

Universidad de las Ciencias Informáticas

Facultad 7



Título: Implementación del Módulo de Búsqueda del Sistema de Monitoreo y Análisis de Noticias

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

Autor: Adniel Quintana Muñoz

Tutora: Ing. Daneysi Granado Rodríguez

Asesor: Ing. Eduardo Manuel Macías Sotolongo

Ciudad de la Habana, Junio 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 17 días del mes de junio del año 2009.

Adniel Quintana Muñoz

Autor

Daneysi Granado Rodríguez

Tutora

**"La mayoría de las ideas fundamentales de la ciencia son esencialmente sencillas
y, por regla general pueden ser expresadas en un lenguaje comprensible
para todos."**

Albert Einstein.

AGRADECIMIENTOS

QUIERO AGRADECER A MI MADRE, POR HABERSE HECHO CARGO DE MÍ DURANTE TODA MI VIDA, CON EL ÚNICO DESEO DE VERME CONVERTIDO EN UN PROFESIONAL.

A MI FAMILIA, A YAUMARA Y EDDY QUE SON PARTE DE LA FAMILIA.

A MI PADRE, QUE CONVERSANDO CONMIGO, ME DIO CONSEJOS QUE ME AYUDARON MUCHO EN MI VIDA COMO ESTUDIANTE.

A NEKY, LA CUAL VINE A CONOCER EN MI ÚLTIMO AÑO DE LA CARRERA, PERO QUE EN CAMBIO SUPO COMPRENDERME CUANDO QUIZÁS POCOS LO HACÍAN, A TI MI VIDA MUCHAS GRACIAS POR ESTAR A MI LADO TODO EL TIEMPO BRIDÁNDOME TU APOYO.

A TODOS MIS COMPAÑEROS DE AULA.

A MIS HERMANOS NADIER Y RUBÉN, LOS CONSIDERO ASÍ PORQUE SON LOS HERMANOS DE SANGRE QUE NUNCA TUVE.

A MI TUTORA, QUE MUCHAS VECES SE ACOSTÓ TARDE EN LA NOCHE POR REVISAR MI TRABAJO PARA MÁS TARDE DECIRME LOS ERRORES Y AYUDARME A CORREGIRLOS.

A LA REVOLUCIÓN, A FIDEL, A RAÚL, POR HABER PROPICIADO QUE NACIERA, ME CRIARA Y ME FORMARA EN UNA PATRIA LIBRE Y SOBERANA COMO LA NUESTRA.

GRACIAS A DIOS QUE LLEGUÉ A LA META PORQUE MUCHAS VECES LA VI INALCANZABLE. MENCIONO A DIOS PORQUE PIENSO QUE UN HOMBRE SIN CREENCIAS SIMPLEMENTE VIVE SIN HALLARLE SENTIDO A SU VIDA.

DEDICATORIA

LE DEDICO ESTE TRABAJO A MI MADRE, PORQUE SÉ QUE ELLA TAMBIÉN SIENTE ESTE TRABAJO COMO SUYO.

A TODA MI FAMILIA QUE NO VEÍAN EL DÍA QUE ME GRADUARA.

A NEKÝ YA QUE EN OCASIONES HASTA ME CORREGISTE MIS ERRORES DE CÓDIGO.

A MI TUTORA POR SU PACIENCIA Y COMPRENSIÓN.

A TODOS MIS AMIGOS POR DARME SU APOYO INCONDICIONAL CUANDO MÁS FALTA ME HIZO.

EN FIN A TODO EL QUE DE UNA FORMA U OTRA HIZO POSIBLE LA REALIZACIÓN DE ESTE TRABAJO. Y POR SUPUESTO A TODO EL QUE LE INTERESE.

RESUMEN

Internet es uno de los principales medios de información que existe en la actualidad, pero cuenta con un gran volumen de información, muy dispersa. Por ello, el monitoreo de noticias y las herramientas que sirven de soporte a dicha actividad son de vital importancia. En la Universidad de las Ciencias Informáticas, el proyecto GIDI (Grupo de Investigación y Desarrollo sobre Internet) cuenta con un grupo de vigilancia que requiere de una herramienta para informatizar el proceso de búsqueda.

El presente trabajo de diploma tiene como objetivo: Implementar un componente que permita la búsqueda mediante canales de sindicación en el Sistema de Monitoreo y Análisis de Noticias (SIMAN).

La forma más eficiente de recibir los contenidos es a través de los canales de sindicación: archivos XML con el extracto de la información del sitio en cuestión. El componente de búsqueda para el Sistema de Monitoreo y Análisis de Noticias constituye un lector RSS que permite la búsqueda, edición y almacenamiento de noticias para su posterior análisis.

La aplicación fue desarrollada con Java como lenguaje de programación sobre el IDE de desarrollo NetBean 6.0. Así como, DB4O (Data Base for Object) que es el motor de base de datos escogido, por su alto rendimiento y bajo consumo de recursos.

El componente de búsqueda desarrollado para el Sistema de Análisis y Monitoreo de Noticias permite la búsqueda, edición y almacenamiento del extracto de las noticias publicadas en los canales de sindicación de los medios de información seleccionados por el usuario.

Palabras claves

Monitoreo de noticias, Bases de datos orientadas a objetos, DB4O, lectores RSS, canales de sindicación.

ÍNDICE

INTRODUCCIÓN.....	1
Capítulo 1. Fundamentación Teórica.....	11
1.1 Conceptos relacionados con el tema	11
1.2. Antecedentes.....	16
1.3. Canales de Sindicación.	29
1.4. ¿Qué es la Programación?	34
1.5. Entorno de Desarrollo Netbeans 6.0.....	44
1.6. Base de datos	45
1.7. Tecnologías a Utilizar	49
1.8. Metodologías de Desarrollo de Software	49
1.9. Fundamentación de la solución propuesta.....	64
Capítulo 2. Elementos de la Arquitectura.....	66
2.1. Requerimientos No Funcionales.....	66
2.2. Descripción de la Arquitectura. Fundamentación.....	70
2.3. Análisis de posibles implementaciones, componentes o módulos ya existentes.....	74
2.4. Análisis de las estructuras de datos utilizadas en la solución	79
2.5. Aspectos técnicos del DB4O.....	80
2.6. Vista de Despliegue	87
2.7. Estilos de Programación	90
2.8. Estándar de Codificación	100
Capítulo 3. Descripción y Análisis de la solución propuesta.....	104
3.1. Modelo de Diseño.....	104
3.2. Diagrama de clases del Diseño.....	104

3.3. Descripción de las Clases.....	107
3.4. Diagrama de Interacción.....	112
3.5. Diseño de la Base de Datos.....	124
3.6. Diagrama de Componentes	136
CONCLUSIONES.....	139
RECOMENDACIONES	140
BIBLIOGRAFÍA.....	141
4. ANEXOS	149
4.1 Anexo 1. Descripción de las Clases del Diseño	149
GLOSARIO DE TÉRMINOS.....	163

INTRODUCCIÓN

En los últimos años, han ocurrido grandes transformaciones y reformas en el campo de las tecnologías a nivel mundial. Con el surgimiento de Internet, grandes volúmenes de información son generados cada día. Por lo que la búsqueda de un determinado tema puede resultar infinita si no se sabe dónde o cómo buscar. Además, la facilidad de publicación con la que cuentan los cibernautas, provoca que cualquier persona pueda publicar artículos sin ningún basamento ni rigor científico. Por lo que la lectura en el ciberespacio se convierte en una actividad de confirmación continua de lo que se lee, o de lo contrario el lector podría ser desinformado.

Para evitar estos errores que pueden ser fatales si afectan la toma de decisiones, han ganado auge actividades como: la vigilancia tecnológica, el monitoreo y la observación de temas específicos. Estos pueden ser: noticias, tecnologías o tendencias del mercado. Estas actividades siguen el mismo objetivo: facilitar la toma de decisiones a la hora de trazar estrategias ya sean económicas, políticas o sociales. Lo que está en dependencia del tema en cuestión y todas requieren de atención detallada y continua.

El monitoreo es una actividad que consiste en realizar análisis tanto cualitativo como cuantitativo de un determinado tema de interés, esta actividad puede tener distintos tipos de duración ya sean a corto, mediano o largo plazo. Todo depende de la prioridad del tema que se esté monitoreando. Por supuesto se realiza sobre la base de información precisa y acertada, por lo que la elección de los medios a monitorear debe ser bien analizada y detallada, para garantizar la transparencia del proceso.

Existen tantos tipos de monitoreo como temas a monitorear. En el caso del monitoreo de noticias, se enfoca en el seguimiento de medios noticiosos. Los que constituyen la forma más rápida de analizar tendencias del mercado, así como la repercusión de un determinado producto. En fin, se puede hacer seguimiento de temas tanto científico-técnicos, como sociales o políticos, todo en dependencia de lo que se necesita.

Hoy en día, una de la vía más rápida para acceder a una información en los monitoreo de noticias, son canales de sindicación presentes en casi todos los medios noticiosos. Mediante los mismos, la notificación de actualización llega mucho más rápido a los usuarios, y sin necesidad de navegar en la web. Lo que es posible mediante la utilización de programas informáticos especializados en estas cuestiones, conocidos como agregadores.

Los observatorios constituyen un órgano rector que se dedica a realizar seguimientos de temas en específico, ya sean tecnológicos, sociales o políticos. Como resultado de este publican una serie de informes estadísticos, boletines, a los cuales pueden acceder sus miembros o usuarios registrados. Todo depende de los niveles de seguridad que se les hayan implementado al observatorio, los cuales aparecen generalmente en forma de portales web. Incluso pueden formar parte de un sistema de vigilancia tecnológica.

La vigilancia tecnológica (de ahora en adelante se va a hacer referencia a la vigilancia tecnológica como VT) no es algo nuevo. Es una actividad que se realiza desde hace muchas centurias, quizás bajo otros nombres en dependencia de la etapa en que se enmarque. Data desde que antiguos emperadores y reyes, mandaban a espiar a otras regiones del mundo. A sus oídos llegaban informaciones referentes al desarrollo de otras regiones, y por supuesto ellos querían llevar ese desarrollo a sus dominios.

Así ocurrió en el Japón de la dinastía Toh (siglos VII al IX) que se enviaron más de una decena de misiones de estudio a Choan, China, entonces probablemente la ciudad más desarrollada e internacional del mundo para captar información sobre su avance. De igual manera, la dinastía japonesa Meiji decidió transformar su modo de desarrollo no por simple fascinación de sus élites respecto a la revolución industrial occidental, sino para preservar su independencia.

Hoy en día muchos confunden la VT con el espionaje industrial. Lo que está errado ya que la VT es una actividad en la cual se capta, se analiza la información referente a un determinado tema. Esta información se les hace llegar a los niveles superiores para

que ellos en conjunto con sus asesores tomen las decisiones pertinentes y tracen una estrategia a seguir, con la mínima posibilidad de riesgo. Para lograr esto, no hay necesidad de sobrepasar los límites legales ni de salirse de los planos éticos por los que se rige un profesional.

Es por esta razón que en el mundo las empresas estilan por darles cursos de superación a sus empleados sobre temas de VT. Así como sobre las diversas técnicas y métodos que sigue para obtener información valiosa. Por supuesto, se les deja bien claro que información es vital para la institución y que no deben saber otras instituciones que representen un potencial peligro económico para ellos, debido a la competencia existente. Por lo tanto, se forma un personal capacitado para realizar la vigilancia y para evitar caer en ella.

Por otra parte, este personal en muchas situaciones no cuenta con las herramientas adecuadas, o con herramientas que no tienen todas las funcionalidades requeridas para facilitar la VT lo que complejiza la tarea y aumenta la probabilidad de errores u omisiones en la información obtenida. Lo anterior pone de manifiesto que la preparación del personal no lo es todo, si existiese una herramienta que automatice la actividad se podrían obtener mejores resultados y minimizar los errores humanos.

Por esta razón cada día se hace más necesario que las actividades de VT tengan un enfoque más automatizado. Que permita avanzar en cuanto a rapidez y calidad en el trabajo. Para lograrlo muchas empresas e instituciones tanto gubernamentales como no gubernamentales, se han trazado diversas estrategias.

En el caso de Japón, este país destina el 1,5% de sus ventas a la vigilancia tecnológica. Quizás se piense que este país destina demasiados recursos a esta actividad, pero quizás lo más sorprendente es que muchos países en el mundo pierden cuantiosos recursos por no realizar adecuadamente esta actividad.

Lo anterior se confirma en el hecho de que las empresas europeas pierden al año alrededor de 20 000 millones de dólares norteamericanos por innovar y desarrollar

productos que ya están patentados. Lo que trae consigo el abandono del proyecto y por supuesto la pérdida de todos los recursos destinados para el trabajo. Además, el tiempo empleado en esta actividad no se va a recuperar y puede ser aprovechado al máximo por empresas de la competencia, lo que significa un atraso considerable para una institución.

Ante esta situación, muchos países se han creado una especie de guías de prácticas a seguir por sus empresas, las cuales pueden consultar muchas veces gratis otras muchas no. Pero que al final logran reducir las pérdidas de éstas y por supuesto el consiguiente alivio a la economía no solo a nivel de empresa sino a nivel nacional. Es por esto que los cuatro mejores modelos a seguir en el mundo sean:

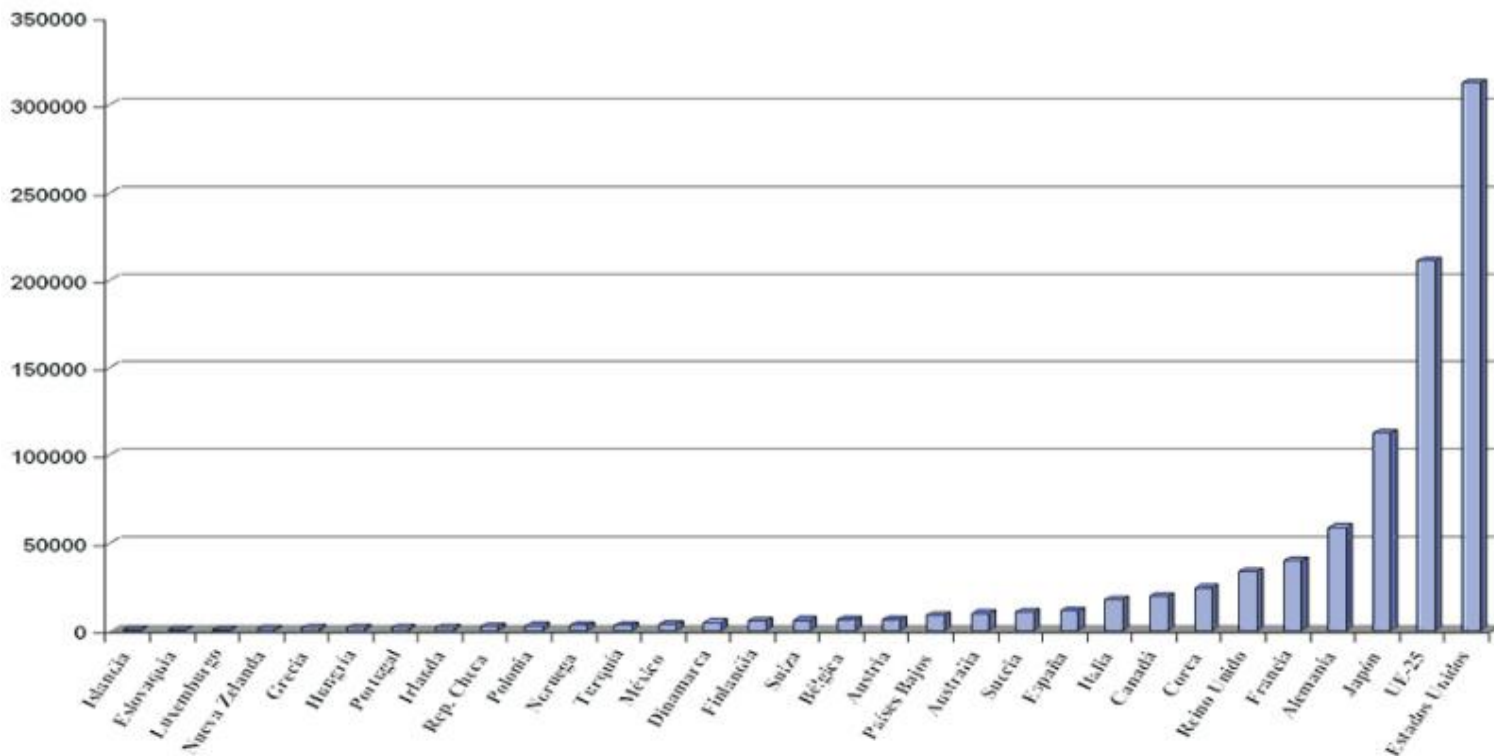
- **Modelo Francés:** El gobierno francés ha impulsado diferentes modos para fomentar la función de inteligencia. En 1988, el CNRS (Centre National de la Recherche Scientifique) creó el INIST (Institut de l'Information Scientifique et Technique) con la misión de recoger y tratar los resultados de la investigación científica mundial para hacerlos accesibles a la comunidad investigadora. Entre sus múltiples actividades, el INIST produce las bases de datos Pascal (tiene acceso a más de tres millones de registros de ciencias, tecnología y medicina) y Francis (economía y ciencias sociales) y presta especial atención a las actividades de Vigilancia Tecnológica. Cuenta con una plantilla superior a 400 personas. Así como agregados tecnológicos en red mundial de oficinas comerciales.
- **Modelo de EE.UU:** este contiene ofertas masivas de información tecnológica desde agencias federales, articulación de un poderoso asociacionismo privado con: Society of Competitive Intelligence Professionals (SCIP) cuyos objetivos son fundamentalmente establecer y promover la inteligencia competitiva (la cual es un tipo de vigilancia tecnológica) como una profesión, favorecer el desarrollo profesional de sus miembros y anticiparse a sus intereses. Entre sus actividades se encuentran la organización de cursos, encuentros internacionales, conferencias, etc. Además de publicar una revista de forma cuatrimestral,

llamada Competitive Intelligence Review (la primera revista internacional dedicada a este tema).

- **Modelo Sueco:** Se realizan conferencias que reúnen a los grandes grupos industriales y a la administración y organizan debates sobre el sistema nacional de inteligencia y acciones de colaboración.
- **Modelo Japonés:** A finales de los años 50 este gobierno creó dos organizaciones para fomentar la adquisición tecnológica: el Japan Information Center of Science and Technology (JICST), principal entidad gubernamental responsable de obtener y difundir información sobre tecnología extranjera, a través del estudio de patentes y revistas técnicas; y el Japan External Trade Organization (JETRO), que estableció el célebre Ministry of International Trade and Industry (MITI), para obtener y transmitir información comercial sobre empresas nacionales y extranjeras, fomentar exportaciones, apoyar acuerdos de transferencia tecnológica e identificar los mecanismos que emplean otras naciones para gestionar sus negocios y economía.

Es cierto que estos países invierten mucho en investigación y desarrollo pero estos gastos están íntimamente relacionados con los procesos de Vigilancia Tecnológica. Puesto que existe un paralelismo entre ambos y un objetivo común: aumentar la productividad empresarial. La gráfica a continuación muestra los gastos más significativos en investigación y desarrollo a nivel mundial en el año 2004.

Gastos en I+D (millones de dólares) durante 2004



En Cuba estas actividades son asumidas por muy pocas empresas, como es el caso de todas las filiales del Instituto de Información Científico y Tecnológica (IDICT), distribuidas a lo largo de todo el territorio nacional. Conocidas como Centro de Información y Gestión Tecnológica (CIGET). Estas tienen como misión satisfacer la demanda en sus territorios de resultados de la ciencia asociados a la información, la tecnología, el conocimiento y el medio ambiente, mediante la prestación de servicios científico-tecnológicos de este tipo, con la adecuada actualización y de acuerdo con la especificidad de cada cliente.

Además existe Consultorías Biomundi (la cual es la dirección de inteligencia corporativa del IDICT) que se encarga de hacer VT en el campo de las Vacunas Humanas, a instituciones de prestigio como el Centro de Ingeniería Genética y Biotecnología (CIGB), Centro de Inmunología Molecular (CIM), Instituto "Finlay", aunque también

provee de asesoramiento empresarial a otros ministerios como el MINFAR (Ministerio de las Fuerzas Armadas Revolucionaria), MINAGRI (Ministerio de la Agricultura) entre otros.

De manera particular, algunas entidades brindan servicios especializados de búsquedas de información o la elaboración de compendios informativos, tal es el caso de las Consultorías DELFOS y AVANTE, entidades especializadas del Ministerio de la Informática y las Telecomunicaciones en la capital de Cuba. Sumado a ello, la Oficina Cubana de la Propiedad Industrial (OCPI) ofrece búsquedas temáticas relacionadas con patentes de invención y el estado del arte para temáticas específicamente solicitadas.

Dentro de los esfuerzos que Cuba por elevar los niveles de informatización asociados al suministro de información dirigido al sector empresarial, se ha estructurado y puesto en marcha la creación de la Red de Información Industrial de Cuba (RIIC), que ya muestra su primer producto, el Portal de la Industria Cubana (www.cubaindustria.cu), que también posee enlace con el sitio web dedicado a La Nueva Empresa en Cuba (www.nuevaempresa.cu). Esta red está integrada por los ministerios e instituciones nacionales vinculados con los sectores Industrial, Financiero y Bancario.

Otras empresas, utilizan métodos un poco más sencillo como: la empresa “TRD Caribe” (creada 1994 para satisfacer necesidades materiales de consumo directo por parte de la población) estableció la cuenta de correo electrónico “SINCO” (Sistema Inteligencia Competitiva) en la red de la cadena, que constituye el canal más importante de información, porque permite que todos los usuarios de la red, incluidas las tiendas, envíen y reciban información proveniente de distintas fuentes: de los trabajadores de la cadena, de los clientes, de la competencia, de proveedores, publicaciones, internet, etc.

Incluso en la Universidad de las Ciencias Informáticas (UCI), universidad creada para el impulso de la industria del software en Cuba, no cuenta con un sistema adecuado de vigilancia tecnológica. Actividad que actualmente la realizan dos trabajadoras de la Biblioteca de la UCI. La cual realizan de forma manual. Trayendo consigo el respectivo

atraso de un trabajo, de un informe o de cualquier artefacto generado por la realización de la vigilancia.

En esta universidad existe un proyecto conocido como Grupo de Investigación y Desarrollo de Internet (GIDI). A la línea de Vigilancia Tecnológica y Política perteneciente al proyecto mencionado anteriormente, siempre se le asignan actividades de levantamiento de información, las cuales generalmente traen consigo el seguimiento de noticias en la Web.

Este trabajo se realiza de forma totalmente manual, sin una herramienta que facilite la actividad de búsqueda y procesamiento de noticias. La organización del trabajo se hace bastante difícil, se reparte el personal en varios turnos de trabajo. Lo que trae consigo que exista al menos un editor por turno, encargado de generar un informe final de toda la información que se ha logrado recopilar. Muchas veces estas informaciones se repiten en los distintos informes que se generan. Ya que los editores trabajan en turnos diferentes y no saben cuales fueron las noticias que se generaron en el informe anterior.

Y la realidad es que no existe un sistema en el país, ni siquiera una herramienta que trabaje sobre la base de la vigilancia o el monitoreo y que se pueda adaptar a las necesidades requeridas; sintiéndose esta falta también en la universidad donde aún no se a socializado mucho este novedoso tema.

Mientras que el monitoreo de noticias en Cuba es algo que se realiza en los medios de comunicación, como la prensa escrita, la radio o la televisión aunque los profesionales también han tenido que hacer algún tipo de monitoreo de noticias, en el seguimiento de un tema de interés, ya sea para su trabajo o para su cultura general. Por supuesto todo esto sin una herramienta adecuada, que agilice la búsqueda y la gestión de la información en general.

Cosa que podría dársele solución al combinar la vigilancia tecnológica con el monitoreo y los observatorios mediante los canales de sindicación existentes, dando a lugar a una herramienta capaz de automatizar y hacer más eficiente esta importante labor.

Labor que no solo se realiza en base de datos especializadas, y en base de datos de patentes, sino también que se puede realizar en los distintos medios noticiosos. Ya que una noticia no solo trata temas políticos y sociales, sino también científicos-técnicos. Además de contar con los canales de sindicación mencionados anteriormente. Característica que debe ser explotada al máximo por sus grandes ventajas en cuanto a rapidez y eficiencia se trata.

Después de haber hecho un profundo análisis de la **Situación Problemática** existente, y ante la necesidad de encontrar una solución para dar respuesta a estas dificultades, se plantea resolver el siguiente **problema**: ¿Cómo obtener un sistema funcional a partir de los requerimientos identificados en el proceso de búsqueda mediante canales de sindicación en el Sistema de Monitoreo y Análisis de Noticias (SIMAN)?

En correspondencia con el problema a resolver planteado se define como **Objeto de estudio**: Los sistemas de Monitoreo y Análisis de Noticias. Restringiendo el **Campo de acción** a: Los procesos de búsqueda de información mediante canales de sindicación.

Planteando como **Objetivo de la investigación**: Implementar un componente que permita la búsqueda mediante canales de sindicación en el Sistema de Monitoreo y Análisis de Noticias (SIMAN).

Para cumplir con el objetivo planteado, se deben cumplir con las siguientes **tareas de la investigación**.

1. Valorar las nuevas tendencias de las tecnologías de la información relacionadas con los canales de sindicación y lectores RSS¹.
2. Analizar la tecnología a utilizar.
3. Analizar el diseño del sistema y valorar cambios.
4. Definir el estándar de codificación a utilizar.
5. Diseñar e implementar la Base de datos.
6. Desarrollar los artefactos establecidos para el flujo de trabajo Implementación.

¹ Son herramientas capaces de leer el contenido de un canal de sindicación, y traducirlo a un lenguaje entendible por el usuario.

7. Implementar el componente.

Este trabajo propone como solución a las dificultades analizadas, realizar un software que permita buscar información mediante los medios de sindicación y almacenarla en una base de datos para su posterior análisis y utilización.

El desarrollo de este sistema, representa un paso de avance en campos como la vigilancia tecnológica, la observación y el monitoreo. Como el proyecto se encuentra dentro del Grupo de Investigación y Desarrollo de Internet (GIDI), permitirá a los estudiantes que pertenecen al grupo de vigilancia política y tecnológica realizar la búsqueda de información de forma automática y posteriormente realizar un análisis de dicha información.

Capítulo 1. Fundamentación Teórica.

En este capítulo se valoran los aspectos teóricos que fundamentan la implementación del módulo propuesto. Se explican y justifican las tecnologías y herramientas usadas para su desarrollo. Así como las técnicas de programación empleadas para solucionar diversas situaciones. También se da una panorámica de las herramientas existentes utilizadas en la vigilancia tecnológica, en el monitoreo de noticias, y de observatorios temáticos existentes y se explican conceptos básicos para un mejor entendimiento del contenido de este trabajo.

1.1 Conceptos relacionados con el tema.

XML: (*Extensible Markup Language*) parecido al HTML de las páginas web, es el lenguaje en el que está descrito técnicamente el contenido de los canales de sindicación. Creado por la W3C (Word Wide Web Consortium).

Se trata de un metalenguaje que ordena, estructura y describe los documentos de las páginas web, permitiendo una descripción más minuciosa. En un principio, no rivalizarán HTML y XML, se complementarán el uno al otro anudándose ambas gramáticas. Este Lenguaje de marcas extensible (XML) es una versión abreviada del SGML (Standard Generalized Markup Language).

Estructura del XML

El metalenguaje XML consta de cuatro especificaciones (el propio XML sienta las bases sintácticas y el alcance de su implementación):²

- **DTD** (Document Type Definition): Definición del tipo de documento. Es, en general, un archivo/s que encierra una definición formal de un tipo de documento y, a la

² GONZÁLEZ, F. S. *XML (EXTENSIBLE MARKUP LANGUAGE): Nuevo Estándar para la Descripción de Documentos en la Word Wide Web*, [En línea 2008]. [Consultada Marzo 8, 2009]. Disponible en: http://fesabid98.florida-uni.es/Comunicaciones/f_santamaria/f_santamaria.htm

vez, especifica la estructura lógica de cada documento. Define tanto los elementos de una página como sus atributos. El DTD del XML es opcional. En tareas sencillas no es necesario construir una DTD, entonces se trataría de un documento "bien formado" y si lleva DTD será un documento "validado".

- **XSL** (eXtensible Stylesheet Language): Define o implementa el lenguaje de estilo de los documentos escritos para XML. Desde el verano de 1997 varias empresas informáticas como Arbortext, Microsoft e Inso vienen trabajando en una propuesta de XSL (antes llamado "XML-style") que presentaron a W3C.

Permite modificar el aspecto de un documento. Se puede lograr múltiple columnas, texto girado, orden de visualización de los datos de una tabla, múltiples tipos de letra con amplia variedad en los tamaños. Este estándar está basado en el lenguaje de semántica y especificación de estilo de documento (DSSSL, Document Style Semantics and Specification Language, ISO/IEC 10179) y, por otro lado, se considera más potente que las hojas de estilo en cascada (CSS, Cascading Style Sheets), usado en un principio con el lenguaje DHTML.

Se espera que el CSS sea usado para visualizar simples estructuras de documentos XML (actualmente se ha conseguido mayor integración en XML con el protocolo CSS2 (Cascading Style Sheets, level 2) ofreciendo nuevas formas de composición y una más rápida visualización) y, por otra parte, XSL pueda ser utilizado donde se requiera más potencia de diseño como documentos XML que encierran datos estructurados (tablas, organigramas, etc.)

- **XLL** (eXtensible Linking Language): Define el modo de enlace entre diferentes enlaces. Se considera que es un subconjunto de HyTime (Hipermedia/Timed-based structuring Language o Lenguaje de estructuración hipermedia/basado en el tiempo, ISO 10744) y sigue algunas especificaciones del TEI (Text Encoding Initiative o Iniciativa de codificación de texto).

Desde marzo de 1998 el W3C trabajó en los enlaces y direccionamientos del XML. Provisionalmente se le renombró como Xlink y a partir de junio se le nombra como XLL. Este lenguaje de enlaces extensible tiene dos importantes componentes: Xlink y el Xpointer. Va más allá de los enlaces simples que sólo soporta el HTML.

- **XUA** (XML User Agent): Estandarización de navegadores XML. Todavía está en proceso de creación de borradores de trabajo. Se aplicará a los navegadores para que compartan todas las especificaciones XML.

Características del lenguaje XML.³

- ❖ Es una arquitectura más abierta y extensible. No se necesita versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet/intranet por medio de un validador de documentos.
- ❖ Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.
- ❖ Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje permitirá agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- ❖ Gestión y manipulación de los datos desde el propio cliente web.
- ❖ Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible.

³ Ver referencia 2

- ❖ Se desarrollarán de manera extensible las búsquedas personalizables y subjetivas para robots y agentes inteligentes. También conllevará que los clientes web puedan ser más autónomos para desarrollar tareas que actualmente se ejecutan en el servidor.
- ❖ El concepto de "hipertexto" se desarrollará ampliamente (permitirá denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc. Creado a través del Lenguaje de enlaces extensible (XLL).
- ❖ Exportabilidad a otros formatos de publicación (papel, web, cd-rom, etc.). El documento maestro de la edición electrónica podría ser un documento XML que se integraría en el formato deseado de manera directa.

Canal de sindicación: es una selección de contenidos de una web a la que el usuario puede "suscribirse". Cada Canal tiene una dirección (URL) similar a la de las páginas web. Los canales contienen toda la información seleccionada (titulares, textos, enlaces, etc.) en **XML**, que los programas Agregadores o Lectores de Canales pueden entender. Un mismo sitio web puede ofrecer varios canales.

Sindicación de Contenido: proceso mediante el cual un productor o un distribuidor de contenidos proporcionan información en formato digital a un suscriptor o a una red de suscriptores, generalmente con el ánimo de que los integren en sus propios sitios web.⁴

Es una forma que tienen algunos sitios web de distribuir contenidos a los que los usuarios acceden frecuentemente. El contenido se distribuye a través de los **canales** de sindicación y los usuarios pueden leer esos canales con un software denominado

⁴ CATALÁN, J. F. Y. M. A. *Bitácoras y sindicación de contenidos: dos herramientas para difundir información*, [En línea 2005]. [Consultada Marzo 8, 2009]. Disponible en: <http://www.ub.es/bid/15frang2.htm>

programa **agregador** (también llamado Lector de Canales o Lector de Noticias, en inglés: *Newsreader* o *Feed Reader*).

RSS: es una familia de formatos de fuentes web codificados en XML. Se utiliza para suministrar a suscriptores de información actualizada frecuentemente. El formato permite distribuir contenido sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador o lector RSS).

ATOM: es otro sublenguaje XML. No se basa en ninguna versión de RSS, pero es un formato muy similar y tiene el mismo objetivo: permitir la distribución de contenidos y noticias de blogs y sitios web.

El Resource Description Framework o Marco de Descripción de Recursos (RDF): es un formato universal para el intercambio de información. Se utiliza para poner a disposición de usuarios o aplicaciones los contenidos más novedosos de un sitio.

Monitoreo: el monitoreo es un sistema de seguimiento continuo, en el que la información se recoge permanentemente.⁵

Es el desarrollo de tareas de supervisión capaz de controlar resultados de operaciones diversas, programas o funciones específicas por medio de circuitos programados, permitiendo la observación de un proceso en ejecución, en funcionamiento, examinando el transcurso del mismo por medio de ciertas variables seleccionadas.

Un observatorio: es un lugar dispuesto o adecuado para hacer cualquier análisis; es una estructura que posibilita una amplia visión de lo que lo rodea.⁶

⁵ PERÚ, M. D. E. D. L. *Orientaciones para el Monitoreo y Evaluación de Planes Operativos en Regiones*, [En línea 2007]. [Consultada Marzo 8, 2009]. Disponible en: http://www.minedu.gob.pe/normatividad/reglamentos/instructivo_monitoreo_evaluacion.pdf

⁶ SANCHO, J. M. *LOS OBSERVATORIOS DE LA SOCIEDAD DE LA INFORMACIÓN: EVALUACIÓN O POLÍTICA DE PROMOCIÓN DE LAS TIC EN EDUCACIÓN*, [En línea 2004]. [Consultada Marzo 8, 2009]. Disponible en: <http://www.rieoei.org/rie36a02.pdf>

Consiste en la recopilación, análisis e interpretación de información relevante para el conocimiento de un mercado u objetivo a observar dentro de un marco en concreto. Su pretensión es obtener información de todos los factores que intervienen en el mercado: producción, importación-exportación, distribución, logística, legislación, consumidores, prescriptores, etc. El observatorio va más allá del diagnóstico de la situación de una determinada marca. Es un instrumento que permite obtener y analizar información sobre el grado de desarrollo y utilización de las Tecnologías de la Información y la Comunicación (TIC), de forma sistemática. Muchos de los observatorios existentes están hechos en forma de portales web.

1.2. Antecedentes

1.2.1. Antecedentes de Observatorios Temáticos

Observatorio del Plástico:

Es un Sistema de Vigilancia Tecnológica (VT) para las empresas industriales del sector de los materiales plásticos y sus procesos de transformación, realizado por AIMPLAS Instituto Tecnológico del Plástico, a partir de un software de creación propia. A través de un proceso de localización, captura, validación y difusión, el Observatorio es capaz de dar respuesta a las necesidades de información de los grupos de investigación de AIMPLAS, empresas del sector o empresas de otros sectores industriales.⁷

Este sistema de VT está montado en un sitio web, mediante el cual brinda una serie de servicios como consultorías, boletines tecnológicos, informes, alertas, incluso cuenta con canales de sindicación por los cuales el usuario puede recibir noticias actualizadas de temas relacionados con la industria del plástico.

Para poder acceder a todos los servicios que presta este observatorio el usuario interesado tiene que estar registrado y pagar de acuerdo a la tarifa establecida. Solamente prestan sus servicios gratis a las empresas o instituciones que son

⁷ PLÁSTICO, A. I. T. D. *Observatorio del Plástico*, [Consultada Marzo 9, 2009]. Disponible en: http://www.observatorioplastico.com/generador.php?pag=paginas/que_es.php

asociadas al Instituto Tecnológico del Plástico (AIMPLAS), que en definitiva serían la misma empresa.

