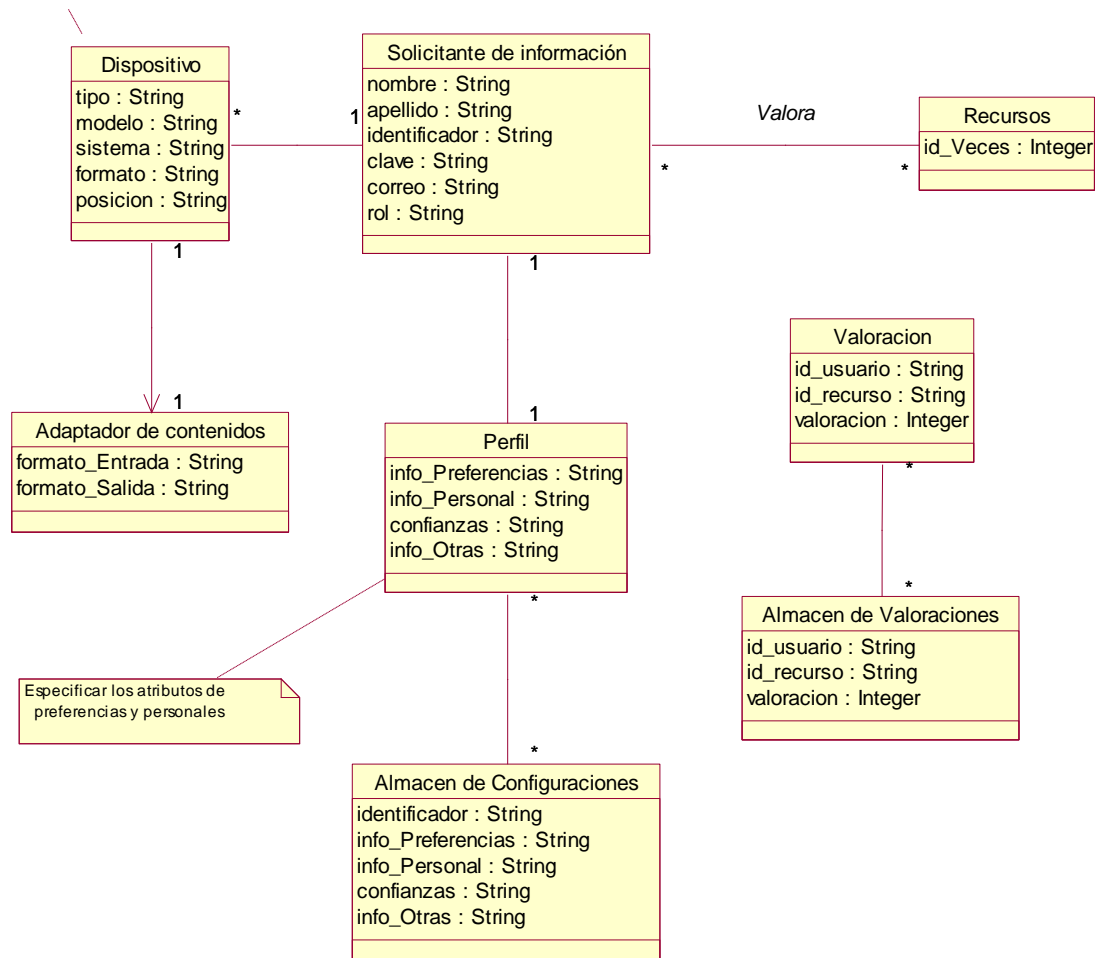
I. Adquisición de Información



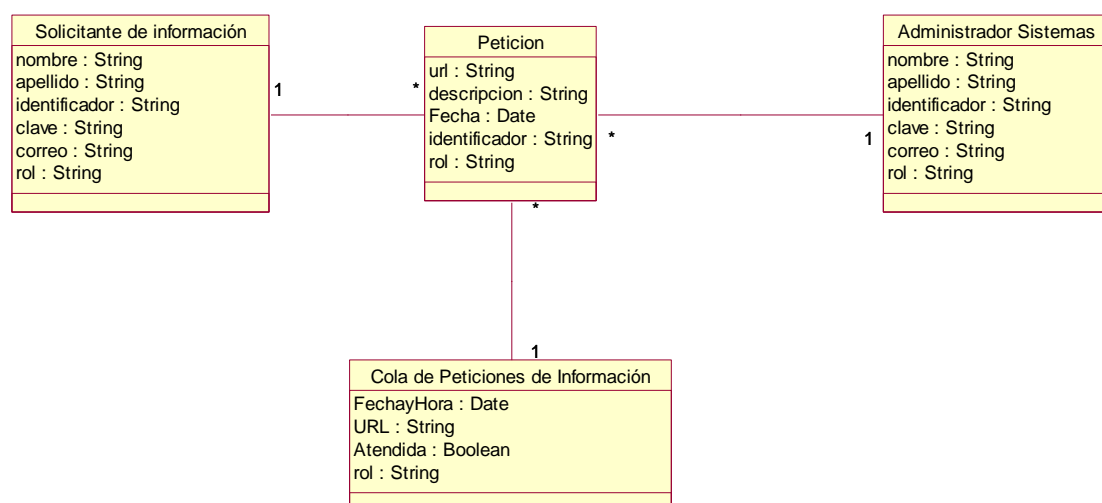
II. Generación de Recomendaciones



III. Configuración del sistema para un usuario



IV. Generación de Peticiones de Información



3.- Conclusiones

Para conseguir el éxito en cualquier proyecto, es fundamental disponer de las especificaciones del sistema a desarrollar. No importa lo bien diseñado o codificado que pueda estar, si no conocemos la situación actual y los objetivos a alcanzar, el resultado puede defraudar al cliente y frustrar al desarrollador.

La información que la gente comparte, a través de las redes sociales u otros medios, nos permite comprender mejor las tendencias o gustos del personal usuario. Su aplicación en los Sistemas de Recomendación, es utilizada para mejorar el tratamiento personalizado del cliente, para las estrategias de marketing o mejorar la posición de una empresa en el mercado.

En el desarrollo del proyecto, se han utilizado UML que permiten, una mejor comprensión del sistema y mayor calidad en los documentos de especificaciones.

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

- La especificación de un Sistema de Recomendación basado en Etiquetado Social.
- El documento de especificación que incluye:
 - Casos de Uso
 - Diagramas de Secuencia
 - Diagrama de Clases
 - Código en C++ y Java de las Clases y sus métodos, obtenida de forma automática con Rational Rose.

