Universidad de las Ciencias Informáticas

Facultad 3

Título: Observatorio de Información

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor: Alain Hernández López

Tutor: Manuel Vázquez Acosta

Junio de 2007

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Alain Hernández López

_____

Manuel Vázquez Acosta

_____

DATOS DE CONTACTO

Datos del Tutor:

Nombre: Lic. Manuel Vázquez Acosta

Área de Trabajo: Universidad de las Ciencias Informáticas Facultad 3.

Cargo: Visedecano de Producción Facultad 3

Correo Electrónico: manuelva@uci.cu

Datos del Autor:

Nombre: Alain Hernández López

Área de Trabajo: Universidad de las Ciencias Informáticas Facultad 3.

Cargo: Estudiante

Correo Electrónico: ahernandezlop@estudiantes.uci.cu

alainhl@gmail.com

AGRADECIMIENTOS

A mis padres, por haberme guiado siempre.

A Darel Camps mi amigo, que me ha tendido la mano en todo momento cuando lo he necesitado.

A Yarisleidis Fernández y Dailin Benavides por hacer que este año fuera diferente.

A mi tutor Manuel Vásquez por haberme guiado en todo el proceso de desarrollo de la tesis.

A los muchachos de tercer año que trabajan en el laboratorio 12, por su ayuda desinteresada, en especial a Daily Miranda, Juniel Tamayo, Ariel Morales, Fernando Rodríguez, Alberto La Rosa (rosita), Rainer Rodríguez.

DEDICATORIA

A mis padres

RESUMEN

Este trabajo se realiza por la necesidad que tienen los integrantes de los polos científicos de la Universidad de Ciencias Informáticas (UCI), de contar con un producto Observatorio de la Información, que les permita monitorear los principales nichos de información de utilidad para sus investigaciones. La tecnología usada para el desarrollo del producto fue Zope/Plone, de ahí que la aplicación es un producto para Plone, el cual descarga de forma proactiva los documentos que existen en las fuentes de información, las que son previamente definidas por los usuarios y los hace persistente para que posteriormente, se muestren a los usuarios finales. Se diseñaron dos módulos, que en su conjunto conforman el observatorio de información:

El módulo principal del sistema, Knowledge Collector, implementa el proceso de consulta y tiene la responsabilidad de conectarse a las fuentes de información para descargar sus contenidos. La herramienta Portal Scheduler Tool es el módulo encargado de todo el mecanismo de concurrencia de hilos, el cual permite que el sistema pueda de forma proactiva, ejecutar los procesos de consultas. Se utilizó la herramienta ArgoUml para modelar el sistema, en la implementación se usó el Eclipse3.2 y el lenguaje de programación seleccionado fue Python.

PALABRAS CLAVE

Observatorio de Información, Monitoreo, Plone, Zope, Python, Recolectores de Noticias, Agregadores de Noticias, Archetype, Content Type, Tool, CMF

TABLA DE CONTENIDOS

INTRODUCCIÓN

A raíz del surgimiento de Internet, muchos diarios de noticias, revistas, y empresas de diferentes áreas, se insertaron de forma inmediata en ese proyecto con el objetivo de tener un espacio más donde mostrar su información y hacerse valer en el mercado. Al principio hubo algunos que se cohibieron y temieron lanzarse a ese propósito, pero la realidad es que una empresa que se respete, actualmente, no se concibe sin un portal en la red que pueda mostrar al mundo todos los contenidos de sus publicaciones.

A medida que Internet fue madurando, el número de empresas en la red fue aumentando (ZAKON 2006), y para que una persona pudiera mantenerse al día de los últimos acontecimientos, tenía que consultar demasiadas fuentes de información y ello, como es de esperar, consumía mucho tiempo. A este problema más tarde se le dio solución con los agregadores de noticias y los recolectores de noticias.

Algunos ejemplos de agregadores de noticias son: Bloglines (BLOGLINES 2007), Rojo (ROJO 2007), FeedShow (FEEDSHOW 2007), Google Reader (GOOGLE 2007b), WinRSS (WINRSS 2007), Raggle (RAGGLE 2007), Snownews (FEILEX 2005). Estos son programas, en los que el usuario puede especificar el RSS (Really Simple Syndication) de la fuente de información que desea monitorear.