Observatorio de Prospectiva Tecnológica Industrial (OPTI):

Es una fundación, que bajo protectorado del Ministerio de Industria, Turismo y Comercio, genera información inteligente sobre la evolución y futuro de la ciencia y la tecnología a medio y largo plazo. En colaboración con la Oficina Española de Patentes y Marcas (OEPM) ha iniciado un Servicio de Vigilancia Tecnológica destinado a la empresa, que facilita información sobre avances y novedades tecnológicas.⁸

OPTI es un observatorio montado en un portal web que se dedica a realizar VT en los campos de agroalimentación, calzado, plástico, metal y acuicultura, que trimestralmente publica de forma online boletines relacionados con estos temas. Además de ofrecer consultorías sobre temas que los usuarios puedan tener dudas o quieran ampliar el campo de conocimiento. Este servicio es vía correo electrónico para agilizar el proceso de respuesta y por supuesto romper la barrera geográfica que pueda separar al cliente, como puede ser grandes distancias.

Este observatorio no cuenta con canales de sindicación por lo que el usuario deberá conformarse con los boletines publicados trimestralmente lo que puede resultar insuficiente en dependencia de las exigencias y necesidades que tenga el cliente en cuestión.

Observatorio Español de Acuicultura (OESA):

Se encamina, además de servir de enlace entre los distintos actores implicados en el sector de la acuicultura, a la elaboración de estudios que detecten tanto las necesidades tecnológicas del sector industrial como los distintos escenarios creados en torno a la acuicultura.

⁸ OPTI, F. *Observatorio de Prospectiva Tecnológica Industrial*, [Consultada Marzo 9, 2009]. Disponible en: <http://www.opti.org/index/que.asp>

Este observatorio nace con el objetivo de servir de plataforma para el análisis y seguimiento permanente del desarrollo de la acuicultura en España, tanto en lo que se refiere a las actividades de I+D+i como a las realizadas por las distintas administraciones públicas y por las empresas.⁹

Además de contar con un portal web para publicar de forma trimestral los boletines informativos y estadísticos, en este observatorio los usuarios no tienen que registrarse para tener acceso a los servicios del mismo. Incluso si el usuario quiere recibir boletines de noticias solamente tiene que suscribirse dejando plasmado su correo electrónico, medio por el cual le llegan los boletines de noticias.

Observatorio CEDITEC (Centro de Difusión de Tecnología):

Es un observatorio que surge en el año 2002 con el propósito de seguir el vertiginoso desarrollo de la telefonía móvil, tanto en sus aspectos tecnológicos (patentes, proyectos, estándares) como sociales y de mercado (nuevos servicios y aplicaciones, legislación, competencia, impacto social, etc.).

El Centro de Difusión de Tecnologías (CEDITEC) se constituye en el seno de la Universidad Politécnica de Madrid (UPM), dentro de la estructura de la Fundación para el Desarrollo de las Telecomunicaciones (FUNDETEL).

La misión principal de CEDITEC es la difusión de las oportunidades de las Tecnologías de la Información y las Comunicaciones (TIC) en sectores empresariales, y la generación de nuevas ideas de negocio basadas en la innovación a través de las TIC.

¹⁰

Los servicios que brinda **CEDITEC** son totalmente gratuitos, además de brindar información confiable respecto al tema de vigilancia que siguen. Esta información fluye a través de boletines e informes detallados.

⁹ OESA. *Observatorio Español de Acuicultura*, [Consultada Marzo 9, 2009]. Disponible en: http://www.observatorio-acuicultura.org/es/es/index.php?option=com_content&task=view&id=6&Itemid=20

¹⁰ CEDITEC. *Centro de difusión de tecnologías ETSIT-UPM*, [Consultada Marzo 9, 2009]. Disponible en: <http://www.ceditec.etsit.upm.es/index.php/Que-es-CEDITEC.html>

Para acceder a estos documentos hay que ser usuario registrado, acción que se realiza totalmente gratis. Este observatorio además les hace llegar a sus usuarios boletines a través de correo electrónico dándole de esta forma un valor agregado.

Observatorio Virtual de Transparencia de Tecnología (OVTT):

Se presenta a sus clientes mediante un portal web el cual es bastante amplio y brinda una gran cantidad de información en forma de boletines e informes, a los cuales los usuarios pueden acceder libremente, sin necesidad de registrarse en el sitio.

Los usuarios registrados tienen la ventaja o beneficio de poder acceder a información especializada de acuerdo al perfil que construyó a la hora de registrarse.

Cuenta con una biblioteca donde se publican artículos de diversos temas, todos organizados por temática para una mejor localización por parte del internauta. Así como con un asistente virtual que se encarga de realizar búsquedas de acuerdo a los parámetros de búsqueda que defina el usuario. También cuenta con un buscador que realiza la misma función pero en bases de datos especializadas.

Su objetivo fundamental es el de impulsar la Investigación, el Desarrollo y la Innovación (I+D+I) como el elemento esencial de progreso y competitividad.¹¹

Los usuarios no pueden almacenar información en sus perfiles solamente acceder a las información que tiene el sitio y realizar un monitoreo de forma individual ya que no se brinda la posibilidad de realizar un monitoreo en grupos de trabajo.

Observatorio Tecnológico de la Soldadura (OBTESOL):

Es una herramienta de apoyo a la I+D+I, desarrollada y gestionada por el Centro Tecnológico AIMEN. Su principal objetivo es el de proporcionar información de todo

¹¹ OVTT. *Observatorio Virtual de Transparencia de Tecnología*, [Consultada Marzo 9, 2009]. Disponible en: <http://www.ovtt.es/objetivos.jsp>

tipo y proveniente de múltiples fuentes de información, que permita conocer de forma actualizada cuál es el estado de desarrollo de las Tecnologías de Unión.¹²

Sus servicios se brindan a través de un portal web, los mismos solo se ofrecen a usuarios registrados. Los que reciben de forma periódica boletines informativos y estadísticos sobre las novedades de la tecnología de soldadura. Estos servicios se prestan principalmente vía correo electrónico, o el cliente puede descargarlo directamente a su computadora desde el sitio web una vez registrado anteriormente.

El usuario tiene que tener bien claro que es lo que busca o de lo contrario tiende a perderse dentro del sitio con otras informaciones que pudiera tener un valor agregado para el cliente pero que al final pudiera servir para distraer al mismo y perder tiempo valioso.

1.2.1.1. Consideraciones Generales de los Observatorios

Los observatorios son sin lugar a dudas, herramientas que ayudan mucho en el proceso de toma de decisiones. Esto sin mencionar que generalmente se emplea, para el trabajo con la misma, un personal preparado fundamentalmente en el tema sobre el que se va a hacer seguimiento, así como personal calificado para confeccionar informes informativos o estadísticos para darle la seriedad que requiere tal actividad y por supuesto ganar en aceptación por parte de los usuarios.

Estas herramientas se desarrollan sobre tecnología web, utilizando los lenguajes de programación existentes para ello como: PHP, JAVASCRIPT, JAVA, ASP.net, entre otros. Cuentan con bases de datos donde almacenan todas las cuentas de sus usuarios y otras informaciones importantes, tratando de mantener lo más seguro posible estas informaciones para lograr un correcto funcionamiento.

Al construirse sobre sitios web generalmente se colocan otras informaciones que en un principio pueden tener valor agregado, pero que después suelen perderlo, ya que

¹² OBTESOL. *Observatorio Tecnológico de la Soldadura*, [Consultada Marzo 9, 2009]. Disponible en: http://www.obtesol.es/index.php?option=com_content&task=view&id=5&Itemid=35

tienden a convertirse en temas publicitarios y de propaganda que pueden ser molestos para el internauta que esté utilizando los servicios del observatorio en cuestión. Además de que estos portales web no brindan la posibilidad de realizar seguimientos u observar ciertos sectores de interés, en grupos de trabajo, sino que sus servicios se brindan de forma personal.

1.2.1.2. Antecedentes de monitorización de noticias

Centroamericajuven.org: Su objetivo general es ofrecer y promover el intercambio de información sobre la juventud centroamericana, que sea de utilidad para los propios jóvenes y para quienes trabajan con esta población.¹³

Women's Media Watch: Sus actividades comprenden el monitoreo de los medios de comunicación, la promoción de los derechos de la mujer, el activismo, la organización de talleres sobre los medios de comunicación y la conciencia de género, y la producción de boletines, documentales y programas radiales sobre la igualdad de oportunidades para las mujeres en los medios.¹⁴

directorio.gob.mx: es un directorio creado para suministrar a los usuarios información sobre el sistema federal en México pero además hacen seguimiento de medios de comunicación a los cuales los usuarios pueden acceder.

Información de medios para Colombia Infomedios Colombia Ltda. : es una empresa colombiana que ofrece servicios de monitoreo de medios escritos, análisis, síntesis, digitalización y escaneo de toda clase de documentos para las necesidades de información y toma de decisiones en la ejecución de la comunicación social de particulares e instituciones.¹⁵

¹³ CENTROAMERICAJOVEN.ORG. *Intercambiando conocimiento sobre juventud en la región*, [Consultada Marzo 9, 2009]. Disponible en: <http://centroamericajuven.org/quienes-somos.html>

¹⁴ WMW. *Women's Media Watch*, [Consultada Marzo 9, 2009]. Disponible en: http://www.womensmediawatch.org/index.php?option=com_content&task=view&id=5&Itemid=6

¹⁵ INFOMEDIOS. *Información de medios para Colombia Infomedios Colombia Ltda.*, [Consultada Marzo 9, 2009]. Disponible en:

En la Mira: Una firma guayaquileña de monitoreo noticioso de medios. Su valor agregado es que la operación la realizan de manera digital para automatizar los procesos. Para En la Mira, la idea es capturar la información, analizarla y enviarla a las PC de sus clientes en un lapso de 10 minutos.

Plan Puebla Panamá: Es una propuesta de los ocho países mesoamericanos para fortalecer la integración regional e impulsar los proyectos de desarrollo social y económico en los estados del Sur-Sureste de México y el Istmo Centroamericano. Ellos hacen un Monitoreo de Noticias y emiten un informe semanal que intenta recopilar vía Internet las noticias periodísticas sobre el Plan Puebla Panamá difundidas a nivel nacional, regional e internacional. La veracidad y precisión de la información contenida en dichas noticias son responsabilidad del periodista y medio de comunicación que las divulga, y para su elaboración no necesariamente han sido consultadas fuentes institucionales del Plan Puebla Panamá.

1.2.2.1. Características Principales

Estos medios o centros de monitorización se caracterizan por hacer seguimientos de determinados temas de interés. Generalmente se generan boletines como resultado de toda una jornada de trabajo, de una investigación determinada, en fin todo depende del tema del que trata el monitoreo.

La arquitectura escogida para este tipo de actividad ha sido muy acertada, ya que las empresas u organizaciones, que se dedican al monitoreo utilizan portales web para llegar de una forma más rápida y entendible al cliente. Esta es una excelente idea ya que se hace mucho más fácil llegar a un cliente sin importar la distancia que se encuentre, siempre y cuando se tenga la posibilidad de publicar el portal web en Internet y se logre posicionarlo adecuadamente.

El buen posicionamiento de un portal web en el ciberespacio, no es muy sencillo, ya que el sitio web debe ser muy intuitivo para los usuarios y por supuesto muy amigable,

que cuando el usuario quiera llegar a cualquier información lo haga en el menor tiempo posible y en el menor número de clics, en fin hay que seguir toda una guía de buenas prácticas ó de lo contrario se corre el riesgo de no ser aceptado y no cumplir el principal objetivo de un centro de monitoreo: brindar información valiosa y confiable sobre un tema en específico.

Pero estos sitios además de tener muy buenas funcionalidades y contar con espacios como los foros, donde los usuarios pueden intercambiar ideas y opiniones. Carecen de espacios donde los usuarios puedan crear sus perfiles y guardar por ejemplo todos los resultados que han obtenido con un seguimiento de un determinado tema. Sino que todos los boletines generados se publican en el sitio web y los usuarios al registrarse pueden bajar estos boletines a su ordenador. Además de tener que conectarse al portal web cada vez que quiera actualizarse sobre un tema de interés, lo que puede resultar tedioso para el usuario.

1.2.3. Antecedentes de Vigilancia

Knowledge Works: es un sistema, que trabaja con varios servidores de bases de datos, en el cual, se puede organizar información relacionada con los competidores, el estado del arte del sector tecnológico en el que se desenvuelve la empresa, la legislación, opiniones de expertos, etc.

Wincite: es un sistema personalizable, que trabaja con varios servidores de bases de datos. Permite la organización de la información, puede ser accedido vía Internet o intranet, comparar características entre productos, se garantiza la fase de divulgación al permitir personalizar los reportes.

VantagePoint: (EE.UU.) Es un software de minería de textos, basado en más de una década de investigación que ha sido desarrollado por Search Technology. Sus características más relevantes son: la navegación rápida en grandes colecciones abstractas, la exhibición visual de relaciones mediante matrices de co-ocurrencia o de

factores, mapas tecnológicos y la creación de tesauros para reducir datos. Brinda un servicio de diseño, gestión y análisis de encuestas.

GoldFire: es un software analizador sintáctico- semántico que lee los contenidos de Internet, de bases de datos locales, de bases de artículos científicos y de patentes. Identificando las soluciones a preguntas hechas en lenguaje natural. El software permite la extracción de todo el valor de los documentos, para usarlas como soluciones innovadoras.

Permite establecer la evolución de una tecnología, el perfil de una organización en cuanto a su actividad de patentes; permite identificar las organizaciones implicadas en el desarrollo de determinadas patentes, la citación de patentes, de inventores, etc.

Matheo: (Francia) Es una herramienta avanzada para profesionales especializados/as en el análisis de informaciones obtenidas de bases de datos. Utilizada en algunos observatorios.

- **Matheo Patent:** Es un software de uso personal diseñado para automatizar las tareas de búsqueda, recuperación y análisis de patentes de las bases de datos de la Oficina de Patentes de los EE.UU. (USPTO) y de la Oficina Europea de Patentes (EPO). Ambos sitios dan información gratuita y de calidad, pero requieren mucho tiempo y paciencia si se desean hacer las búsquedas a mano.
- **Matheo-Analyzer:** Es una herramienta avanzada para profesionales especializados en el análisis de informaciones obtenidas de bases de datos.

Competitor Analyst: es un servicio web, con una parte gratuita, que permite identificar la posición de los competidores en Internet.

Tetralogie: (Francia) Es un paquete muy completo especializado en analizar grandes volúmenes de información científica y de patentes. Su función más avanzada es la

generación de mapas tecnológicos a partir de análisis estadísticos de co-ocurrencias de términos

Aurigin-aureka: (EE.UU.) Permite hacer búsquedas, análisis de datos y de texto, hacer anotaciones y guardar los resultados. Incluye:

- **Aureka Themescape:** Genera mapas con aspecto geográfico (valles, montañas, desniveles) que permiten entender intuitivamente los conceptos más relevantes, su grado de concentración y grado de relación.
- **Aureka Citation Tree:** Construye un árbol de citas a partir de una patente seleccionada.

WatchSite: es un software que vigila determinadas páginas webs e indica si éstas han sido modificadas o no. Es ideal para realizar la vigilancia de las páginas webs de los competidores.

1.2.3.1 Análisis General de las Herramientas

Las herramientas vistas anteriormente son herramientas especializadas en el campo de la vigilancia tecnológica. Las cuales se basan fundamentalmente en la búsqueda de información en base de datos especializadas, realizan vigilancia sobre sitios web predefinidos por el usuario, es decir avisan al usuario cada vez que se actualizan los sitios web que el cliente introdujo dentro de su criterio de búsqueda o vigilancia, entre muchas otras funcionalidades que son muy útiles en el análisis de la información.

Pero estas herramientas tienen un gran inconveniente y es que son herramientas comerciales creadas para las grandes empresas en el mundo que tienen los recursos y el capital para adquirirlas así como para darles el soporte técnico que necesitan, además de las empresas que se dedican al negocio de la VT para dar consultorías sobre el tema a otras empresas.

Los precios de estas herramientas oscilan entre los 7000 u 8000 dólares precios que pueden llegar a ser bastante alto para las pequeñas empresas.

1.2.4. Herramientas lectoras de Canales de Sindicación.

WinRSS:

Ventaja

Es un completo y a la vez sencillo sistema de lectura de noticias disponibles en Internet. Con una interfaz web. Soporta noticias en todos los formatos más populares existentes: RSS, ATOM, y RDF. Además de contar con una versión gratuita.

Desventaja

Tiene una interfaz poco amigable, cada vez que se realiza una operación determinada sobre la herramienta se abre una nueva pestaña en el navegador (lo cual es muy molesto para el usuario) y funciona de forma individual, es decir no ofrece la opción de integrarse junto con otros clientes a un mismo servidor central. Por lo que la administración de los canales RSS debe ser controlada manualmente.

FeedDemon:

Ventaja

Es una aplicación de escritorio que organiza los sitios web en canales y muestra la información extraída de ellos en forma de artículos de noticias, fácil y cómodo de leer. Te permite controlar la llegada de noticias sobre determinados temas, realizar búsquedas y tiene además soporte para varios idiomas, entre ellos español.

Desventaja

Una vez que se instala en una computadora no se puede acceder desde otra, esto podría mejorar la seguridad de la aplicación, pero atenta contra la aceptación del producto por parte del cliente. Ya que siempre tendría que volver a la misma

computadora donde instaló la aplicación. Solo cuenta con una versión de prueba que dura 30 días para probarlo gratuitamente.

FeedReader:

Ventajas

Soporta diversas versiones de la sindicación RSS y utiliza un sistema de caché avanzado para reducir al máximo el consumo de ancho de banda de la conexión.

Incluye un listado de sitios web que ofrecen información de última hora mediante este sistema, organizados en categorías, y además se puede leer cómodamente en la propia interfaz del programa.

Desventaja

Tarda demasiado en actualizar los sumarios que el usuario ha adicionado previamente al agregador.

Feedmania:

Ventajas

Es un servicio online totalmente en castellano. Soporta las distintas versiones de RSS y ATOM, permite importar y exportar listas de suscripciones en el formato OPML.

También los usuarios pueden compartir sus suscripciones entre ellos.

Desventajas:

No existe estructuración en carpetas, por lo que en caso de que el usuario quiera leer todos los feeds relacionados con el diseño gráfico tiene que buscarlos en la lista de feeds. En cambio, los mismos están ordenados alfabéticamente.

Muestra todos los feeds independientemente de que se hayan leído o no.

Google Reader:

Una herramienta web bastante potente, con una rapidez sobresaliente, así como una interfaz muy amigable e interactiva. Es gratis pero es online.

Desventajas:

No se puede crear filtros de palabras, esto se utiliza para guardar noticias que concuerden con ciertas palabras definidas por el usuario.

Actualiza automáticamente sin dar la posibilidad de cambiarlo a manual, lo que puede ser molesto para algunos usuarios ya que podrían marcar noticias como leídas sin haberlas leído.

Una vez que se nombran las etiquetas o tags no se pueden renombrar.

1.2.4.1. Lectores Web vs Lectores de Escritorio

Con lo expuesto en el epígrafe anterior, no se quiere imponer el lector a utilizar, sino simplemente exponer varias características que dejen bien claro cual sería la mejor opción a elegir basada en las funcionalidades de los dos tipos de herramientas lectoras de RSS.

Aunque entre los cibernautas existe la polémica de cual herramienta es mejor que la otra. Todo está en las necesidades que tenga el usuario para determinar según su criterio cuál lector RSS es mejor utilizar.

Es cierto que los lectores web tienen características que los de escritorio no tienen y viceversa como es el hecho de que en los lectores RSS web se pueden consultar los feed del usuario desde cualquier PC conectado a internet pero tienen como desventaja la falta de confidencialidad, tienen menor número de opciones y vistas que los agregadores de escritorio y no despliegan avisos en la esquina inferior izquierda de la pantalla al recibir novedades.

Otro aspecto de peso a tener en cuenta es que los lectores o agregadores de escritorio son más rápidos que los web, y además la mayoría de los agregadores web no trabajan offline.

Mediante el estudio riguroso y a profundidad de las herramientas tratadas en los antecedentes, se hace evidente que las mismas a pesar de tener muchas funcionalidades, muchas veces no se complementan una con otra. Lo ideal sería una herramienta que fuera capaz de realizar la búsqueda y el procesamiento de las noticias por sí misma sin necesidad de necesitar de otra herramienta que complemente el trabajo.

Aunque hay que destacar que aportan mucho desde el punto de vista de diseño. Fundamentalmente las aplicaciones de escritorio ya que se acercan bastante al sistema que se quiere lograr. Sobre todo a la hora de diseñar una interfaz amigable y fácil de usar como la que se quiere implementar. Sin dejar a un lado las potencialidades que presentan los canales de sindicación en cuanto a rapidez de actualización se trata. Es decir, todas estas herramientas analizadas vienen a formar parte del cimiento de la aplicación desarrollada, contribuyendo con la robustez de la misma.

1.3. Canales de Sindicación.

1.3.1. ¿En qué consiste la sindicación de contenido?

Los antecedentes de la sindicación de contenidos se encuentran en las tecnologías *push and pull* que se popularizaron en la década de los años 90. La tecnología *push* describe un sistema de distribución de contenidos por Internet donde la información se distribuye desde un servidor central hasta una computadora cliente, sobre la base de un conjunto predefinido de parámetros de solicitud, definido por la computadora cliente; de esta manera, el servidor *empuja* la información hasta la máquina cliente. Por su parte,

la tecnología *pull* se refiere a aquella que permite a los usuarios “halar” contenidos hasta su máquina, por ejemplo, por medio de la suscripción.¹⁶

La sindicación de contenido es una técnica que se implementa actualmente por casi todos los medios noticiosos, incluyendo blogs, agencias de noticias, agencias de anuncios publicitarios, incluso los servicios de correo electrónico poseen esta funcionalidad. La razón por la que la sindicación es tan popular es sencilla, ya que el usuario no tiene que hacer ninguna petición a los servidores de estos medios sino que la información es empujada hacia ellos, tan pronto como se actualice el medio, a través de los canales RSS que tienen implementados.

En rigor, a partir de la inclusión de algún nuevo contenido en un sitio, lo que se distribuye es una lista de enlaces junto con cierta cantidad de información adicional o metadato.

Los enlaces apuntarán a esos nuevos contenidos y la información adicional permitirá a los receptores evaluar si los contenidos son de su interés, en cuyo caso accederá a la versión completa simplemente siguiendo el enlace.

1.3.2. Los archivos RSS

El término RSS corresponde a Rich Site Summary o Really Simple Syndication. Es interesante destacar que se trata de un formato que no está concebido para su visualización (como el HTML) sino para la interacción entre computadoras, ofreciendo la información en un formato estandarizado. Escrito en XML.

Para que este proceso resulte posible, un sitio web debe generar un feed o canal, que no es más que el RSS, este es un archivo más que debe estar alojado en el servidor del sitio. Una vez que el RSS se encuentre habilitado otros sistemas podrán acceder a él y enterarse de las actualizaciones del sitio en cuestión.

¹⁶ TARRAGÓ, L. N. S. *Sindicación de contenidos con canales RSS: aplicaciones actuales y tendencias* [En línea 2007]. [Consultada Marzo 9, 2009]. Disponible en: http://bvs.sld.cu/revistas/aci/vol15_3_07/aci03307.htm

Para leer estos archivos es necesario tener un programa especializado en esta tarea conocidos genéricamente como agregadores o lectores de canales de sindicación.

Antes del surgimiento de formato RSS, existían varios formatos de sindicación, sin embargo, ninguno había ganado la popularidad que ha ganado éste. Existen opiniones divergentes sobre el origen de los canales RSS, pero sus antecedentes más documentados son el Channel Definition Format, creado por Microsoft para el Internet Explorer 4.0 en 1997 y el formato ScriptingNews, creado por Dave Winer, de UserLand ese mismo año.¹⁷

Características generales de sus versiones:

- **RSS 0.91**

Publicada en Junio del 2000 por Dave Winner. Esta especificación supone adoptar la especificación XML 1.0 y abandonar la especificación original de documentos RDF. De la misma forma se cambia el significado de las siglas RSS (se modifica por Really Simply Syndication).

- **RSS 0.92**

Publicada en Diciembre del 2000 por Dave Winner. Esta especificación es totalmente compatible con RSS 0.91 ya que los nuevos elementos incorporados por esta versión son opcionales. Por tanto, un fichero RSS 0.91 es también un fichero RSS 0.92 válido.

- **RSS 1.0**

Publicada en Diciembre del 2000 por los miembros del grupo de trabajo RSS-DEV. Esta especificación supone adoptar la especificación XML 1.0 y mantiene la compatibilidad original con documentos RDF. De la misma forma se mantiene el significado de las siglas RSS (RDF Site Summary).

- **RSS 2.0**

Publicada en Octubre del 2002 por Dave Winner. Esta especificación es compatible con RSS 0.91 y RSS 0.92. Por tanto, un fichero RSS 0.91 es también un fichero RSS 2.0 válido.

¹⁷ Ver referencia 16

Los agregadores o Lectores de canales

Gracias a los agregadores RSS o lectores de feeds (aplicación local o basada en Web, que permiten leer e interpretar fuentes Web) se puede obtener resúmenes de todos los sitios que se desee desde el escritorio del ordenador, programas de correo electrónico o por medio de aplicaciones que funcionan como agregadores, sin necesidad de abrir el navegador y visitar decenas de Web. Un agregador es muy similar en sus prestaciones a los anteriores lectores de noticias, pero la tecnología XML y la Web semántica los ha hecho más populares.¹⁸

Los archivos RSS al ser cargados en el explorador solo se verían una página con código XML que es por lo que están compuestos. Mientras que los agregadores muestran la información de forma organizada, con una estructura bien definida e incluso ocultan las noticias que ya han sido leídas, en dependencia de la configuración que le haya dado el usuario.

1.3.3 ¿Para qué sirve la sindicación?

Mediante la sindicación los usuarios no tienen que visitar ningún sitio para actualizarse de un tema determinado ya que todo le queda a un clic de distancia, es cierto que en un principio parece que atenta contra la visibilidad de los sitios pero en realidad es una estrategia para incrementar y fidelizar visitantes, pues mediante este método se ahorra las constantes visitas a los sitios de interés para ver si ya fueron actualizados.

Con la sindicación de contenidos las bibliotecas y centros de documentación han podido perfilar y mejorar sus servicios de alerta y difusión selectiva de información. Además de contribuir a la ampliación de plataformas de acceso a la información por parte del usuario, el cual, tan sólo debe preocuparse de agregar los contenidos que le

¹⁸ HUIDOBRO, J. M. *La Sindicación Web: RSS* [En línea 2008]. [Consultada Marzo 9, 2009]. Disponible en: <http://www.coit.es/publicaciones/bit/bit167/rincondeinternet.pdf>

interesan en un agregador o lector de feeds y podrá consumir la información desde casi cualquier plataforma (un teléfono móvil, una PDA, un iPod, etc.).¹⁹

1.3.4. ATOM.

Atom, formato para la sindicación de contenidos, similar al RSS. Nació para resolver la confusión creada por la existencia de estándares similares para la distribución de contenidos. Pero más que eliminar el problema de la multiplicidad de estándares, ha creado un formato nuevo, condenado a convivir con los que pretendía sustituir. La principal característica distintiva de Atom es su flexibilidad: es un formato consistente en cuanto a la sindicación, el almacenamiento y la edición de contenidos, capaz de transportar información más compleja. Los artículos sindicados con Atom acostumbran a incluir el texto completo, y de esta manera es posible ejercer un control adicional sobre la cantidad de información a representar en los agregadores.²⁰

A pesar de que nació para suplir la incompatibilidad de algunas versiones del RSS, porque muchos pensaban que le faltaba interoperabilidad entre sus elementos. Hoy en día se les llama RSS o Atom indiferentemente.

1.3.5 RDF.

Se refiere a una familia de especificaciones, destinadas a definir un modelo de metadatos, es decir, que pretende establecer la forma en que se debe expresar la información que describe los recursos de la Red. Propone, en definitiva, una manera estándar para codificar las abstracciones que permitirán describir los elementos informacionales que constituyen la infraestructura de la Web. Es, por tanto, un componente clave para el desarrollo de la Web Semántica, con el que se ha

¹⁹ GARCÍA, C. S. *La Web 2.0 y sus implicaciones en el mundo de la gestión de la información*, [En línea 2007]. [Consultada Marzo 9, 2009]. Disponible en: <http://www.iuetaeb.tec.ve/caa/html/La%20Web%202.0%20y%20sus%20implicaciones%20en%20el%20mundo%20de%20la%20gesti%C3%B3n%20de%20la%20informaci%C3%B3n.pdf>

²⁰ CATALÁN, J. F. Y. M. A. *Bitácoras y sindicación de contenidos: dos herramientas para difundir información*, [En línea 2005]. [Consultada Marzo 8, 2009]. Disponible en: <http://www.ub.es/bid/15frang2.htm>

comprometido el W3C desde hace ya más de una década y que promete transformar por completo la fisiología de la Web.²¹

Se utiliza para poner a disposición de otros, ya sea web o aplicaciones, los contenidos más novedosos de un sitio. RDF es un framework para metadatos ya que mediante él se pueden implementar o desarrollar otros canales como el RDF Site Summary que vienen siendo las versiones de 1.x de los RSS, aunque hay que aclarar que las versiones 0.9x y 2.x no son RDF sino Really Simple Syndication.

1.4. ¿Qué es la Programación?

La programación es la codificación de las órdenes y datos que permiten la creación de un programa o aplicación.²²

La programación desde el punto de vista informático es un proceso mediante el cual se puede crear programas mediante códigos generados y definidos a través de lenguajes de programación. Estos programas a su vez forman parte de programas más grandes conocidos como software. Por supuesto todo está en dependencia del tamaño del sistema que se quiera lograr. El software al final no es más que instrucciones que va a seguir el hardware de una computadora determinada.

De ahí que la programación sea una de las principales áreas en el mundo de la informática.

1.4.1 ¿Qué son los lenguajes de programación?

Un lenguaje natural es un instrumento de comunicación utilizado de forma común entre personas. Para que exista comunicación, debe existir una comprensión mutua de cierto conjunto de símbolos y reglas del lenguaje. Los lenguajes de programación tienen como

²¹ FUMERO, A. *El abecé del universo blog*, [Consultada Marzo 9, 2009]. Disponible en: <http://antoine.eduangi.com/MyWeb/Papeles/Blogotutorial.pdf>

²² WORDREFERENCE.COM. *Diccionario de la lengua española*, [Consultada Marzo 1, 2009]. Disponible en: <http://www.wordreference.com/definicion/programacion>

objetivo la construcción de programas, normalmente escritos por personas. Estos programas se ejecutarán por un computador que realizará las tareas descritas. El programa debe ser comprendido tanto por personas como por computadores. La utilización de un lenguaje de programación requiere, por tanto, una comprensión mutua por parte de personas y máquinas. Este objetivo es difícil de alcanzar debido a la naturaleza diferente de ambos.²³

Al aparecer la computadora, nació junto con él, pues el lenguaje conocido como lenguaje máquina, el cual consistía en 0 y 1, para decir que una llave estaba abierta se definía con un 1 y para decir lo contrario con un 0, todo esto se hacía para lograr una acción determinada, al querer realizar una acción que desencadenara una serie de llaves mecánicas esto traía consigo una cadena de 0 y 1 (por ejemplo 001110111000) que son difíciles de recordar.

A raíz de esta necesidad surge el lenguaje ensamblador, el que está constituido por instrucciones más fáciles de memorizar como ADD (sumar), SUB (restar), MUL (multiplicar), entre otras.

Después surgen los lenguajes de programación, los que se denominan así porque su estructura sintáctica es similar a la escritura que usan los humanos.

En la web se define como que es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones, y es utilizado para controlar el comportamiento físico y lógico de una máquina.

1.4.2 ¿Qué es la Programación Orientada a Objetos (OOP)?

Los lenguajes de programación Orientada a Objetos están teniendo un enorme éxito en el desarrollo del software, debido principalmente al esfuerzo empleado en proporcionar

²³ GAYO, J. E. L. *Desarrollo Modular de Procesadores de Lenguajes a partir de Especificaciones Semánticas reutilizables.*, [En línea 2001]. [Consultada Marzo 1, 2009]. Disponible en: <http://www.di.uniovi.es/~labra/tesis/tesis.pdf>

capacidades de reutilización de software. Simula puede considerarse el primer lenguaje Orientado a Objetos, creado en los años 60 con el propósito de describir y programar simulaciones computacionales. Smalltalk se desarrolla posteriormente como un sistema uniforme Orientado a Objetos. Aunque originalmente interpretado y poco eficiente, Smalltalk incorporaba un completo entorno gráfico de desarrollo y formó una importante base conceptual para la ingeniería del software Orientado a Objetos.²⁴

La OOP ha venido a agilizar considerablemente el tiempo de desarrollo del software, ya que en este tipo de programación, se piensa de una forma distinta. Se conciben las cosas de una forma más cercana a la realidad.

Se piensa en todo en formas de clases, métodos, propiedades o atributos, objetos, los cuales se modelan como en la vida real. Por ejemplo: un coche sería un objeto, que tiene como propiedades o atributos: el color, número de la chapa y el modelo; y como métodos o funcionalidades echarlo a andar, pararlo, todo lo que se quiere que haga el mismo.

Algunas de las estructuras de la OOP son:

- **Clase:**

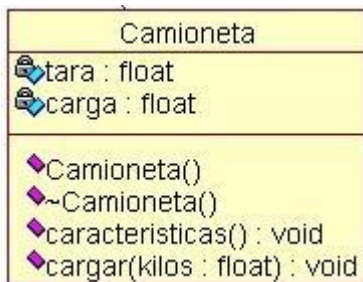
Es una abstracción que describe propiedades importantes para una aplicación y que ignora el resto. La selección de clases es arbitraria y depende de la aplicación.

Una clase contiene la descripción de las características comunes de todos los objetos que pertenecen a ella: la especificación del comportamiento, la definición de la estructura interna y la implementación de los métodos.²⁵

²⁴ Ver referencia 23

²⁵ BIBIANA ROSSI, P. B. Y. R. G. M. *MODELADO DE OBJETOS*, pág. 3, Consultada Marzo 2, 2009]. Disponible en: http://www.itba.edu.ar/archivos/secciones/art_revistas_2.pdf

Las clases vienen siendo como los moldes que serán usados para sacar objetos con las mismas funcionalidades y atributos o campos. Al definir un objeto se estaría definiendo una clase ya que la misma es una abstracción del objeto. Además de que estas funcionalidades y atributos tienen tipos de datos que son los que definen su comportamiento. Este es un ejemplo de clase (Representado en Lenguaje de Modelado Universal (UML siglas en inglés)).



- **Campos o Atributos:**

Estas son características del objeto, por lo que se encuentran declaradas en las clases, y al final no hacen más que almacenar y devolver datos.

- **Métodos:**

Los métodos constituyen las funcionalidades que van a tener los objetos, los cuales también se encuentran implementados en la clase.

- **Objetos :**

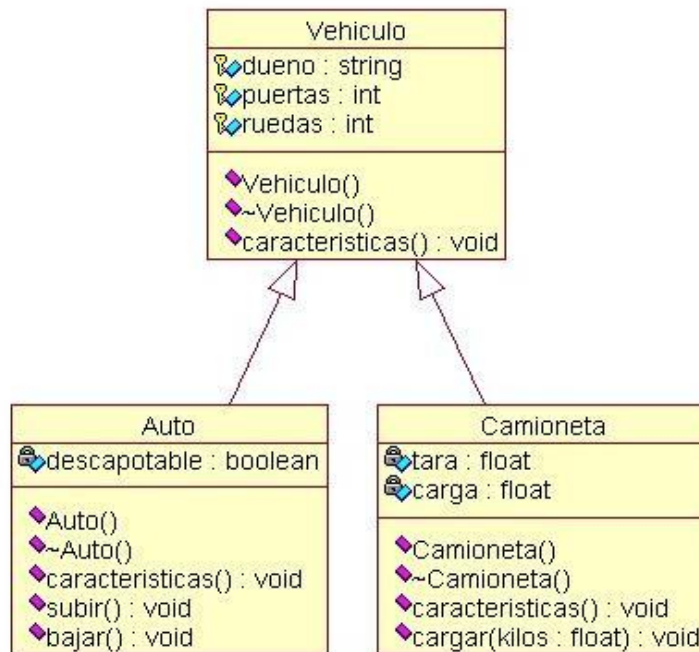
Los objetos son ejemplares de las clases, que tienen todas las funcionalidades y atributos de la clase que los creó, el proceso de crear un objeto se le conoce como instanciar que es una mala traducción de la palabra en inglés instance que significa ejemplar.