Como trabajos futuros se podría realizar lo siguiente:

- La implementación de un caso de uso y seguir un proceso de refinamiento cíclico (Catalysis).
- Compartir la información con otros Sistemas de Recomendación, podría mejorar o incrementar las recomendaciones realizadas, por ser mayor los datos a consultar.

El estudio realizado, me ha permitido adquirir nuevos conocimientos, además de motivarme a profundizar sobre la necesidad de definir las especificaciones de un sistema, sobre todo por aplicar dicha solución en el sector comercial.

Señalar que definir las especificaciones de un sistema de recomendación, sin contar con un sistema real, ha implicado una dificultad añadida al no contar con personas que formen parte del sistema real a quien consultar. He aprendido una amplia variedad de tecnologías, sobretudo la notación UML y un método de desarrollo enfocado a la generación de modelos de los procesos de negocio (As Is y To Be).

Otro aspecto positivo a destacar es el hecho de cómo, un proyecto nacido de una conversación con mi tutor, ha sido capaz de llegar a buen puerto y sobre todo ver como mediante la utilización de unas técnicas y herramientas (totalmente desconocidas por mí), ha quedado plasmado en papel y se puedan difundir instrucciones sencillas y precisas.

Finalmente, comentar que el haber realizado este proyecto, ha supuesto para mí un verdadero reto desde los primeros momentos de su realización, ya que en un principio la temática era totalmente desconocida, pero a la vez y sobre todo, ha sido una experiencia muy positiva y enriquecedora.

4.- Bibliografía

- UML y PATRONES. Una introducción al análisis y diseño orientado a objetos y al proceso unificado, Segunda edición por Graig Larman.
- IEEE Standard Glossary of Software Engineering Terminology 610.12-1990
- Aprendiendo UML en 24 horas por Joseph Schmuller
- Metodología MÉTRICA Versión 3, © Ministerio de Hacienda y Administraciones Públicas.
- Diseño orientado a objetos con UML por Raúl Alarcón
- UML con Rational Rose por Gesvin Romero Moreno
- Artículo Privacidad y Seguridad Redes Sociales – Daniel Ortega 2011
- Informe Chaos: Visión y críticas sobre el éxito de los proyectos software (realizado por Standish Group 1994)
- Ingeniería de Requerimientos, Ingeniería del Software por Lizka Johany Herrera J.

5.- Anexos

5.1.- Código Java generado con Rational Rose del diagrama de clase “Generación de peticiones de información”

5.1.1. Solicitante de Información.

```
/**
 * Entidad que solicita una nueva captura de información. Estas peticiones
 las
 * gestiona el gestor de peticiones de información.
 * Para que los usuarios puedan obtener seguridad, acceso al sistema,
 * administración de recursos, etc, dichos usuarios deberán identificarse.
 * El usuario necesita una cuenta (una cuenta de usuario) y una contraseña.
 * Los usuarios utilizan una interfaz de usuario para acceder a los
 sistemas, el
 * proceso de identificación es conocido como identificación de usuario o
 acceso
 * del usuario al sistema .
 * Se corresponde con cada uno de los usuarios que se han registrado en el
 sistema
 * para poder realizar tareas propias de ellos como pedir, buscar, evaluar,
 * modificar sus datos personales y de perfil.
 */
public class SolicitanteDeInformacion
{
    private String nombre;
    private String apellido;
    private String identificador;
    private String clave;
    private String correo;

    /**
     * Administrador
     * Usuario Registrado
     * Usuario Anónimo
     */
    private String rol;
    public AlmacenDeConfiguraciones theAlmacenDeConfiguraciones[];

    /**
     * @roseuid 4FD6F8D7035C
     */
    public SolicitanteDeInformacion()
    {

    }

    /**
     * @return Boolean
     * @roseuid 4E6B1E360128
     */
    public Boolean Consultar()
    {
        return null;
    }

    /**
```

```
* @return Boolean
* @roseuid 4E6B1E1601D4
*/
public Boolean Insertar()
{
    return null;
}

/**
 * @return Boolean
 * @roseuid 4E6B1E20003E
 */
public Boolean Modificar()
{
    return null;
}

/**
 * @return Boolean
 * @roseuid 4FD6F4DF00AC
 */
public Boolean SolicitanteDeInformación()
{
    return null;
}
}
```

5.1.2. Administrador.

```
/**
 * Persona responsable de asegurar el correcto funcionamiento del sistema.
 *
 * Hace referencia al responsable de la aplicación, es el que se encarga de
 * asegurar el correcto funcionamiento del sistema y su mejora.
 */
public class AdministradorSistemas
{
    private String nombre;
    private String apellido;
    private String identificador;
    private String clave;
    private String correo;
    private String rol;

    /**
     * @roseuid 4FD6FC2F0143
     */
    public AdministradorSistemas()
    {

    }

    /**
     * @param error
     * @return Boolean
     * @roseuid 4E8373730399
     */
    public Boolean enviarAvisoError(String error)
    {
        return null;
    }
}
```

```
/**
 * @return Boolean
 * @roseuid 4E80D00F0000
 */
public Boolean Insertar()
{
    return null;
}

/**
 * @return Boolean
 * @roseuid 4E80D0240196
 */
public Boolean Modificar()
{
    return null;
}

/**
 * @return Boolean
 * @roseuid 4E80D02C0119
 */
public Boolean Consultar()
{
    return null;
}

/**
 * @return Boolean
 * @roseuid 4E80D0310280
 */
public Boolean Eliminar()
{
    return null;
}
}
```

5.1.3. Petición.

```
/**
 * El cliente realiza una petición de información, aportando una serie de
 * datos.
 */
public class Peticion
{
    private String url;
    private String descripcion;
    private Date Fecha;
    private String identificador;
    private String rol;
    public GestorDePeticonesDeInformación
theGestorDePeticonesDeInformación;

    /**
     * @roseuid 4FD6F64203AC
     */
    public Peticion()
    {
    }
}
}
```