El software, a través de los mecanismos que tiene implementados, se conecta a estas fuentes de información y descarga las últimas actualizaciones del sitio Web. Lo interesante de los agregadores es que con ellos el internauta no tiene que consultar tantas fuentes de información, de manera que puede actualizarse rápidamente en un tema específico.

Los recolectores de noticias también se han vuelto famosos, existen muchos de ellos como son, Yahoo News (YAHOO 2007), MSN Newsbot (MICROSOFT 2007), Techmeme (TECHMEME 2007), Findory (FINDORY 2007). El más utilizado por los internautas es Google News (GOOGLE, NEWS 2007c). Actualmente existe un total de 35 sitios de Google News en diversos países y lenguas, que en su conjunto reciben una media de 38 millones de usuarios (GIBSON 2006).

A diferencia de los agregadores de noticias, los recolectores de información son aplicaciones Web que permiten conocer las noticias más importantes del día en un breve espacio de tiempo. Estos tienen

diferentes mecanismos para mostrar las noticias y para personalizar a los usuarios, de ahí que unos sean más eficientes que otros.

Estos sistemas son en su conjunto observatorios de información, porque permiten hacer un estudio y análisis de indicadores en cualquier campo del conocimiento, y pueden abarcar los datos presentados por organismos nacionales e internacionales. Además les permiten a las instituciones que los utilicen, mantenerse informadas sobre todos los adelantos tecnológicos y científicos que acontecen en el mundo sobre un tema específico, ya sea informática, medicina, telecomunicaciones, turismo, humanidades, biología, ecología, noticias, ocio, etc.

Los observatorios de información son utilizados para difundir:

- Estudios e informes sobre la implantación de las tecnologías de la información y las comunicaciones.
- Indicadores sobre la Sociedad de la Información.
- Información sobre actualidad nacional e internacional.
- Seguimiento y evaluación de políticas públicas en cuanto a Sociedad de la Información.

Un ejemplo de observatorio de información es Google News, el cual no monitorea todo Internet (algo inabarcable) sino 7000 fuentes de información (4.500 de habla inglesa), después reúne y da prioridad a las noticias en grupos de artículos (GONZÁLEZ 2004).

Un sistema que integre todo lo planteado antes, es decir, que sea capaz de recolectar conocimiento de las diferentes fuentes de información, eficientemente, para sus usuarios finales, sería ideal para el país. Cuba no cuenta actualmente con muchos sistemas que le permitan realizar estas operaciones, y que sean capaces de brindar las prestaciones que el país necesita para el desarrollo de sus múltiples investigaciones, en disímiles campos del conocimiento.

Con la existencia de observatorios de información en cada una de las instituciones cubanas, por ejemplo: de educación, de turismo, de medicina, de informática, etc., estas se mantendrían informadas de los

últimos acontecimientos sobre un tema específico que les interese. Además, podrían ahorrar mucho tiempo a la hora de llevar a cabo sus investigaciones, porque a través de los observatorios tendrían la información de última hora a su disposición.

Actualmente las universidades cubanas no cuentan con observatorios que le permitan monitorear los diferentes campos de conocimientos sobre los que está interesada. En Internet se encuentra gran cantidad de información de todo tipo, pero esta se encuentra a la vez, muy dispersa, por lo que el proceso de buscarla, clasificarla y agruparla es muy engorroso.

Este trabajo de diploma surge debido a la necesidad que tiene el país de dar solución a las situaciones antes expuestas, de ahí que se plantee la siguiente **situación problemática:** ¿Cómo crear un sistema observatorio de información capaz de monitorear distintos puntos de la red, y obtener información actualizada?

La Universidad de las Ciencias Informáticas (UCI) ha mostrado interés en desarrollar este sistema porque lo necesita como complemento de otros proyectos productivos, como son Operación Verdad y otros proyectos de la prensa que se están desarrollando. El proyecto en sí abarca el desarrollo de tres módulos fundamentales, el módulo observatorio de información desarrollado en el marco de este trabajo de diploma y los módulos: implementación de algoritmos de agrupamiento y biblioteca de clases para la creación automática de resúmenes extractos, los cuales son trabajos de diplomas no publicados, que se integrarán en un futuro con el observatorio de información que se desarrollará en los próximos capítulos.