- **Herencia:**

Es compartir atributos y operaciones entre clases tomando como base una relación jerárquica. En términos generales se puede definir una clase que después se irá refinando sucesivamente para producir subclases. Todas las subclases poseen o heredan todas y cada una de las propiedades de su superclase y añaden, además, sus propiedades exclusivas. No es necesario repetir las propiedades de las superclases en cada subclase.²⁶

Es una propiedad que permite que los objetos sean creados a partir de otros ya existentes, obteniendo características (métodos y atributos) similares a los ya existentes.

Se representa de la siguiente forma:



²⁶ BIBIANA ROSSI, P. B. Y. R. G. M. *MODELADO DE OBJETOS*, pág. 4, Consultada Marzo 2, 2009]. Disponible en: http://www.itba.edu.ar/archivos/secciones/art_revistas_2.pdf

Donde Vehículo es la clase base de las que heredan Auto y Camioneta, en la figura se puede apreciar los atributos de cada cual junto con su tipo de dato.

- **Polimorfismo:**

Significa que una operación puede comportarse de modos distintos en distintas clases teniendo el mismo nombre de método.²⁷

Se denomina polimorfismo a la capacidad que tienen los objetos de una clase de responder al mismo mensaje o evento en función de los parámetros utilizados durante su invocación. Un objeto polimórfico es una entidad que puede contener valores de diferentes tipos durante la ejecución del programa.

Capacidad de usar un objeto sin saber su tipo exacto. Formalmente, el polimorfismo es la capacidad de que un elemento de código pueda denotar, en tiempo de ejecución, objetos de dos o más tipos distintos.

También es conocido en los métodos de sobrecarga de operadores de muchos lenguajes aunque por supuesto es un polimorfismo especial. Aunque no es preferidos por todos ya que se requiere programar un método por cada operador que vaya a sobrecargar.

1.4.3 Lenguaje JAVA

Es un lenguaje de Programación Orientado a Objetos (POO) desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.²⁸

²⁷ Ver referencia 25

²⁸ ANÓNIMO. *DESCRIPCIÓN DE JAVA 2 PLATFORM MICRO EDITION*, pág. 2, [Consultada Marzo 2, 2009]. Disponible en:

Es toda una tecnología orientada al desarrollo de software con el cual se puede realizar cualquier tipo de programa. En la industria para dispositivos móviles hay una gran acogida para este lenguaje.

El lenguaje Java es reconocido mundialmente por su capacidad de adaptación a cualquier ambiente. Los desarrolladores de Java optaron por dotar al lenguaje de un conjunto de capacidades orientadas a objetos que permitiesen añadir librerías de apoyo a medida que fuera necesario y en función de las necesidades tecnológicas.

La tecnología Java está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. La plataforma es la máquina virtual de Java (Java Virtual Machine). En el mundo es muy preferido por sus potencialidades en cuanto a programación web se refiere.

Además de ser muy difundido en el mundo entero, por lo que cuenta con una comunidad bastante amplia, en las que se puede encontrar un muy buen repositorio de información y códigos fuentes, así como documentación respecto al lenguaje como tal. Es decir es una comunidad que trabaja sobre la reusabilidad de código, cosa que agiliza mucho el desarrollo de aplicaciones.

1.4.4 Características de JAVA

Entre las principales características que JAVA tiene u ofrece son²⁹:

❖ **Es Simple:**

Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por

<http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1147/5/T%2011010%20CAP%202.pdf>

²⁹ PROGRAMACIONENJAVA.COM. *Programación en Java*, [Consultada Marzo 3, 2009]. Disponible en:

<http://www.programacionenjava.com/blog/2008/06/10/tutorial/tutorial-caracteristicas-de-java/>

ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.

❖ **Es Orientado a Objetos:**

Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, *clases* y sus copias, *instancias*. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria.

❖ **Es Distribuido:**

Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como *http* y *ftp*. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que los programas corran en varias máquinas, y permita la interacción entre ellas.

❖ **Es Robusto:**

Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.

❖ **Es de Arquitectura Neutral:**

Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (*run-time*) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado.

❖ **Es Seguro:**

El código Java pasa muchos *tests o pruebas* antes de ejecutarse en una máquina. El código se pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal (código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto).

❖ **Es Interpretado:**

El intérprete Java (sistema run-time) puede ejecutar directamente el código objeto. Enlazar un programa, normalmente, consume menos recursos que compilarlo, por lo que los desarrolladores con Java pasarán más tiempo desarrollando y menos esperando por el ordenador. Por ahora, que todavía no hay compiladores específicos de Java para las diversas plataformas, Java es más lento que otros lenguajes de programación, como C++, ya que debe ser interpretado y no ejecutado como sucede en cualquier programa tradicional.

❖ **Es Multithreaded:**

Al ser multithreaded (multihilvanado o multihilo), Java permite muchas actividades simultáneas en un programa. Los threads (a veces llamados, procesos ligeros), son básicamente pequeños procesos o piezas independientes

de un gran proceso. Al estar los threads contruidos en el lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++.

El beneficio de ser multithreaded consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.), aún supera a los entornos de flujo único de programa (single-threaded) tanto en facilidad de desarrollo como en rendimiento.

1.4.5 Ventajas y Desventajas de JAVA

Ventajas

- ✓ Existe soporte dado por Sun Microsystem.
- ✓ Debido a que existen diferentes productos de Java, hay más de un proveedor de servicios.
- ✓ Es independiente de la plataforma de desarrollo.
- ✓ Existen dentro de sus librerías clases gráficas como awt y swing, las cuales permiten crear objetos gráficos comunes altamente configurables y con una arquitectura independiente de la plataforma.
- ✓ Java permite a los desarrolladores aprovechar la flexibilidad de la Programación Orientada a Objetos en el diseño de sus aplicaciones.
- ✓ El conocimiento sobre tecnología Java está en alto crecimiento en el mercado.

Desventajas

- ✓ Los programas hechos en Java no tienden a ser muy rápidos. Como los programas de Java son interpretados nunca alcanzan la velocidad de un verdadero ejecutable.
- ✓ Hay diferentes tipos de soporte técnico para la misma herramienta, por lo que el análisis de la mejor opción se dificulta
- ✓ Para manejo a bajo nivel deben usarse métodos nativos, lo que limita la portabilidad.

- ✓ El diseño de interfaces gráficas con awt y swing no es simple.
- ✓ Existen herramientas como el JBuilder que permiten generar interfaces gráficas de manera sencilla, pero tienen un costo adicional.

Este lenguaje al igual que la mayoría tiene sus ventajas pero también tiene sus desventajas, todo está en el tipo de aplicación que se quiera crear, dependiendo de sus requerimientos y de las funcionalidades que deba cumplir el mismo.

1.5. Entorno de Desarrollo Netbeans 6.0.

El NetBeans es un entorno de desarrollo integrado (IDE), basado en estándares, escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones, que puede ser utilizado como un marco genérico para construir cualquier tipo de aplicación. El enfoque de NetBeans 6.0 es la mejora de la productividad a través de un editor más inteligente, más rápido, y la integración de todos los productos en un IDE.³⁰

El NetBeans 6.0 amplía el soporte más allá de Java y C/C++ al ofrecer una serie de funcionalidades para los lenguajes de programación dinámicos Ruby y Java Script, así como soporte para el último Java Standards, manteniendo, a su vez, la facilidad de uso y alta productividad de NetBeans.

Además esta versión incluye una mejor edición del código, capacidades de navegación e inspección, historia local, soporte integrado para Subversion (un controlador de versiones), y mayores capacidades de personalización.

³⁰ NETBEANS. *Sitio Oficial del Netbeans*, [Consultada Marzo 3. 2009]. Disponible en: <http://www.netbeans.org/community/releases/60/>

1.6. Base de datos

Un sistema de bases de datos es básicamente un sistema computarizado para llevar registros. Es posible considerar a la propia base de datos como una especie de armario electrónico para archivar, es decir, es un depósito o contenedor de una colección de archivos computarizados. Los usuarios del sistema pueden realizar una variedad de operaciones sobre dichos archivos, por ejemplo³¹:

- Agregar nuevos archivos vacíos a la base de datos;
- Insertar datos dentro de los archivos existentes;
- Recuperar datos de los archivos existentes;
- Modificar datos en archivos existentes;
- Eliminar datos de los archivos existentes;
- Eliminar archivos existentes de la base de datos.

Las bases de datos es un soporte digital que tiene como fin el almacenamiento masivo en formato texto plano. En el caso de querer almacenar imágenes o archivos de sonido u otro tipo de dato por el estilo, lo que se almacena es la ruta de donde se encuentra el archivo como tal.

En la actualidad existen dos tipos de base de datos: las base de datos orientados a objetos con su gestor correspondiente más conocido como OODBMS (Object Oriented Database Management System, lo que significa en español: sistema de administración de base de datos orientadas a objetos). Y las base de datos relacionales con sus

³¹ MARÍA, S. L. *Introducción a los sistemas de base de datos*, [Consultada Marzo 4, 2009]. Disponible en: http://books.google.com/books?hl=es&lr=lang_es&id=Vhum351T-K8C&oi=fnd&pg=PA1&dq=base+de+datos+definicion&ots=fwJ9RVb75h&sig=vHa9cFEmpUriWy7qkvjJleZa5XQ#PPA2,M1

respectivos gestores o RDBMS (Relational Database Management System, en español sistema de administración de base de datos relacionales).

Característica de las RDBMS

- ✓ Los datos constituyen un ente propio y van completamente separados de las funciones que los manejan.
- ✓ Los datos pueden existir sin la existencia de las funciones que los manejan y viceversa.
- ✓ Debido a esa característica se crea una complejidad añadida a la hora de manejar los datos.

Característica de las OODBMS

- ✓ Los datos coexisten junto a los procesos que los tratan.
- ✓ Las entidades son los objetos.
- ✓ Simplifica el tratamiento de los datos.

Base de Datos Orientada a Objetos

Las bases de datos orientadas a objetos permiten el desarrollo y mantenimiento de aplicaciones complejas con un costo significativamente menor. Permiten que el mismo modelo conceptual se aplique al análisis, diseño, programación, definición y acceso a la base de datos.

Las bases de datos orientadas a objetos toman la idea de las bases inteligentes de datos a su conclusión lógica. No se tiene acceso a dato alguno sino es a través de los

métodos almacenados en la base de datos. Estos métodos están listos para entrar en acción al momento en que reciben una solicitud. Los datos de todos los objetos quedan entonces encapsulados. En general los datos son activos más que pasivos.³²

Historia de las BD Orientadas a Objetos

Las bases de datos orientadas a objetos surgieron en un principio para soportar la programación orientada a objetos. Los programadores de Smalltalk y C++ necesitaban almacenar lo que llamaban datos persistentes. Las BDOO se volvieron importantes para cierto tipo de aplicaciones con datos complejos, como CAD (diseño apoyado por computadora) y CAE (Ingeniería apoyada por computadora).³³

DB4O

Db4o (Data Base for Object, por sus siglas en inglés) es una base de datos orientada a objetos de código abierto para Java y .NET. Se afirma ser la única base de datos que es nativa para ambos lenguajes de programación, de Java y .Net.³⁴

DB4o es una solución de persistencia ideal cuando se necesita una base transparente y fácil de utilizar, ésta se puede incorporar rápidamente en un proyecto Java o .NET sin sacrificar desempeño. Logra acortar los tiempos de entrega de las aplicaciones de forma significativa, ya que permite concentrarse directamente en el dominio del problema (los objetos). Además, ofrece características avanzadas, como soporte

³² MERINO, C. U. *Desarrollo de sistema Orientada a Objetos para la Biblioteca del Seminario Metropolitano de Concepción*. pág. 23, Universidad del BIO-BIO, 2007. 360. p.

³³ MERINO, C. U. *Desarrollo de sistema Orientada a Objetos para la Biblioteca del Seminario Metropolitano de Concepción*. pág. 32, Universidad del BIO-BIO, 2007. 360. p.

³⁴ WEEKLY, C. *Hot skills: cut costs and coding time with Database for Objects.(Career Moves)*. [En línea 2007]. [Consultada Marzo 4, 2009]. Disponible en: http://find.galegroup.com/itx/retrieve.do?contentSet=IAC- Documents&resultListType=RESULT_LIST&qrySerId=Locale%28es%2C%29%3AFQE%3D%28KE%2CNone%2C4%29DB4O%3AAnd%3ALQE%3D%28AC%2CNone%2C8%29fulltext%24&sgHitCountType=None&inPS=true

transparente a cambios en las versiones de los objetos (evolución del esquema), cero administración y consultas nativas.

Es un novedoso motor de base de datos orientada a objetos. Las claves innovadoras de este producto son su alto rendimiento (sobre todo en modo embebido) y el modelo de desarrollo que proporciona a las aplicaciones para su capa de acceso a datos, el cual propugna un abandono completo del paradigma relacional de las bases de datos tradicionales.

Actualmente este producto funciona como una librería. La mayor clave del éxito que está teniendo este motor de base de datos frente a otros competidores que han desarrollado tecnologías similares, es que se ha optado por la licencia dual GPL/comercial. Es decir, si se quiere desarrollar software libre con esta librería, su uso no conlleva ningún coste por licencia. DB4O reduce el tiempo y costo de desarrollo y provee un desempeño superior. Almacena incluso las estructuras de objetos más complejas con una sola línea de código.

Dentro de sus principales características se encuentran:

- ✓ **Alto rendimiento:** Ofreciendo notables ventajas con sistemas que utilizan objetos anidados o compuestos, o en donde existen referencias cruzadas, herencia o interacciones ricas entre los objetos.
- ✓ Por su **bajo consumo de recursos**, es especialmente apta para dispositivos móviles y entornos Clientes/Servidor, aunque no necesariamente limitada sólo a ellos.
- ✓ **Doble licencia:** GPL (Open Source) y Comercial (que incluye soporte).
- ✓ **Gran comunidad de usuarios.** Alto nivel de respuesta y participación.

- ✓ **Documentación:** Clara, amplia y ordenada. Orientada a ejemplos y de fácil lectura.
- ✓ **Dos modos de trabajo:** Embebido y Cliente/Servidor
- ✓ **Transparencia Persistente / Consultas Nativas.**
- ✓ Soporte de **versionado.**
- ✓ **Portabilidad** entre .Net, Mono y Java.
- ✓ **Cientes de peso** avalan su uso: BMW, Intel, Boeing, Ricoh, Seagate, Bosch, Novell, etc.

1.7. Tecnologías a Utilizar

Después de la descripción de las tecnologías anteriores se ha llegado a la conclusión de que para realizar el desarrollo de la herramienta la arquitectura queda definida de la siguiente forma: Obtener un producto sobre Windows programado en JAVA mediante el IDE de desarrollo Netbeans 6.0 por las facilidades que el entorno de trabajo de éste brinda. Además se utilizará como base de datos para la aplicación DB4O.

Las tecnologías escogidas anteriormente responden cabalmente a las funcionalidades que se quiere que tenga la herramienta. Además de convertirse en un aporte del Sistema de Monitoreo y Análisis de Noticias (SIMAN) del Grupo de Investigación y Desarrollo de Internet (GIDI) al desarrollo del software libre.

1.8. Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software describen los pasos que se deben seguir para la producción de un determinado producto informático, a través de patrones, que surgieron gracias a la experiencia en un determinado campo del desarrollo de software,

como es el caso de los patrones de diseño, los cuales sugieren una serie de pasos a la hora de realizar un diseño determinado.

Las metodologías se clasifican en dos grupos fundamentales: metodologías estructuradas y las orientadas a objetos, aunque también existen las tradicionales y las metodologías ágiles.

Hay que destacar que existen tantas metodologías como forma de desarrollar productos existen. Es decir, una metodología no es aplicable a todos los proyectos, ya que puede que no se ajuste al proceso de desarrollo del mismo. Por esa razón antes de comenzar a desarrollar cualquier software se hace un estudio de cuál metodología es la más apropiada para llevar a cabo la construcción del producto software.

Metodologías Estructuradas

Estas metodologías fueron creadas en los años 70 según surgió la programación estructurada, por lo que es potencialmente efectivo en proyectos que utilizan como lenguajes de desarrollo los de tercera y cuarta generación. Ejemplos de estas metodologías son: MERISE2 (Francia), MÉTRICA3 (España), SSADM4 (Reino Unido), Gane & Sarson5, Ward & Mellor6, Yourdon & DeMarco7 e Information Engineering8.

La metodología estructurada está basada en la representación de las funciones que debe realizar el sistema y los datos que fluyen entre ellas.³⁵

Metodologías Orientadas a Objetos:

En la Metodología Orientada a Objetos se utiliza el UML, mediante el cual se puede representar diagramas (casos de uso) que permiten definir el sistema desde el punto de vista del usuario estableciendo las relaciones entre el futuro sistema y su entorno. Estas relaciones se establecen en forma de acciones del usuario y reacciones del sistema.

³⁵ AGUT, R. M. *Especificación de Requisitos Software según el estándar de IEEE 830*, [En línea 2000-2001]. [Consultada Marzo 4, 2009]. Disponible en: http://www3.uji.es/~coltell/Docs/IngSw_Apuntes/ISW_2004_D08.pdf

Las Metodologías Orientadas a Objetos están muy ligadas a los lenguajes de programación orientados a objetos como el caso de JAVA 9, C#, C++, etc. Con el fin de poder unificar los métodos y notaciones de este gran paradigma. Booch y Rumbaugh proponen el método unificado mediante el cual surge la que es hoy (a consideración de muchos expertos además de ser la más popular) la mejor notación orientada a objetos, el Unified Modeling Language (UML), que en español quiere decir lenguaje de modelado unificado. Muchas metodologías utilizan esta notación como es el caso del Rational Unified Process (RUP), OPEN12 y MÉTRICA, OMT. Las cuales son orientadas a objetos.

Metodologías tradicionales

Al inicio el desarrollo de software era artesanal en su totalidad. La ausencia de procesos formales, lineamientos claros, determinaron que se importara la concepción y fundamentos de metodologías existentes en otras áreas, y adaptarlas al desarrollo de software. Esta nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera secuencial, que de algo mejoraba la necesidad latente en el campo del software.

Entre las principales metodologías tradicionales existen los tan conocidos RUP y MSF entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto, definido en la fase inicial del desarrollo del proyecto.

Estas metodologías se caracterizan por ser:

- Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
- Cierta resistencia a los cambios.
- Impuestas externamente.

- Proceso mucho más controlado, con numerosas políticas/normas.
- Existe un contrato prefijado.
- El cliente interactúa con el equipo de desarrollo mediante reuniones.
- Grupos grandes y posiblemente distribuidos.
- Tienen más artefactos que las metodologías ligeras.
- Tienen más roles que las metodologías ligeras.
- La arquitectura del software es esencial y se expresa mediante modelos.

Otra de las características importantes dentro de este enfoque, son los altos costos al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil. Las metodologías tradicionales (formales) se focalizan en documentación, planificación y procesos (plantillas, técnicas de administración, revisiones, etc.).

Metodologías ágiles

Estas metodologías se basan en dos aspectos puntuales, el retrasar las decisiones y la planificación adaptativa; permitiendo potenciar aún más el desarrollo de software a gran escala. Ya que tienen un desarrollo de software incremental, es decir se van entregando pequeñas porciones de software en períodos de tiempo cortos.

Además de que los desarrolladores y clientes trabajan juntos, permitiendo de esta forma una cercana comunicación entre ellos, logrando controlar de una forma efectiva cualquier cambio que se produzca en el proceso de desarrollo del software. Y por sobre todo es muy sencillo pues es muy fácil de aprender, de modificar y muy bien documentado.

Dentro de estas metodologías se encuentran el Extreme Programming (XP), el SCRUM, incluso el RUP podría llegar a ser ágil realizándole una configuración adecuada dado por el especial énfasis que presenta en cuanto a su adaptación a las condiciones del proyecto, Dynamic Systems Development Method (DSDM), entre otros.

Estas metodologías se caracterizan por ser³⁶:

- Basadas en heurísticas provenientes de prácticas de producción de código.
- Especialmente preparados para cambios durante el proyecto.
- Impuestas internamente (por el equipo).
- Proceso menos controlado que las metodologías pesadas, con pocos principios.
- No existe contrato tradicional o al menos es bastante flexible.
- El cliente es parte del equipo de desarrollo.
- Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.
- Presenta pocos artefactos.
- Presenta pocos roles.
- Tienen menos énfasis en la arquitectura del software.

³⁶ ISSI, G. *Metodologías Ágiles en el Desarrollo de Software*, [En línea 2003]. [Consultada Marzo 5, 2009]. Disponible en:
<http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf#page=9>

Metodologías más Usadas

Las metodologías más usadas en la actualidad dados por las potencialidades y las capacidades de adaptabilidad que tienen son el XP, el RUP, el SCRUM, MSF y la más reciente la OMT.

Rational Unified Process (RUP)

El RUP (Rational Unified Process) especifica un framework para el desarrollo de un proyecto, en particular un proyecto de desarrollo de software, definiendo: etapas, actividades a realizar por un equipo de desarrollo, secuencia y lógica necesaria para obtener el producto final.³⁷

RUP es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto).

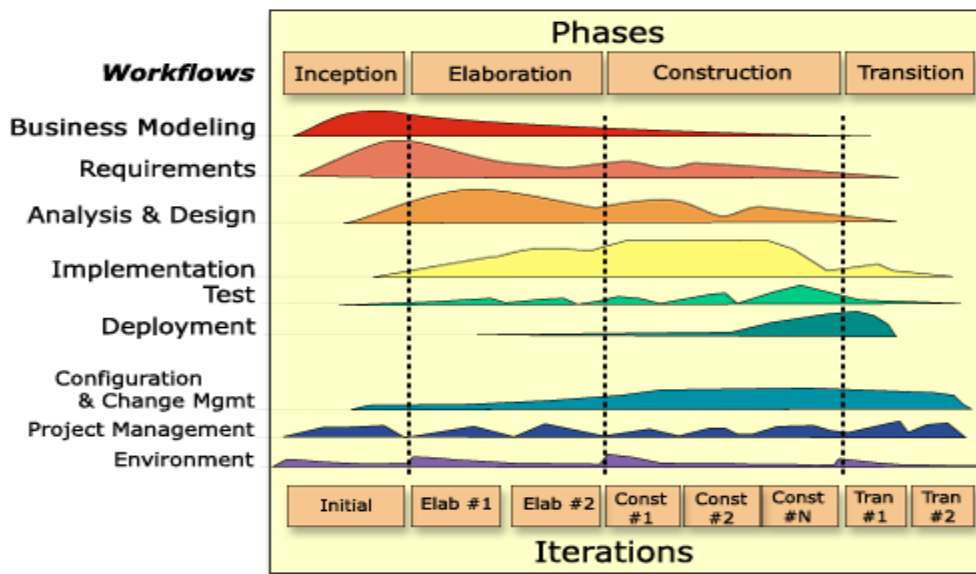
Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación.

El RUP y los productos similares, como Object-Oriented Software Process (OOSP), y el OPEN Process, son herramientas de la ingeniería del software que combinan los aspectos procesales de desarrollo (como fases definidas, técnicas, y prácticas) con otros componentes de desarrollo (como documentos, modelos, manuales, código fuente, etc.) dentro de un framework unificado.

Está formado por cuatro fases y 9 flujos de trabajo. Cada una de estas etapas o fases es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo

³⁷ RIESCO, D. *Un Workflow que Automatice los Procesos de Negocios del Proceso Unificado Rational*, [En línea 2004]. [Consultada Marzo 5, 2009]. Disponible en: <http://users.dsic.upv.es/~dromero/papers/wicc2004.pdf>

de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Además de ser guiado por casos de uso, centrado en la arquitectura y se caracteriza por ser iterativo e incremental.



Esquema de Trabajo RUP

Fases de RUP

Las cuatro fases del ciclo de vida son:

- Concepción (Inception)
- Elaboración (Elaboration)
- Construcción (Construction)
- Transición (transition)

Flujos de Trabajo

Los flujos de trabajo son nueve, de ellos:

Seis son de ingeniería

1. **Modelamiento del Negocio:** que es donde se entiende las necesidades del negocio.

2. **Requerimientos:** Trasladando las necesidades del negocio a un sistema automatizado.
3. **Análisis y Diseño:** Trasladando los requerimientos dentro de la arquitectura de software.
4. **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
5. **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
6. **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

Y tres son de soporte.

7. **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
8. **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
9. **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización

Ventajas

- Evaluación en cada fase que permite cambios de objetivos
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.

Desventajas

- La evaluación de riesgos es compleja
- Excesiva flexibilidad para algunos proyectos
- Se corre el riesgo de poner al cliente en una situación que puede ser muy incómoda para él.
- El cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

Microsoft Solution Framework (MSF)

Microsoft Solutions Framework (MSF) ha desarrollado un proceso para identificar y valorar ininterrumpidamente los riesgos de un proyecto, dar prioridad a estos riesgos e implementar las estrategias para tratarlos de forma proactiva a lo largo del ciclo de vida del proyecto, tal como se define en el Modelo de procesos de MSF.³⁸

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información.

El modelo de equipos de MSF (MSF team model) fue desarrollado para compensar algunas de las desventajas impuestas por las estructuras jerárquicas de los equipos en los proyectos tradicionales.

Los equipos organizados bajo este modelo son pequeños y multidisciplinarios, en los cuales los miembros comparten responsabilidades y balancean las destrezas del equipo para mantenerse enfocados en el proyecto que están desarrollando. Comparten una visión común del proyecto y se enfocan en implementar la solución, con altos estándares de calidad y deseos de aprender.

³⁸ ROMERO, A. *Gestión de riesgos con CMMI, RUP e ISO en ingeniería de software minero* [En línea 2007]. [Consultada Marzo 6, 2009]. Disponible en: http://www.scielo.org.pe/scielo.php?pid=S1561-08882007000100005&script=sci_arttext&tlng=es

MSF tiene las siguientes características:

- ✓ Adaptable: es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- ✓ Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- ✓ Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.
- ✓ Tecnología Agnóstica: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

Extreme Programming (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.³⁹

Es la más destacada de los procesos ágiles de desarrollo de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

³⁹ LETELIER, P. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)* [Consultada Marzo 6, 2009]. Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Las características fundamentales del método son:

- ✓ Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- ✓ Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- ✓ Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone una mayor calidad del código escrito de esta manera, ya que el código es revisado y discutido mientras se escribe.
- ✓ Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- ✓ Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- ✓ Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.

- ✓ Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores sean detectados.
- ✓ Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Ventajas

- ✓ Apropiado para entornos volátiles
- ✓ Estar preparados para el cambio, significa reducir su coste.
- ✓ Planificación más transparente para los clientes, conocen las fechas de entrega de funcionalidades vitales para su negocio
- ✓ Permitirá definir en cada iteración cuales son los objetivos de la siguiente
- ✓ Permite tener retroalimentación de los usuarios lo cual es muy útil.
- ✓ La presión esta a lo largo de todo el proyecto y no en una entrega final

Desventajas

- ✓ Delimitar el alcance del proyecto con el cliente. Para mitigar esta desventaja se plantea definir un alcance a alto nivel basado en la experiencia.

SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos.

El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.⁴⁰

SCRUM es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas.

SCRUM se utiliza como marco para otras prácticas de ingeniería de software como RUP o Extreme Programming.

SCRUM se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad.

Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el

⁴⁰ LETELIER, P. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)* [Consultada Marzo 6, 2009]. Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

producto que se está construyendo a las necesidades del cliente. Se busca entregar software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.

En SCRUM, el equipo se focaliza en una única cosa: construir software de calidad. Por otro lado, la gestión de un proyecto SCRUM se basa en definir cuales son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en remover cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. Se busca que los equipos sean lo más efectivos y productivos posible.

OMT

La metodología OMT (Object Modeling Technique) fue creada por James Rumbaugh y Michael Blaha en 1991, mientras James dirigía un equipo de investigación de los laboratorios General Electric.

OMT es una de las metodologías de análisis y diseño orientados a objetos, más maduras y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software.

La OMT intenta abstraer la realidad utilizando tres clases de modelos Orientados a Objetos (OO): el modelo de objetos, que describe la estructura estática, el modelo dinámico, con el que describe las relaciones temporales entre objetos; y el modelo funcional que describe las relaciones funcionales entre valores. Mediante estas tres fases de construcción de modelos, se consigue una abstracción de la realidad que tiene en si misma información sobre las principales características de ésta.⁴¹

Las fases que conforman a la metodología OMT son:

⁴¹ ANÓNIMO. *Ingeniería de Software con UML Unified Modeling Language - Lenguaje Unificado de Modelado*, [Consultada Marzo 6, 2009]. Disponible en: <http://www.eduardoleyton.com/apuntes/Uml.pdf>

Análisis. El analista construye un modelo del dominio del problema, mostrando sus propiedades más importantes. El modelo de análisis es una abstracción resumida y precisa de lo que debe de hacer el sistema deseado y no de la forma en que se hará. Los elementos del modelo deben ser conceptos del dominio de aplicación y no conceptos informáticos tales como estructuras de datos. Un buen modelo debe poder ser entendido y criticado por expertos en el dominio del problema que no tengan conocimientos informáticos.

Diseño del sistema. El diseñador del sistema toma decisiones de alto nivel sobre la arquitectura del mismo. Durante esta fase el sistema se organiza en subsistemas basándose tanto en la estructura del análisis como en la arquitectura propuesta. Se selecciona una estrategia para afrontar el problema.

Diseño de objetos. El diseñador de objetos construye un modelo de diseño basándose en el modelo de análisis, pero incorporando detalles de implementación. El diseño de objetos se centra en las estructuras de datos y algoritmos que son necesarios para implementar cada clase. OMT describe la forma en que el diseño puede ser implementado en distintos lenguajes (orientados y no orientados a objetos, bases de datos, etc.).

Implementación. Las clases de objetos y relaciones desarrolladas durante el análisis de objetos se traducen finalmente a una implementación concreta. Durante la fase de implementación se debe tener en cuenta los principios de la ingeniería del software de forma que la correspondencia con el diseño sea directa y el sistema implementado sea flexible y extensible.

Resultado del estudio de las metodologías

Después de un detallado estudio de las metodologías anteriores se decidió utilizar para el desarrollo del sistema a RUP y la notación UML, por las potencialidades que presentan. Además de propiciar una buena organización y utilización de la información

producida a lo largo de todo el proceso de desarrollo del software pudiendo darle el alcance que se espera tenga esta herramienta.

1.9. Fundamentación de la solución propuesta.

Luego del análisis y estudio realizado en este capítulo, se ha decidido como propuesta para la solución del problema, desarrollar una aplicación de escritorio. Debido a que se necesita un alto nivel de procesamiento y que la versión que se propone desarrollar es la autónoma, es decir una aplicación que se utilice en un puesto de trabajo, no en un grupo de trabajo. Además de que con una aplicación desktop o de escritorio todas las máquinas, no importa las prestaciones que tengan, podrán ser capaces de conectarse a internet y descargar la información a sus bases de datos locales.

Por lo que para la realización de la aplicación de manera más eficiente y que cumpla con los requerimientos propuestos por el proyecto, se proponen las siguientes herramientas:

- ❖ Entorno de desarrollo: Netbeans 6.0
- ❖ Lenguaje de Programación: JAVA
- ❖ Motor de Base de Datos: DB4O

La aplicación se desarrollará en el lenguaje de programación Java aprovechando las potencialidades del Netbeans 6.0 como entorno de desarrollo y a la vez explotar las funcionalidades del motor de base de datos DB4O para lograr una base de datos con un buen rendimiento .

En este capítulo se analizaron las características principales de algunos programas, sistemas y herramientas que realizan vigilancia, monitoreo y observación, se ha dado una panorámica de los existentes actualmente en Cuba y el mundo. Se realizó un estudio de los canales de sindicación con el objetivo de detectar tendencias y tecnologías existentes que sirvan de apoyo. Se fundamentó la utilización del lenguaje, el sistema gestor de base de datos y la metodología de desarrollo de software a utilizar en el desarrollo de la propuesta de la solución, así como algunos conceptos fundamentales a tener en cuenta.

Capítulo 2. Elementos de la Arquitectura.

En este capítulo se presentan los elementos de arquitectura utilizados para llevar a cabo el desarrollo del Módulo de Búsqueda del Sistema de Monitoreo y Análisis de Noticias. Abordando lo requerimientos no funcionales imprescindibles para que el sistema funcione correctamente y con buen rendimiento, se fundamenta la arquitectura a utilizar, también se realiza un análisis crítico de posibles implementaciones y componentes existentes que se reutilizaron, así como las estrategias de integración del mismo, aspectos de seguridad, la vista de despliegue del sistema y por último pero no menos importante el estándar de codificación a utilizar.

2.1. Requerimientos No Funcionales.

Los requisitos o requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

No se refieren a funciones específicas que proporciona el sistema. Son restricciones de los servicios o funciones ofrecidas por el sistema y surgen de las necesidades del usuario.

Por el hecho de que se llamen “no funcionales” no quiere decir que no aporten a la funcionalidad del sistema que se esté desarrollando ya que sirven como complemento a los requisitos funcionales que debe cumplir el sistema. Un ejemplo de lo anteriormente planteado sería que un sistema cualquiera podría tener como requisito funcional que toda la información con la que se trabaje en el mismo se debe guardar en una base de datos, y como requisito no funcional que las salvadas deben realizarse en un intervalo de tiempo determinado.

De no cumplir con ese requisito no funcional, a pesar de estar cumpliendo con la funcionalidad, el sistema no estaría funcionando correctamente y por ende no cumplir con las necesidades del cliente o del usuario.

Además los requerimientos no funcionales pueden marcar la diferencia entre productos que cumplan con todas las funcionalidades requeridas. Ya que son los que pueden

Llegar a definir el grado de aceptación de un producto determinado como pueden ser: la fácil manipulación del sistema por parte del usuario, que el usuario pueda realizar una operación determinada por más de una vía, en fin son los que le dan un rostro al producto a mostrar.

A continuación se detallan los requerimientos no funcionales del Sistema de Monitoreo y Análisis de Noticias:

Requerimientos de apariencia o interfaz externa:

- El sistema debe tener un ambiente amigable y entendible para los usuarios finales, de forma tal que no les sea muy complicado utilizar el software.
- El diseño de la interfaz debe ser sobrio, formal y con colores que no sean agresivos a la vista.
- La información estará bien distribuida en cada una de las pantallas.
- Identificación de colores y formatos con acciones del sistema.

Requerimientos de usabilidad:

- El sistema podrá ser usado por cualquier persona que tenga conocimientos básicos en el manejo de la computadora y esté interesada trabajar en la aplicación.
- Debe brindarse comodidad a la hora de acceder a las diferentes funcionalidades que brinda la aplicación mediante teclas de acceso rápido, la navegabilidad no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por el usuario.

Requerimientos de rendimiento:

- Se debe garantizar que la respuesta a solicitudes de los usuarios del sistema se realice en un breve período de tiempo para la toma de decisiones.
- Se debe tener en cuenta que se trabaja con conexiones a internet, debido a que parte de las actualizaciones de la información dependen de la actualización que se lleve a cabo en su lugar de origen.

Requerimientos de Soporte:

- Se debe garantizar la inserción de módulos nuevos que añadan funcionalidad al sistema, sin afectar lo realizado hasta el momento. El sistema debe estar bien documentado de forma tal que el tiempo de mantenimiento sea mínimo.
- Se le debe dar mantenimiento periódicamente a las bases de datos de los clientes controlando la integridad de la información.

Requerimientos de portabilidad:

- El producto podrá ser usado bajo cualquier sistema operativo ya sea Linux o Windows.