5.1.4. Cola de peticiones de información.

```
/**
 * Registro de la información solicitada a los proveedores y que aún no ha
 sido
 * atendida.
 *
 * Registro de las llegadas de información solicitada a los proveedores o
 * peticiones de los solicitantes de información y que aún no han sido
 atendidas.
 *
 */
public class ColaDePeticionesDeInformación
{
    private Date FechaHora;
    private String URL;
    private Boolean Atendida;
    private String rol;

    /**
     * @roseuid 4FD70B02012B
     */
    public ColaDePeticionesDeInformación()
    {

    }

    /**
     * @param source
     * @param url
     * @param etiqueta
     * @param usuario
     * @param prioridad
     * @return Boolean
     * @roseuid 4DAC55C20147
     */
    public Boolean añadirPeticionInformacionCola(String source, String url,
String etiqueta, String usuario, Integer prioridad)
    {
        return null;
    }

    /**
     * @param url - Direccion de conexion a las redes sociales
     *
     * @return Boolean
     * @roseuid 4E25BBD30157
     */
    public Boolean lanzarPeticion(String url)
    {
        return null;
    }

    /**
     * @param atendida
     * @return Boolean
     * @roseuid 4E25C1D10157
     */
    public Boolean peticionNoAtendida(Boolean atendida)
```



```
{
    return null;
}

/**
 * @param atendida
 * @return Boolean
 * @roseuid 4E25D1A40203
 */
public Boolean peticionAtendida(Boolean atendida)
{
    return null;
}

/**
 * @return Boolean
 * @roseuid 4E83792200FA
 */
public Boolean Eliminar()
{
    return null;
}

/**
 * @return Boolean
 * @roseuid 4E8378F501F4
 */
public Boolean Insertar()
{
    return null;
}

/**
 * @return Boolean
 * @roseuid 4E83788D033C
 */
public Boolean Consultar()
{
    return null;
}
}
```

5.2.- Código C++ generado con Rational Rose del diagrama de clase “Generación de peticiones de información”

5.2.1. Solicitante de Información.

```
//## begin module%1.10%.codegen_version preserve=yes
// Read the documentation to learn more about C++ code generator
// versioning.
//## end module%1.10%.codegen_version

//## begin module%4DAC50E40308.cm preserve=no
// %X% %Q% %Z% %W%
//## end module%4DAC50E40308.cm

//## begin module%4DAC50E40308.cp preserve=no
//## end module%4DAC50E40308.cp

//## Module: SolicitanteDeInformacion%4DAC50E40308; Pseudo Package
specification
//## Source file: D:\TEMP\C++\SolicitanteDeInformacion.h

#ifndef SolicitanteDeInformacion_h
#define SolicitanteDeInformacion_h 1

//## begin module%4DAC50E40308.additionalIncludes preserve=no
//## end module%4DAC50E40308.additionalIncludes

//## begin module%4DAC50E40308.includes preserve=yes
//## end module%4DAC50E40308.includes

// Gestor de Configuraciones
#include "Gestor de Configuraciones.h"
// AlmacenDeConfiguraciones
#include "AlmacenDeConfiguraciones.h"
// Recomendacion
#include "Recomendacion.h"
// Valoracion
#include "Valoracion.h"
//## begin module%4DAC50E40308.additionalDeclarations preserve=yes
//## end module%4DAC50E40308.additionalDeclarations

//## begin SolicitanteDeInformacion%4DAC50E40308.preface preserve=yes
//## end SolicitanteDeInformacion%4DAC50E40308.preface

//## Class: SolicitanteDeInformacion%4DAC50E40308
// Entidad que solicita una nueva captura de información.
// Estas peticiones las
// gestiona el gestor de peticiones de información.
// Para que los usuarios puedan obtener seguridad, acceso
// al sistema,
// administración de recursos, etc, dichos usuarios deberán
// identificarse.
// El usuario necesita una cuenta (una cuenta de usuario) y
// una contraseña.
// Los usuarios utilizan una interfaz de usuario para
// acceder a los sistemas, el
// proceso de identificación es conocido como
// identificación de usuario o acceso
```

```
// del usuario al sistema .
// Se corresponde con cada uno de los usuarios que se han
// registrado en el sistema
// para poder realizar tareas propias de ellos como pedir,
// buscar, evaluar,
// modificar sus datos personales y de perfil.
//## Category: <Top Level>
//## Persistence: Transient
//## Cardinality/Multiplicity: n

class SolicitanteDeInformacion
{
  //## begin SolicitanteDeInformacion%4DAC50E40308.initialDeclarations
preserve=yes
  //## end SolicitanteDeInformacion%4DAC50E40308.initialDeclarations

public:
  //## Constructors (generated)
  SolicitanteDeInformacion();

  SolicitanteDeInformacion(const SolicitanteDeInformacion &right);

  //## Constructors (specified)
  //## Operation: SolicitanteDeInformacion%4FD6F8D7035C
  SolicitanteDeInformacion ();

  //## Destructor (generated)
  ~SolicitanteDeInformacion();

  //## Assignment Operation (generated)
  SolicitanteDeInformacion & operator=(const SolicitanteDeInformacion
&right);

  //## Equality Operations (generated)
  int operator==(const SolicitanteDeInformacion &right) const;

  int operator!=(const SolicitanteDeInformacion &right) const;

  //## Other Operations (specified)
  //## Operation: Consultar%4E6B1E360128
  Boolean Consultar ();

  //## Operation: Insertar%4E6B1E1601D4
  Boolean Insertar ();

  //## Operation: Modificar%4E6B1E20003E
  Boolean Modificar ();

  //## Operation: SolicitanteDeInformacion%4FD6F4DF00AC
  Boolean SolicitanteDeInformacion ();

  //## Get and Set Operations for Associations (generated)

  //## Association: <unnamed>%4F318401016E
  //## Role:
  SolicitanteDeInformacion::theAlmacenDeConfiguraciones%4F3184050239
  const UnboundedSetByReference<Gestor_de_Configuraciones>
get_the_Gestor_de_Configuraciones () const;
```

```
void set_the_Gestor_de_Configuraciones
(UnboundedSetByReference<Gestor_de_Configuraciones> value);

// Additional Public Declarations
//## begin SolicitanteDeInformacion%4DAC50E40308.public preserve=yes
//## end SolicitanteDeInformacion%4DAC50E40308.public

protected:
// Additional Protected Declarations
//## begin SolicitanteDeInformacion%4DAC50E40308.protected
preserve=yes
//## end SolicitanteDeInformacion%4DAC50E40308.protected

private:
//## Get and Set Operations for Class Attributes (generated)

//## Attribute: nombre%4E6B1D140203
const String get_nombre () const;
void set_nombre (String value);

//## Attribute: apellido%4E6B1D26037A
const String get_apellido () const;
void set_apellido (String value);

//## Attribute: identificador%4E6B1D41009C
const String get_identificador () const;
void set_identificador (String value);

//## Attribute: clave%4E6B1D5D0119
const String get_clave () const;
void set_clave (String value);

//## Attribute: correo%4E6B1D440177
const String get_correo () const;
void set_correo (String value);

//## Attribute: rol%4E7A1DE50222
// Administrador
// Usuario Registrado
// Usuario Anónimo
const String get_rol () const;
void set_rol (String value);

// Additional Private Declarations
//## begin SolicitanteDeInformacion%4DAC50E40308.private preserve=yes
//## end SolicitanteDeInformacion%4DAC50E40308.private

private: //## implementation
// Data Members for Class Attributes

//## begin SolicitanteDeInformacion::nombre%4E6B1D140203.attr
preserve=no private: String {U}
String nombre;
//## end SolicitanteDeInformacion::nombre%4E6B1D140203.attr

//## begin SolicitanteDeInformacion::apellido%4E6B1D26037A.attr
preserve=no private: String {U}
String apellido;
```

```

    /// end SolicitanteDeInformacion::apellido%4E6B1D26037A.attr

    /// begin SolicitanteDeInformacion::identificador%4E6B1D41009C.attr
preserve=no private: String {U}
String identificador;
    /// end SolicitanteDeInformacion::identificador%4E6B1D41009C.attr

    /// begin SolicitanteDeInformacion::clave%4E6B1D5D0119.attr
preserve=no private: String {U}
String clave;
    /// end SolicitanteDeInformacion::clave%4E6B1D5D0119.attr

    /// begin SolicitanteDeInformacion::correo%4E6B1D440177.attr
preserve=no private: String {U}
String correo;
    /// end SolicitanteDeInformacion::correo%4E6B1D440177.attr

    /// begin SolicitanteDeInformacion::rol%4E7A1DE50222.attr
preserve=no private: String {U}
String rol;
    /// end SolicitanteDeInformacion::rol%4E7A1DE50222.attr

// Data Members for Associations

    /// Association: <unnamed>%4F318401016E
    /// begin
SolicitanteDeInformacion::theAlmacenDeConfiguraciones%4F3184050239.role
preserve=no public: AlmacenDeConfiguraciones {1 -> *RHNX}
    UnboundedSetByReference<Gestor_de_Configuraciones>
the_Gestor_de_Configuraciones;
    /// end
SolicitanteDeInformacion::theAlmacenDeConfiguraciones%4F3184050239.role

// Additional Implementation Declarations
    /// begin SolicitanteDeInformacion%4DAC50E40308.implementation
preserve=yes
    /// end SolicitanteDeInformacion%4DAC50E40308.implementation

};

/// begin SolicitanteDeInformacion%4DAC50E40308.postscript preserve=yes
/// end SolicitanteDeInformacion%4DAC50E40308.postscript

// Class SolicitanteDeInformacion

/// Get and Set Operations for Class Attributes (inline)

inline const String SolicitanteDeInformacion::get_nombre () const
{
    /// begin SolicitanteDeInformacion::get_nombre%4E6B1D140203.get
preserve=no
    return nombre;
    /// end SolicitanteDeInformacion::get_nombre%4E6B1D140203.get
}

inline void SolicitanteDeInformacion::set_nombre (String value)
{
    /// begin SolicitanteDeInformacion::set_nombre%4E6B1D140203.set
preserve=no
    nombre = value;
    /// end SolicitanteDeInformacion::set_nombre%4E6B1D140203.set
}