El **objeto de estudio** de este trabajo son los observatorios de información que existen en el mundo. El **campo de acción** es la creación de herramientas automatizadas para crear observatorios de información nacionales.

Se plantea como **hipótesis** que: Si se implementa un prototipo funcional de un producto para Plone que permita la creación de las bases de un observatorio de sitios de información se podrá monitorear distintos puntos de la red y obtener información actualizada.

El **objetivo general** de este trabajo de diploma es: Implementación de un prototipo funcional de un producto para Plone que permita la creación de las bases de un observatorio de sitios de información.

Como **objetivos específicos** que se derivan del objetivo general tenemos los siguientes:

1. Construir un marco teórico sobre los observatorios de información.
2. Diseñar e implementar un producto para Plone, que sirva como base para la creación de observatorios de información.
3. Evaluación de los resultados.

Este trabajo de diploma estará organizado de la siguiente manera:

En el Capítulo1: Fundamentación teórica. Se describen y explican conceptos relacionados con los observatorios de información, recolectores y agregadores de noticias, que será la principal guía en el desarrollo de este proyecto. Además, se analizan los principales sistemas que actualmente dominan el mercado, se define la tecnología sobre la cual se va a desarrollar el sistema, y la metodología que se utilizará para llevar a cabo todo el proceso de desarrollo.

En el Capítulo 2: Modelo, propuesta concreta de la solución. Se detectan los principales requisitos necesarios para desarrollar un observatorio de información, se describen los casos de usos detectados, se muestran los diagramas de subsistemas y clases que se diseñaron, así como los principales métodos desarrollados en la implementación. Finalmente se hace un análisis de los resultados obtenidos a través del caso de prueba desarrollado.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se brindan diferentes aspectos y conceptos relacionados con los observatorios información. Además se hace un análisis de la importancia que brindan cada uno de estos sistemas y la falta que hace la implementación de un observatorio de información nacional.

También se analizarán varios tipos de observatorios de información, donde se verá la tipología, los mecanismos que utilizan, y los servicios que brindan cada uno, lo que permitirá tomar la decisión sobre que tipo de observatorio se debe desarrollar, o introducir características híbridas.

Además se hace la propuesta de la tecnología en la cual se va a desarrollar el producto. Se describe la misma, y se plantean los argumentos que se tuvieron en cuenta a la hora de seleccionar este tipo de tecnología. Así como la descripción y ventajas de desarrollar con esta tecnología. Y finalmente se describe la metodología y las herramientas en las cuales se va a desarrollar el producto.

1.2 ¿Qué es un observatorio de información?

Es un sistema organizado de observación y análisis del entorno, seguido de una transmisión precisa de los conocimientos útiles a los órganos encargados de tomar decisiones. Se basa en el empleo de técnicas y procedimientos de vigilancia tecnológica y esta dirigido a un colectivo bien definido de usuarios. (GERMÁN 2003) (WIKIPEDIA 2007c) El escenario en el que actualmente hay que realizar esa vigilancia es muy complejo y se caracteriza por:

- Saturación de información como resultado de una sobreproducción científico tecnológica.
- Dificultad para estar en contacto con todos aquellos que generan información, pues la información circula a través de los denominados "cauces invisibles", grupos de expertos, académicos o profesionales, de diferentes comunidades o países que se comunican entre sí mediante relaciones virtuales (correo electrónico, Internet) o se encuentran en documentos que no se distribuyen a través de los canales convencionales (tesis, actas de congresos, etc...).

- Fuerte incremento de los costes para conseguir el liderazgo tecnológico.

Existen muchos tipos de observatorios de información, por ejemplo, los observatorios espaciales, astronómicos, meteorológicos, climatológicos, geológicos, vulcanológicos, etc. Este trabajo de diploma no va a concentrase en un tipo específico de observatorio, sino que se van a dar las herramientas para la creación de un observatorio tecnológico, el cual a través de la red podrá monitorear toda la información que se genere en un campo de conocimiento específico, ya sea el campo de la medicina, informática, educación, turismo, etc., de manera que la universidad pueda aplicar esta herramienta a los diferentes polos científicos a los cuales esté orientado. Una vez aclarado el concepto de observatorio de información se hace prudente explicar cual es la importancia que representa la creación de un observatorio de información.