Requerimientos de seguridad y privacidad:

- Se garantizará la seguridad a todos los niveles (Interfaz, negocio y Acceso a datos), ya que todos los usuarios del sistema deberán estar registrados en el mismo y contarán con un escritorio de trabajo personal, aunque podrán acceder a las noticias de los demás usuarios.
- Para la conexión a internet se realizará mediante la cuenta del dominio uci de cada usuario.

Requerimientos de confiabilidad:

- Deberá prevenir los posibles fallos y/o errores que pudieran presentarse y posibilitar una rápida recuperación en dichos casos.

Requerimientos de ayudas y documentación en línea:

- Se debe brindar una interfaz amigable que explique las diferentes funcionalidades con que cuenta el sistema de manera rápida, además el sistema contará con una ayuda integrada.

Requerimientos de hardware:

- Requerimientos para una estación de trabajo: 256Mb RAM (Recomendado 512Mb), 1GHz (o superior) de micro, 10Gb HDD.
- Requerimientos para un servidor: 512Mb RAM (Recomendado 1Gb RAM o superior), 1GHz o superior, 60Gb HDD.

Requerimientos de software:

- El sistema debe correr en sistemas operativos Windows 98 o superior y sistemas Unix, Linux utilizando la máquina virtual de java 6 o superior.

2.2. Descripción de la Arquitectura. Fundamentación.

“La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.”

Kruchten, Philippe

La arquitectura del software es una disciplina que a pesar de conocerse desde los años '60 por grupos de personalidades en la informática como Dijkstra. No se hace popular hasta 1990 con la crisis del software, que azotó en ese año. Entre otros factores, debido a que se realizaba la producción de software sin un patrón o una buena práctica a seguir. Como consecuencia, el resultado final después de grandes inversiones y empleo de un tiempo considerable, se obtenían productos altamente acoplados y en muchos casos con muy alta cohesión.

Lo que provocaba que si se detectaba un error en algunos de los módulos desarrollados paralelamente, la corrección del mismo resultaba una tarea titánica. En muchos casos significaba rehacer gran parte del trabajo, porque existía una dependencia total o parcial entre los módulos. Por esta razón es que alcanzan gran auge la arquitectura en capas, la de cliente/servidor, la modelo vista controlador (MVC), etc.

Modelo Vista Controlador (MVC):

Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Esto no quiere decir que no se pueda aplicar a aplicaciones de escritorio, ya que las mismas al aplicar esta arquitectura ganan mucho en claridad y legibilidad en el código, además de lograr una organización óptima de las clases utilizadas en el desarrollo del producto que se desee implementar. Sin mencionar que un buen uso del MVC, contribuye en gran medida a la robustez de la aplicación en general.

El patrón de diseño software MVC establece que la arquitectura se divide en tres partes fundamentales:

Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

La principal ventaja de esta separación reside en la facilidad para realizar cambios en la aplicación puesto que:

- Cuando se realiza un cambio de bases de datos, programación o interfaz de usuario solo se trabaja con uno de los componentes sin que afecte al resto de los componentes.
- Se puede modificar uno de los componentes sin conocer cómo funcionan los otros.

En el caso del módulo de búsqueda del Sistema de Monitoreo y Análisis de Noticias (SIMAN) se aplicará el modelo de arquitectura analizado anteriormente por lo que se tendrán las clases persistentes y las de acceso a datos en un paquete llamado Modelo, la clase controladora en otro paquete llamado Controlador, y las clases correspondientes a la interfaz de usuario se ubicarán en el paquete Vista.

Patrones de Diseño

Los patrones de diseño se han ganado un lugar muy especial dentro de la lógica de programación de prácticamente todos los programadores. Incluso hoy en día no existen programadores que no los hayan aplicado a sus trabajos. Quizás hasta lo hayan hecho de forma inconsciente, simplemente porque al hacerlo de una forma determinada el código adquiere más robustez y a la vez le da más seriedad al producto que presentan.

Es real que existe una amplia gama de patrones de diseño, ya que muchos implementadores según les iba surgiendo la necesidad, iban creando el suyo propio. Hoy en día no es necesario hacerlo, ya que para eso existen los patrones. Los mismos son una especie de guías prácticas basadas en la experiencia de otras personas en el proceso de desarrollo de software. Por lo que simplemente con buscar los patrones que más le acomoden al implementador y por supuesto sean aplicables al tipo de proyecto que se llevará a cabo, se logra obtener componentes realmente reutilizables y robustos en menor tiempo.

Por otro lado, los patrones de diseño, facilitan el aprendizaje al programador inexperto, pudiendo establecer parejas problema-solución. Además, los patrones de diseño, también ayudarán a especificar las interfaces, identificando los elementos claves en las interfaces y las relaciones existentes entre distintas interfaces.

Patrones de Diseño Utilizados

En el desarrollo de la aplicación se decidió utilizar los patrones de diseño GRASP (General Responsibility Assignment Software Patterns, por sus siglas en inglés), definidos anteriormente en el análisis y diseño del módulo de búsqueda del SIMAN, estos describen los principios fundamentales de la asignación de responsabilidades a objetos. Constituyen el fundamento de cómo se va a diseñar el sistema finalmente. Es importante que tanto el diseñador de software como el implementador dominen y apliquen estos conocimientos antes y durante el desarrollo de un sistema cualquiera. Los patrones GRASP son:

Patrón Experto: ¿Cómo asignar responsabilidades de la forma más eficiente?

Asignar una responsabilidad al experto en la información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto provee un bajo nivel de acoplamiento.

Patrón Creador: ¿Quién debería ser responsable de crear una nueva instancia de alguna clase? Las responsabilidades de crear una instancia de la **clase A** se le dará a aquella **clase B**, en los siguientes casos:

- B agrega los objetos de A
- B contiene los objetos de A
- B registra las instancias de los objetos de A.
- B tiene los datos de inicialización que serán enviados a A cuando este objeto sea creado.

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.

Patrón Controlador: ¿Quién debería encargarse de atender un evento del sistema? El controlador es un intermediario entre la interfaz de usuario y el núcleo de las clases donde reside la lógica de la aplicación. El controlador no realiza mucho trabajo por si mismo; más bien coordina la actividad de otros objetos. Asignar la responsabilidad del manejo de mensajes de los eventos del sistema a una clase que represente alguna de las siguientes opciones:

- El sistema global.
- La empresa u organización global.
- Algo activo en el mundo real que pueda participar en la tarea.
- Un manejador artificial de todos los eventos del sistema de un caso de uso (controlador de casos de uso)

Patrón Bajo Acoplamiento: ¿Cómo dar soporte a una mínima dependencia y a un aumento de la reutilización?

Una clase con bajo acoplamiento no depende de “muchas otras” clases. Las clases con alto acoplamiento recurren a muchas clases y no es conveniente. Son más difíciles de mantener, entender y reutilizar.

Patrón Alta Cohesión: Es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño.

Una clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. Su alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización.

2.3. Análisis de posibles implementaciones, componentes o módulos ya existentes.

En la actualidad la reusabilidad de componentes que ya han sido desarrollados con anterioridad, se ha convertido en una práctica a seguir por casi todos los desarrolladores del mundo. Ya que se logra reducir el tiempo de desarrollo al máximo, se construyen aplicaciones más robustas y eficientes. Además de que se puede

implementar en paralelo y reducir las formaciones de “cuello de botella” (término asignado en la ingeniería de software, al proceso de esperar a que se implemente o desarrolle una funcionalidad determinada para comenzar a desarrollar otra) en la producción.

En el módulo de búsqueda del Sistema de Monitoreo y Análisis de Noticias no se emplearon componentes ya desarrollados, pero si se hizo uso de librerías externas al IDE de desarrollo, que sirvieron para optimizar el tiempo de desarrollo del mismo. Estas librerías se detallan a continuación.

Lucene:

Lucene es una librería escalable de alto rendimiento para la recuperación de información, que permite agregar capacidades de indexación y búsqueda a las aplicaciones. Puede indexar⁴² y buscar cualquier dato que pueda ser convertido a formato textual. Se trata de una API de indexación y búsqueda con simplicidad en su utilización que requiere básicamente que el usuario aprenda el uso de sus clases.

Lucene está implementado originalmente en el lenguaje de programación Java, y es miembro de la familia Apache Jakarta, siendo un proyecto de código abierto gratuito, muy maduro y fácil de utilizar. Durante muchos años ha sido la librería gratuita más utilizada bajo Java para recuperación de información.

Lucene, siendo una librería, no es una aplicación finalizada lista para ser utilizada, como lo pudiera ser un programa buscador de archivos o un buscador de sitios Web. Sino más bien es una librería de software utilizada como herramienta para incorporar capacidades de búsqueda a las aplicaciones.

Esta librería fue creada por Doug Cutting, creador de Nutch, primer buscador online de código abierto. Son muchas las empresas y entidades que utilizan esta librería, debido al éxito que ha tenido.

⁴² Proviene de la palabra en inglés index que dentro de una de sus acepciones significa ponerle un índice a algo según WordReference.com.

El éxito del software de código abierto puede medirse por la cantidad de veces que ha sido traducido a otros lenguajes de programación. En el caso de Lucene ha sido portada a otros lenguajes de programación como: Java, C, C++, Python, Perl, Ruby, PHP, entre otros.

En el caso del desarrollo del módulo del SIMAN se escogió esta librería y se optimizó en gran medida el tiempo de desarrollo, ya que no se tuvo que desarrollar ningún sistema de recuperación de información. Sino simplemente utilizar lucene, la que resalta por la simplicidad con la que se indexan y se buscan objetos. Para más información sobre la herramienta dirigirse a su sitio oficial <http://lucene.apache.org>

Xerces

Xerces es una librería que se encarga de la generación o parseo de archivos XML, es miembro de la familia de software libre y además es uno de los más populares y utilizados a nivel mundial en cuanto a parseadores se trata.

Un parseador XML (XML Parser, en inglés) no es más que un programa capaz de leer una secuencia de XML. Esta secuencia de datos obtenida es convertida a una estructura de datos útil para el programa, el mismo provee llamadas para informar al parseador sobre nuevos datos o de lo contrario obtener datos de la secuencia de XML, incluida la estructura jerárquica de las etiquetas XML disponibles para el programa.⁴³

En el módulo de búsqueda del SIMAN se utiliza Xerces para poder parsear los canales RSS con los que cuentan los medios noticiosos (sumarios), los cuales se ponen en la Web en forma de archivos XML. En estos archivos se escriben los principales elementos de una noticia como son el título, la descripción, la dirección electrónica, el autor, la fecha de publicación, además de los datos generales del sitio, como el nombre del medio, la dirección electrónica y demás datos del portal Web.

⁴³ MORIN, A. M. *XML Parser*, [En línea] 2009. [Consultada: Marzo 5, 2009]. Disponible en: <http://ruby.about.com/od/sz/g/xmlparse.htm>

Xerces facilita esta actividad desglosando las distintas partes por las que está compuesto el archivo XML, una vez hecho esto el programador sólo tiene que trabajar en base a la información devuelta por esta librería.

SAX Parser

SAX Parser (de ahora en adelante SAX) fue originalmente definido como un API de Java⁴⁴, pero ya ha sido portado a otros lenguajes orientados a objetos. La utilización de SAX es sencilla, el analizador lee el documento XML desde el inicio y avisa a la aplicación qué es lo que encontró, una vez hecho esto pues toda la información que se recupera es almacenada en estructuras de datos⁴⁵. Hay que aclarar que el SAX es un API implementado por muchos parseadores XML, incluyendo Xerces.

SAX proporciona mecanismos para leer desde los documentos XML. Tiene como una de sus principales potencialidades que es orientada a eventos desde la versión 2.0. No conlleva a la generación de estructuras internas, es decir no crea un árbol en memoria del documento que esté tratando, sino que simplemente según va leyendo el archivo XML va guardando lo leído en las estructuras de datos definidas, haciéndolo ideal para leer documentos muy extensos, ya que sobrecarga lo menos posible la memoria de la computadora.

Ventajas del SAX Parser

- Los datos se pueden comenzar a procesar tan pronto como se reciben. Ganando en rapidez de respuesta.

- No se espera a leer todo el documento antes de actuar sobre los datos.

⁴⁴ Un API Java es una Interfaz de Programación de Aplicaciones provista por los creadores del lenguaje Java, y que da a los programadores los medios para desarrollar aplicaciones Java.

⁴⁵ Una estructura de datos es una entidad que posee varias funcionalidades tales como adicionar, eliminar insertar, etc. Existen varios tipos de estas entidades como: listas, pilas y colas. Además pueden ser reutilizables.

- El documento entero no tiene que residir en memoria, de esta forma se crean lectores XML mucho más rápidos a la hora de dar una respuesta.
- Es conveniente para documentos largos ya que sobrecarga menos el ordenador.

Parseador XML

EL parseador XML es uno de los principales componentes que se implementó en el módulo de búsqueda de SMAN. Para llevar a cabo esta actividad se investigó y analizó las estructuras de los parseadores XML existentes.

También se analizó línea por línea el código de los mismos para entender el funcionamiento de este tipo de herramientas. Para la construcción del parseador se empleó la API SAX, por las características que presenta la misma, las cuales se analizaron anteriormente.

Al principio el parseador sólo parseaba archivos locales, por lo que se le implementó la funcionalidad de conectarse a internet para poder parsear archivos colgados en la Web. Esto se logró a través del uso de sockets.

Los sockets no son más que puntos o mecanismos de comunicación entre procesos que permiten que un proceso hable (emita o reciba información) con otro proceso incluso estando estos procesos en distintas máquinas. Esta característica de interconectividad entre máquinas hace que el concepto de socket sea de gran utilidad. Un socket es al sistema de comunicación entre ordenadores lo que un buzón o un teléfono es al sistema de comunicación entre personas: un punto de comunicación entre dos agentes (procesos o personas respectivamente) por el cual se puede emitir o recibir información.⁴⁶

⁴⁶ BARRANCO, M. R. *Doctorado en informática sockets: comunicación entre procesos distribuidos*, [En línea] 1996. [Consultada Marzo 5, 2009]. Disponible en: <http://es.tldp.org/Universitarios/seminario-2-sockets.html>

Los sockets proporcionan una comunicación de dos vías, punto a punto entre dos procesos. Son muy versátiles y constituyen un componente básico de comunicación entre interprocesos e intersistemas. Un socket es un punto final de comunicación al cual se puede asociar un nombre. Este tiene un tipo y uno o más procesos asociados.

2.4. Análisis de las estructuras de datos utilizadas en la solución

En del módulo de búsqueda del SIMAN se utilizó la estructura de datos predefinida del Netbean (Entorno de desarrollo utilizado), listas genéricas. Este tipo de estructura es utilizada por casi todos los entornos de desarrollo. Ya que constituye parte del conocimiento básico de la Programación Orientada a Objetos (POO).

Las listas genéricas son utilizadas para almacenar información en tiempo de ejecución, por lo que constituyen la base principal en el desarrollo del módulo que se quiere lograr. Ya que en el mismo se puede encontrar, desde almacenando una simple colección de datos como cadenas de texto, hasta la representación de relaciones de agregación/composición.

Sobre todo hay que resaltar que las listas genéricas que brinda Netbean tienen muchas funcionalidades que son de vital importancia en el desarrollo de cualquier aplicación, y más aún cuando no se tiene que invertir tiempo en la implementación de las mismas. Dentro de sus principales funcionalidades se encuentran adicionar un elemento, eliminar un elemento en una posición determinada en la lista, así como insertar y devolver los elementos que posee en la posición en la que el programador desee obtener o insertar.

También se utilizó en menor cuantía los arreglos, sobre todo a la hora de crear colecciones de datos con un número de elementos conocidos con anterioridad, para realizar operaciones de dividir por partes cadenas de texto, para su posterior procesamiento.

2.5. Aspectos técnicos del DB4O

DB4O (Data Base for Object) es la base de datos (BD en adelante) que se utilizó en el módulo de búsqueda del SIMAN. Este motor de BD orientado a objetos reduce el tiempo de diseño de la BD, ya que el diseño es el mismo que el de las clases que van a persistir, no existen tablas por las que el diseñador de BD tenga que emplear grandes cantidades de tiempo para representar las posibles y a la vez difíciles relaciones entre tablas, las que si son muy complicadas, pueden llegar a generar nuevas relaciones, con sus respectivas complicaciones. Además de que al terminar todo este proceso, el diseñador de BD debe normalizarla.

La normalización⁴⁷ generalmente se debe llevar hasta la tercera forma normal ya que a partir de ahí es que se considera que es una base de datos bastante seria. En todo este proceso el diseñador podría tardar meses diseñando la BD.

Mientras que con DB4O se puede reducir el tiempo a menos de la mitad ya que brinda funcionalidades que no son muy complejas de utilizar y que agilizan el proceso de desarrollo, a continuación se brindan algunos ejemplos de la misma.

Evitar duplicidad de objetos

La integridad de objetos es algo que debe ser básico en cualquier BD y DB4O también lo garantiza de la siguiente forma:

⁴⁷ Es el análisis de dependencias funcionales entre atributos, su propósito es reducir complejas vistas de usuario a un conjunto de pequeñas y estables estructuras de datos, las estructuras de datos normalizadas son más flexibles, estables y más fáciles de mantener, que las estructuras no normalizadas.

ROTTA, I. L. Z. *Normalización*, [En línea] 2009. [Consultada Marzo 8, 2009]. Disponible en: <http://sebx.files.wordpress.com/2008/05/norm3ra.pdf>


```
45 public Configuration ConfigurarBD() {
46     Configuration config = Db4o.newConfiguration();
47     config.objectClass(CUsuario.class).generateUUIDs(true);
48     config.objectClass(CUsuario.class).generateVersionNumbers(true);
49     config.objectClass(CUsuario.class).objectField("usuario").indexed(true);
50     config.add(new UniqueFieldValueConstraint (Clases.CUsuario.class, "usuario"));
51     config.objectClass(CGrupoSumario.class).objectField("nombre").indexed(true);
52     config.add(new UniqueFieldValueConstraint (CGrupoSumario.class, "nombre"));
53     config.objectClass (Clases.CSumario.class).objectField("titulo").indexed(true);
54     config.add(new UniqueFieldValueConstraint (Clases.CSumario.class, "titulo"));
55     config.objectClass (CConfiguracionSistema.class).objectField("host").indexed(true);
56     config.add(new UniqueFieldValueConstraint (CConfiguracionSistema.class, "host"));
57     config.objectClass (CUsuario.class).updateDepth(50);
58     config.objectClass (CUsuario.class).cascadeOnUpdate(true);
59     config.objectClass (CUsuario.class).cascadeOnDelete(true);
60     config.objectClass (CGrupoSumario.class).cascadeOnUpdate(true);
61     config.objectClass (CGrupoSumario.class).cascadeOnDelete(true);
62     config.objectClass (CMedio.class).cascadeOnUpdate(true);
63     config.objectClass (CMedio.class).cascadeOnDelete(true);
64     config.objectClass (CSumario.class).cascadeOnUpdate(true);
65     config.objectClass (CSumario.class).cascadeOnDelete(true);
66     config.objectClass (CNoticia.class).cascadeOnUpdate(true);
67     config.objectClass (CNoticia.class).cascadeOnDelete(true);
68     return config;
69 }
```

Configurar la Base de Datos

En la imagen anterior se muestra de que forma DB4O crea llaves únicas para identificar a los objetos y evitar la duplicidad de los mismos. Se realiza mediante la creación de un objeto de tipo Configuration (línea 46 de la imagen) y después a este objeto se le van pasando las clases que serán indexadas así como los campos a indexar de las mismas (líneas de la 49 a la 56). También se define el nivel de profundidad de actualización que tendrá la BD (línea 57 de la imagen). Esta configuración que se presenta en la imagen es el método con el cual se configura la base de datos del módulo de búsqueda del SIMAN.

Crear objeto contenedor

El objeto contenedor es el encargado de realizar todas las operaciones en la base de datos, para crearlo existen dos formas:

Una es construirlo pasándole la configuración (realizada en la imagen anterior) que se desea que tenga y la otra es no pasarle ninguna configuración (no aconsejable ya que no se evita la duplicidad de objetos).

```
29     private ObjectContainer db = Db4o.openFile(ConfigurarBD(), "C:/prueba2.yap");
```

Creando objeto contenedor

En la imagen anterior se crea un contenedor llamado “db”, con una configuración determinada (llamada al método de la imagen anterior, “ConfigurarBD ()”), y se le dio la responsabilidad de crear una base de datos llamada “prueba2”, en la unidad C⁴⁸ del ordenador.

Para poder realizar cualquier operación sobre estos contenedores, primero se debe abrir el contenedor como se explicó anteriormente y una vez se termine cerrar el mismo.

```
public static void accessDb4o() {
    ObjectContainer db=Db4o.openFile("nombre de la base de datos");
    try {
        // se pone todo el código que se quiera ejecutar
    }
    finally {
        db.close();
    }
}
```

Adicionar un objeto

Para adicionar un objeto determinado se llama a la función “set” que facilita el contenedor declarado anteriormente:

⁴⁸ Partición del disco duro de una computadora, la cual puede tener un tamaño variable en dependencia del gusto del dueño de la computadora.

```
260 public void AdicionarUsuario(CUsuario pusuario)
261     throws Exception{
262     try
263     {
264         db.set(pusuario);
265         db.commit();
266     }
267     catch(UniqueFieldValueConstraintViolationException ex)
268     {
269         db.rollback();
270         throw new Exception("Usuario registrado anteriormente: "+ ex.getMessage());
271     }
272 }
```

Adicionando un objeto de tipo CUsuario

En este ejemplo se está pasando por parámetro un objeto de tipo CUsuario (una clase implementada en el sistema), para adicionarlo se llama a la función “set” del contenedor “db” (línea 264 de la imagen).

En este ejemplo también se usa transacciones las cuales son las que dicen cuando y si los cambios en la base de datos se hacen permanentes, hasta los cambios que hayan sido entregados pueden ser revertidos. Una transacción es una unidad de trabajo compuesta por uno o más inserciones, actualizaciones o borrados.⁴⁹

En el caso de ejemplo se usó la transacción “commit” para realizar el adición del objeto (línea 265 de la imagen), de no poderse llevar a cabo pues se regresa al estado anterior antes de adicionar el elemento (línea 269 de la imagen) y se lanza un error (línea 270 de la imagen).

⁴⁹ SALISBURY., S. *SQL transaction management with JDBC: roll back database transactions to undo errors.(Programming & Development: In-Depth ARTICLE)*. [En línea 2007]. [Consultada Marzo 6, 2009]. Disponible en: [http://find.galegroup.com/itx/retrieve.do?contentSet=IAC-Documents&resultListType=RESULT_LIST&qrySerId=Locale\(es%2C%2C\)%3AFQE%3D\(KE%2CNone%2C21\)transactions+database%24&sgHitCountType=None&inPS=true&sort=DateDescend&searchType=BasicSearchForm&tabID=T003&prodId=CDB&searchId=R1¤tPosition=2&userGroupName=ucinf&docId=A172525164&docType=IAC](http://find.galegroup.com/itx/retrieve.do?contentSet=IAC-Documents&resultListType=RESULT_LIST&qrySerId=Locale(es%2C%2C)%3AFQE%3D(KE%2CNone%2C21)transactions+database%24&sgHitCountType=None&inPS=true&sort=DateDescend&searchType=BasicSearchForm&tabID=T003&prodId=CDB&searchId=R1¤tPosition=2&userGroupName=ucinf&docId=A172525164&docType=IAC)

Borrar un objeto

Para borrar un objeto primeramente se debe recuperar el objeto deseado de la base de datos y después llamar a la función “delete” mediante el contenedor.

```
443 public void BorrarMedio(String psumario)
444 throws Exception{
445     CSumario auxiliar = new CSumario();
446     auxiliar = BuscarSumario(psumario);
447     CMedio medio = auxiliar.GetMedio();
448     db.delete(medio);
449 }
```

Borrado de un objeto

En este ejemplo se pasa por parámetro el identificador del objeto que se quiere eliminar (psumario) y se realiza la búsqueda del mismo en la BD (línea 446 de la imagen), se obtiene un objeto que es atributo del objeto pasado por parámetro (línea 447 de la imagen) y luego se le pasa a la función “delete” (en la línea 448 de la imagen).

Consultas en DB4O

Existen tres formas de realizar consultas las cuales son :

- Consulta por Ejemplo (en inglés QBE (Query By Example))

Donde se construye una plantilla del objeto que se quiera recuperar, actualizar o eliminar.

```
22 public void DevuelvePiloto(){
23     Piloto proto=new Piloto(null,0);
24     ObjectSet result=db.get(proto);
25     MostrarResultados(result);
```

Muestra los datos de un objeto piloto

En este caso se utiliza la función “get” (línea 24 de la imagen) para devolver todos los objetos pilotos que están en la base de datos, para devolver un piloto específico se construye la plantilla del objeto (línea 23 de la imagen) con el nombre del piloto o el campo que tenga el objeto como identificador. En la línea 25 de la imagen se llama al método “MostrarResultados” y se le pasa como parámetro el objeto “result” (el cual tiene el resultado de la búsqueda), y se muestra el resultado por pantalla.

➤ Consultas Nativas (en inglés Native Query (NQ)):

Las consultas nativas son la principal interfaz de consulta que tiene DB4O, son la forma recomendada para realizar consultas a la base de datos de las aplicaciones. Porque las NQ simplemente usan la semántica del lenguaje⁵⁰ de programación que se esté utilizando. Están perfectamente estandarizadas, y constituyen una opción segura para el futuro.⁵¹

```
29 public void DevuelveCarro(){
30     final String pilotName = "Rubens Barrichello";
31     ObjectSet results = db.query(new Predicate<Carro>() {
32         public boolean match(Carro carro){
33             return carro.GetPiloto().GetNombre().equals(pilotName);
34         }
35     });
36     MostrarResultados(results);
37 }
```

Devuelve el carro que tiene como piloto a Rubén Barriachelo

⁵⁰ La semántica en lingüística es el estudio del significado de una palabra tanto a nivel individual como en el contexto de una frase. En los lenguajes de programación es el conjunto de reglas que especifican el significado de cualquier sentencia, sintácticamente correcta y escrita en un determinado lenguaje.

LOVELLE, J. M. C. *Conceptos Básicos de Procesadores de Lenguaje* [En línea 1998]. [Consultada Marzo 6, 2009]. Disponible en: http://avellano.fis.usal.es/~lalonso/procesadores/10_conceptos_basicos_uniovi.pdf

⁵¹ DEVELOPER.DB4O.COM. *db4objects*, [En línea 2007]. [Consultada Marzo 6, 2009]. Disponible en: <http://www.db4o.com>

Como se puede observar las consultas nativas suelen ser bastante sencillas de implementar, además de poder optimizarse para un mejor rendimiento.

➤ API de Consultas (en inglés Soda Query API)

Este tipo de consulta es de muy bajo nivel, pero esto no significa que no tenga potencialidades, una de las desventajas que presenta el mismo es que no es de tipo seguro, ya que se accede directamente a los nodos del grafo de consultas, a los cuales se accede a través de los campos identificadores de las clases, pero este tipo de consultas es muy útil cuando hay que generar consultas dinámicas, o en casos donde la aplicación de las NQ no sea tan clara para el programador y se le haga más sencillo hacerlo con consultas de bajo nivel. A continuación un ejemplo de este tipo de consultas.

```
373     public CGrupoSumario BuscarGrupoSumario(String pgrupoSumario)
374     {
375         Query query = db.query();
376         query.constrain(CGrupoSumario.class);
377         query.descend("nombre").constrain(pgrupoSumario);
378         ObjectSet resultado = query.execute();
379         return (CGrupoSumario)resultado.next();
380     }
```

Buscar un objeto en la base de datos

En este ejemplo se busca un grupo de sumarios (sumario es un grupo de noticias), que tiene un determinado nombre (pasado por parámetro), lo primero que se hace es restringir la búsqueda por la clase que se quiera buscar y que contenga ese objeto (línea 376 de la imagen), se desciende por el campo nombre, el cual es el identificador de este tipo de objeto (línea 377 en la imagen) y se compara con el pasado por parámetro (en este caso pgrupoSumario) y por último se ejecuta la consulta y se retorna el objeto deseado (líneas de la 378 a la 379).

2.6. Vista de Despliegue

Para el desarrollo de una aplicación informática cualquiera, el arquitecto de software debe prestarle especial atención a la configuración de hardware en la cual se desplegará el sistema, identificando nodos procesadores (computadoras), dispositivos y protocolos; lo que en su conjunto conforman el modelo de despliegue, el cual define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos hardware sobre los cuales pueden ejecutarse los elementos software y para representar esta información se emplea el elemento de UML denominado Diagrama de Despliegue.

Un diagrama de despliegue muestra la configuración de los procesos en tiempo de corrida y los componentes que se encuentran en los nodos.⁵²

Los nodos son elementos físicos, que existen en tiempo de ejecución y representan un recurso computacional que generalmente tienen alguna memoria y, a menudo, capacidad de procesamiento. Los nodos sirven para modelar la topología del hardware sobre el que se ejecuta el sistema.

Un nodo representa normalmente un procesador o un dispositivo sobre el que se pueden desplegar los componentes. Un nodo debe tener un nombre asignado que lo distinga del resto de nodos.⁵³

Y se representan de la siguiente forma:

⁵² SANTANDER, U. F. D. P. En el Futuro de Todos, En línea 2002. [Consultada Marzo 7, 2009]. Disponible en: <http://www.ufps.edu.co/paginas/respuestas/pdf/revista7.pdf#page=25>

⁵³ XAVIER FERRÉ GRAU, M. I. S. S. Desarrollo Orientado a Objetos con UML, [En línea 2009]. [Consultada Marzo 7, 2009]. Disponible en: <http://www.willydev.net/descargas/Articulos/General/umlTotal.pdf>



También se le pueden aplicar estereotipos, los que no son más que formas que reflejan la funcionalidad del nodo o lo que el nodo representa, desde un punto de vista gráfico. Por ejemplo en el caso de que el nodo represente una PC (computadora personal) cliente se puede mostrar así:



Mientras que los componentes pertenecen al mundo físico, es decir, representan un bloque de construcción al modelar aspectos físicos de un sistema. Una característica básica de un componente es que:

Debe definir una abstracción precisa con una interfaz bien definida, y permitiendo reemplazar fácilmente los componentes más viejos con otros más nuevos y compatibles.

Existen básicamente tres tipos de componentes:

- **Componentes de despliegue:** componentes necesarios y suficientes para formar un sistema ejecutable, como pueden ser las bibliotecas dinámicas (DLLs) y ejecutables (EXEs).
- **Componentes producto del trabajo:** estos componentes son básicamente productos que quedan al final del proceso de desarrollo. Consisten en archivos de código fuente y de datos a partir de los cuales se crean los componentes de despliegue.
- **Componentes de ejecución:** son componentes que se crean como consecuencia de un sistema en ejecución. Es el caso de un objeto que se instancia (o se crea) a partir de una DLL.⁵⁴

En el caso del módulo de búsqueda del SIMAN, se cuenta con un servidor de aplicaciones web que es donde el sistema se conectará mediante el protocolo HTTP, y descargará los elementos que interesan del RSS, entiéndase por esto, el título de la noticia, la descripción, la URL, el autor o creador y la fecha de publicación.

Además contará con una PC cliente, en la cual estará corriendo el sistema, en este nodo es donde se encontrará la base de datos de la aplicación, respondiendo de forma individual por cada cliente.

A continuación se muestra el Diagrama de Despliegue correspondiente al módulo de búsqueda del SIMAN:

⁵⁴ XAVIER FERRÉ GRAU, M. I. S. S. *Desarrollo Orientado a Objetos con UML*, [En línea 2009]. [Consultada Marzo 7, 2009]. Disponible en: <http://www.willydev.net/descargas/Articulos/General/umITotal.pdf>

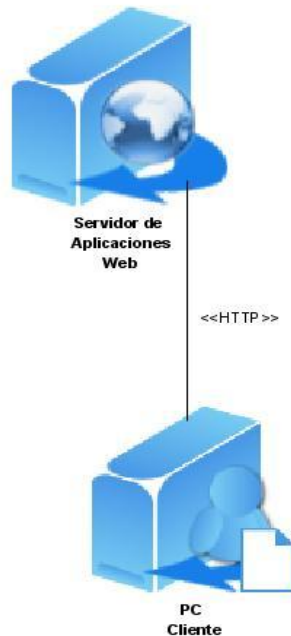


Diagrama de Despliegue

Servidor de Aplicaciones Web:

En este servidor es donde radican los RSS a los que se conectará el sistema.

PC Cliente:

Desde donde se accede al sistema a través del programa cliente del sistema.

2.7. Estilos de Programación

Cada lenguaje de programación tiene un estilo de codificación característico, y en ocasiones, más de uno. Por estilo de codificación, convenciones de codificación, o directamente, estándares de codificación, se hace referencia a la forma de elegir y disponer físicamente los caracteres que componen la sintaxis del lenguaje; por ejemplo, la profundidad del sangrado, la posición de las llaves de apertura y cierre de bloque, los saltos de línea o el convenio de elección de nombres para los identificadores (variables, clases, funciones, paquetes, etc.).

Algunos estilos tienen nombre propio, como las Java Coding Conventions de Sun, el estilo K&R para C (inspirado en los ejemplos del libro correspondiente), el estilo ANSI

C, el estilo Win32 definido por Microsoft (notación húngara para los identificadores), etc. Los proyectos deben elegir un estilo y seguirlo. No se trata de un capricho: utilizar un estilo uniforme facilita la propiedad colectiva del código, simplifica el control de versiones, y en general, aumenta la productividad.⁵⁵

Las convenciones de código son importantes para los programadores por numerosas razones:

- El 80% del tiempo de vida de un programa se dedica al mantenimiento del mismo.
- En pocas ocasiones, el programa es mantenido durante toda su vida útil por su autor original.
- Las convenciones de código aumentan la legibilidad de los programas, permitiendo a los desarrolladores comprender nuevo código rápida y perfectamente.
- Si se distribuye el código fuente como un producto, se necesita asegurar que está bien empaquetado y limpio.

A continuación se muestran algunos estándares de programación del lenguaje Java presentados en la Especificación del Lenguaje Java de Sun Microsystems, inc.⁵⁶

Nombres de Ficheros:

Java utiliza los siguientes sufijos de fichero:

⁵⁵ BERRUETA, D. *Programación extrema y software libre*, [En línea 2006]. [Consultada Marzo 7, 2009]. Disponible en: http://www.asturlinux.org/archivos/jornadas2006/ponencias/ProgExtrema_Berrueta/ponencia-sl-y-xp.pdf

⁵⁶ ALCAIDE, J. P. *Convenciones de código Java*, [En línea 2008]. [Consultada Marzo 7, 2009]. Disponible en: <http://jpereza.nom.es/convenciones-de-codigo-java>

Tipo de fichero	Sufijo
Código fuente Java	.java
Código compilado Java	.class

Nombres de ficheros comunes

Nombre de fichero	Uso
LEEME	Nombre preferido para el fichero que resume el contenido de un directorio en particular.

Organización de ficheros

Un fichero consiste en secciones que deberían estar separadas por líneas en blanco y un comentario opcional identificando cada sección.

Los ficheros de más de 2000 líneas son demasiado largos y deberían evitarse.

Ficheros de código fuente Java

Cada fichero de código fuente Java contiene una única clase o interfaz pública. Cuando una clase pública tiene clases privadas e interfaces asociados, se pueden poner en el mismo fichero de código fuente que la clase pública. La clase pública debería ser la primera clase o interfaz en el fichero.

Los ficheros de código fuente Java tienen la siguiente ordenación:

- Comentarios iniciales
- Sentencias package e import
- Declaraciones de clase e interfaz

Comentarios iniciales

Todos los ficheros de código fuente deberían comenzar con un comentario que muestre el nombre de la clase, información sobre la versión, la fecha y el copyright ó derecho de autor.

```
/*  
 * Nombre de la clase  
 *  
 * Información sobre la versión  
 *  
 * Fecha  
 *  
 * Copyright  
  
*/
```

Esto lo genera automáticamente el IDE de desarrollo Netbeans 6.0.

Sentencias package e import

La primera línea que no sea un comentario de todos los ficheros de código fuente Java es una sentencia package. Después, puede haber sentencias import. Por ejemplo:

```
package es.nom.jpereza;  
  
import java.util.List;
```

Declaraciones de clase e interfaz

La siguiente tabla describe las partes de una declaración de clase o interfaz, en el orden que deben aparecer.