```

```
}

inline const String SolicitanteDeInformacion::get_apellido () const
{
    ///# begin SolicitanteDeInformacion::get_apellido%4E6B1D26037A.get
    preserve=no
    return apellido;
    ///# end SolicitanteDeInformacion::get_apellido%4E6B1D26037A.get
}

inline void SolicitanteDeInformacion::set_apellido (String value)
{
    ///# begin SolicitanteDeInformacion::set_apellido%4E6B1D26037A.set
    preserve=no
    apellido = value;
    ///# end SolicitanteDeInformacion::set_apellido%4E6B1D26037A.set
}

inline const String SolicitanteDeInformacion::get_identificador () const
{
    ///# begin SolicitanteDeInformacion::get_identificador%4E6B1D41009C.get
    preserve=no
    return identificador;
    ///# end SolicitanteDeInformacion::get_identificador%4E6B1D41009C.get
}

inline void SolicitanteDeInformacion::set_identificador (String value)
{
    ///# begin SolicitanteDeInformacion::set_identificador%4E6B1D41009C.set
    preserve=no
    identificador = value;
    ///# end SolicitanteDeInformacion::set_identificador%4E6B1D41009C.set
}

inline const String SolicitanteDeInformacion::get_clave () const
{
    ///# begin SolicitanteDeInformacion::get_clave%4E6B1D5D0119.get
    preserve=no
    return clave;
    ///# end SolicitanteDeInformacion::get_clave%4E6B1D5D0119.get
}

inline void SolicitanteDeInformacion::set_clave (String value)
{
    ///# begin SolicitanteDeInformacion::set_clave%4E6B1D5D0119.set
    preserve=no
    clave = value;
    ///# end SolicitanteDeInformacion::set_clave%4E6B1D5D0119.set
}

inline const String SolicitanteDeInformacion::get_correo () const
{
    ///# begin SolicitanteDeInformacion::get_correo%4E6B1D440177.get
    preserve=no
    return correo;
    ///# end SolicitanteDeInformacion::get_correo%4E6B1D440177.get
}

inline void SolicitanteDeInformacion::set_correo (String value)
{
```

```

    ///## begin SolicitanteDeInformacion::set_correo%4E6B1D440177.set
preserve=no
    correo = value;
    ///## end SolicitanteDeInformacion::set_correo%4E6B1D440177.set
}

inline const String SolicitanteDeInformacion::get_rol () const
{
    ///## begin SolicitanteDeInformacion::get_rol%4E7A1DE50222.get preserve=no
return rol;
    ///## end SolicitanteDeInformacion::get_rol%4E7A1DE50222.get
}

inline void SolicitanteDeInformacion::set_rol (String value)
{
    ///## begin SolicitanteDeInformacion::set_rol%4E7A1DE50222.set preserve=no
rol = value;
    ///## end SolicitanteDeInformacion::set_rol%4E7A1DE50222.set
}

///## Get and Set Operations for Associations (inline)

inline const UnboundedSetByReference<Gestor_de_Configuraciones>
SolicitanteDeInformacion::get_the_Gestor_de_Configuraciones () const
{
    ///## begin
SolicitanteDeInformacion::get_the_Gestor_de_Configuraciones%4F3184050239.ge
t preserve=no
    return the_Gestor_de_Configuraciones;
    ///## end
SolicitanteDeInformacion::get_the_Gestor_de_Configuraciones%4F3184050239.ge
t
}

inline void SolicitanteDeInformacion::set_the_Gestor_de_Configuraciones
(UnboundedSetByReference<Gestor_de_Configuraciones> value)
{
    ///## begin
SolicitanteDeInformacion::set_the_Gestor_de_Configuraciones%4F3184050239.se
t preserve=no
    the_Gestor_de_Configuraciones = value;
    ///## end
SolicitanteDeInformacion::set_the_Gestor_de_Configuraciones%4F3184050239.se
t
}

///## begin module%4DAC50E40308.epilog preserve=yes
///## end module%4DAC50E40308.epilog

#endif