1.3 Importancia de los observatorios de información

La importancia de los observatorios puede medirse en relación a la información que es observada, aunque hay aspectos que son independientes de esta:

- Los observatorios permiten obtener un sistema de información continuo y sistematizado sobre la situación actual y tendencias de la ciencia y la tecnología en sectores relevantes del entorno en el cual la empresa se desarrolla (VIVERO 2005).
- También filtran, analizan y ponen dicha información en un formato asequible a los agentes interesados y la difunden entre ellos continuamente.
- Además las empresas ahorran mucho tiempo cuando realizan sus investigaciones debido a que a través del observatorio, pueden tener el conocimiento de forma inmediata, y no tienes que ir a Internet a buscar entre tanta información que a menudo, se encuentra dispersa.

A continuación se realizará un análisis de los principales tipos de observatorios que existen y de los criterios que se utilizarán para clasificarlos.

1.4 Tipos de observatorios

Existen tantos observatorios como personas hay en el mundo, se dice esto porque un observatorio no tiene porqué seguir unas normas preestablecidas. Cada persona o institución puede construirse un observatorio adaptado a sus necesidades, por ejemplo: especies a observar, condiciones del terreno, ecología, medio ambiente, tecnología, espacio exterior, meteorología, etc.

A la hora de proponer cualquier clasificación, el primer paso consiste en establecer una serie de criterios que nos permitan distinguir entre los observatorios. En este caso, los criterios de clasificación parten del conjunto de variaciones que se registran en los observatorios. Se ha atendido solo a los criterios que se consideran más significativos. De esta forma, los criterios que permitan diferenciar a los observatorios son (SUSANA 2005):

- La temática que investigan.
- Su origen y composición social.
- La cobertura de los medios que analizan.
- El tipo de medios que observan.
- El método de obtención de la información.

La tabla 1 resume los criterios que se mencionaron anteriormente.

Criterio	Tipos de observatorios
En función de la temática que investigan	• Observatorios generales. • Observatorios especializados.

En función de su composición social	• Vinculados a la gente de la profesión:
	  o Vinculados a facultades de comunicación.
	  o Vinculados a profesionales en activo.
	  o Mixtos.
	• Vinculados a gente de fuera de la profesión.

	• Mixtos.
En función de la cobertura de medios que analizan	• Observatorios que analizan medios locales. • Observatorios que analizan medios regionales. • Observatorios que analizan medios nacionales. • Observatorios que analizan medios internacionales. • Observatorios con cobertura geográfica múltiple.
En función del tipo de medios que observan	• Observatorios que analizan prensa digital. • Observatorios que analizan blogs. • Observatorios que analizan Internet. • Observatorios que analizan Base de Datos. • Observatorios con cobertura de tipos de medios múltiples.
En función del método de obtención de la información	• Activo. • Pasivo.

Tabla 1 Tipología de observatorios

1.4.1 En función de la temática que investigan

En función de la temática que investigan, cabe hacer una primera distinción entre los observatorios generales—dedicados al monitoreo de una gran cantidad de temas, siempre que estos estén relacionados con las profesiones de comunicación—y los observatorios especializados que se centran en el rastreo de uno o varios temas acotados así desde el comienzo. En este último caso, las posibles focalizaciones temáticas de los observatorios son muy variadas y, a nivel mundial, incluyen—por ejemplo—la investigación en temas de elecciones, convivencia entre el poder político y los intereses de las corporaciones mediáticas, crimen y violencia, derechos humanos, salud, intimidad, comunicación local, formas de trabajo de los periodistas y condiciones de producción de los mensajes, cibersociedad y nuevas tecnologías de la información, políticas de comunicación, propiedades de los medios y aspectos empresariales de los medios de comunicación, etc.

1.4.2 En función a su composición social

Atendiendo al criterio de su composición social, se puede distinguir entre los observatorios integrados por gente de la profesión, por gente de fuera de la profesión y, en tercer lugar, los observatorios mixtos que están compuestos por profesionales de la comunicación pero también por público general.

Además, dentro del primer grupo—el de los observatorios integrados por gente de la profesión—cabe diferenciar a su vez entre los observatorios vinculados, por un lado, a facultades de comunicación social y , por otro, los observatorios en los que participan fundamentalmente profesionales en activo.