	Parte de la declaración de Clase/Interfaz	Notas
1	Comentario de documentación de la clase/interfaz (/** ... */)	
2	Sentencia class ó interface	
3	Comentario de la implementación de la clase/interfaz, si fuera necesario (* ... *)	Este comentario debería contener cualquier información relativa a toda la clase o interfaz, que no sea apropiada para el comentario de documentación.
4	Variables de clase (estáticas) (static)	Primero las variables públicas (public), luego las protegidas (protected), después las de paquete (sin modificador de acceso) y por último las privadas (private).
5	Variables de instancia	Primero las variables públicas (public), luego las protegidas (protected), después las de paquete (sin modificador de acceso) y por último las privadas (private).
6	Constructores	
7	Métodos	Estos métodos deberían estar agrupados por funcionalidad en lugar de por ámbito o accesibilidad. Por ejemplo, un método estático privado puede estar entre dos métodos de instancia públicos. El objetivo es hacer la lectura y comprensión del código más fácil.

Tabulación

Longitud de línea

Evitar las líneas de más de 80 caracteres, ya que algunas herramientas no las manejan bien.

Ruptura de líneas (Wrapping lines)

Cuando una expresión no cabe en una única línea, se debe romper de acuerdo a estos principios generales:

- Romper después de una coma.
- Romper antes de un operador.
- Preferir las rupturas de alto nivel a las de bajo nivel.
- Alinear la nueva línea con el principio de la expresión al mismo nivel que la línea anterior.
- Si las reglas anteriores llevan a un código confuso o demasiado pegado al margen derecho, entonces tabular sólo con 8 espacios.

Ejemplos de llamadas a métodos en varias líneas:

```
someMethod(longExpression1, longExpression2, longExpression3,  
           longExpression4, longExpression5);  
  
var = someMethod1(longExpression1,  
                 someMethod2(longExpression2,  
                             longExpression3));
```

A continuación, dos ejemplos de cómo romper una expresión aritmética. El primero es más recomendable, puesto que la ruptura ocurre fuera de la expresión entre paréntesis, la cual es de mayor nivel.

```
longName1 = longName2 * (longName3 + longName4 - longName5)
    + 4 * longname6; // RECOMENDADA

longName1 = longName2 * (longName3 + longName4
    - longName5) + 4 * longname6; // EVITAR
```

A continuación hay dos ejemplos de cómo tabular declaraciones de métodos. El primero es el caso convencional. El segundo dejaría la segunda y tercera líneas demasiado pegadas al margen derecho si se usara la tabulación convencional, por eso en cambio se tabula sólo con 8 espacios.

```
// TABULACION CONVENCIONAL
someMethod(int anArg, Object anotherArg, String yetAnotherArg,
    Object andStillAnother) {
    ...
}

// USAR 8 ESPACIOS PARA EVITAR PEGARSE AL MARGEN DERECHO
private static synchronized horkingLongMethodName(int anArg,
    Object anotherArg, String yetAnotherArg,
    Object andStillAnother) {
    ...
}
```

La ruptura de líneas para las sentencias if debería usar generalmente la regla de los 8 espacios, ya que la tabulación convencional (4 espacios) dificulta la lectura del cuerpo de la sentencia if. Por ejemplo:

```
// NO UTILIZAR ESTA TABULACION
if ((condition1 && condition2)
    || (condition3 && condition4))
```



```
    ||!(condition5 && condition6)) { // MALAS RUPTURAS
    doSomethingAboutIt();           // HACEN QUE ESTA LINEA SE PIERDA
FACILMENTE
}

// USAR ESTA TABULACION EN SU LUGAR
if ((condition1 && condition2)
    || (condition3 && condition4)
    ||!(condition5 && condition6)) {
    doSomethingAboutIt();
}

// O USAR ESTA OTRA
if ((condition1 && condition2) || (condition3 && condition4)
    ||!(condition5 && condition6)) {
    doSomethingAboutIt();
}
```

Aquí se muestran tres maneras aceptables de escribir expresiones ternarias:

```
alpha = (aLongBooleanExpression) ? beta : gamma;

alpha = (aLongBooleanExpression) ? beta
                                     : gamma;

alpha = (aLongBooleanExpression)
        ? beta
        : gamma;
```

Ejemplo de código fuente Java

El siguiente ejemplo muestra el formato de un fichero de código fuente Java que contiene una clase pública.

```
/*
 * @(#)Blah.java          1.82 99/03/18
 *
 * Copyright (c) 1994-1999 Sun Microsystems, Inc.
 * 901 San Antonio Road, Palo Alto, California, 94303, U.S.A.
 * All rights reserved.
 *
 * This software is the confidential and proprietary information of Sun
 * Microsystems, Inc. ("Confidential Information"). You shall not
 * disclose such Confidential Information and shall use it only in
 * accordance with the terms of the license agreement you entered into
 * with Sun.
 */

package java.blah;

import java.blah.blahdy.BlahBlah;

/**
 * La descripción de la clase va aquí.
 *
 * @version      1.82 18 Mar 1999
 * @author      Nombre Apellido
 */
public class Blah extends SomeClass {
    /* Un comentario de implementación de clase puede ir aquí. */

    /** Comentario de documentación de classVar1 */
    public static int classVar1;

    /**
     * Comentario de documentación de classVar2
     * que ocupa más de una línea
     */
    private static Object classVar2;
```

```
/** Comentario de documentación de instanceVar1 */
public Object instanceVar1;

/** Comentario de documentacion de instanceVar2 */
protected int instanceVar2;

/** Comentario de documentación de instanceVar3 */
private Object[] instanceVar3;

/**
 * ...comentario de documentación del constructor de Blah...
 */
public Blah() {
    // ...la implementación va aquí...
}

/**
 * ...comentario de documentación del método doSomething...
 */
public void doSomething() {
    // ...la implementación va aquí...
}

/**
 * ...comentario de documentación del método doSomethingElse...
 * @param someParam descripción del parámetro
 */
public void doSomethingElse(Object someParam) {
    // ...la implementación va aquí...
}
}
```

2.8. Estándar de Codificación

En todo proyecto que se esté desarrollando debe existir un estándar de codificación, que guíe todo el trabajo de implementación. El principal objetivo de esta tarea es la de obtener un código claro y entendible por cualquier programador, por supuesto que esté familiarizado con el estándar que se esté utilizando. Debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código del sistema. Además con esto se pretende lograr que todos los programadores del proyecto trabajen de forma coordinada.

¿Para qué se necesita un estándar?

La respuesta se resume en:

- Eleva la mantenibilidad del código.
- Sirve como punto de referencia para los programadores.
- Mantiene un estilo de programación.
- Ayuda a mejorar el proceso de codificación, haciéndolo entre otras cosas más eficientes.

En el caso del módulo de búsqueda del SIMAN se cumplirá con el siguiente estándar de codificación:

- Idioma: Se debe emplear un solo idioma y se utilizará el español, las palabras no se acentuarán. Sólo se podrá utilizar el idioma inglés en el caso de sobrescribir métodos de clases propias del entorno de desarrollo y en las propiedades de las clases.
- Palabras Reservadas: Las palabras reservadas van en minúsculas sin excepción alguna.
- Indentación: En el caso del Netbeans (IDE utilizado en el desarrollo de éste sistema) se encarga de ajustar los espacios de forma automática. Los inicios “(“,”

- {“, y cierre “}”, “)” de bloque deben estar alineados debajo de la declaración a la que pertenecen y es a gusto utilizarse o no en el caso de que exista una sola instrucción. Nunca colocar {en la línea de un código cualquiera, esto requiere una línea propia. El cuerpo de declaraciones compuestas (declaraciones como “if”, “while”, “for”, etc.) debe anidarse a una profundidad de 4 espacios.
- Líneas en blanco: Se colocará una línea en blanco antes y después de la declaración de una estructura o de una clase.
 - Uso de espacios en blanco: Los signos lógicos y de operación deberán estar separados por un espacio antes y después del mismo, ejemplo: `int Total = 15 - 8 + 9`; no se debe utilizar entre el nombre de un método y el paréntesis abierto, `void HacerAlgo(int Posicion)`; entre el cast y la expresión, ejemplo: `float Promedio = (float)(5 + 4) / 2`; después del corchete abierto y antes del cerrado de un arreglo, ejemplo `int Cantidad = arrElementos[i]`.
 - Comentarios de línea: Consiste en un “//” seguido de un texto. Debe incluirse un espacio simple entre el “//” y el texto. Se debe evitar comentar cada línea de código. Es mejor colocar el comentario precediendo al bloque de líneas de código. Si el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco, si se refiere sólo a la siguiente instrucción, se suprime la línea. Este tipo de comentario también puede seguir al código que referencia, debiendo aparecer en la misma línea.
 - Declaraciones de clases: Las directivas de visibilidad deben estar indentadas con la definición de la clase. Estas directivas deben declararse en el orden siguiente: `private`, `protected`, `public`.
 - Variables locales: Las variables deben ser declaradas justo antes de ser utilizadas. Cuando sea posible, la inicialización de las variables locales ocurrirá

en el momento de su declaración. Se debe evitar la declaración dentro del cuerpo de un ciclo.

- Declaraciones condicionales: Las declaraciones “if” – “else” no deben poseer más de 4 niveles de profundidad, de tenerlos no deben estar en una línea sino repartidos en igual número de líneas como niveles tenga, esto se explica porque es más fácil leer una declaración “if” – “else”de arriba para abajo que de izquierda a derecha. La cláusula else, siempre debe estar alineada con la cláusula “if”. ” //” Se sugiere utilizar cuando las condiciones sobrepasan los márgenes.

```
if (condition1 && condition2 && condition3)
```

```
{.....en caso de tener más de una línea de código
```

```
.....código aquí
```

```
}
```

```
else
```

```
{.....en caso de tener más de una línea de código
```

```
.....código aquí
```

```
}
```

- Declaraciones switch: Las expresiones constantes dentro de las declaraciones “switch” deben estar ordenadas numérica o alfabéticamente dependiendo del tipo al que pertenecen. La cláusula default deberá alinearse con la cláusula “case “.
- Declaraciones “for”, “foreach”: El código de inicialización debe realizarse en la declaración del mismo a menos que sean necesario declararlo fuera para darle más tiempo de vida a la variable de control.
- Declaraciones “while”: Las declaraciones “while” deben usarse en lugar de “for” cuando el lenguaje natural a la hora de pensar en algo que se repita lo indique.

- Declaraciones “do”...”while”: El código de la declaración está acotado por las palabras reservadas: “do”... “while”. Se utilizará cuando se necesite explícitamente que se cumpla al menos una vez ciclo post condicional.
- Regla para identificadores: Los nombres deben formarse a partir de palabras en español únicamente. En el caso de variables y atributos en minúscula, en caso de ser atributos compuestos la segunda palabra es en mayúscula ejemplo: estaActiva, y en los nombres de métodos o propiedades la primera letra en Mayúscula en caso de ser una palabra compuesta la primera letra de la que empieza a continuación de la anterior, ejemplo: CambiarEstado (...)

En este capítulo se fundamentó la arquitectura empleada: el uso de los modelos y patrones utilizados. Se analizaron aspectos técnicos de la base de datos utilizada, se explicaron los estilos y estándar de codificación a utilizar. También se mostró la vista de despliegue del sistema. Así como conceptos que son de vital importancia.

Capítulo 3. Descripción y Análisis de la solución propuesta.

En el presente capítulo se expone la descripción y se analiza la solución propuesta para desarrollar la aplicación modelándose los artefactos necesarios que contribuyen a la implementación del sistema. Se refinan los principales diagramas de clases y secuencia del modelo de análisis y diseño.

3.1. Modelo de Diseño

La fase de diseño (y los modelos UML resultantes) expande y detalla los modelos de análisis tomando en cuenta todas las implicaciones y restricciones técnicas. El propósito del diseño es especificar una solución que trabaje y pueda ser fácilmente convertida en código fuente y construir una arquitectura simple y fácilmente extensible.

57

En el diseño se obtiene un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva se le da respuesta a la pregunta de: ¿Cómo cumple el sistema sus objetivos para preparar la implementación y prueba del sistema?.

En el diseño se modela el sistema y se le da forma (incluida la arquitectura) para que soporte todos los requisitos y las restricciones que se le suponen. La entrada principal para el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos. Además de plantear una estructura del sistema la cual debe mantenerse lo más fielmente posible cuando se le dé forma al sistema.

3.2. Diagrama de clases del Diseño.

Clases del Diseño: Una clase de diseño es una abstracción de una clase o construcción en la implementación del sistema.

⁵⁷ NAVARRO, J. M. *Diseño del Sistema de Tarjeta de Crédito con UML*, Consultada Abril 10, 2009]. Disponible en:
http://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Basic/mendoza_nj/Cap5.pdf

Diagrama de Clases del Diseño: Una clase de diseño es una construcción similar en la implementación del sistema. Los diagramas de clases de diseño exponen un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema.

Al igual que los demás diagramas, los diagramas de clases pueden contener notas y restricciones. Los diagramas de clases también pueden contener paquetes o subsistemas, los cuales se usan para agrupar los elementos de un modelo en partes más grandes.

Dadas las características particulares del sistema por ser una aplicación de escritorio se mostrará el diagrama de clases de una manera general agrupando la interrelación de todas clases. Además de representar todos los cambios llevados a cabo en el diagrama de clases debido al refinamiento que se le realizó al diseño del sistema.

3.3. Descripción de las Clases.

La descripción de las clases da una idea más específica de cómo estarán constituidas las clases. A continuación se muestra la descripción de la clase controladora. (Ver anexos)

Nombre: CC_Sistema	
Tipo de clase: Clase Controladora	
Atributo	Tipo
consultasBD	CC_ConsultaBD
temporizarActualizacion	CC_TemporizadorActualizacion
temporizarLimpieza	CC_TemporizadorLimpieza
usuario	CE_Usuario
buscador	CC_Buscador
Para cada responsabilidad:	
Nombre:	ArrancarTemporizadores()
Descripción:	Hecha a andar los temporizadores de tiempo de la actualización y la limpieza de la base de datos. Para que realice estas funciones de forma automática cada cierto tiempo definido por el usuario.
Nombre:	EjecutarDescargaCompleta(CSumario psumario)
Descripción:	Método que se encarga de realizar la descarga de las noticias del sumario pasado por parámetro.
Nombre:	DameNoticiasQueCoincidan(String pvalorBuscar)
Descripción:	Se encarga de buscar una palabra determinada en el índice de noticias

Capítulo3. Descripción y Análisis de la Solución Propuesta.

	para devolver la noticia que contenga esa palabra ya sea en el título o la descripción.
Nombre:	Logueo(String pusuario, String pcontrasenna)
Descripción:	Verifica que el usuario y contraseña suministrada exista en la base de datos.
Nombre:	Registrarse(String pnombre, String papellidos, String pcorreo, String pusuario, String pcontrasenna)
Descripción:	Crea una cuenta de usuario en la base de datos.
Nombre:	EntrarAlSistema(String pusuario, String pcontrasenna)
Descripción:	Si son válidos el usuario y la contraseña permite que el usuario entre al sistema.
Nombre:	RealizarSalvaRespaldo(String pcaminoSalva)
Descripción:	Realiza una salva de respaldo de la base de datos en el camino especificado por el usuario.
Nombre:	CambiarBaseDatos(String pnuevoCaminoBaseDatos)
Descripción:	Cambia la base de datos del sistema por la especificada por el usuario, una vez cerrado el sistema, volverá a retomar la base de datos anterior.
Nombre:	ActualizarSumarioSeleccionado(String pTituloSumario)
Descripción:	Permite realizar la actualización de un sumario seleccionado por el usuario.
Nombre:	ActualizarGrupoSumario(String pgrupoSumario)
Descripción:	Actualiza un grupo de sumarios seleccionado por el usuario.

Capítulo3. Descripción y Análisis de la Solución Propuesta.

Nombre:	RealizarActualizacionAutomatica()
Descripción:	Realiza la actualización automática de todos los sumarios del usuario conectado en el sistema.
Nombre:	EjecutarLimpieza()
Descripción:	Realiza la limpieza automáticamente de la base de datos según los criterios definidos por el usuario.
Nombre:	ActualizarTodosSumariosDelUsuario()
Descripción:	Actualiza todos los sumarios del usuario logueado.
Nombre:	AdicionarSumario(String pgrupo ,CSumario psumario)
Descripción:	Adiciona un sumario en un grupo definido por el usuario.
Nombre:	BuscarSumario(String psumario)
Descripción:	Busca un sumario determinado por el usuario y lo retorna
Nombre:	AdicionarUsuario(CUsuario pusuario)
Descripción:	Adiciona un usuario a la base de datos
Nombre:	CambiarContrasenna(String pcontrasennaV, String pcontrasennaN)
Descripción:	Cambia la contraseña del usuario conectado.
Nombre:	DesconectarUsuario()
Descripción:	Desconecta del sistema al usuario conectado pasando a ser usuario no conectado.
Nombre:	DevuelveUsuarioConectado()
Descripción:	Devuelve el usuario que está conectado en el sistema.

Capítulo3. Descripción y Análisis de la Solución Propuesta.

Nombre:	CambiarPropiedadesUsuario(String pnombre, String papellidos, String pcorreo)
Descripción:	Cambia las propiedades del usuario es decir permite que el usuario conectado pueda cambiar el nombre, apellidos o correo.
Nombre:	AdicionarGrupoSumario (CGrupoSumario pgrupoSumario)
Descripción:	Adiciona un grupo de sumarios a la base de datos
Nombre:	BuscarGrupoSumario(String pgrupoSumario)
Descripción:	Busca y devuelve un grupo de sumarios definido por el usuario.
Nombre:	BorrarGrupoSumario(String pnombreGrupoSumario)
Descripción:	Borra un grupo de sumarios de la base de datos.
Nombre:	AdicionarMedio(String psumario, CMedio pmedio)
Descripción:	Adiciona un medio al sistema.
Nombre:	BorrarMedio(String psumario)
Descripción:	Borra un medio definido por el usuario del sistema.
Nombre:	ConfigurarSumario(String psumario, CConfiguracionSumario pconfiguracion)
Descripción:	Permite configurar un sumario definido por el usuario.
Nombre:	CambiarNombreSumario(String ptituloViejo, String ptitulo)
Descripción:	Permite cambiarle el nombre a un sumario determinado por el usuario.
Nombre:	ActualizarPropiedadesSumario(String psumario, String ptitulo,String pdescripcion, Calendar pfechaadicion, String purl, int pprioridad, CCategoriaSumario pcategoria)

Capítulo3. Descripción y Análisis de la Solución Propuesta.

Descripción:	Actualiza las propiedades del sumario determinado por el usuario.
Nombre:	BorrarSumario(String pnombreGrupo, String psumario)
Descripción:	Borra del sistema un sumario determinado por el usuario.
Nombre:	DevuelveSumario(String ptituloSumario)
Descripción:	Busca y retorna un sumario determinado por el usuario.
Nombre:	ConfigurarSistemaPorDefecto()
Descripción:	Configura por defecto el sistema
Nombre:	CambiarConfiguracionSistema(CConfiguracionSistema pconfigSistema)
Descripción:	Cambia la configuración del sistema.
Nombre:	DevuelveConfiguracionSistema()
Descripción:	Devuelve la configuración del sistema.
Nombre:	EditarElementosNoticia(String ptitulo,List<String> ppalabrasClaves)
Descripción:	Edita las palabras claves de una noticia determinada definida por el usuario
Nombre:	DevuelveNoticiasSumario(String ptituloSumario)
Descripción:	Devuelve las noticias de un sumario determinado por el usuario.
Nombre:	CambiarEstadoNoticia(String ptituloNoticia, boolean pestado)
Descripción:	Cambia el estado de leída o no leída de una noticia determinada por el usuario.
Nombre:	DevuelveNoticia(String ptitulo)
Descripción:	Busca y devuelve una noticia determinada por el usuario.

Nombre:	Categoria(String psumario)
Descripción:	Devuelve la categoría que tiene el sumario pasado por parámetro.
Nombre:	CambiarRelevancia(String ptituloNoticia)
Descripción:	Cambia la relevancia de la noticia pasada por parámetro. Ya sea a tiene relevancia o no tiene relevancia.
Nombre:	IndexarNoticiasBaseDato()
Descripción:	Crea un índice de las noticias de la base de datos para posteriormente poder recuperarlas más rápidamente.

3.4. Diagrama de Interacción

Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos, o se pueden utilizar para modelar un flujo de control particular de un caso de uso. Los diagramas de interacción no son sólo importantes para modelar los aspectos dinámicos de un sistema, sino también para construir sistemas ejecutables por medio de ingeniería directa e inversa.

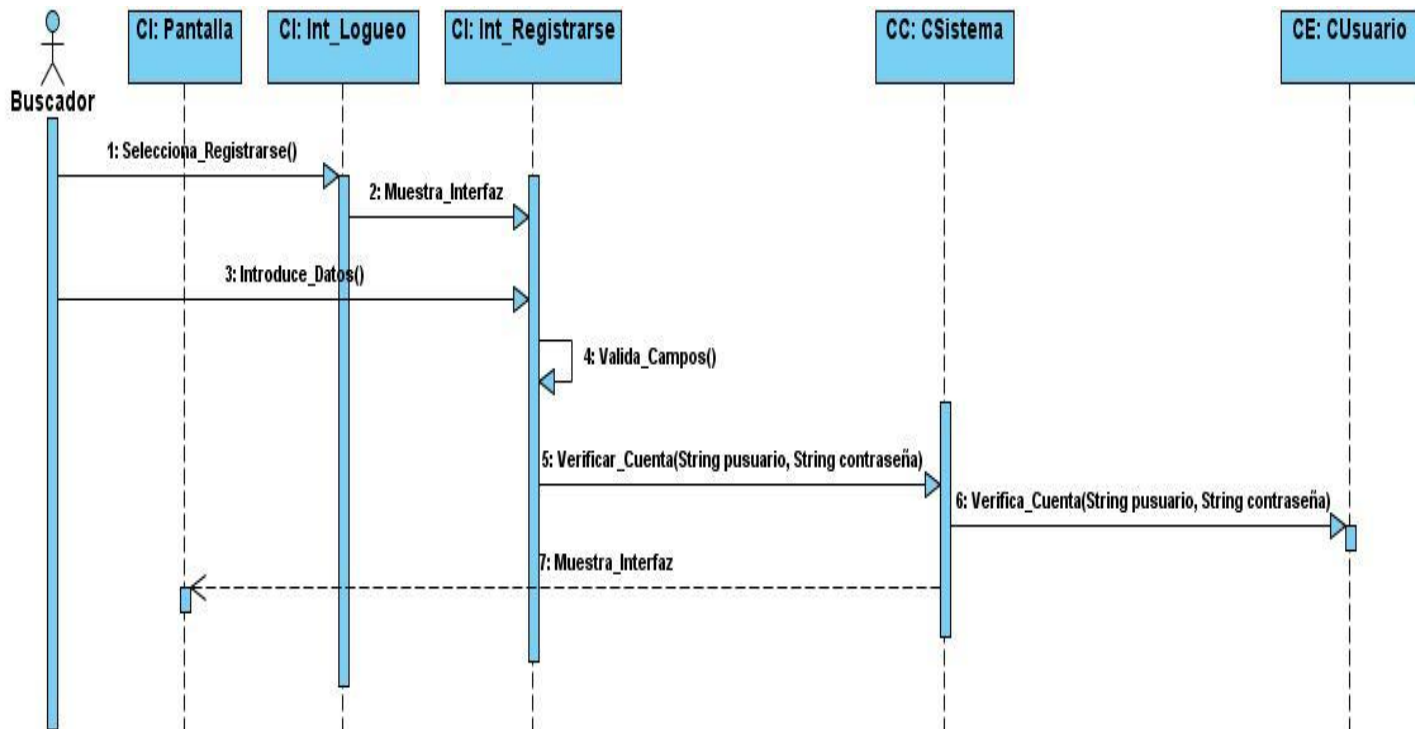
Un **diagrama de secuencia**: es un diagrama de interacción que destaca la ordenación temporal de los mensajes.

Los diagramas de secuencia se pueden utilizar a distintos niveles de abstracción, agrupando o separando componentes en función de las necesidades, obteniendo así protocolos para componentes individuales o para agrupaciones de componentes. Estos protocolos, especificados mediante grafos y álgebras de procesos, se pueden guardar

en repositorios junto a la restante información sobre el componente, y permiten estudiar su compatibilidad con el entorno donde va a ser integrado.⁵⁸

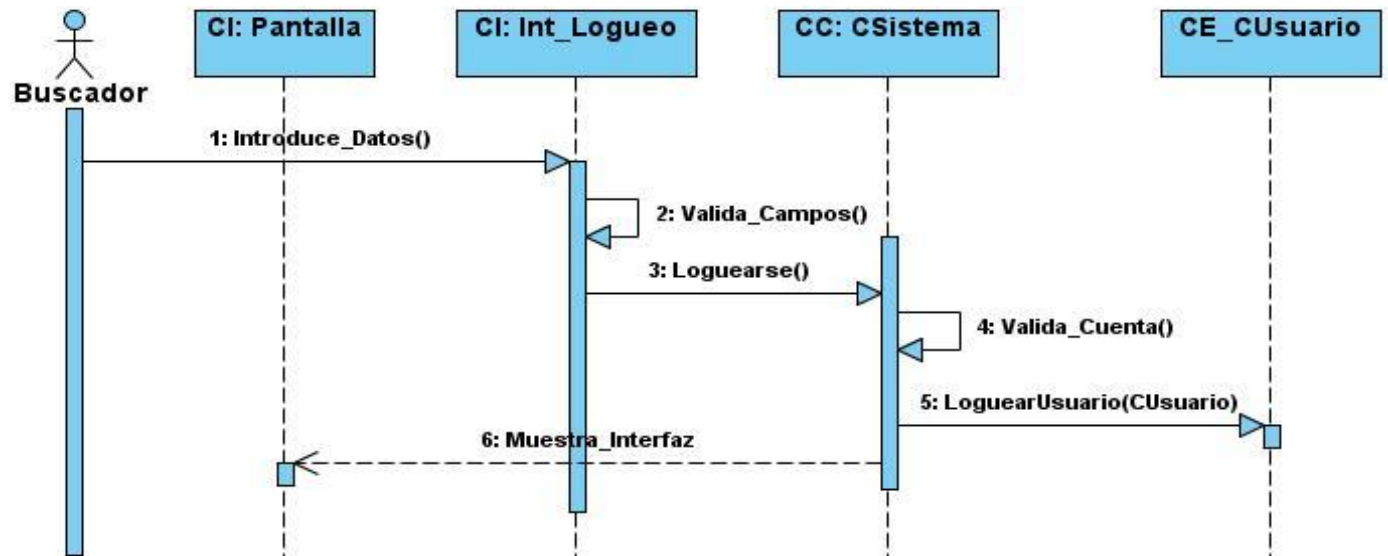
A continuación se muestran los diagramas de interacción del sistema.

DS: Registrar Usuario

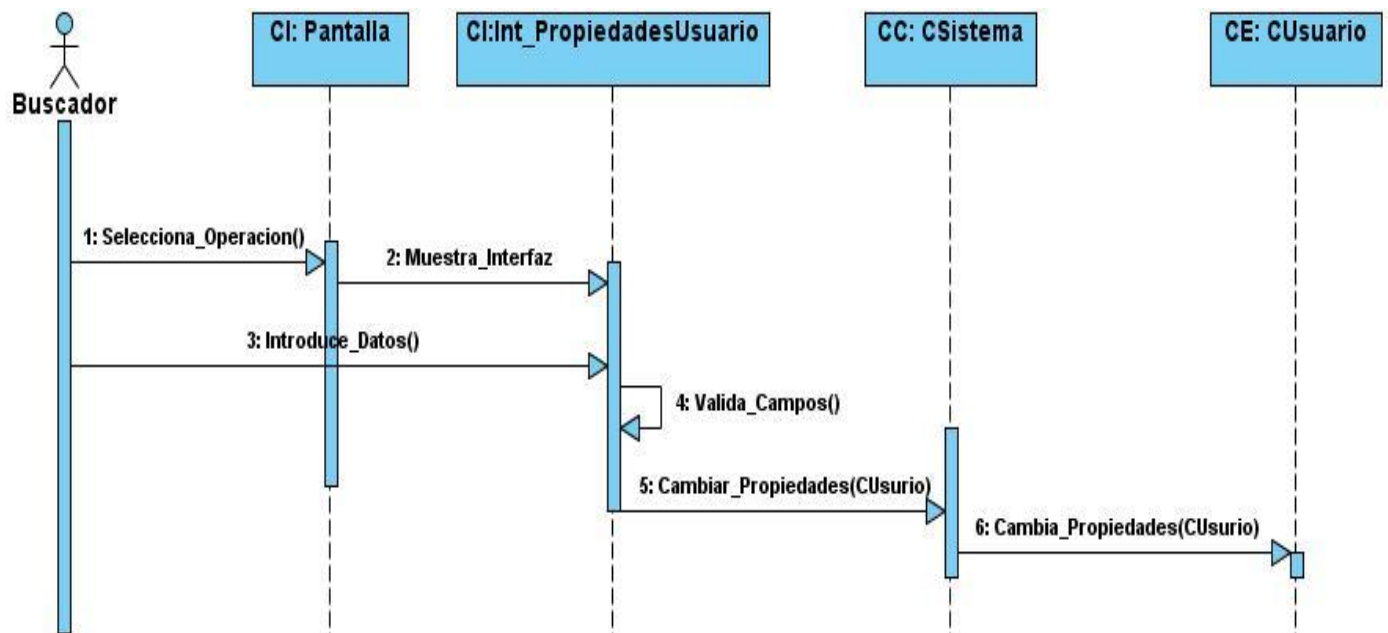


⁵⁸ MIGUEL A. PÉREZ TOLEDANO, A. N. M., CARLOS CANAL, JUAN M. MURILLO RODRÍGUEZ and *Desarrollo de Sistemas Basados en Componentes Utilizando Diagramas de Secuencia.*, Consultada Abril 11, 2009]. Disponible en: http://www.lcc.uma.es/~canal/papers/ideas05/tncm_ideas05.pdf