```

5.2.2. Administrador.

```

///## begin module%1.10%.codegen_version preserve=yes
// Read the documentation to learn more about C++ code generator
// versioning.
///## end module%1.10%.codegen_version

```

```
//## begin module%4E836D0002CE.cm preserve=no
//      %X% %Q% %Z% %W%
//## end module%4E836D0002CE.cm

//## begin module%4E836D0002CE.cp preserve=no
//## end module%4E836D0002CE.cp

//## Module: AdministradorSistemas%4E836D0002CE; Pseudo Package
specification
//## Source file: D:\TEMP\C++\AdministradorSistemas.h

#ifndef AdministradorSistemas_h
#define AdministradorSistemas_h 1

//## begin module%4E836D0002CE.additionalIncludes preserve=no
//## end module%4E836D0002CE.additionalIncludes

//## begin module%4E836D0002CE.includes preserve=yes
//## end module%4E836D0002CE.includes

//## begin module%4E836D0002CE.additionalDeclarations preserve=yes
//## end module%4E836D0002CE.additionalDeclarations

//## begin AdministradorSistemas%4E836D0002CE.preface preserve=yes
//## end AdministradorSistemas%4E836D0002CE.preface

//## Class: AdministradorSistemas%4E836D0002CE
//      Persona responsable de asegurar el correcto
//      funcionamiento del sistema.
//
//      Hace referencia al responsable de la aplicación, es el
//      que se encarga de asegurar el correcto funcionamiento
//      del sistema y su mejora.
//## Category: <Top Level>
//## Persistence: Transient
//## Cardinality/Multiplicity: n

//## Uses: <unnamed>%4E836D79030D;AdministradorSistemas { -> }

class AdministradorSistemas
{
    //## begin AdministradorSistemas%4E836D0002CE.initialDeclarations
preserve=yes
    //## end AdministradorSistemas%4E836D0002CE.initialDeclarations

public:
    //## Constructors (generated)
    AdministradorSistemas();

    AdministradorSistemas(const AdministradorSistemas &right);

    //## Constructors (specified)
    //## Operation: AdministradorSistemas%4FD6FC2F0143
    AdministradorSistemas ();

    //## Destructor (generated)
    ~AdministradorSistemas();
};
```



```
//## Assignment Operation (generated)
AdministradorSistemas & operator=(const AdministradorSistemas
&right);

//## Equality Operations (generated)
int operator==(const AdministradorSistemas &right) const;

int operator!=(const AdministradorSistemas &right) const;

//## Other Operations (specified)
//## Operation: enviarAvisoError%4E8373730399
Boolean enviarAvisoError (String error);

//## Operation: Insertar%4E80D00F0000
Boolean Insertar ();

//## Operation: Modificar%4E80D0240196
Boolean Modificar ();

//## Operation: Consultar%4E80D02C0119
Boolean Consultar ();

//## Operation: Eliminar%4E80D0310280
Boolean Eliminar ();

//## Get and Set Operations for Associations (generated)

// Additional Public Declarations
//## begin AdministradorSistemas%4E836D0002CE.public preserve=yes
//## end AdministradorSistemas%4E836D0002CE.public

protected:
// Additional Protected Declarations
//## begin AdministradorSistemas%4E836D0002CE.protected preserve=yes
//## end AdministradorSistemas%4E836D0002CE.protected

private:
//## Get and Set Operations for Class Attributes (generated)

//## Attribute: nombre%4F6718910155
const String get_nombre () const;
void set_nombre (String value);

//## Attribute: apellido%4F6718CC0277
const String get_apellido () const;
void set_apellido (String value);

//## Attribute: identificador%4F6718DB03C9
const String get_identificador () const;
void set_identificador (String value);

//## Attribute: clave%4F6718EF0057
const String get_clave () const;
void set_clave (String value);

//## Attribute: correo%4F6718FF013C
const String get_correo () const;
void set_correo (String value);
```

```
    /// Attribute: rol%4F6719130069
    const String get_rol () const;
    void set_rol (String value);

// Additional Private Declarations
/// begin AdministradorSistemas%4E836D0002CE.private preserve=yes
/// end AdministradorSistemas%4E836D0002CE.private

private: /// implementation
// Data Members for Class Attributes

    /// begin AdministradorSistemas::nombre%4F6718910155.attr
preserve=no private: String {U}
    String nombre;
    /// end AdministradorSistemas::nombre%4F6718910155.attr

    /// begin AdministradorSistemas::apellido%4F6718CC0277.attr
preserve=no private: String {U}
    String apellido;
    /// end AdministradorSistemas::apellido%4F6718CC0277.attr

    /// begin AdministradorSistemas::identificador%4F6718DB03C9.attr
preserve=no private: String {U}
    String identificador;
    /// end AdministradorSistemas::identificador%4F6718DB03C9.attr

    /// begin AdministradorSistemas::clave%4F6718EF0057.attr preserve=no
private: String {U}
    String clave;
    /// end AdministradorSistemas::clave%4F6718EF0057.attr

    /// begin AdministradorSistemas::correo%4F6718FF013C.attr
preserve=no private: String {U}
    String correo;
    /// end AdministradorSistemas::correo%4F6718FF013C.attr

    /// begin AdministradorSistemas::rol%4F6719130069.attr preserve=no
private: String {U}
    String rol;
    /// end AdministradorSistemas::rol%4F6719130069.attr

// Data Members for Associations

// Additional Implementation Declarations
/// begin AdministradorSistemas%4E836D0002CE.implementation
preserve=yes
    /// end AdministradorSistemas%4E836D0002CE.implementation
};

/// begin AdministradorSistemas%4E836D0002CE.postscript preserve=yes
/// end AdministradorSistemas%4E836D0002CE.postscript

// Class AdministradorSistemas

/// Get and Set Operations for Class Attributes (inline)

inline const String AdministradorSistemas::get_nombre () const
{
    /// begin AdministradorSistemas::get_nombre%4F6718910155.get preserve=no
```

```
    return nombre;
    ///## end AdministradorSistemas::get_nombre%4F6718910155.get
}

inline void AdministradorSistemas::set_nombre (String value)
{
    ///## begin AdministradorSistemas::set_nombre%4F6718910155.set preserve=no
    nombre = value;
    ///## end AdministradorSistemas::set_nombre%4F6718910155.set
}

inline const String AdministradorSistemas::get_apellido () const
{
    ///## begin AdministradorSistemas::get_apellido%4F6718CC0277.get
preserve=no
    return apellido;
    ///## end AdministradorSistemas::get_apellido%4F6718CC0277.get
}

inline void AdministradorSistemas::set_apellido (String value)
{
    ///## begin AdministradorSistemas::set_apellido%4F6718CC0277.set
preserve=no
    apellido = value;
    ///## end AdministradorSistemas::set_apellido%4F6718CC0277.set
}

inline const String AdministradorSistemas::get_identificador () const
{
    ///## begin AdministradorSistemas::get_identificador%4F6718DB03C9.get
preserve=no
    return identificador;
    ///## end AdministradorSistemas::get_identificador%4F6718DB03C9.get
}

inline void AdministradorSistemas::set_identificador (String value)
{
    ///## begin AdministradorSistemas::set_identificador%4F6718DB03C9.set
preserve=no
    identificador = value;
    ///## end AdministradorSistemas::set_identificador%4F6718DB03C9.set
}

inline const String AdministradorSistemas::get_clave () const
{
    ///## begin AdministradorSistemas::get_clave%4F6718EF0057.get preserve=no
    return clave;
    ///## end AdministradorSistemas::get_clave%4F6718EF0057.get
}

inline void AdministradorSistemas::set_clave (String value)
{
    ///## begin AdministradorSistemas::set_clave%4F6718EF0057.set preserve=no
    clave = value;
    ///## end AdministradorSistemas::set_clave%4F6718EF0057.set
}

inline const String AdministradorSistemas::get_correo () const
{
    ///## begin AdministradorSistemas::get_correo%4F6718FF013C.get preserve=no
    return correo;
}
```

```

    ///## end AdministradorSistemas::get_correo%4F6718FF013C.get
}

inline void AdministradorSistemas::set_correo (String value)
{
    ///## begin AdministradorSistemas::set_correo%4F6718FF013C.set preserve=no
    correo = value;
    ///## end AdministradorSistemas::set_correo%4F6718FF013C.set
}

inline const String AdministradorSistemas::get_rol () const
{
    ///## begin AdministradorSistemas::get_rol%4F6719130069.get preserve=no
    return rol;
    ///## end AdministradorSistemas::get_rol%4F6719130069.get
}

inline void AdministradorSistemas::set_rol (String value)
{
    ///## begin AdministradorSistemas::set_rol%4F6719130069.set preserve=no
    rol = value;
    ///## end AdministradorSistemas::set_rol%4F6719130069.set
}

///## begin module%4E836D0002CE.epilog preserve=yes
///## end module%4E836D0002CE.epilog

#endif