1.4.3 En función de la cobertura de los medios que analizan

Junto a los criterios de temática y composición social, la cobertura de los medios que analizan constituye otro criterio que permite clasificar a los observatorios de medios. En este caso, los observatorios se diferencian entre si según registren la actividad de los medios locales, regionales, nacionales o internacionales.

1.4.4 En función del tipo de medios que observen

Según el tipo de medios que analicen, los observatorios se pueden clasificar en diferentes grupos como son: observatorios de prensa digital, observatorios de blogs, observatorios de noticias y observatorios de base de datos.

1.4.5 En función al método de obtención de la información

Finalmente, según el método de obtención de la información pueden ser activos o pasivos. Activos cuando ellos mismos a través de diferentes mecanismos obtienen la información que publican y pasivos cuando no obtienen la información por ellos mismos, es decir, obtienen la información a través de la intervención del hombre.

Una vez definidos los principales criterios a tener en cuenta a la hora de clasificar un observatorio de información, se analizarán a continuación los principales sistemas automatizados existentes.

1.5 Sistemas automatizados existentes

En Internet existen muchos sistemas que permiten la observación de diferentes medios de información, los cuales se agrupan en dos grupos principales, los agregadores de noticias y los recolectores de noticias.

1.5.1 Agregadores de noticias

“Un agregador o agregador de noticias (también llamado "lector de *feeds*") es un tipo de software para sindicar contenidos Web en forma de *feed* (RSS, Atom y otros formatos derivados de XML/RDF). El agregador recoge ese *feed* con las noticias o historias publicadas en los distintos *weblogs* o bitácoras que se elijan, y muestra las novedades o ediciones que se han producido en ese *feed*; es decir, avisan qué noticias o historias son nuevas desde la última lectura.” (WIKIPEDIA 2007a)

1.5.2 Recolectores de noticias

Los recolectores de noticias son aplicaciones Web que filtran las noticias con alta eficacia, estos implementan diferentes mecanismos para presentar sus noticias, por ejemplo: existen sitios que rastrean las estructuras vinculadas de los mejores blogs, para presentar una vista en primera plana de las conversaciones que acontecen en Internet y motores personalizados de recomendación de noticias que estudian los hábitos de lectura que usted tiene, a fin de llevarle las noticias con un enfoque más directo.

Para lograr un mayor entendimiento de lo que se quiere mostrar, se analizarán varios de estos sistemas, donde se especificarán los siguientes datos:

- Tipo de observatorio.
- Mecanismo que utiliza para mostrar la información.
- Servicios que brinda.

Los ejemplos que se muestran a continuación basan su mecanismo de búsqueda en el filtrado de fuentes de información.

1.5.3 Techmeme (TECHMEME 2007)

Techmeme es una aplicación Web, la cual utiliza su colección de fuentes de información compuesta por los sitios más importantes de noticias y blogs, para identificar los artículos que más vínculos tienes asociados. Y finalmente mostrarle estas al usuario final junto con las discusiones de esas historias, así como los artículos que se relacionan con ellas.

El propietario del sitio, usa los mismos algoritmos para crear una lista creciente de filtros de noticias dedicados a la política (MEMEORANDUM 2007), béisbol (BALLBUG 2007) y chismes de la farándula (WESMIRCH 2007).

La Tabla 2 muestra la clasificación que se le dio al observatorio después de ser analizado.

Criterio	Tipos de observatorios
En función de la temática que investigan	Observatorios generales.
En función de su composición social	Mixtos
En función de la cobertura de medios que analizan	Observatorios con cobertura geográfica múltiple
En función del tipo de medios que observan	Observatorio que analiza prensa digital y blogs.
En función del método de obtención de la información	Activo

Tabla 2 Tipología de Techmeme

1.5.4 Tailrank (TAILRANK 2007)

Tailrank encuentra los artículos entre millares de blogs, funciona rastreando conversaciones entre blogs, vínculos y citaciones altamente ligados y discutidos. Toma en consideración el comportamiento de los vínculos, el texto de los artículos, los vínculos en común con otros usuarios, la relevancia del texto, el ranking del weblog, y otros factores. Una vez que ha rastreado los artículos los almacena en sus índices. Si Tailrank descubre algo importante lo promueve para la página principal (TAILRANK 2007).

Permite visualizar artículos tanto generales como específicos. Entre los temas específicos se encuentran tecnología, política, entretenimiento, videos y personalizado, en el cual se pueden seleccionar los weblogs sobre los cuales Tailrank buscará sus artículos importantes.