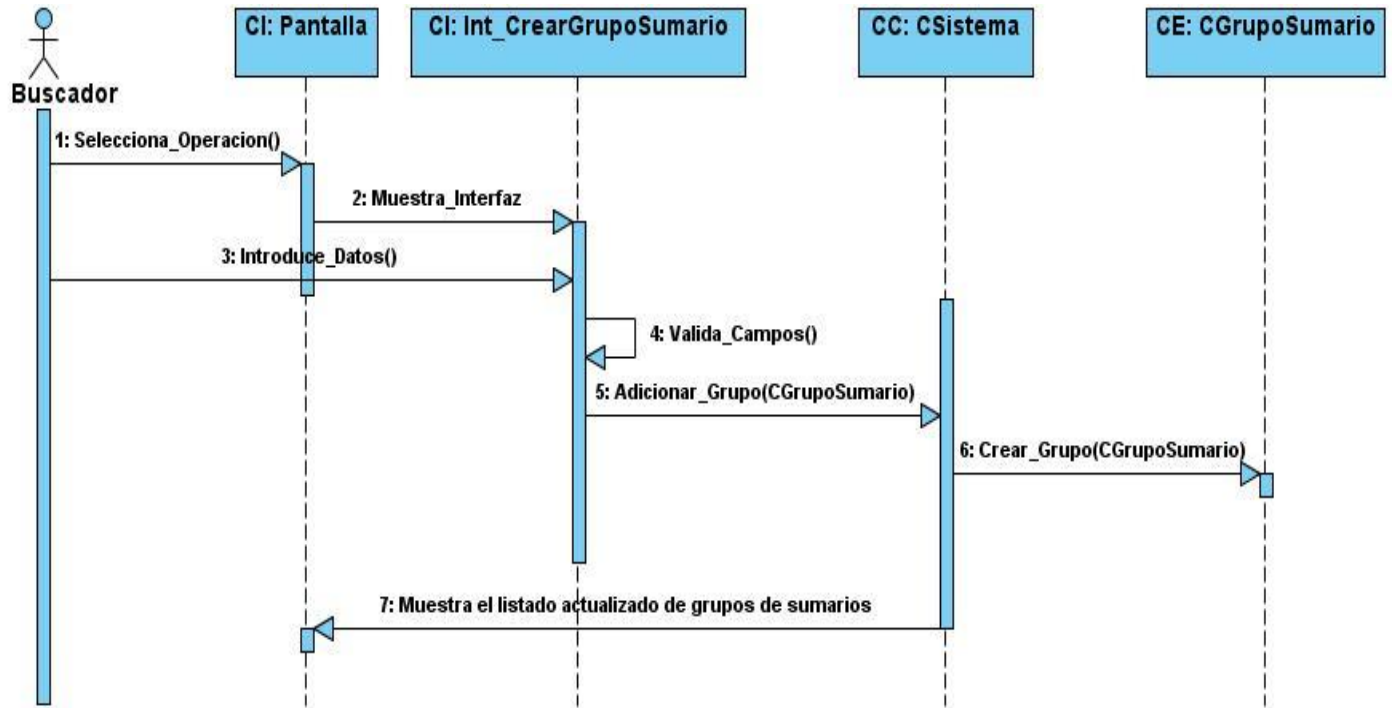
DS: Entrar Usuario al Sistema



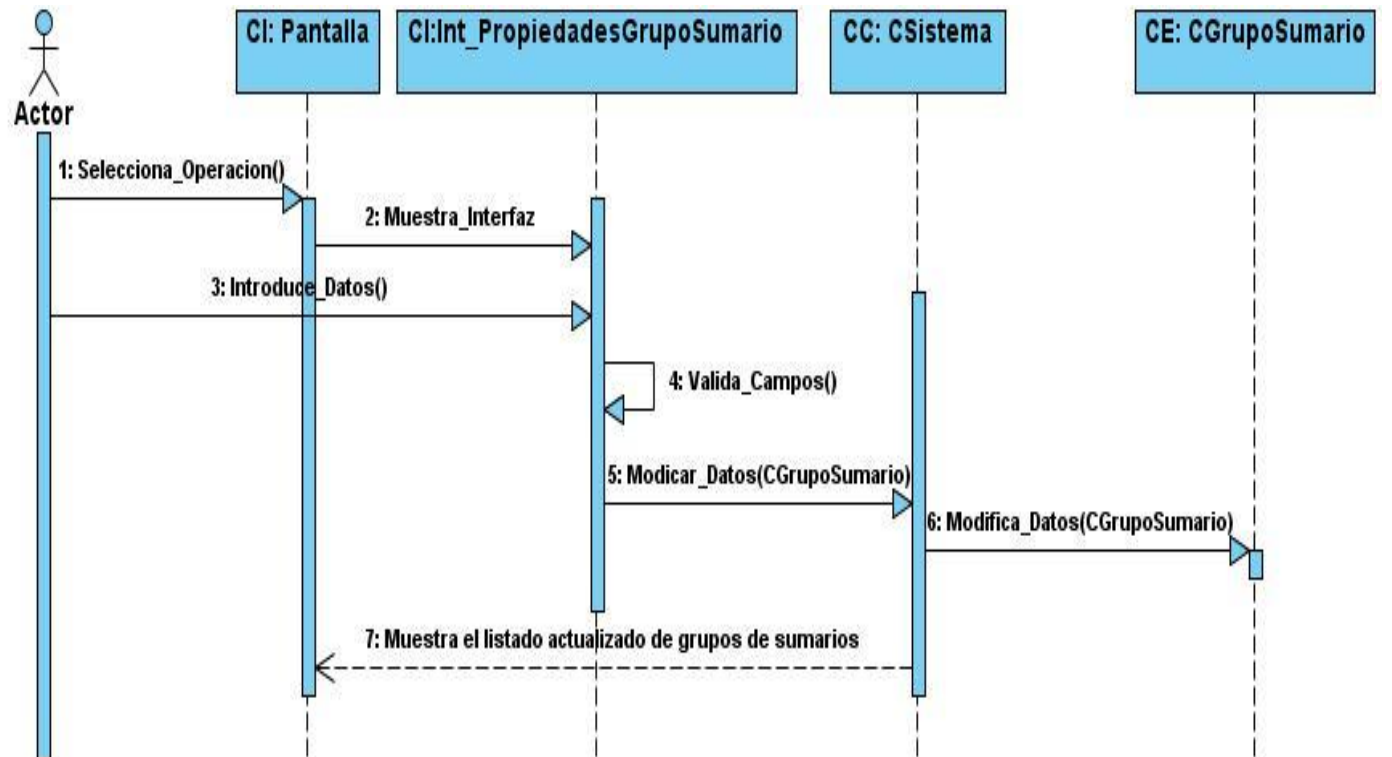
DS: Cambiar Propiedades al Usuario



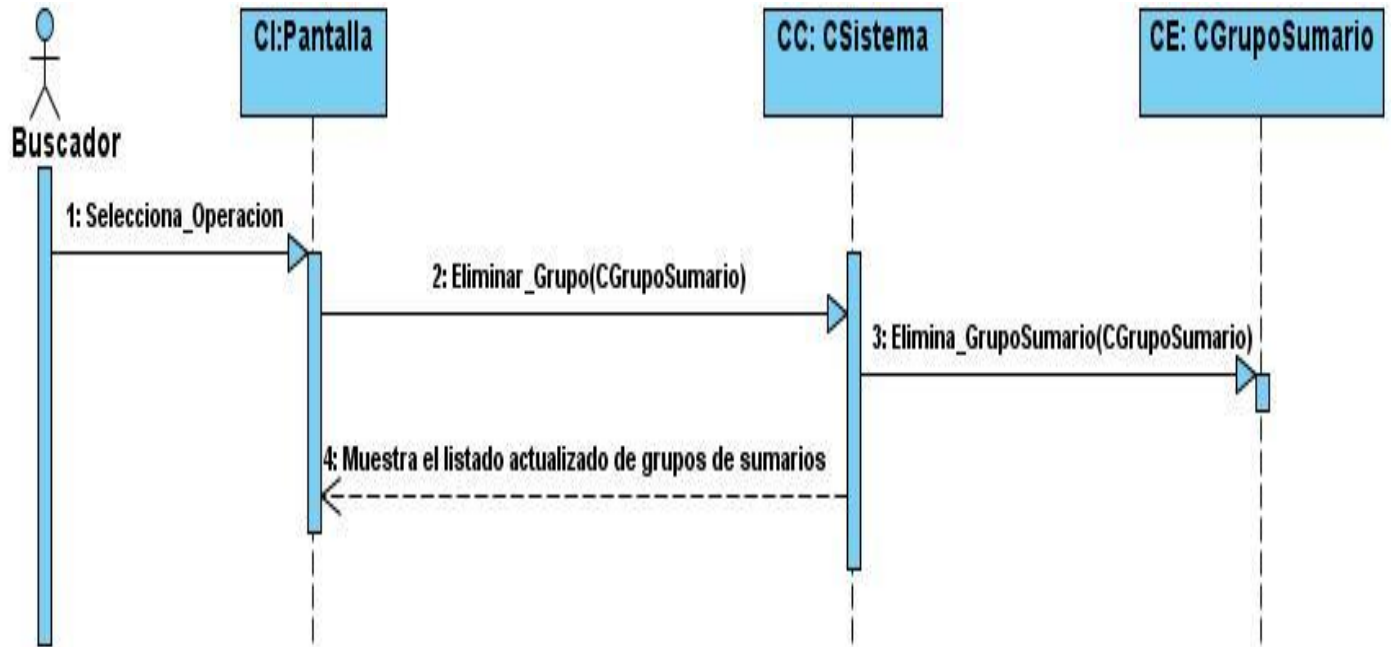
DS: Adicionar Grupo de Sumarios



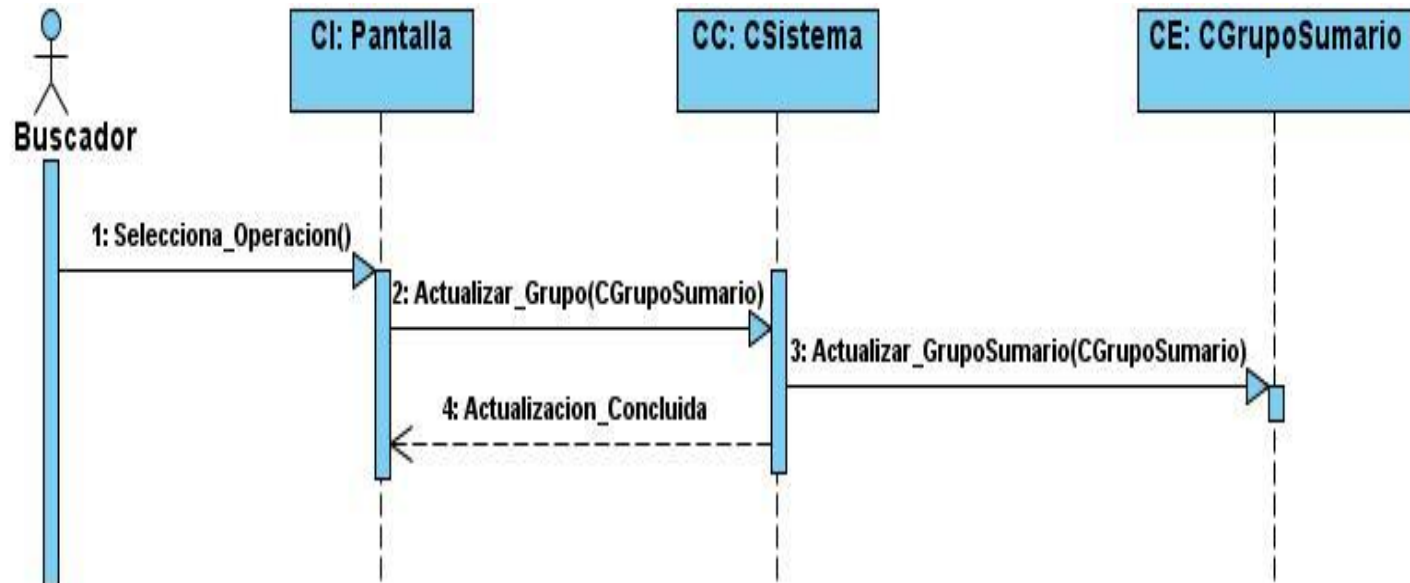
DS: Cambiar Propiedades de un Grupo Sumario