```

5.2.3. Petición.

```

///## begin module%1.10%.codegen_version preserve=yes
// Read the documentation to learn more about C++ code generator
// versioning.
///## end module%1.10%.codegen_version

///## begin module%4F53BEB002FD.cm preserve=no
// %X% %Q% %Z% %W%
///## end module%4F53BEB002FD.cm

///## begin module%4F53BEB002FD.cp preserve=no
///## end module%4F53BEB002FD.cp

///## Module: Peticion%4F53BEB002FD; Pseudo Package specification
///## Source file: D:\TEMP\C++\Peticion.h

#ifndef Peticion_h
#define Peticion_h 1

///## begin module%4F53BEB002FD.additionalIncludes preserve=no
///## end module%4F53BEB002FD.additionalIncludes

///## begin module%4F53BEB002FD.includes preserve=yes
///## end module%4F53BEB002FD.includes

// Almac\351n de Informaci\363n Inicial
#include "Almacn de Informacin Inicial.h"
// Recomendacion
#include "Recomendacion.h"
///## begin module%4F53BEB002FD.additionalDeclarations preserve=yes

```

```
//## end module%4F53BEB002FD.additionalDeclarations

//## begin Peticion%4F53BEB002FD.preface preserve=yes
//## end Peticion%4F53BEB002FD.preface

//## Class: Peticion%4F53BEB002FD
// El cliente realiza una petición de información,
// aportando una serie de datos.
//## Category: <Top Level>
//## Persistence: Transient
//## Cardinality/Multiplicity: n

class Peticion
{
  //## begin Peticion%4F53BEB002FD.initialDeclarations preserve=yes
  //## end Peticion%4F53BEB002FD.initialDeclarations

  public:
    //## Constructors (generated)
    Peticion();

    Peticion(const Peticion &right);

    //## Constructors (specified)
    //## Operation: Peticion%4FD6F64203AC
    Peticion ();

    //## Destructor (generated)
    ~Peticion();

    //## Assignment Operation (generated)
    Peticion & operator=(const Peticion &right);

    //## Equality Operations (generated)
    int operator==(const Peticion &right) const;

    int operator!=(const Peticion &right) const;

    // Additional Public Declarations
    //## begin Peticion%4F53BEB002FD.public preserve=yes
    //## end Peticion%4F53BEB002FD.public

  protected:
    // Additional Protected Declarations
    //## begin Peticion%4F53BEB002FD.protected preserve=yes
    //## end Peticion%4F53BEB002FD.protected

  private:
    //## Get and Set Operations for Class Attributes (generated)

    //## Attribute: url%4F53BEF603B9
    const String get_url () const;
    void set_url (String value);

    //## Attribute: descripcion%4F683D8C024A
    const String get_descripcion () const;
    void set_descripcion (String value);
```

```

    /// Attribute: Fecha%4F53BF09002E
    const Date get_Fecha () const;
    void set_Fecha (Date value);

    /// Attribute: identificador%4F53BF1F0119
    const String get_identificador () const;
    void set_identificador (String value);

    /// Attribute: rol%4F6719BA02D5
    const String get_rol () const;
    void set_rol (String value);

    // Additional Private Declarations
    /// begin Peticion%4F53BEB002FD.private preserve=yes
    /// end Peticion%4F53BEB002FD.private

private: /// implementation
    // Data Members for Class Attributes

    /// begin Peticion::url%4F53BEF603B9.attr preserve=no private:
String {U}
    String url;
    /// end Peticion::url%4F53BEF603B9.attr

    /// begin Peticion::descripcion%4F683D8C024A.attr preserve=no
private: String {U}
    String descripcion;
    /// end Peticion::descripcion%4F683D8C024A.attr

    /// begin Peticion::Fecha%4F53BF09002E.attr preserve=no private:
Date {U}
    Date Fecha;
    /// end Peticion::Fecha%4F53BF09002E.attr

    /// begin Peticion::identificador%4F53BF1F0119.attr preserve=no
private: String {U}
    String identificador;
    /// end Peticion::identificador%4F53BF1F0119.attr

    /// begin Peticion::rol%4F6719BA02D5.attr preserve=no private:
String {U}
    String rol;
    /// end Peticion::rol%4F6719BA02D5.attr

    // Additional Implementation Declarations
    /// begin Peticion%4F53BEB002FD.implementation preserve=yes
    /// end Peticion%4F53BEB002FD.implementation

};

/// begin Peticion%4F53BEB002FD.postscript preserve=yes
/// end Peticion%4F53BEB002FD.postscript

// Class Peticion

/// Get and Set Operations for Class Attributes (inline)

inline const String Peticion::get_url () const
{
    /// begin Peticion::get_url%4F53BEF603B9.get preserve=no
    return url;