La Tabla 3 muestra la clasificación que se le dio al observatorio después de ser analizado.

Criterio	Tipos de observatorios
En función de la temática que investigan	Observatorios generales.
En función de su composición social	Mixtos
En función de la cobertura de medios que analizan	Observatorios con cobertura geográfica múltiple
En función del tipo de medios que observan	Observatorio que analiza bolgs.
En función del método de obtención de la información	Activo

Tabla 3 Tipología de Tailrank

Ahora se analizarán unos ejemplos que basan sus mecanismos de búsqueda de información en la observación de todo lo que el lector lee.

1.5.5 Google Noticias (GOOGLE, NEWS 2007c)

Google Noticias es un sitio de noticias generado automáticamente que reúne titulares de más de 7000 fuentes de noticias (4.500 de habla inglesa) de todo el mundo, agrupa artículos similares y los muestra en función de los intereses personalizados de cada lector (GOOGLE 2007a).

Este sitio se basa en observar los hábitos de lectura de noticias, y búsqueda en la Web que tienen los usuarios, para ajustar los artículos que muestra en su página de noticias Google Noticias la que se genera algorítmicamente. El sitio destaca las últimas noticias más importantes sin la intervención de editores humanos. Cada noticia ofrece vínculos a varios artículos, de modo que en primer lugar puede elegir el tema que le interese y, a continuación, seleccionar la publicación (GOOGLE 2007a).

La Tabla 4 muestra la clasificación que se le dio al observatorio después de ser analizado.

Criterio	Tipos de observatorios
En función de la temática que investigan	Observatorios generales.
En función de su composición social	Mixtos
En función de la cobertura de medios que analizan	Observatorios con cobertura geográfica múltiple
En función del tipo de medios que observan	Observatorio que analiza prensa digital.
En función del método de obtención de la información	Activo

Tabla 4 Tipología de Google News

1.5.6 Digg (DIGG 2007a)

Digg es un sitio que se enfoca nada más al tratamiento de noticias relacionadas con tecnología. El mecanismo que implementa este sitio es muy simple y a la vez inteligente, los usuarios envían vínculos y otros comentan y votan a favor o en contra. En la medida que un artículo vaya ganando votos, va subiendo su ranking, de manera que el que más ranking tenga es el que subirá a primera plana. (DIGG 2007b).

Entre los servicios que brindan se encuentran el Digg (para promover el artículo), el Bury (para descartar la historia o clasificarla como Spam) y los comentarios sobre historias.

La Tabla 5 muestra la clasificación que se le dio al observatorio después de ser analizado.

Criterio	Tipos de observatorios
En función de la temática que investigan	Observatorios especializados
En función de su composición social	Mixtos
En función de la cobertura de medios que analizan	Observatorios con cobertura geográfica múltiple
En función del tipo de medios que observan	Observatorio que analiza prensa digital.
En función del método de obtención de la información	Pasivo

Tabla 5 Tipología de Digg

Una vez concluido con los recolectores de noticias se analizará dos casos de agregadores de noticias. Se comenzará con:

1.5.7 Bloglines (BLOGLINES 2007)

Bloglines es un servicio en línea para buscar, suscribir, crear y compartir *feeds*. El usuario lo único que tiene que hacer es subscribirse e inmediatamente puede tener acceso a su cuenta, desde cualquier computadora o dispositivo móvil. Se puede usar un número ilimitado de direcciones de correo electrónico especializadas, para inscribirse en

La Tabla 6 muestra la clasificación que se le dio al observatorio después de ser analizado.

Criterio	Tipos de observatorios
En función de la temática que investigan	Observatorios generales.
En función de su composición social	Mixtos
En función de la cobertura de medios que analizan	Observatorios con cobertura geográfica múltiple
En función del tipo de medios que observan	Observatorios con cobertura de tipos de medios múltiples.
En función del método de obtención de la información	Activo

Tabla 6 Tipología de Bloglines

1.5.8 Google Reader (GOOGLE 2007b)

Es una herramienta para gestionar y visualizar todos los *feeds* de sitios web que sean de interés para el usuario. Su funcionamiento es muy simple, el usuario adiciona los *feeds* y el sistema a través de sus algoritmos de búsqueda, recolecta y muestra las noticias de última hora. Google Reader es el equivalente

de Gmail entre los lectores de noticias. Permite mostrar las estadísticas de los *feeds*, escuchar *podcast*, entre otros servicios.