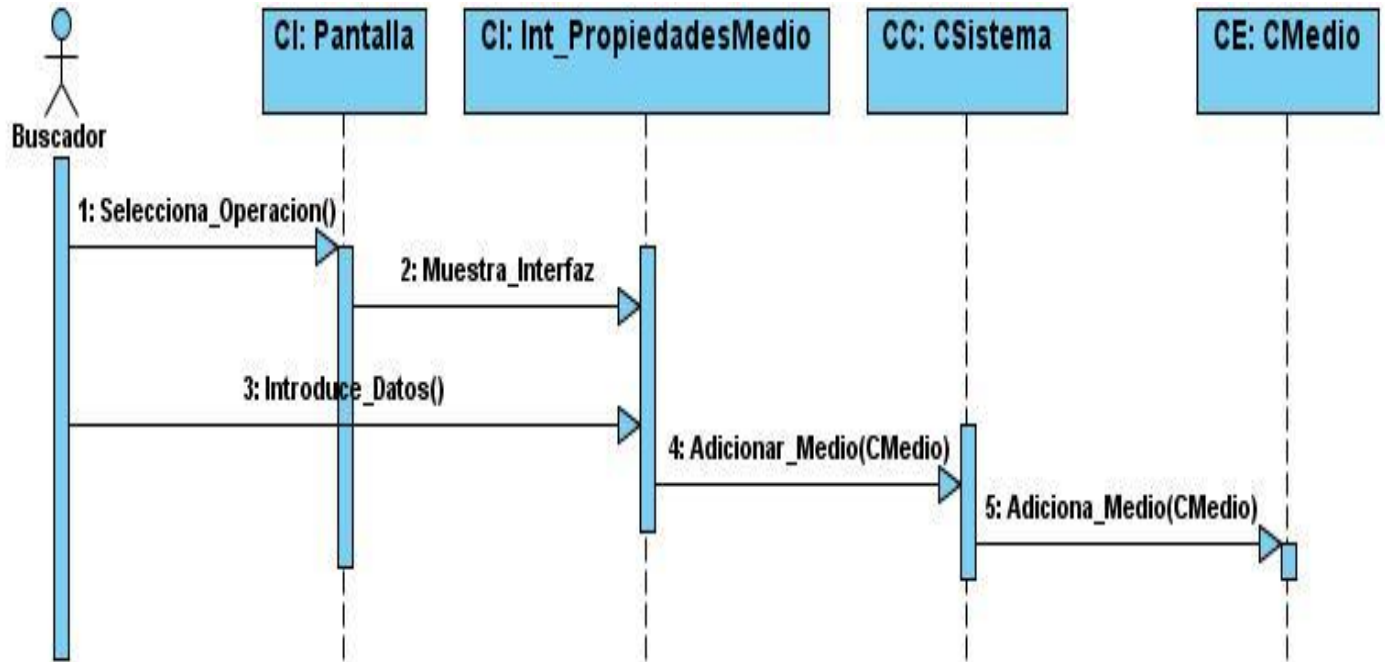
DS: Elimina un Grupo de Sumarios



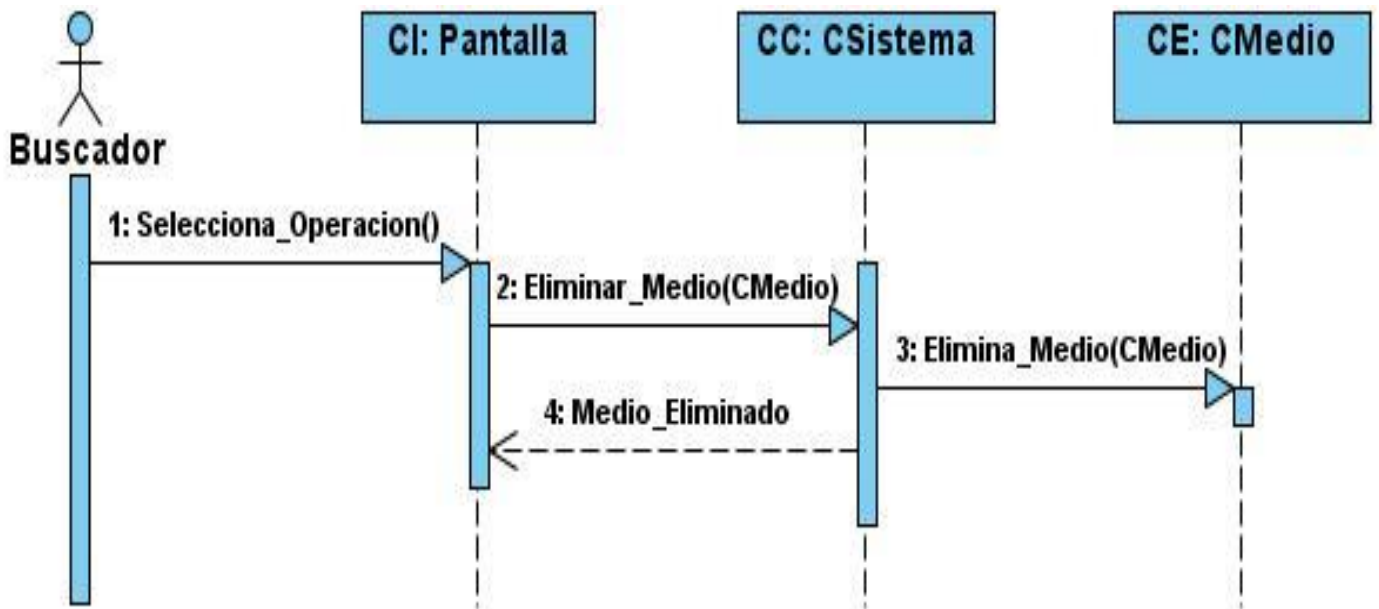
DS: Actualizar Grupo de Sumarios



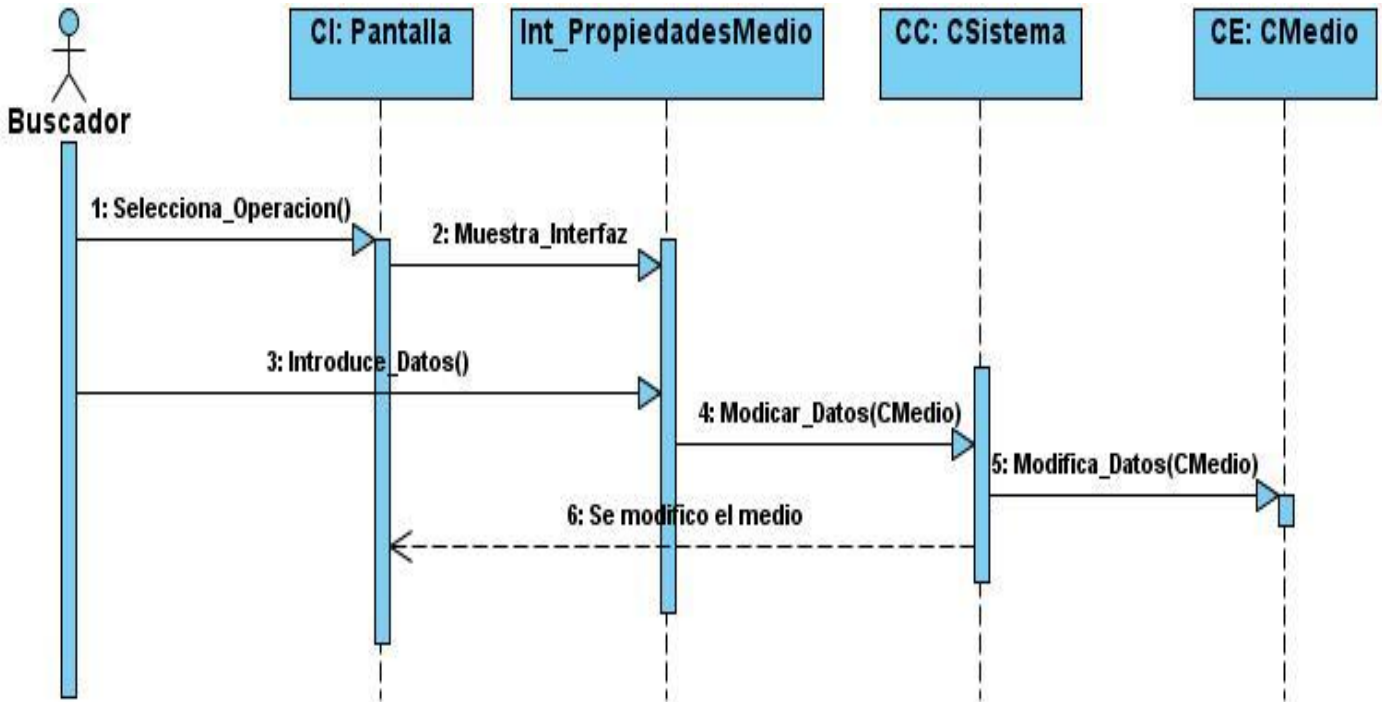
DS: Adicionar Medio



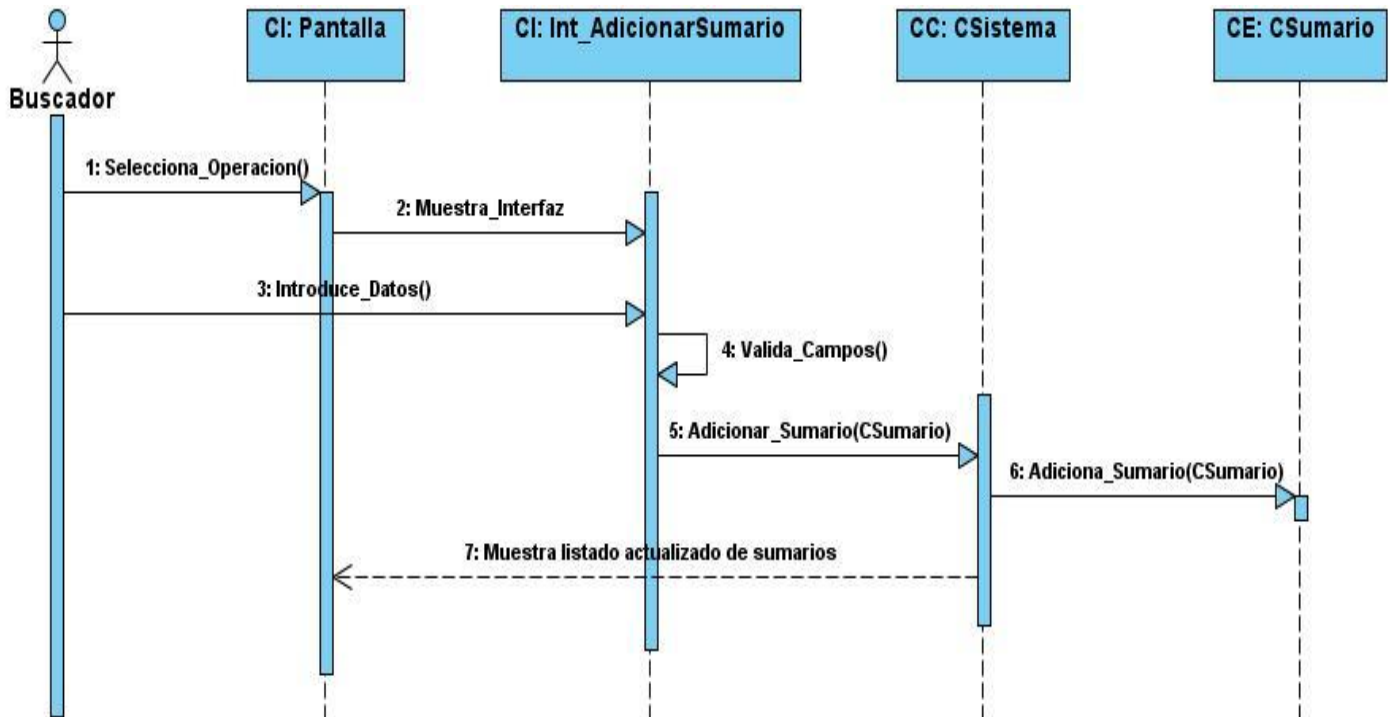
DS: Eliminar Medio



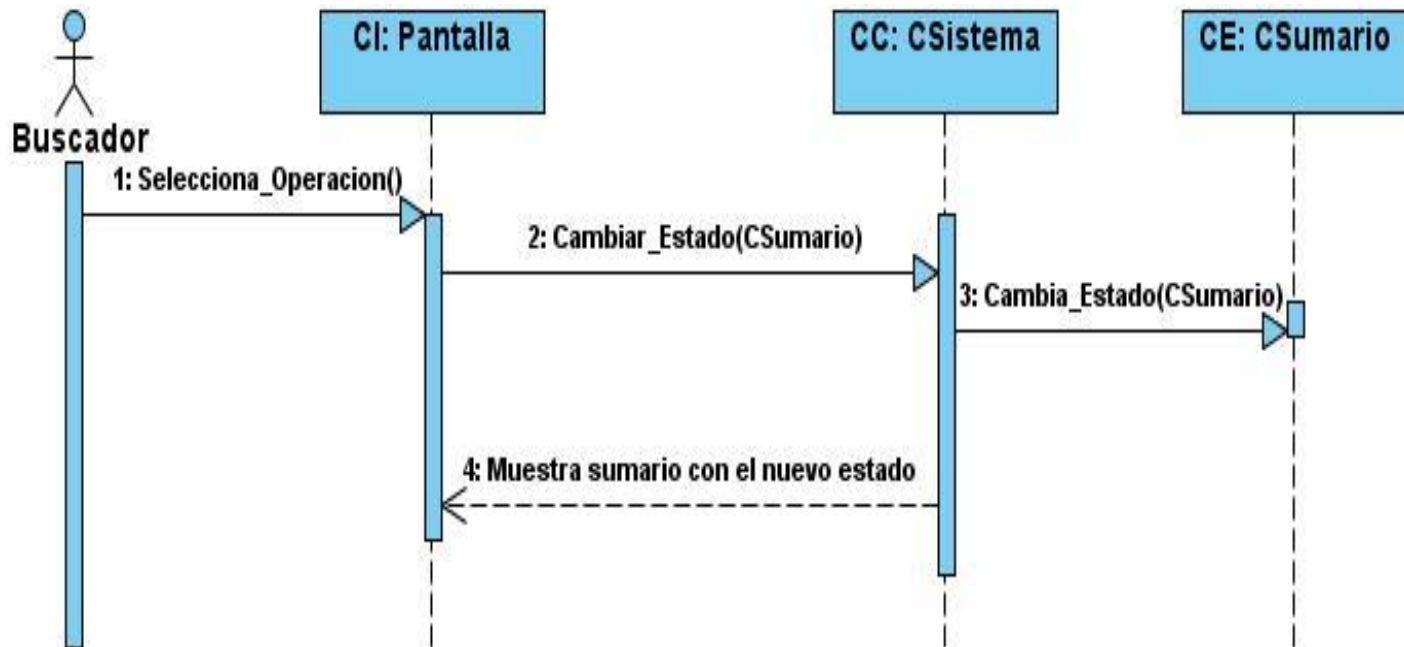
DS: Cambiar Propiedades Medio



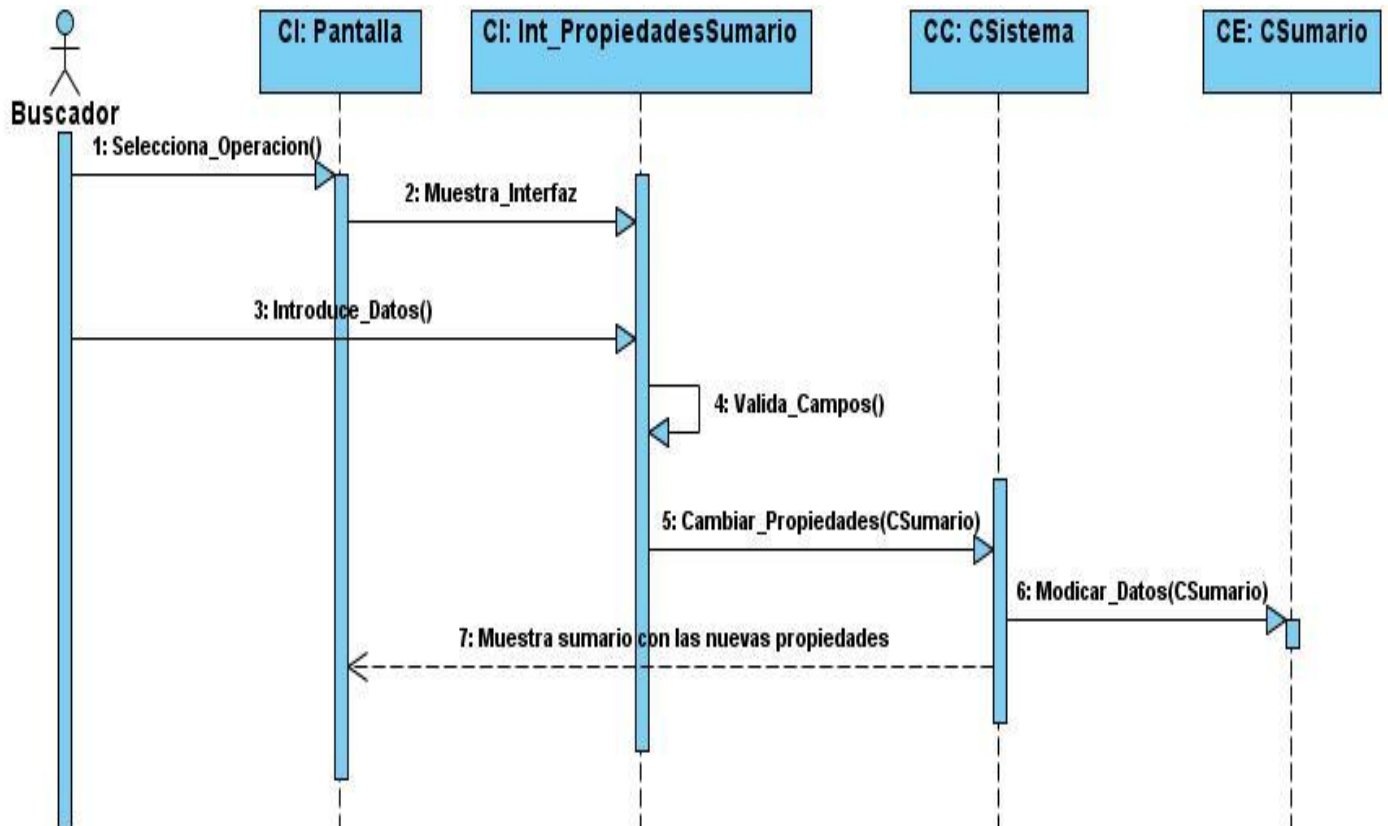
DS: Adicionar Sumario



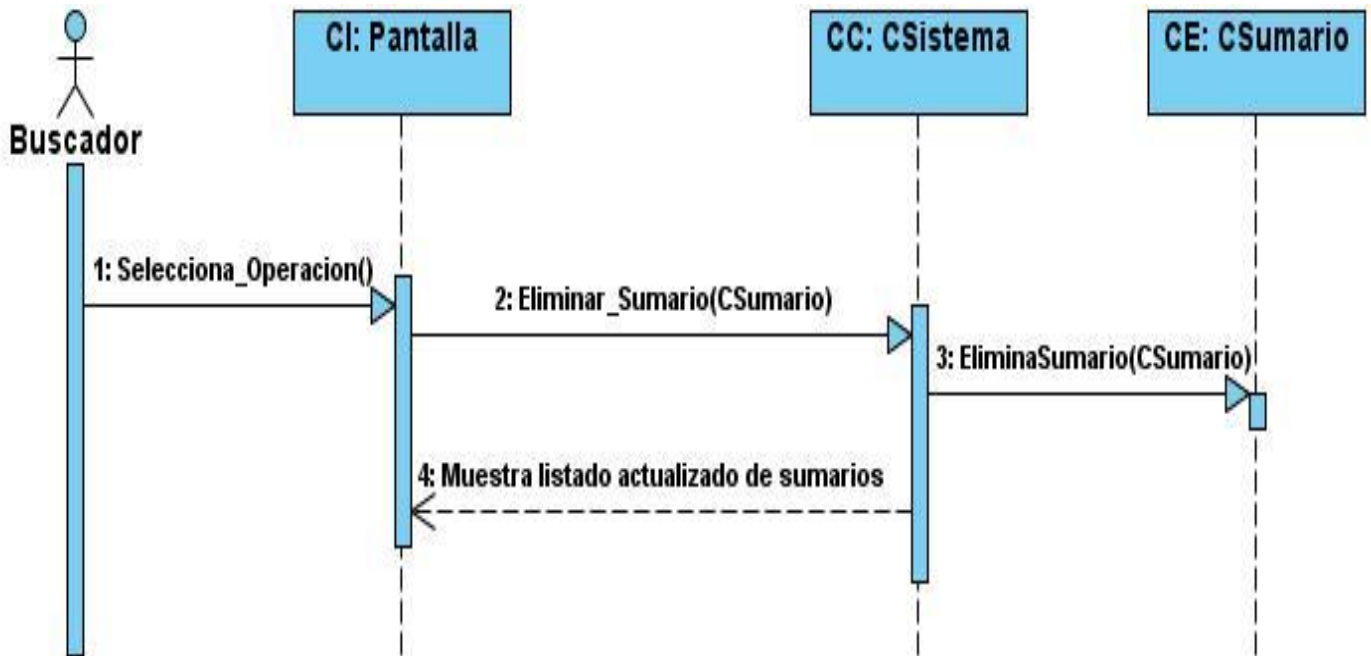
DS: Cambiar Estado Sumario



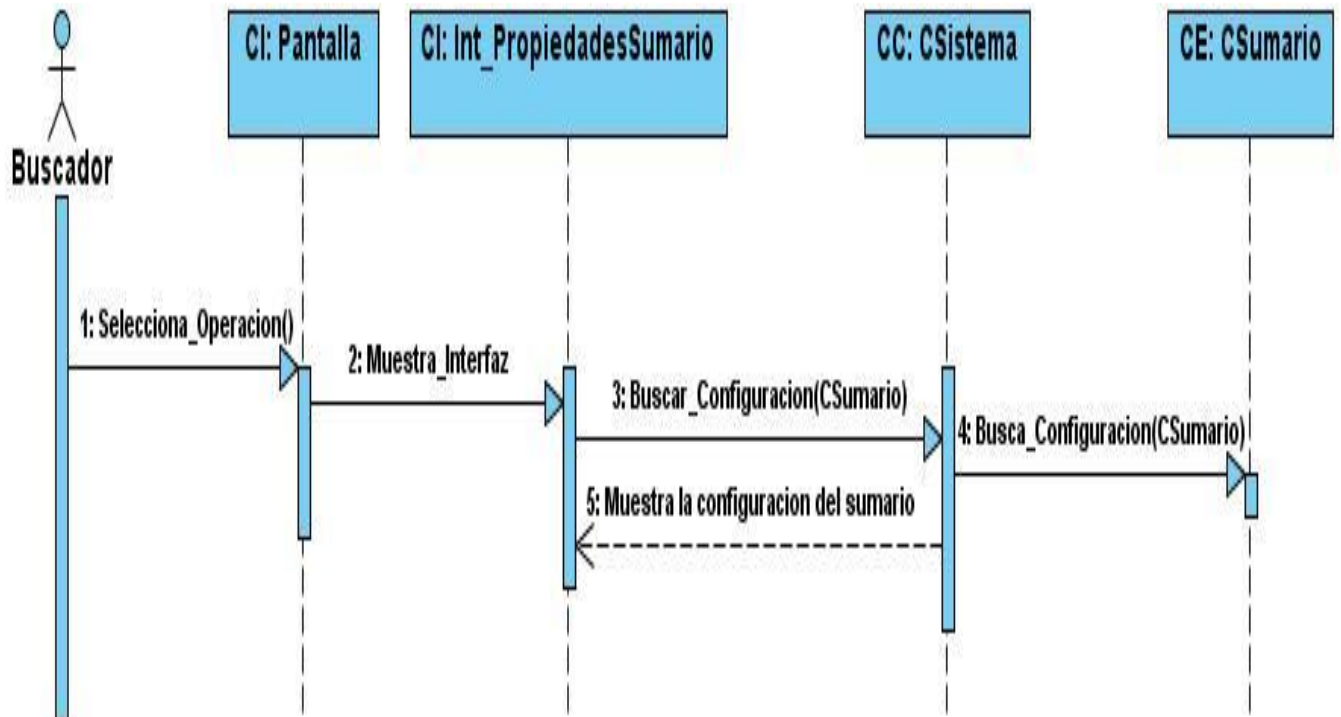
DS: Cambiar Propiedades Sumario



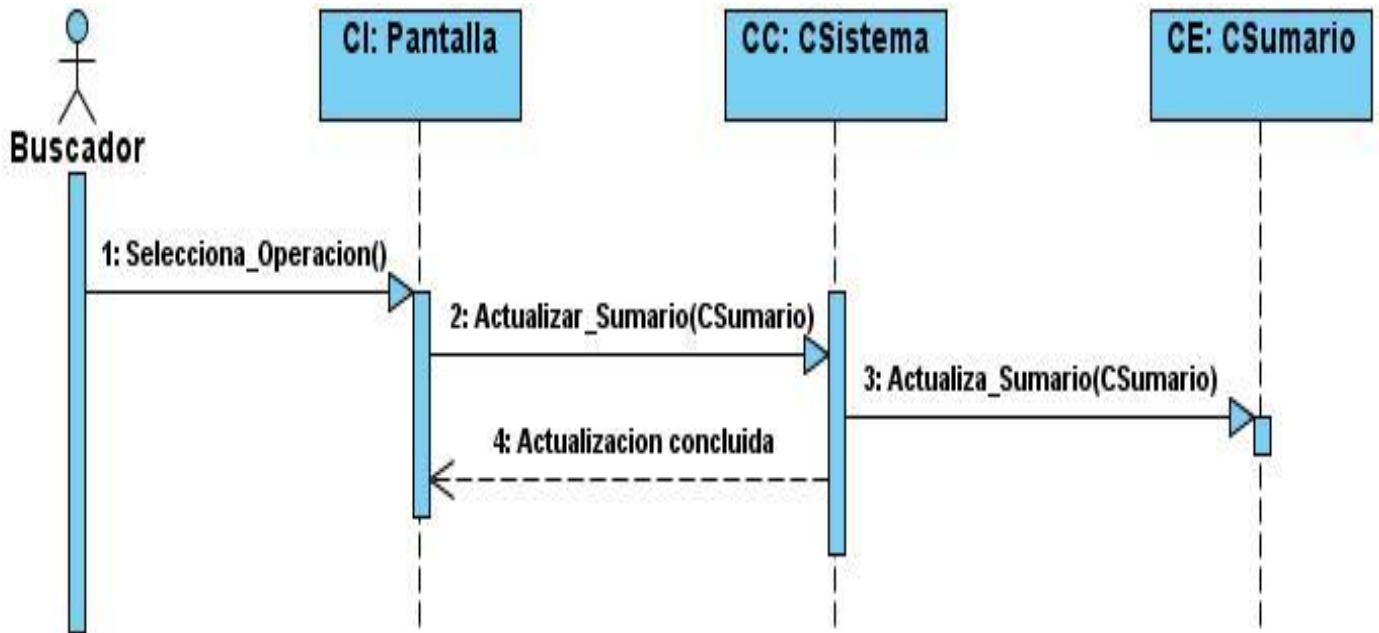
DS: Eliminar Sumario



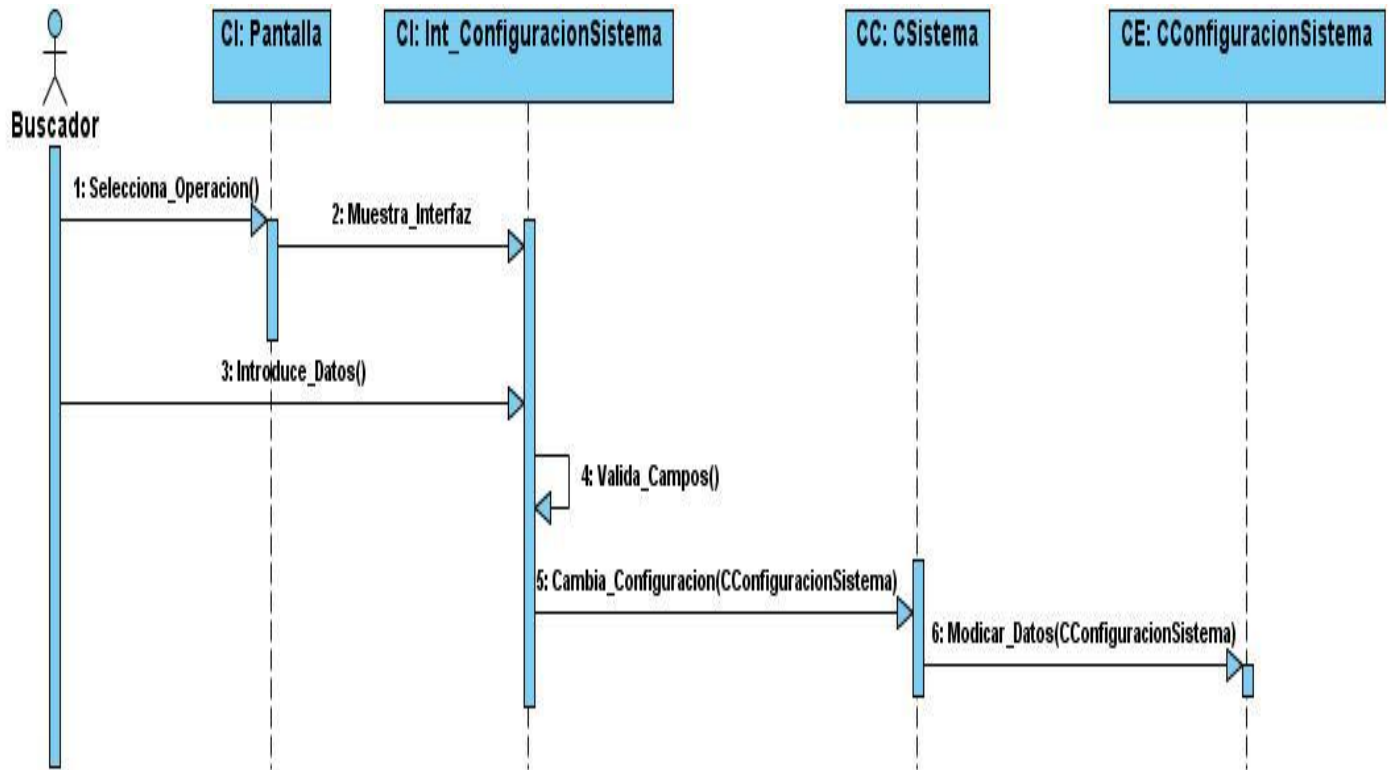
DS: Solicitar Configuración Sumario



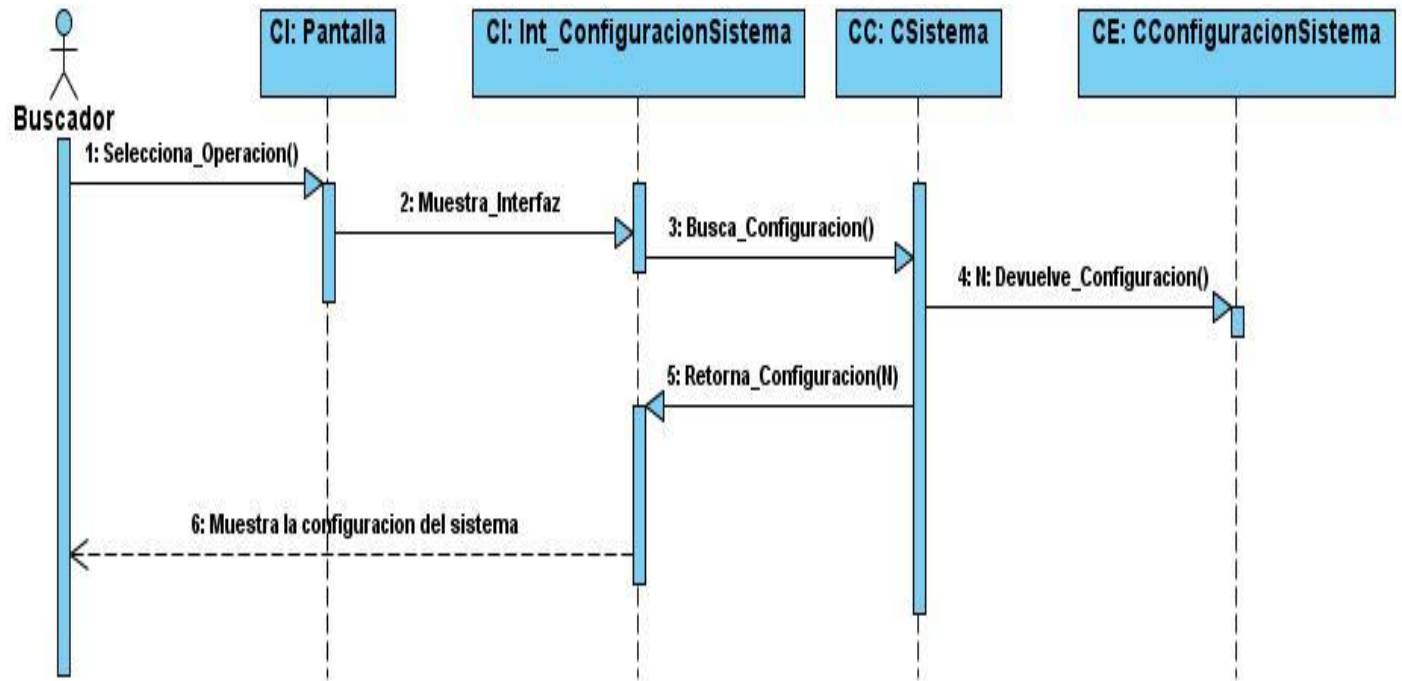
DS: Actualizar Sumario



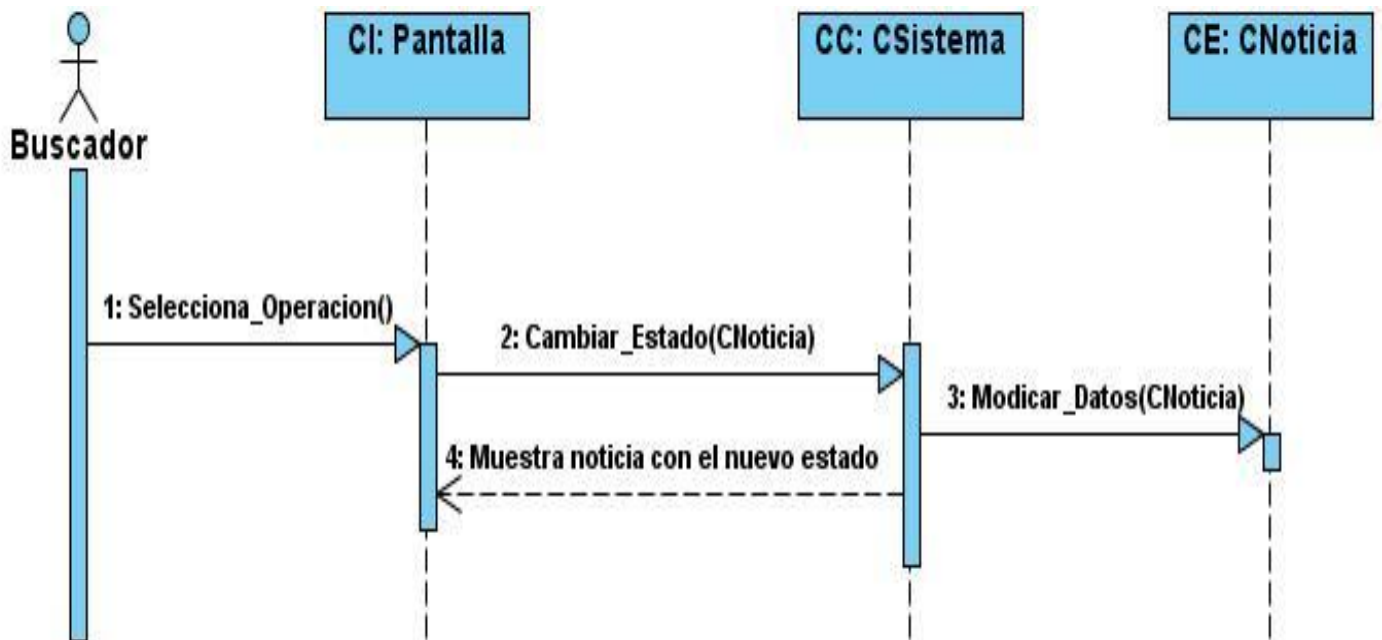
DS: Cambiar la configuración del sistema



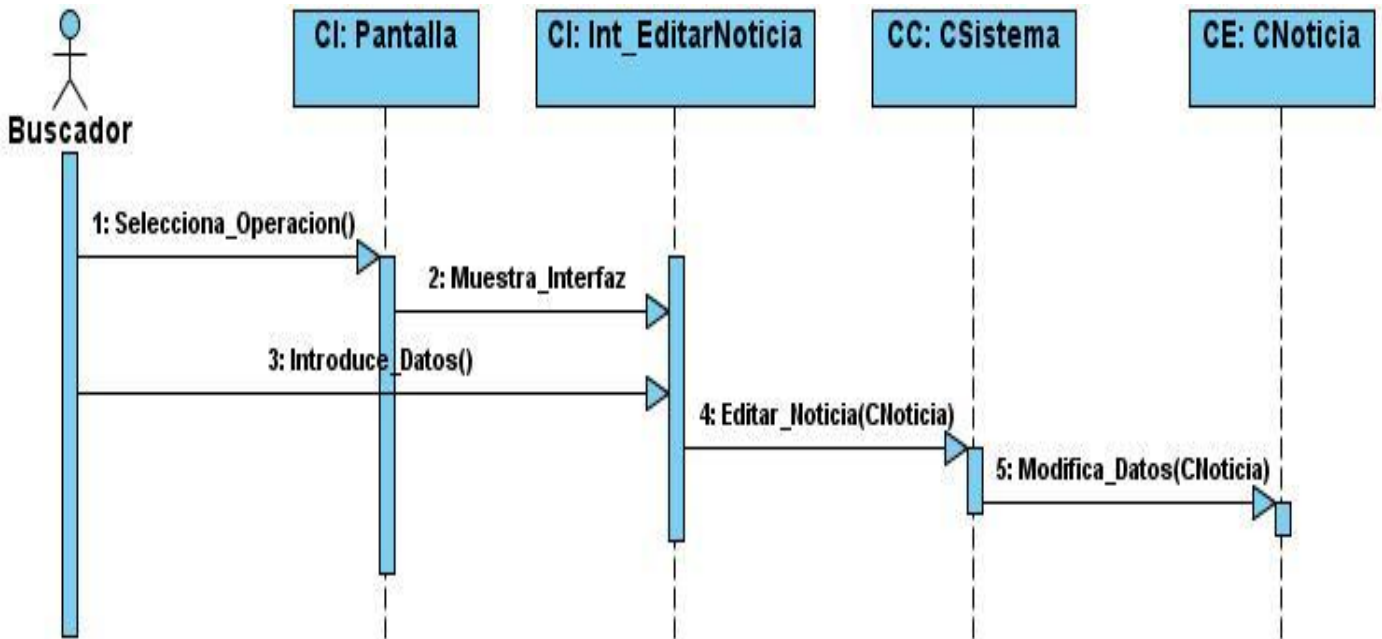
DS: Mostrar la Configuración del Sistema



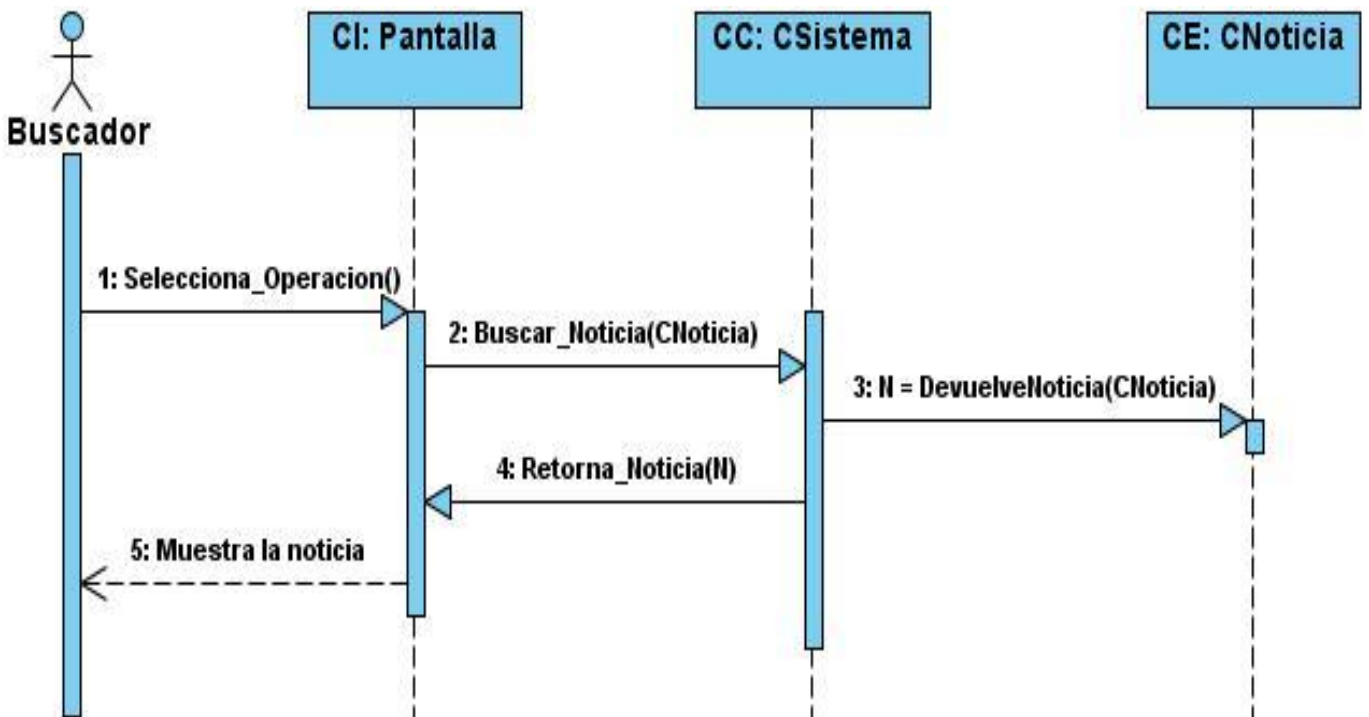
DS: Cambiar Estado Noticia



DS: Editar Noticia



DS: Mostrar Noticia Vista Normal



3.5. Diseño de la Base de Datos

En el Sistema de Monitoreo de Análisis de Noticias se utilizará una base de datos orientada a objetos (DB4O) como se trató en epígrafes anteriores. Por lo que en vez de tener un modelo de datos como los tienen las aplicaciones que utilizan base de datos relacionales, pues se utiliza un modelo de objetos.

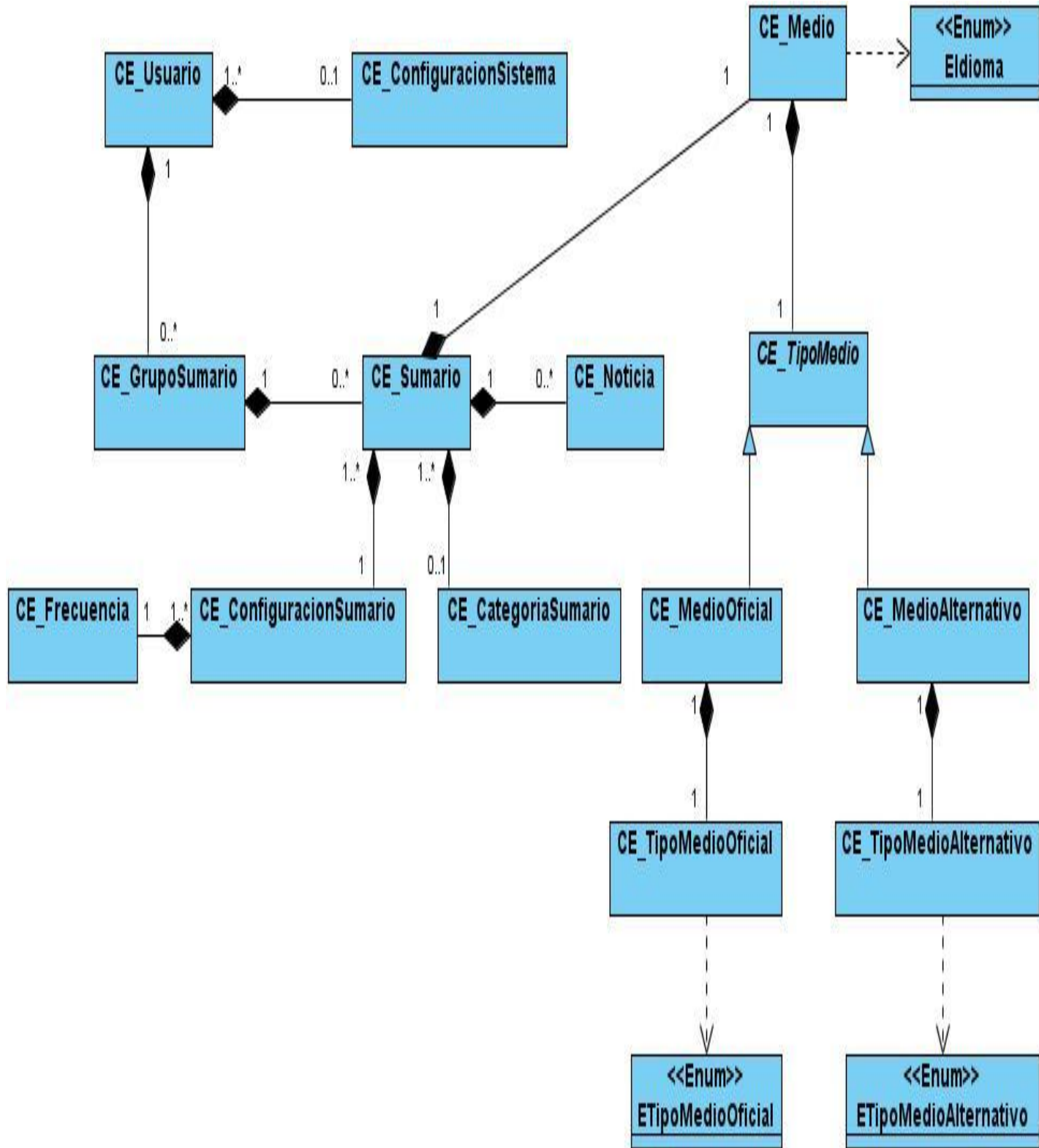
El diseño de la base de datos está dado por su diagrama de clases persistentes. El modelo diseñado representa una base de datos OO (Orientada a Objetos), manteniendo información acerca de sus usuarios, grupos de sumarios, sumarios, configuraciones del sistema, etc.

A diferencia del modelado Entidad-Relación de las bases de datos relacionales, el diagrama de una base de datos OO no necesita pasar por los modelos conceptual, lógico y físico, lo que representa una ventaja a la hora de la mantención del modelo, puesto que significa menor esfuerzo. Esto también significa que no se deben realizar operaciones especiales para representar relaciones N: N o recursivas.⁵⁹

A continuación se le presenta el modelo de objetos del Sistema de Monitoreo y Análisis de Noticias.

⁵⁹ MOLINA, A. A. S. *HERRAMIENTA PARA EL CONTROL DE INTEGRIDAD DE DATOS EN DB4O* UNIVERSIDAD AUSTRAL DE CHILE, CAMPUS PUERTO MONTT , ESCUELA DE INGENIERIA EN COMPUTACION 2007. p.

Modelo de Objetos



Al modelo anterior se le aplicó la técnica de “Normalización de Clases”, la cual tiene por finalidad mejorar la calidad del modelo.

Esta normalización es similar a la “Normalización de datos”, y se denominan “Primera forma normal del objeto” (1ONF), “Segunda forma normal del objeto” (2ONF) y “Tercera forma normal del objeto” (3ONF).

La normalización de clases fue definida por Ambler en su libro: Agile DatabaseTechniques: Effective Strategies for the Agile Software Developer, publicado en el año 2003.

En toda base de datos es muy importante la integridad de los datos. Ya que es la que se encarga de la corrección y completitud de los datos. A los que al realizar diversas operaciones sobre ellos, como es el caso de una actualización, un insertado o un borrado, se podrían estar modificando datos existentes por valores incorrectos.

DB4O tiene una forma especial de tratar la integridad de sus datos. La misma se implementa en la configuración de la base de datos antes de abrir la conexión. Unos de los aspectos fundamentales de la integridad consisten en evitar la duplicidad de objetos. Es decir, que un mismo objeto no esté almacenado más de una vez.

En el motor de base de datos DB4O se evita la duplicidad de objetos mediante la indexación de los campos identificadores de las clases persistentes. Un ejemplo sería:

```
config.objectClass(CE_Usuario.class).objectField("usuario").indexed(true);  
config.add(new UniqueFieldValueConstraint(CE_Usuario.class,"usuario"));
```

En el ejemplo anterior se está indexando el atributo “usuario” de la clase “CE_Usuario”, por lo que en la base de datos no existirán dos objetos de CE_Usuario que tengan el mismo “usuario”.

Otras formas de lograr la integridad de los datos son a través de las transacciones tratadas en epígrafes anteriores. Como son: el commit, y el rollback, entre otros.

Descripción de las Clases Persistentes

Al utilizarse un motor de base de datos como DB4O, el diseño de la base de datos consiste, como se explicó anteriormente, en las clases persistentes del sistema. Ya que al ser una base de datos Orientada a Objetos, dejan de existir las tablas y relaciones entre ellas. Por lo que en lugar de presentarse la descripción de las tablas, a continuación se presenta la descripción de las clases persistentes, con una breve descripción de los atributos de los mismos para lograr un mejor entendimiento del modelo de objetos.

Nombre: CE_Usuario		
Tipo de clase: Clase Entidad		
Atributo	Tipo	Descripción
nombre	String	Almacena el nombre del usuario.
apellidos	String	Almacena los apellidos del usuario.
correo	String	Almacena el correo del usuario.
usuario	String	Es el identificador de la clase.
contrasenna	String	Almacena la contraseña del usuario.
conectado	boolean	Almacena el estado de conexión del usuario.
grupoSumarios	List<CE_GrupoSuario>	Almacena la lista de grupos de sumarios del usuario.
configuracionSistema	CE_ConfiguracionSistema	Almacena la configuración del sistema del usuario.
Para cada responsabilidad:		

Capítulo3. Descripción y Análisis de la Solución Propuesta.

En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.

Nombre: CE_Sumario		
Tipo de clase: Clase Entidad		
Atributo	Tipo	Descripción
titulo	String	Es el identificador de la clase.
descripcion	String	Almacena la descripción del sumario.
fechaadicion	Calendar	Almacena la fecha de adición del sumario a la base de datos.
historialdeactualizacion	List<Date>	Almacena el historial de actualizaciones del sumario.
noticias	List<CE_Noticia>	Almacena la lista de noticias pertenecientes al sumario.
actualizandose	boolean	Almacena si el sumario está siendo actualizado en el instante en que se recupera este valor.
noticiasCandidatas	List<CE_Noticia>	Es una lista de noticias que se verifican que no existan antes de almacenarlas.
prioridad	int	Almacena la prioridad del sumario.

Capítulo3. Descripción y Análisis de la Solución Propuesta.

medio	CE_Medio	Almacena el medio al que pertenece el sumario.
categoria	CE_CategoriaSumario	Almacena la categoría del sumario.
configuracion	CE_ConfiguracionSumario	Almacena la configuración del sumario.
leido	boolean	Almacena si el sumario ha sido leído o no.

Para cada responsabilidad:

En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.

Nombre: CE_GrupoSumario

Tipo de clase: Clase Entidad

Atributo	Tipo	Descripción
nombre	String	Es el identificador de la clase.
descripcion	String	Almacena la descripción del grupo de sumarios.
sumarios	List<CE_Sumario>	Almacena la lista de sumarios pertenecientes al grupo de sumario.
leido	boolean	Almacena si el grupo de sumarios ha sido leído o no.

Para cada responsabilidad:

En esta clase solo se definieron métodos de acceso, es decir, operaciones que

Capítulo3. Descripción y Análisis de la Solución Propuesta.

solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.

Nombre: CE_Noticia		
Tipo de clase: Clase Entidad		
Atributo	Tipo	Descripción
titulo	String	Es el identificador de la clase.
descripcion	String	Almacena la descripción de la noticia.
url	String	Almacena la dirección URL de la noticia.
fechaPublicacion	String	Almacena la fecha de publicación de la noticia.
fechaAdicionBaseDatos	Date	Almacena la fecha en que la noticia se adicionó a la base de datos.
autor	String	Almacena el autor de la noticia.
categoria	String	Almacena la categoría de la noticia.
palabrasClaves	List<String>	Almacena las palabras de la noticia.
leida	boolean	Almacena la noticia fue leída o no.
sumario	String	Almacena el sumario al que pertenece la noticia.
relevancia	int	Almacena la relevancia de la noticia.
Para cada responsabilidad:		
En esta clase solo se definieron métodos de acceso, es decir, operaciones que		

solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.

Nombre: CE_Medio		
Tipo de clase: Clase Entidad		
Atributo	Tipo	Descripción
nombre	String	Es el identificador de la clase.
descripcion	String	Almacena la descripción del medio.
idioma	EIdioma (enumerativo)	Almacena el idioma del medio.
tipoMedio	CE_TipoMedio	Almacena el tipo de medio del medio.
url	String	Almacena la dirección URL del medio.
pais	EPaises (enumerativo)	Almacena el país al que pertenece el medio.
Para cada responsabilidad:		
<p>En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.</p>		

Nombre: CE_MedioOficial		
Tipo de clase: Clase Entidad		
Atributo	Tipo	Descripción
tipoMedio	CE_TipoMedioOficial	Almacena el tipo de medio oficial del sumario a que

Capítulo3. Descripción y Análisis de la Solución Propuesta.

		pertenezca.
Para cada responsabilidad:		
<p>En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.</p>		

Nombre: CE_TipoMedioOficial		
Tipo de clase: Clase Entidad		
Atributo	Tipo	Descripción
medioOficial	ETipoMedioOficial (enumerativo)	Es un enumerativo que se le asigna a los medios oficiales y se almacena en la base de datos.
Para cada responsabilidad:		
<p>En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.</p>		

Nombre: CE_TipoMedioAlternativo		
Tipo de clase: Clase Entidad		
Atributo	Tipo	Descripción
medioAlternativo	ETipoMedioAlternativo (enumerativo)	Es un enumerativo que se le asigna a los medios alternativos y se almacena en

Capítulo3. Descripción y Análisis de la Solución Propuesta.

		la base de datos.
Para cada responsabilidad:		
<p>En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.</p>		

Nombre: CE_Frecuencia		
Tipo de clase: Clase Entidad		
Atributo	Tipo	Descripción
tiempoDescarga	int	Almacena cada que tiempo de realizará la descarga.
tiempoLimpieza	int	Almacena cada que tiempo se realizará la limpieza automática de la base de datos.
tipoTiempo	int	Almacena cual es el tipo de tiempo que se va a emplear, ya sean minutos, horas o días.
Para cada responsabilidad:		
Nombre:	TiempoDescarga(Date pfechaAdicion)	
Descripción:	Método que avisa si ya es tiempo para levantar el hilo de actualización del sumario.	
Nombre:	TiempoLimpieza(Date pfechaAdicion)	
Descripción:	Método que avisa si ya es tiempo para levantar el hilo de limpieza del	

Capítulo3. Descripción y Análisis de la Solución Propuesta.

	sumario.
Nombre:	TiempoTranscurridoDescarga(Date pfechaAdicion)
Descripción:	Calcula el tiempo que ha transcurrido para la descarga
Nombre:	TiempoTranscurridoLimpieza(Date pfechaAdicion)
Descripción:	Calcula el tiempo que ha transcurrido para la limpieza
Además posee las propiedades que devuelven o permiten cambiar los valores de los campos o atributos.	

Nombre: CE_ConfiguracionSistema		
Tipo de clase: Clase Entidad		
Atributo	Tipo	Descripción
puerto	int	Almacena el puerto del proxy que se va a utilizar.
host	String	Almacena el host del proxy que se va a utilizar por ejemplo 10.0.0.1
usuario	String	Almacena el usuario del dominio que se va a utilizar.
contrasenna	String	Almacena la contraseña del dominio que se va a utilizar.
caminoBaseDatos	String	Almacena el camino a la base de datos.
Para cada responsabilidad:		
En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son		

conocidas como propiedades.

Nombre: CE_CategoriaSumario

Tipo de clase: Clase Entidad

Atributo	Tipo	Descripción
nombre	String	Almacena el nombre de la categoría.
color	Color	Almacena el color de la categoría.

Para cada responsabilidad:

En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.

Nombre: CE_ConfiguracionSumario

Tipo de clase: Clase Entidad

Atributo	Tipo	Descripción
frecuenciaDescarga	CE_Frecuencia	Almacena la frecuencia con la que se realizará la descarga del sumario.
cantNoticiasMostradas	int	Almacena la cantidad de noticias que se mostrarán en el sumario.
nuevaNoticia	boolean	Almacena si el usuario desea recibir notificaciones en la bandeja del sistema cada vez que el sumario se actualice.

Para cada responsabilidad:

En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.

Nombre: CE_MedioAlternativo

Tipo de clase: Clase Entidad

Atributo	Tipo	Descripción
tipoMedio	CE_TipoMedioAlternativo	Almacena el medio alternativo del sumario al que pertenezca.

Para cada responsabilidad:

En esta clase solo se definieron métodos de acceso, es decir, operaciones que solamente se limitan a devolver o cambiar atributos. Este tipo de operaciones son conocidas como propiedades.

3.6. Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de

programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.⁶⁰

Para el desarrollo del Módulo de Búsqueda del Sistema de Monitoreo y Análisis de Noticias se emplearon varias librerías las cuales son consideradas componentes del sistema, en conjunto con el ejecutable general del mismo. Por lo que el diagrama de componentes quedaría de la siguiente forma:

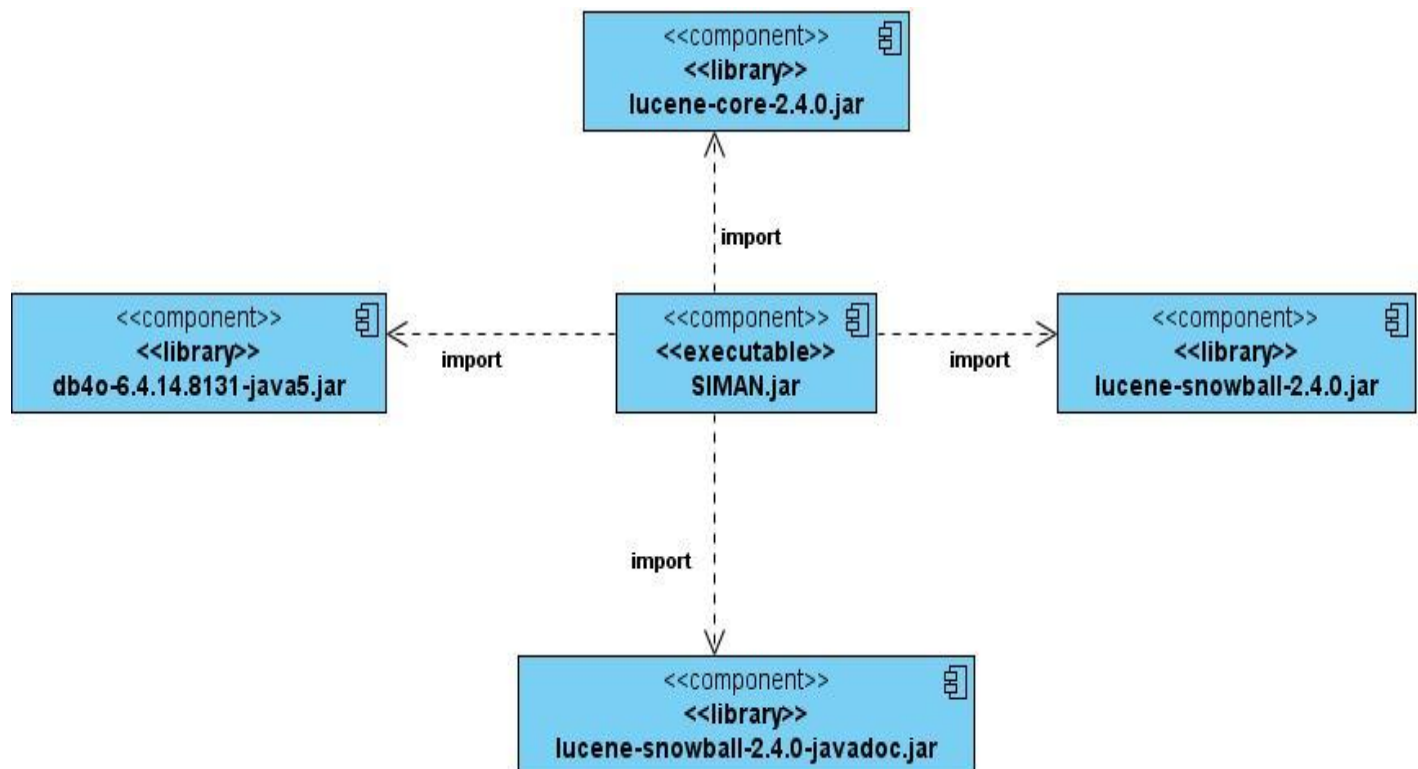


Diagrama de Componentes

⁶⁰ VILAS, A. F. *Diagrama de Componentes* En línea 2001. [Consultada Abril 7, 2009]. Disponible en: <http://tvdi.det.uvigo.es/~avilas/UML/node49.html#sub:componentes>

Capítulo3. Descripción y Análisis de la Solución Propuesta.

En este capítulo se realizó una modelación y representación del sistema en términos del diseño, para cada uno de los casos de uso del sistema y por escenarios de los mismos. Lo que permitió lograr una mejor comprensión de las funcionalidades de la solución propuesta. También se identificaron las clases interfaces, de control y de entidad de este flujo de trabajo. Se obtuvieron el diagrama de clases del diseño y los diagramas de secuencia. Se logró realizar el diseño de la base de datos del sistema.

CONCLUSIONES

El monitoreo de noticias se ha convertido en una actividad fundamental para facilitar la toma de decisiones. Se destaca el uso de aplicaciones de búsqueda mediante canales de sindicación pues agilizan la obtención de información de forma rápida, resumida y de medios confiables.

Mediante la aplicación de la programación multihilo se garantiza la ejecución concurrente de operaciones evitando la congestión en el sistema.

Para el procesamiento y análisis de información es necesario archivar noticias durante determinado período de tiempo, para lo cual se requiere un motor de base de datos ligero y de alto rendimiento, el DB4O se consideró el más indicado.

Con las potencialidades del metalenguaje XML, los canales de sindicación y el DB4O se construyó un componente de búsqueda para el Sistema de Análisis y Monitoreo de Noticias que permite la búsqueda, edición y almacenamiento del extracto de las noticias publicadas en los canales de sindicación de los medios de información seleccionados por el usuario.

RECOMENDACIONES

Se cumplieron los objetivos planteados en la investigación, aunque se considera que por lo novedoso del tema, se pueden realizar ampliaciones, modificaciones y mejoras de este Módulo de Búsqueda. También se pueden desarrollar nuevas versiones donde se incluyan mejoras al diseño propuesto para dicha aplicación.

Por lo que se proponen las siguientes recomendaciones:

- Continuar el estudio del motor de base de datos DB4O.
- Realizar las pruebas al software.
- Incluir las API de Google para lograr la sindicación de google news.
- Desarrollar la versión cliente-servidor del sistema para lograr obtener los beneficios que el trabajo en equipo pueden proveer en tareas como el monitoreo de noticias.
- Implementar la descarga del cuerpo de las noticias seleccionadas.

BIBLIOGRAFÍA

- ❖ AGUT, R. M. Especificación de Requisitos Software según el estándar de IEEE 830, [En línea 2000-2001]. [Consultada Marzo 4, 2009]. Disponible en: http://www3.uji.es/~coltell/Docs/IngSw_Apuntes/ISW_2004_D08.pdf
- ❖ ALCAIDE, J. P. Convenciones de código Java, [En línea 2008]. [Consultada Marzo 7, 2009]. Disponible en: <http://jpereza.nom.es/convenciones-de-codigo-java>
- ❖ ANÓNIMO. Descripción de Java 2 Plataform Micro Edition, [Consultada Marzo 2, 2009]. Disponible en: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1147/5/T%2011010%20CAP%202.pdf>
- ❖ ANÓNIMO. Ingeniería de Software con UML Unified Modeling Language - Lenguaje Unificado de Modelado, [Consultada Marzo 6, 2009]. Disponible en: <http://www.eduardoleyton.com/apuntes/Uml.pdf>
- ❖ BARRANCO, M. R. Doctorado en informática sockets: comunicación entre procesos distribuidos, [En línea] 1996. [Consultada Marzo 5, 2009]. Disponible en: <http://es.tldp.org/Universitarios/seminario-2-sockets.html>
- ❖ BERRUETA, D. Programación extrema y software libre, [En línea 2006]. [Consultada Marzo 7, 2009]. Disponible en: http://www.asturlinux.org/archivos/jornadas2006/ponencias/ProgExtrema_Berrueta/ponencia-sl-y-xp.pdf

- ❖ BIBIANA ROSSI, P. B. Y. R. G. M. MODELADO DE OBJETOS, pag 3, Consultada Marzo 2, 2009]. Disponible en:
http://www.itba.edu.ar/archivos/secciones/art_revistas_2.pdf
- ❖ CALANDRIA, A. d. Monitoreo de Medios. CALANDRIA. [En línea] 2000, Febrero. [Consultada: Noviembre 7, 2008]
Disponible en: <http://www.calandria.org.pe>.
- ❖ CATALÁN, J. F. Y. M. A. Bitácoras y sindicación de contenidos: dos herramientas para difundir información, [En línea 2005]. [Consultada Marzo 8, 2009].
Disponible en: <http://www.ub.es/bid/15frang2.htm>
- ❖ CEDITEC. Centro de difusión de tecnologías ETSIT-UPM, [Consultada Marzo 9, 2009]. Disponible en: <http://www.ceditec.etsit.upm.es/index.php/Que-es-CEDITEC.html>
- ❖ CENTROAMERICAJOVEN.ORG. Intercambiando conocimiento sobre juventud en la región, [Consultada Marzo 9, 2009]. Disponible en:
<http://centroamericajuven.org/quienes-somos.html>
- ❖ Contenidos, G. d. LA VIGILANCIA TECNOLÓGICA. Innovación. [En línea] 2006, Junio 21. [Consultada: Diciembre 1, 2008] Disponible en:
http://www.innovacion.com.es/index.php?option=com_content&task=view&id=43&Itemid=28.
- ❖ DEVELOPER.DB4O.COM. db4objects, [En línea 2009]. [Consultada Marzo 6, 2009]. Disponible en: <http://www.db4o.com>
- ❖ Dora, C. Haz tu programa. [En línea] 2008 [Consultada: Marzo 10, 2009] Disponible en: <http://www.haztuprograma.com/tiposTecnologias.html#java>.