```

```
    ///## end Peticion::get_url%4F53BEF603B9.get
}

inline void Peticion::set_url (String value)
{
    ///## begin Peticion::set_url%4F53BEF603B9.set preserve=no
    url = value;
    ///## end Peticion::set_url%4F53BEF603B9.set
}

inline const String Peticion::get_descripcion () const
{
    ///## begin Peticion::get_descripcion%4F683D8C024A.get preserve=no
    return descripcion;
    ///## end Peticion::get_descripcion%4F683D8C024A.get
}

inline void Peticion::set_descripcion (String value)
{
    ///## begin Peticion::set_descripcion%4F683D8C024A.set preserve=no
    descripcion = value;
    ///## end Peticion::set_descripcion%4F683D8C024A.set
}

inline const Date Peticion::get_Fecha () const
{
    ///## begin Peticion::get_Fecha%4F53BF09002E.get preserve=no
    return Fecha;
    ///## end Peticion::get_Fecha%4F53BF09002E.get
}

inline void Peticion::set_Fecha (Date value)
{
    ///## begin Peticion::set_Fecha%4F53BF09002E.set preserve=no
    Fecha = value;
    ///## end Peticion::set_Fecha%4F53BF09002E.set
}

inline const String Peticion::get_identificador () const
{
    ///## begin Peticion::get_identificador%4F53BF1F0119.get preserve=no
    return identificador;
    ///## end Peticion::get_identificador%4F53BF1F0119.get
}

inline void Peticion::set_identificador (String value)
{
    ///## begin Peticion::set_identificador%4F53BF1F0119.set preserve=no
    identificador = value;
    ///## end Peticion::set_identificador%4F53BF1F0119.set
}

inline const String Peticion::get_rol () const
{
    ///## begin Peticion::get_rol%4F6719BA02D5.get preserve=no
    return rol;
    ///## end Peticion::get_rol%4F6719BA02D5.get
}

inline void Peticion::set_rol (String value)
{

```

```

    //## begin Peticion::set_rol%4F6719BA02D5.set preserve=no
    rol = value;
    //## end Peticion::set_rol%4F6719BA02D5.set
}

```

```

//## begin module%4F53BEB002FD.epilog preserve=yes
//## end module%4F53BEB002FD.epilog

```

```

#endif

```

5.2.4. Cola de peticiones de información.

```

//## begin module%1.10%.codegen_version preserve=yes
//  Read the documentation to learn more about C++ code generator
//  versioning.
//## end module%1.10%.codegen_version

//## begin module%4DAC4A6E0255.cm preserve=no
//  %X% %Q% %Z% %W%
//## end module%4DAC4A6E0255.cm

//## begin module%4DAC4A6E0255.cp preserve=no
//## end module%4DAC4A6E0255.cp

//## Module: ColaDePeticionesDeInformaci\363n%4DAC4A6E0255; Pseudo Package
specification
//## Source file: D:\TEMP\C++\ColaDePeticionesDeInformacin.h

#ifndef ColaDePeticionesDeInformacin_h
#define ColaDePeticionesDeInformacin_h 1

//## begin module%4DAC4A6E0255.additionalIncludes preserve=no
//## end module%4DAC4A6E0255.additionalIncludes

//## begin module%4DAC4A6E0255.includes preserve=yes
//## end module%4DAC4A6E0255.includes

//## begin module%4DAC4A6E0255.additionalDeclarations preserve=yes
//## end module%4DAC4A6E0255.additionalDeclarations

//## begin ColaDePeticionesDeInformaciXn%4DAC4A6E0255.preface preserve=yes
//## end ColaDePeticionesDeInformaciXn%4DAC4A6E0255.preface

//## Class: ColaDePeticionesDeInformaci\363n%4DAC4A6E0255
//  Registro de la información solicitada a los proveedores
//  y que aún no ha sido atendida.
//
//  Registro de las llegadas de información solicitada a los
//  proveedores o peticiones de los solicitantes de
//  información y que aún no han sido atendidas. ¿FIFO o
//  LIFO?
//## Category: <Top Level>
//## Persistence: Transient
//## Cardinality/Multiplicity: n

//## Uses: <unnamed>%4F67530802B1;Geolocalizador { -> }