La Tabla 7 muestra la clasificación que se le dio al observatorio después de ser analizado.

Criterio	Tipos de observatorios
En función de la temática que investigan	Observatorios generales.
En función de su composición social	Mixtos
En función de la cobertura de medios que analizan	Observatorios con cobertura geográfica múltiple
En función del tipo de medios que observan	Observatorios con cobertura de tipos de medios múltiples.
En función del método de obtención de la información	Activo

Tabla 7 Tipología de Google Reader

Una vez analizados los principales sistemas que existen en Internet, se analizará un último ejemplo de un observatorio información cubano llamado Predictor el cual es desarrollado y mantenido por la empresa Consultoría del Ministerio de la Informática y las Comunicaciones (Delfos).

1.5.9 Predictor

La Consultoría del Ministerio de la Informática y las Comunicaciones, en su nombre abreviado DELFOS brinda servicios integrales de consultoría, representados en gran medida por el ejercicio de la Vigilancia Estratégica, Red de Información como vía de integración de las Unidades de Información y la Gestión Documental.

El sistema Predictor es un observatorio de información desarrollado por esta empresa el cual tiene como metas facilitar la descarga, procesamiento, análisis de grandes volúmenes de información para contribuir al conocimiento de las implicaciones de los cambios tecnológicos para el MIC (Ministerio de la Informáticas y las Comunicaciones) y disminuir el tiempo de descargas, procesamiento, análisis de la

información en temas relacionados con las nuevas tecnologías, en instituciones científicas, en el sector empresarial, disminuyendo los niveles de respuesta en apoyo a las decisiones. El sistema se conecta a Internet y descarga información publicada en bases de datos en Internet y utiliza un analizador sintáctico con un fin específico de obtener una serie de parámetros.

La Tabla 8 muestra la clasificación que se le dio al observatorio después de ser analizado.

Criterio	Tipos de observatorios
En función de la temática que investigan	Observatorios especializados
En función de su composición social	Vinculados a la gente de la profesión
En función de la cobertura de medios que analizan	Observatorios con cobertura geográfica múltiple
En función del tipo de medios que observan	Observatorios que analiza base de datos
En función del método de obtención de la información	Pasivo

Tabla 8 Tipología de Predictor

Después de analizar este grupo de observatorios de información se puede concluir que existen muchos mecanismos para implementar, los cuales deben ser seleccionados con mucha cautela, para no fracasar a la hora de construir un producto. Según los criterios analizados se puede observar que, prácticamente sólo existen dos criterios que varían, en función del tipo de medios que observan y en función del método de obtención de la información. Esta variedad existe de acuerdo a los objetivos e intereses de cada una de las empresas.

En el caso particular de la UCI se necesita un observatorio que de forma activa y automática realice todo el proceso, de manera que no exista la intervención del hombre. Por lo que según los criterios analizados deberá cumplir con las especificaciones siguientes.

Criterio	Tipos de observatorios
En función de la temática que investigan	Observatorios generales.
En función de su composición social	Mixtos
En función de la cobertura de medios que analizan	Observatorios con cobertura geográfica múltiple
En función del tipo de medios que observan	Observatorios con cobertura de tipos de medios múltiples.
En función del método de obtención de la información	Activo

Tabla 9 Tipología del Observatorio de Información UCI

En este sentido, el observatorio que se implementará en los próximos capítulos será una combinación de los observatorios Techmeme, Google News y Predictor, es decir, el sistema buscará sobre fuentes de información preestablecidas y recolectará todos los artículos que se publiquen en estas fuentes.

Ya definido el tipo de observatorio que se desarrollará, se hace necesario analizar el conjunto de factores que se tuvieron en cuenta, cuando se decidió llevar a cabo este proyecto.

1.6 Necesidad de los observatorios de información en la UCI

La UCI no cuenta actualmente con observatorios de información, los cuales serían de mucha utilidad porque permitirían monitorear las nuevas tecnologías que salen al mercado, las publicaciones de revistas importantes, grupos de noticias, información de todo tipo que circula por la red y mantenerse al tanto con información actualizada.