- ❖ FUMERO, A. El abecé del universo blog, [Consultada Marzo 9, 2009]. Disponible en: <http://antoine.eduangi.com/MyWeb/Papeles/Blogotutorial.pdf>
- ❖ GARCÍA, C. S. La Web 2.0 y sus implicaciones en el mundo de la gestión de la información, [En línea 2007]. [Consultada Marzo 9, 2009]. Disponible en: <http://www.iuetaeb.tec.ve/caa/html/La%20Web%202.0%20y%20sus%20implicaciones%20en%20el%20mundo%20de%20la%20gesti%C3%B3n%20de%20la%20informaci%C3%B3n.pdf>
- ❖ GAYO, J. E. L. Desarrollo Modular de Procesadores de Lenguajes a partir de Especificaciones Semánticas reutilizables., [En línea 2001]. [Consultada Marzo 1, 2009]. Disponible en: <http://www.di.uniovi.es/~labra/tesis/tesis.pdf>
- ❖ GONZÁLEZ, F. S. XML (EXTENSIBLE MARKUP LANGUAGE): NUEVO ESTÁNDAR PARA LA DESCRIPCIÓN DE DOCUMENTOS EN LA WORD WIDE WEB[En línea 2008]. [Consultada Marzo 8, 2009]. Disponible en: http://fesabid98.florida-uni.es/Comunicaciones/f_santamaria/f_santamaria.htm
- ❖ HUIDOBRO, J. M. La Sindicación Web: RSS [En línea 2008]. [Consultada Marzo 9, 2009]. Disponible en: <http://www.coit.es/publicaciones/bit/bit167/rincondeinternet.pdf>
- ❖ INFOMEDIOS. Información de medios para Colombia Infomedios Colombia Ltda., [Consultada Marzo 9, 2009]. Disponible en: http://www.infomediosco.com/index.php?option=com_content&task=view&id=14&Itemid=33
- ❖ ISSI, G. Metodologías Ágiles en el Desarrollo de Software, [En línea 2003]. [Consultada Marzo 5, 2009]. Disponible en: <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf#page=9>

- ❖ LETELIER, P. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)[Consultada Marzo 6, 2009].
Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- ❖ LOVELLE, J. M. C. Conceptos Básicos de Procesadores de Lenguaje [En línea 1998]. [Consultada Marzo 6, 2009]. Disponible en:
http://avellano.fis.usal.es/~lalonso/procesadores/10_conceptos_basicos_uniovi.pdf
- ❖ MARÍA, S. L. Introducción a los sistemas de base de datos, [Consultada Marzo 4, 2009]. Disponible en:
http://books.google.com/books?hl=es&lr=lang_es&id=Vhum351TK8C&oi=fnd&pg=PA1&dq=base+de+datos+definicion&ots=fwJ9RVb75h&sig=vHa9cFEmpUrIW7qkvjJleZa5XQ#PPA2,M1
- ❖ MERINO, C. U. Desarrollo de sistema Orientada a Objetos para la Biblioteca del Seminario Metropolitano de Concepción., Universidad del BIO-BIO, 2007. 360. p.
- ❖ MIGUEL A. PÉREZ TOLEDANO, A. N. M., CARLOS CANAL, JUAN M. MURILLO RODRÍGUEZ Desarrollo de Sistemas Basados en Componentes Utilizando Diagramas de Secuencia. , Consultada Abril 11, 2009]. Disponible en:
http://www.lcc.uma.es/~canal/papers/ideas05/tncm_ideas05.pdf
- ❖ MOLINA, A. A. S. HERRAMIENTA PARA EL CONTROL DE INTEGRIDAD DE DATOS EN DB4O UNIVERSIDAD AUSTRAL DE CHILE, CAMPUS PUERTO MONTT , ESCUELA DE INGENIERIA EN COMPUTACION 2007. p.
- ❖ MORIN, A. M. XML Parser, [En línea] 2009. [Consultada: Marzo 5, 2009].
Disponible en: <http://ruby.about.com/od/sz/g/xmlparse.htm>

- ❖ NAVARRO, J. M. Diseño del Sistema de Tarjeta de Crédito con UML, Consultada Abril 10, 2009]. Disponible en:
http://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Basic/mendoza_nj/Cap5.pdf
- ❖ NETBEANS. Sitio Oficial del Netbeans, [Consultada Marzo 3.2009]. Disponible en: <http://www.netbeans.org/community/releases/60/>
- ❖ OBTESOL. Observatorio Tecnológico de la Soldadura, [Consultada Marzo 9, 2009]. Disponible en:
http://www.obtesol.es/index.php?option=com_content&task=view&id=5&Itemid=35
- ❖ OESA. Observatorio Español de Acuicultura, [Consultada Marzo 9, 2009]. Disponible en: http://www.observatorio-acuicultura.org/es/es/index.php?option=com_content&task=view&id=6&Itemid=20
- ❖ OPTI, F. Observatorio de Prospectiva Tecnológica Industrial, [Consultada Marzo 9, 2009]. Disponible en: <http://www.opti.org/index/que.asp>
- ❖ OVTT. Observatorio Virtual de Transferencia de Tecnología, [Consultada Marzo 9, 2009]. Disponible en: <http://www.ovtt.es/objetivos.jsp>
- ❖ Pampeana, A. d. Monitoreo de medios. Notas pampeanas. . [En línea] 2006, Diciembre 5. [Consultada: Febrero 20, 2009]
Disponible en: <http://notaspampeanas.com.ar>.

- ❖ PERÚ, M. D. E. D. L. Orientaciones para el Monitoreo y Evaluación de Planes Operativos en Regiones, [En línea 2007]. [Consultada Marzo 8, 2009]. Disponible en:
http://www.minedu.gob.pe/normatividad/reglamentos/instructivo_monitoreo_evaluacion.pdf
- ❖ PLÁSTICO, A. I. T. D. Observatorio del Plástico, [Consultada Marzo 9, 2009]. Disponible en:
http://www.observatorioplastico.com/generador.php?pag=paginas/que_es.php
- ❖ PROGRAMACIONENJAVA.COM. Programación en Java, [Consultada Marzo 3, 2009]. Disponible en:
<http://www.programacionenjava.com/blog/2008/06/10/tutorial/tutorial-caracteristicas-de-java/>
- ❖ Publicidad, G. d. Monitoreo de noticias. Notiblog.[En línea] 2003, Julio 1. [Consultada: Diciembre 2, 2008]
Disponible en: http://notilog.com/notilog_nuevo/home/default.php.
- ❖ RIESCO, D. Un Workflow que Automatice los Procesos de Negocios del Proceso Unificado Rational, [En línea 2004]. [Consultada Marzo 5, 2009]. Disponible en:
<http://users.dsic.upv.es/~dromero/papers/wicc2004.pdf>
- ❖ ROMERO, A. Gestión de riesgos con CMMI, RUP e ISO en ingeniería de software minero, [En línea 2007]. [Consultada Marzo 6, 2009]. Disponible en:
http://www.scielo.org.pe/scielo.php?pid=S1561-08882007000100005&script=sci_arttext&tlng=es
- ❖ ROTTA, I. L. Z. Normalización, [En línea] 2009. [Consultada Marzo 8, 2009]. Disponible en: <http://sebx.files.wordpress.com/2008/05/norm3ra.pdf>

- ❖ SANCHO, J. M. LOS OBSERVATORIOS DE LA SOCIEDAD DE LA INFORMACIÓN: EVALUACIÓN O POLÍTICA DE PROMOCIÓN DE LAS TIC EN EDUCACIÓN, [En línea 2004]. [Consultada Marzo 8, 2009]. Disponible en: <http://www.rieoei.org/rie36a02.pdf>

- ❖ SANTANDER, U. F. D. P. En el Futuro de Todos, En línea 2002. [Consultada Marzo 7, 2009]. Disponible en: <http://www.ufps.edu.co/paginas/respuestas/pdf/revista7.pdf#page=25>

- ❖ SALISBURY., S. SQL transaction management with JDBC: roll back database transactions to undo errors.(Programming & Development: In-Depth ARTICLE). [En línea 2007]. [Consultada Marzo 6, 2009]. Disponible en: [http://find.galegroup.com/itx/retrieve.do?contentSet=IAC- Documents&resultListType=RESULT_LIST&qrySerId=Locale\(es%2C%2C\)%3AFQE%3D\(KE%2CNone%2C21\)transactions+database%24&sgHitCountType=Non e&inPS=true&sort=DateDescend&searchType=BasicSearchForm&tabID=T003&prodId=CDB&searchId=R1¤tPosition=2&userGroupName=ucinf&docId=A172525164&docType=IAC](http://find.galegroup.com/itx/retrieve.do?contentSet=IAC- Documents&resultListType=RESULT_LIST&qrySerId=Locale(es%2C%2C)%3AFQE%3D(KE%2CNone%2C21)transactions+database%24&sgHitCountType=Non e&inPS=true&sort=DateDescend&searchType=BasicSearchForm&tabID=T003&prodId=CDB&searchId=R1¤tPosition=2&userGroupName=ucinf&docId=A172525164&docType=IAC)

- ❖ Sitio, A. d. Sindicación de contenidos. Hola.com. .[En línea] Desde 2000. [Consultada: Diciembre 3, 2008] , Disponible en: <http://www.hola.com/canales>.

- ❖ Suárez, J. S. Consulta al sitio. Idict Villa Clara .[En línea] 2001, Enero 24. [Consultada: Diciembre 10, 2008]Disponible en: <http://www.idict.villaclara.cu/comunidad-cientifica/lineas-estrategicas>.

- ❖ TARRAGÓ, L. N. S. Sindicación de contenidos con canales RSS: aplicaciones actuales y tendencias [En línea 2007]. [Consultada Marzo 9, 2009]. Disponible en: http://bvs.sld.cu/revistas/aci/vol15_3_07/aci03307.htm

- ❖ VILAS, A. F. Diagrama de Componentes En línea 2001. [Consultada Abril 7, 2009]. Disponible en:
<http://tvdi.det.uvigo.es/~avilas/UML/node49.html#sub:componentes>
- ❖ WEEKLY, C. Hot skills: cut costs and coding time with Database for Objects.(Career Moves). [En línea 2007]. [Consultada Marzo 4, 2009]. Disponible en:
http://find.galegroup.com/itx/retrieve.do?contentSet=IAC- Documents&resultListType=RESULT_LIST&qrySerId=Locale%28es%2C%2C%29%3AFQE%3D%28KE%2CNone%2C4%29DB4O%3AAnd%3ALQE%3D%28AC%2CNone%2C8%29fulltext%24&sgHitCountType=None&inPS=true
- ❖ Wikipedia. Estandar de facto. Wikipedia [En línea] 2001, Enero 15. [Consultada: Febrero 20, 2009] Disponible en:
http://es.wikipedia.org/wiki/Est%C3%A1ndar_de_facto.
- ❖ WMW. Women's Media Watch, [Consultada Marzo 9, 2009]. Disponible en:
http://www.womensmediawatch.org/index.php?option=com_content&task=view&id=5&Itemid=6
- ❖ WORDREFERENCE.COM. Diccionario de la lengua española, [Consultada Marzo 1, 2009]. Disponible en:
<http://www.wordreference.com/definicion/programacion>
- ❖ XAVIER FERRÉ GRAU, M. I. S. S. Desarrollo Orientado a Objetos con UML, [En línea 2009]. [Consultada Marzo 7, 2009]. Disponible en:
<http://www.willydev.net/descargas/Articulos/General/umlTotal.pdf>

4. ANEXOS

4.1 Anexo 1. Descripción de las Clases del Diseño

Nombre: CC_Conexion	
Tipo de clase: Clase Controladora	
Atributo	Tipo
caminoBaseDatos	String
contenedor	ObjectContainer
Para cada responsabilidad:	
Nombre:	ConfigurarBD()
Descripción:	Configura la conexión a la base de datos. En este método es donde se garantiza la integridad de los datos y demás características que aumentan la escalabilidad y desempeño de la base de datos.
Nombre:	AbrirBD()
Descripción:	Abre la conexión a la base de datos.
Nombre:	CerrarBD()
Descripción:	Cierra la conexión con la base de datos
Nombre:	EstablecerCaminoBaseDatos()
Descripción:	Establece el camino donde se ubicará la base de datos.

Nombre: CC_ConsultaBD	
Tipo de clase: Clase Controladora	
Atributo	Tipo
indexador	CC_Indexador
Para cada responsabilidad:	
Nombre:	ActualizacionSumario(CE_Sumario psumario)
Descripción:	Actualiza en a base de datos el sumario pasado por parámetro
Nombre:	ActualizarUsuario(CE_Usuario pusuario)
Descripción:	Actualiza en a base de datos el usuario pasado por parámetro
Nombre:	ActualizarUnGrupoSumario(CE_GrupoSumario pgrupoSumarios)
Descripción:	Actualiza en a base de datos el grupo de sumarios pasado por parámetro
Nombre:	ActualizarTodosSumarios(List<CE_Sumario> plistaSumarios)
Descripción:	Actualiza en a base de datos todos los sumarios del usuario.
Nombre:	AdicionarUsuarioBD(CE_Usuario pusuario)
Descripción:	Adiciona los usuarios a la base de datos del sistema
Nombre:	CambiarContrasennaBD(String pcontrasennaV, String pcontrasennaN)
Descripción:	Cambia la contraseña del usuario en el sistema
Nombre:	VerificarContrasenna(String pusuario,String pcontrasennaV)
Descripción:	Verifica que la contraseña y usuario existen en la base de datos

Nombre:	CambiarEstadoLogueoBD(String pusuario,boolean plogueo)
Descripción:	Cambia el estado del usuario a conectado o no conectado en la base de datos del sistema.
Nombre:	Desconectar()
Descripción:	Desconecta del sistema al usuario
Nombre:	ActualizarUsuarioBD
Descripción:	Actualiza al usuario en la base de datos.
Nombre:	SumariosDelUsuarioBD()
Descripción:	Devuelve los sumarios del usuario en la base de datos
Nombre:	AdicionarGrupoSumarioBD(CE_GrupoSumario pgrupoSumario)
Descripción:	Adiciona a la base de datos los grupos de sumarios pasados por parámetros
Nombre:	ActualizarPropiedadesGrupoSumarioBD(String pnombreGrupoSumario, String pnombre, String pdescripcion)
Descripción:	Actualiza en la base de datos las propiedades de los grupos de sumarios pasados por parámetro.
Nombre:	CambiarEstadoGrupo(String pgrupoSumario, boolean pestado)
Descripción:	Cambia el estado de los grupos pasados por parámetro en leído o no leído
Nombre:	BorrarGrupoSumarioBD(String pnombreGrupoSumario)
Descripción:	Borra de la base de datos el grupo de sumario pasado por parámetro
Nombre:	AdicionarMedioBD(String pTituloSumario, CE_Medio pmedio)

Descripción:	Adiciona los medios pasados por parámetros a la base de datos y se los asigna al sumario pasado por parámetro
Nombre:	ActualizarPropiedadesMedio(CE_Sumario psumario)
Descripción:	Actualiza en la base de datos las propiedades del medio que pertenece al sumario pasado por parámetro.
Nombre:	BorrarMedioBD(String pTituloSumario)
Descripción:	Borra de la base de datos el medio que pertenece al sumario pasado por parámetro.
Nombre:	RenombrarMedio(String pTituloSumario, String pnombreMedio)
Descripción:	Renombra el medio del sumario pasado por parámetro.
Nombre:	CambiaEstadoSumarioBD(String psumario, boolean pestado)
Descripción:	Cambia en la base de datos el estado de leído o no leído, del sumario pasado por parámetro
Nombre:	ConfigurarSumarioBD(String pTituloSumario, CE_ConfiguracionSumario pconfiguracion)
Descripción:	Guarda la configuración del sumario pasada por parámetro en la base de datos. Para que sea aplicada al sumario cada vez que se levante el sistema.
Nombre:	AdicionarSumarioBD(String pgrupoSumario,CE_Sumario psumario)
Descripción:	Adiciona a la base de datos el sumario pasado por parámetros y se le asigna al grupo de sumarios pasado por parámetro.
Nombre:	CambiarCategoriaSumario(String ptituloSumario, CE_CategoriaSumario pcategoria)

Descripción:	Cambia en la base de datos la categoría del sumario pasado por parámetro
Nombre:	SumarioLeido(String ptituloSumario)
Descripción:	Verifica que el sumario pasado por parámetro haya sido leído o no
Nombre:	CambiarNombreSumario(String ptituloViejo, String ptitulo)
Descripción:	Cambia el nombre en la base de datos del sumario pasado por parámetro
Nombre:	ActualizarPropiedadesSumarioBD(String pTituloSumario, String ptitulo,String pdescripcion, Calendar pfechaadicion, String purl, int pprioridad, CE_CategoriaSumario pcategoria)
Descripción:	Actualiza en la base de datos las propiedades del sumario pasado por parámetro.
Nombre:	BorrarSumarioBD(String pnombreGrupo, String psumario)
Descripción:	Borra de la base de datos el sumario pasado por parámetro
Nombre:	ConfigurarSistemaPorDefectoBD()
Descripción:	Si el usuario no establece una configuración al sistema, se le da una configuración por defecto.
Nombre:	CambiarConfiguracionSistemaBD(CE_ConfiguracionSistema pconfigSistema)
Descripción:	Cambia la configuración del sistema en la base de datos
Nombre:	EditarElementosNoticiaBD(String ptitulo,List<String> ppalabrasClaves)
Descripción:	Guarda en la base de datos las palabras claves de las noticias suministradas por el usuario.

Nombre:	CambiarEstadoNoticiaBD(String ptituloNoticia, boolean pestado)
Descripción:	Cambia las noticias al estado de leídas o no leídas.
Nombre:	CambiarRelevanciaNoticia(String pnoticia)
Descripción:	Cambia la relevancia de la noticia pasada por parámetro.
Nombre:	SumariosDelGrupo(String pgrupoSumario)
Descripción:	Devuelve los sumarios del grupo de sumario pasado por parámetro
Nombre:	DameNoticia(String ptitulo)
Descripción:	Devuelve de la base de datos la noticia que coincide con el título pasado por parámetro
Nombre:	BuscarSumario(String pvalor, int ptipoBusqueda)
Descripción:	Devuelve de la base de datos el sumario que coincida con el título o dirección URL pasada por parámetro en dependencia del tipo de búsqueda que se realice.
Nombre:	RecuperarUsuario(String pusuario)
Descripción:	Devuelve de la base de datos el usuario que coincida con el pasado por parámetro.
Nombre:	RecuperarUsuarioLogueado()
Descripción:	Devuelve el usuario que se encuentre conectado a la base de datos
Nombre:	BuscarGrupoSumarioBD(String pgrupoSumario)
Descripción:	Devuelve de la base de datos el grupo de sumario que coincida con el pasado por parámetro.
Nombre:	CategoriaBD(String psumario)

Descripción:	Devuelve la categoría almacenada en la base de datos, del sumario pasado por parámetro.
Nombre:	EjecutarDepuracion(CE_Sumario psumario)
Descripción:	Saca de la lista de noticias del sumario las que ya están almacenadas en la base de datos.
Nombre:	DevuelveConfiguracionActualizar()
Descripción:	Devuelve la configuración del sistema para realizar cualquier actualización
Nombre:	BuscaNoticia(String ptitulo)
Descripción:	Busca una noticia y devuelve 1 si la encuentra, en caso contrario -1
Nombre:	EliminarNoticiaBD(CE_Noticia pnoticia)
Descripción:	Elimina de la base de datos la noticia pasada por parámetro.
Nombre:	DevolverListaNoticiaBD(String ptituloSumario)
Descripción:	Devuelve todas las noticias del sumario pasado por parámetro.
Nombre:	Noticias()
Descripción:	Devuelve todas las noticias de la base de datos
Nombre:	GruposSumarios()
Descripción:	Devuelve los grupos de sumarios del usuario que está conectado.
Nombre:	SalvaRespaldoBD(String pcaminoSalva)
Descripción:	Realiza una salva de respaldo de la base de datos donde el usuario prefiera.

Nombre:	SalvaPersonal(String pcaminoSalva)
Descripción:	Realiza una salva de la base de datos donde el usuario prefiera, pero solamente de los datos del usuario que se encuentra conectado.

Nombre: CC_EstablecerConexion	
Tipo de clase: Clase Controladora	
Atributo	Tipo
sistema	CE_ConfiguracionSistema
proxy	Proxy
conexion	HttpURLConnection
url	URL
Para cada responsabilidad:	
Nombre:	GetContrasennaUsuario()
Descripción:	Devuelve el usuario y contraseña del usuario almacenadas en la base de datos.
Nombre:	ConectarConProxy(String purl)
Descripción:	Se conecta a Internet con el proxy definido por el usuario
Nombre:	ConectarSinProxy(String purl)
Descripción:	Trata de establecer una conexión con el proxy por defecto definido en la estación del trabajo.
Nombre:	Autenticarse(String purl)

Descripción:	Verifica la autenticación del usuario para establecer un tipo de comunicación.
---------------------	--

Nombre: CC_LectorSax	
Tipo de clase: Clase Controladora	
Atributo	Tipo
noticias	List<CE_Noticia>
sumario	CE_Sumario
autenticarse	CC_EstablecerConexion
factoriaSax	SAXParserFactory
parseador	SAXParser
Para cada responsabilidad:	
Nombre:	GetNoticias()
Descripción:	Devuelve las noticias que han sido parseadas.
Nombre:	Leer(String purl)
Descripción:	Se conecta el sumario pasado por parámetro y parsea el canal de sindicación (un archivo XML).

Nombre: CC_Buscador	
Tipo de clase: Clase Controladora	
Atributo	Tipo
parametro	String
directorioBuscar	String
stopWords	CC_StopWords
Para cada responsabilidad:	
Nombre:	RealizarBusqueda()
Descripción:	Devuelve las noticias de la base de datos que concuerdan con el criterio de búsqueda.

Nombre: CC_Buscador	
Tipo de clase: Clase Controladora	
Atributo	Tipo
directorioIndexacion	String
stopWords	CC_StopWords
Para cada responsabilidad:	
Nombre:	Indexar(List<CE_Noticia> pnoticias)
Descripción:	Crea un índice con las noticias pasadas por parámetros para que sean recuperadas más rápidamente por el buscador.

Nombre: CC_StopWords	
Tipo de clase: Clase Controladora	
Atributo	Tipo
stopWords	String []
Para cada responsabilidad:	
Nombre:	GetStopWords()
Descripción:	Devuelve el arreglo con las stopwords.

Nombre: CC_TemporizadorActualizacion	
Tipo de clase: Clase Controladora	
Atributo	Tipo
temporizadorActualizarSumarios	Timer
controladoraActualizar	CC_Sistema
iconoBandeja	TrayIcon
Para cada responsabilidad:	
Nombre:	
Descripción:	

Nombre: CC_TemporizadorLimpieza	
Tipo de clase: Clase Controladora	
Atributo	Tipo
temporizadorLimpiarSumarios	Timer
controladoraLimpieza	CC_Sistema
tabla	JTable
Para cada responsabilidad:	
Nombre:	
Descripción:	

Nombre: CC_HiloActualizar	
Tipo de clase: Clase Controladora	
Atributo	Tipo
control	CC_Sistema
iconoBandeja	TrayIcon
Para cada responsabilidad:	
Nombre:	run()
Descripción:	Se sobrescribe de la clase Thread, y se encarga de crear un hilo que ejecuta la actualización automática del sistema.

Nombre: CC_HiloIndexar	
Tipo de clase: Clase Controladora	
Atributo	Tipo
controladora	CC_Sistema
Para cada responsabilidad:	
Nombre:	run()
Descripción:	Se sobrescribe de la clase Thread, y se encarga de crear un hilo que ejecuta la indexación automática de las noticias en la base de dato.

Nombre: CC_CrearHiloDescarga	
Tipo de clase: Clase Controladora	
Atributo	Tipo
actualizarSumario	boolean
hiloIndexar	CC_HiloIndexar
actualizarGrupoSumario	boolean
actualizarTodosSumario	boolean
sumario	String
grupoSumario	String
control	CC_Sistema
Para cada responsabilidad:	
Nombre:	run()

Descripción:	Se sobrescribe de la clase Thread, y se encarga de crear un hilo que ejecuta la descarga a los distintos niveles ya sea a nivel de sumario, grupo de sumarios o a nivel de todos los sumarios.
---------------------	--

Nombre: CC_CrearHiloLimpieza	
Tipo de clase: Clase Controladora	
Atributo	Tipo
tabla	JTable
control	CC_Sistema
Para cada responsabilidad:	
Nombre:	run()
Descripción:	Se sobrescribe de la clase Thread, y se encarga de crear un hilo que ejecuta la limpieza de las noticias de la base de datos y actualiza la tabla que contiene las noticias en la ventana principal del sistema.

GLOSARIO DE TÉRMINOS

1. **Acuicultura:** La **acuicultura** se define como la acción y rubro comercial productivo, en la crianza de recursos hidrobiológicos, conocidos también como peces, moluscos, crustáceos y vegetación acuática, en ambientes físicos controlados, con el fin de reemplazar y mejorar las condiciones que estos organismos encuentran en ambientes normales.
2. **Agregadores o lectores:** Son programas que permiten recopilar contenidos actualizados de los sitios favoritos. También reciben el nombre de lector de noticias.
3. **Arquitectura neutral: Arquitectura Neutral.** Al compilar un programa en Java, el código resultante es un tipo de código binario conocido como "**byte code**". Este código es interpretado por diferentes computadoras de igual manera, solamente hay que implementar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional definida.
4. **Blogs:** Un **blog**, o en español también una bitácora, es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.
5. **C:** es un lenguaje de programación creado en 1972 por Kenneth L. Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL.
6. **C++:** es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup.

7. **Ciberespacio:** alude al ámbito en el que se encuentran tanto los ordenadores como los usuarios. Es muy importante destacar el hecho de que no es un espacio entendido en sentido físico, sino que es un espacio virtual.
8. **Cibernauta:** Persona que navega en la red internet.
9. **Cientes Web:** Se dice que un programa es un "cliente" cuando sirve sólo para obtener información sobre un programa "servidor". Cada programa "cliente" está diseñado para trabajar con uno o más programas "servidores" específicos, y cada "servidor" requiere un tipo especial de "cliente". Un navegador es un programa "cliente".
10. **Clic:** se denomina **clic**, **hacer clic**, **clickar**, **clickear** o **pinchar** a la acción de pulsar cualquiera de los botones de un mouse o ratón de computadora. Como resultado de esta operación, el sistema aplica alguna función o proceso al objeto señalado por el cursor o el puntero en el momento de realizarla.
11. **Correo electrónico:** o en inglés **e-mail** (electronic mail), es un servicio de red que permite a los usuarios enviar y recibir mensajes rápidamente (también denominados **mensajes electrónicos** o **cartas electrónicas**) mediante sistemas de comunicación electrónicos.
12. **Encapsulación:** Es el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales. Esto significa que aquello que está en el interior de la clase está oculto, solo las interfaces externas pueden ver otros objetos.
13. **Fidelizar: Fidelización** es un concepto de marketing, se refiere a la «fidelización de los clientes». La fidelización es el fenómeno por el que un público determinado permanece fiel a la compra de un producto concreto de una marca concreta, de una forma continua o periódica.

14. **FTP:** (sigla en inglés de **File Transfer Protocol - Protocolo de Transferencia de Archivos**) en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.
15. **Herramientas:** Funciones que ofrece un programa a través de una barra con íconos, que representan los distintos recursos del software para realizar una tarea determinada.
16. **Herencia:** Es una propiedad que permite que los objetos sean creados a partir de otros ya existentes, obteniendo características (métodos y atributos) similares a los ya existentes. Es la relación entre una clase general y otra clase más específica. Es un mecanismo que permite crear clases derivadas a partir de la clase base.
17. **HTTP:** El protocolo de transferencia de hipertexto (**HTTP**, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). HTTP fue desarrollado por el consorcio W3C y la IETF.
18. **HTML:** Lenguaje utilizado para armar documentos de hipertexto, como las páginas que se visualizan en la World Wide Web.
19. **I+D+i:** Las siglas I+D+i significan Investigación + Desarrollo + Innovación tecnológica.

Se entiende por **investigación** la indagación original planificada que persiga descubrir nuevos conocimientos y una superior comprensión en el ámbito científico y tecnológico.

Se entiende por **desarrollo** la aplicación de los resultados de la investigación o de cualquier otro tipo de conocimiento científico para la fabricación de nuevos

materiales o productos o para el diseño de nuevos procesos o sistemas de producción, así como para la mejora tecnológica sustancial de materiales, productos, procesos o sistemas preexistentes.

Se entiende por **innovación tecnológica** la actividad cuyo resultado sea un avance tecnológico en la obtención de nuevos productos o procesos de producción o mejoras sustanciales de los ya existentes. Se considerarán nuevos aquellos productos o procesos cuyas características o aplicaciones, desde el punto de vista tecnológico, difieran sustancialmente de las existentes con anterioridad.

20. **Ingeniería del software:** es la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.
21. **Internet:** conjunto de ordenadores, o servidores, conectados en una red de redes mundial, que comparten un mismo protocolo de comunicación, y que prestan servicio a los ordenadores que se conectan a esa red.
22. **Interfaz:** Conexión e interacción entre elementos de hardware, programas y usuarios, es decir como la plataforma o medio de comunicación entre usuario o programa.. Las interfaces de usuario son todos los dispositivos de ingreso de datos (teclado, mouse) y los recursos gráficos de los sistemas operativos (cajas de diálogo, menús descolgables, íconos, etc.) que hacen posible la interacción entre la PC y el operador.
23. **JAVA:** es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Inicialmente Java fue creado por un grupo de investigadores de Sun Microsystems como un lenguaje de programación multipropósito y que funcionara no sólo en computadoras, sino en cualquier dispositivo electrónico que necesite de programación.

24. **Lenguaje de Programación:** Código utilizado para la creación de programas. Lo utilizan los programadores (generalmente especializados en un lenguaje en particular) para la creación de diversas aplicaciones.
25. **Metalinguaje:** es un lenguaje usado para hacer referencia a otros lenguajes.
26. **Máquina virtual:** En informática una **máquina virtual** es un software que emula a un ordenador y puede ejecutar programas como si fuese un ordenador real.
27. **Metadato:** son datos que describen otros datos. En general, un grupo de metadatos se refiere a un grupo de datos, llamado recurso.
28. **Motores de Búsqueda:** Un **motor de búsqueda** es un sistema informático que indexa archivos almacenados en servidores web gracias a su «*spider*» (o Web crawler). Un ejemplo son los buscadores de Internet (algunos buscan sólo en la Web pero otros buscan además en noticias, servicios como Gopher, FTP, etc.) cuando se pide información sobre algún tema.
29. **Multithread:** Un lenguaje que soporta múltiples threads (hilos en español) es un lenguaje que puede ejecutar varios procedimientos en paralelo.
30. **Navegador:** Programa diseñado para acceder a los contenidos textuales, gráficos y audiovisuales publicados en la World Wide Web, el servicio más popular de Internet. El navegador o browser interpreta el código fuente con el que se diseñan las páginas web (lenguaje HTML) Además, posee opciones que facilitan la "navegación".
31. **Paradigma:** Un **paradigma** es —desde fines de la década de 1960— un modelo o patrón en cualquier disciplina científica u otro contexto epistemológico.

32. **PC:** Computadora personal dotada de memoria y procesamiento, capaz de ejecutar aplicaciones comerciales, procesamiento de textos, generación de gráficos, programas de entretenimiento, multimedia, etc. En la actualidad, la gran mayoría de las computadoras de hogares y oficinas son PC.
33. **Polimorfismo:** En programación orientada a objetos se denomina **polimorfismo** a la capacidad que tienen los objetos de una clase de responder al mismo mensaje o evento en función de los parámetros utilizados durante su invocación.
34. **Portal web:** es un sitio web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, documentos, aplicaciones, compra electrónica, etc. Principalmente están dirigidos a resolver necesidades específicas de un grupo de personas o de acceso a la información y servicios de una institución pública o privada.
35. **RDF: El Marco de Descripción de Recursos** (del inglés Resource Description Framework, RDF) es un framework para metadatos en la World Wide Web (WWW), desarrollado por el World Wide Web Consortium (W3C).
36. **Requerimientos:** es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Se usa en un sentido formal en la ingeniería de sistemas o la ingeniería de software. Otro (Comprender todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer).
37. **Relación jerárquica:** Relación entre conceptos en la cual uno de ellos se encuentra conceptualmente superordenado respecto de otro u otros.

38. **RunTime:** Se denomina **Tiempo de ejecución** (Runtime en inglés) al intervalo de tiempo en el que un programa de computadora se ejecuta en un sistema operativo.
39. **Sindicación:** Sistema que recoge los titulares de las noticias o historias publicadas en los weblog, casi todos los weblogs incorporan enlaces de texto, o bien, los famosos botones naranjas con las abreviaturas **XML, RSS, RDF, o ATOM** para que el lector copie la dirección web que traen conocida como feed y la inserten en un **lector/agregador de noticias o feeds** que avisará de las actualizaciones de los weblogs favoritos evitando acudir a la página web de forma que no se tenga que consultar las novedades que hay en cada página individual, al centralizar las novedades de las páginas seleccionadas en un único lugar de consulta.
40. **Sitio web:** Colección de páginas web dotada de una dirección web única.
41. **Sistema:** Grupo de componentes relacionados que interactúan para realizar una tarea determinada. En el campo computacional, un sistema informático se compone de una CPU, el sistema operativo y los dispositivos periféricos.
42. **Software:** Conjunto de programas, documentos, procesamientos y rutinas asociadas con la operación de un sistema de computadoras, es decir, la parte intangible o lógica de una computadora.
43. **Software libre:** **Software libre** (en inglés free software) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

44. **TCP/IP:** La **familia de protocolos de Internet** es un conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se le denomina conjunto de protocolos **TCP/IP**, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia.
45. **Tesoro:** La palabra **tesoro**, derivado del neo latín que significa tesoro, se refiere a listado de palabras o términos empleados para representar conceptos.
46. **UML: Lenguaje Unificado de Modelado (UML,** por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.
47. **Usuario:** Persona que utiliza la computadora para sistematizar tareas, guardar información, conectarse con otras personas para compartir información, etc.
48. **Web semántica:** La **Web semántica** (del inglés semantic web) es la "Web de los datos". Se basa en la idea de añadir metadatos semánticos y ontológicos a la World Wide Web. Esas informaciones adicionales —que describen el contenido, el significado y la relación de los datos— se deben proporcionar de manera formal, para que así sea posible evaluarlas automáticamente por máquinas de procesamiento.
49. **XML:** Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un

lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.