```



```
class ColaDePeticionesDeInformaciXn
{
  ///## begin ColaDePeticionesDeInformaciXn%4DAC4A6E0255.initialDeclarations
  preserve=yes
  ///## end ColaDePeticionesDeInformaciXn%4DAC4A6E0255.initialDeclarations

  public:
    ///## Constructors (generated)
    ColaDePeticionesDeInformaciXn();

    ColaDePeticionesDeInformaciXn(const ColaDePeticionesDeInformaciXn
&right);

    ///## Constructors (specified)
    ///## Operation: ColaDePeticionesDeInformaci\363n%4FD70B02012B
    ColaDePeticionesDeInformaciXn ();

    ///## Destructor (generated)
    ~ColaDePeticionesDeInformaciXn();

    ///## Assignment Operation (generated)
    ColaDePeticionesDeInformaciXn & operator=(const
ColaDePeticionesDeInformaciXn &right);

    ///## Equality Operations (generated)
    int operator==(const ColaDePeticionesDeInformaciXn &right) const;

    int operator!=(const ColaDePeticionesDeInformaciXn &right) const;

    ///## Other Operations (specified)
    ///## Operation: a\361adirPeticionInformacionCola%4DAC55C20147
    ///## Preconditions:
    //   La cola de peticiones debe estar disponible y los datos
    //   de la socilidad en formato correcto
    ///## Postconditions:
    //   Queda añadida una nueva petición en la cola en el orden
    //   de prioridad correcto
    Boolean añadirPeticionInformacionCola (String source, String url,
String etiqueta, String usuario, Integer prioridad);

    ///## Operation: lanzarPeticion%4E25BBD30157
    Boolean lanzarPeticion (String url // Direccion de conexion a las
redes sociales
);

    ///## Operation: peticionNoAtendida%4E25C1D10157
    Boolean peticionNoAtendida (Boolean atendida = FALSE);

    ///## Operation: peticionAtendida%4E25D1A40203
    Boolean peticionAtendida (Boolean atendida = TRUE);

    ///## Operation: Eliminar%4E83792200FA
    Boolean Eliminar ();

    ///## Operation: Insertar%4E8378F501F4
    Boolean Insertar ();

    ///## Operation: Consultar%4E83788D033C
    Boolean Consultar ();
```

```
//## Get and Set Operations for Associations (generated)

// Additional Public Declarations
//## begin ColaDePeticionesDeInformaciXn%4DAC4A6E0255.public
preserve=yes
//## end ColaDePeticionesDeInformaciXn%4DAC4A6E0255.public

protected:
// Additional Protected Declarations
//## begin ColaDePeticionesDeInformaciXn%4DAC4A6E0255.protected
preserve=yes
//## end ColaDePeticionesDeInformaciXn%4DAC4A6E0255.protected

private:
//## Get and Set Operations for Class Attributes (generated)

//## Attribute: FechayHora%4E837944006D
const Date get_FechayHora () const;
void set_FechayHora (Date value);

//## Attribute: URL%4E8378AC00DA
const String get_URL () const;
void set_URL (String value);

//## Attribute: Atendida%4E837972008C
const Boolean get_Atendida () const;
void set_Atendida (Boolean value);

//## Attribute: rol%4F671BF30309
const String get_rol () const;
void set_rol (String value);

// Additional Private Declarations
//## begin ColaDePeticionesDeInformaciXn%4DAC4A6E0255.private
preserve=yes
//## end ColaDePeticionesDeInformaciXn%4DAC4A6E0255.private

private: //## implementation
// Data Members for Class Attributes

//## begin
ColaDePeticionesDeInformaciXn::FechayHora%4E837944006D.attr preserve=no
private: Date {U}
Date FechayHora;
//## end ColaDePeticionesDeInformaciXn::FechayHora%4E837944006D.attr

//## begin ColaDePeticionesDeInformaciXn::URL%4E8378AC00DA.attr
preserve=no private: String {U}
String URL;
//## end ColaDePeticionesDeInformaciXn::URL%4E8378AC00DA.attr

//## begin ColaDePeticionesDeInformaciXn::Atendida%4E837972008C.attr
preserve=no private: Boolean {U}
Boolean Atendida;
//## end ColaDePeticionesDeInformaciXn::Atendida%4E837972008C.attr

//## begin ColaDePeticionesDeInformaciXn::rol%4F671BF30309.attr
preserve=no private: String {U}
String rol;
```

```
    /// end ColaDePeticonesDeInformaciXn::rol%4F671BF30309.attr

// Data Members for Associations

// Additional Implementation Declarations
/// begin ColaDePeticonesDeInformaciXn%4DAC4A6E0255.implementation
preserve=yes
/// end ColaDePeticonesDeInformaciXn%4DAC4A6E0255.implementation
};

/// begin ColaDePeticonesDeInformaciXn%4DAC4A6E0255.postscript
preserve=yes
/// end ColaDePeticonesDeInformaciXn%4DAC4A6E0255.postscript

// Class ColaDePeticonesDeInformaciXn

/// Get and Set Operations for Class Attributes (inline)

inline const Date ColaDePeticonesDeInformaciXn::get_FechayHora () const
{
    /// begin ColaDePeticonesDeInformaciXn::get_FechayHora%4E837944006D.get
preserve=no
    return FechayHora;
    /// end ColaDePeticonesDeInformaciXn::get_FechayHora%4E837944006D.get
}

inline void ColaDePeticonesDeInformaciXn::set_FechayHora (Date value)
{
    /// begin ColaDePeticonesDeInformaciXn::set_FechayHora%4E837944006D.set
preserve=no
    FechayHora = value;
    /// end ColaDePeticonesDeInformaciXn::set_FechayHora%4E837944006D.set
}

inline const String ColaDePeticonesDeInformaciXn::get_URL () const
{
    /// begin ColaDePeticonesDeInformaciXn::get_URL%4E8378AC00DA.get
preserve=no
    return URL;
    /// end ColaDePeticonesDeInformaciXn::get_URL%4E8378AC00DA.get
}

inline void ColaDePeticonesDeInformaciXn::set_URL (String value)
{
    /// begin ColaDePeticonesDeInformaciXn::set_URL%4E8378AC00DA.set
preserve=no
    URL = value;
    /// end ColaDePeticonesDeInformaciXn::set_URL%4E8378AC00DA.set
}

inline const Boolean ColaDePeticonesDeInformaciXn::get_Atendida () const
{
    /// begin ColaDePeticonesDeInformaciXn::get_Atendida%4E837972008C.get
preserve=no
    return Atendida;
    /// end ColaDePeticonesDeInformaciXn::get_Atendida%4E837972008C.get
}

inline void ColaDePeticonesDeInformaciXn::set_Atendida (Boolean value)
{
```

```
    ///## begin ColaDePeticionesDeInformaciXn::set_Atendida%4E837972008C.set
preserve=no
    Atendida = value;
    ///## end ColaDePeticionesDeInformaciXn::set_Atendida%4E837972008C.set
}

inline const String ColaDePeticionesDeInformaciXn::get_rol () const
{
    ///## begin ColaDePeticionesDeInformaciXn::get_rol%4F671BF30309.get
preserve=no
    return rol;
    ///## end ColaDePeticionesDeInformaciXn::get_rol%4F671BF30309.get
}

inline void ColaDePeticionesDeInformaciXn::set_rol (String value)
{
    ///## begin ColaDePeticionesDeInformaciXn::set_rol%4F671BF30309.set
preserve=no
    rol = value;
    ///## end ColaDePeticionesDeInformaciXn::set_rol%4F671BF30309.set
}

///## begin module%4DAC4A6E0255.epilog preserve=yes
///## end module%4DAC4A6E0255.epilog

#endif
```