Además existen problemas cuando las personas comienzas a investigar sobre un tema nuevo al que nunca se han enfrentado, muchas veces resulta un problema el inicio de la investigación debido a que las personas no saben donde buscar la información, ni tampoco saben cual es la información de última hora. Con la existencia de observatorios de información, cada facultad de la UCI podría usarlos para llevar a cabo sus investigaciones. De manera que podrían mantener monitoreados todos los temas que son de

interés paras sus polos productivos. Así cuando comience una investigación, ya no se partiría desde cero, se constaría con un repositorio donde estaría la información clasificada, que representaría la base para cada investigación que se desarrolle en la facultad. En caso de que sea un tema nuevo que no se haya monitoreado nunca, se empieza a monitorear a partir de ese momento, de manera que constituya la base para futuras investigaciones.

Por lo antes expuesto, con la existencia de observatorios de información que notifiquen la aparición de nuevas publicaciones, noticias, invenciones y la aplicación de nuevas tecnologías, se evita que las personas tengan que perseguir la información a través de Internet. Además se establece un mecanismo de alerta en un área específica del conocimiento que será muy útil para la UCI, debido a que con estos se eliminarían gran parte de los problemas que tiene esta institución a la hora de llevar a cabos sus investigaciones. De ahí que se lanza la siguiente propuesta a desarrollar.

1.7 Propuesta a desarrollar

El sistema tendrá una arquitectura cliente/servidor basada en componentes, sobre plataforma Web, debido las ventajas que esta plataforma brinda. Se pudiera pensar en hacer una aplicación de escritorio, pero entonces, los usuarios tendrían que exportar sus configuraciones cuando desearan cambiarse de PC, es decir, no podrían sentarse en otra PC y abrir su observatorio porque sencillamente no lo tendrían instalado, o si lo tuvieran instalado no tendrían la misma configuración. Todo esto se evita desarrollando una aplicación Web, porque el usuario ya no tiene que preocuparse de tener instalado el observatorio en su PC, él solo se conecta al servidor Web e inmediatamente tiene su observatorio actualizado con su configuración.

Para el desarrollo de esta aplicación Web hay que tener en cuenta varios factores, entre los que está la tecnología sobre la cual se va a desarrollar.

Debido a que los principales clientes de esta aplicación desarrollan sus productos sobre la tecnología Zope/Plone y a que la facultad 3 desde hace algún tiempo se ha interesado por desarrollar sobre esta tecnología, por las ventajas que esta brinda. Es o que se decidió optar por esta plataforma para el desarrollo del observatorio de información. A continuación se hace un análisis de los principales aspectos y conceptos a tratar en esta nueva tecnología.

1.7.1 ¿Qué es Zope?

Zope es un servidor de aplicaciones totalmente orientado a objetos escrito en Python. Se publica bajo los términos de la licencia Zope Public License (ZOPE), una licencia de software libre. Zope ofrece una infraestructura general sobre la que se pueden construir aplicaciones Web, Intranets y CMS. De esta manera, muchos conceptos y funcionalidades pueden ser reutilizados. Así, por ejemplo, la herramienta de generación de portales se basa en módulos de gestión de usuarios, de seguridad o de sesiones ofrecidos por la arquitectura Zope. (SITE, ZOPE PROJECT WEB 2007b).

1.7.2 ¿Qué es Plone?

Plone es un sistema de gestión de contenidos o CMS por sus siglas en inglés (Content Management System) construido sobre la sólida base de Zope. Plone permite la creación, personalización y gestión de un sitio Web de manera rápida y fácil. Todas las acciones que se han de realizar para la gestión de Plone se pueden realizar a través de su interfaz Web, una vez instalados Zope y Plone, lo que facilita el trabajo colaborativo y distribuido. Plone es un proyecto desarrollado por una amplia comunidad y su licencia es GNU (LICENSE) *(SITE, PLONE PROJECT WEB 2007a).*

1.7.3 Ventajas de Zope/Plone

Un sitio Web de Zope está compuesto de objetos en lugar de archivos, como es usual con la mayoría de los otros sistemas de servidores Web. Zope posee una base de datos orientada a objetos, llamada ZODB o Zope Object Database. Esta base de datos almacena objetos ordenados en un sistema similar a un sistema de ficheros, pero cada objeto tiene propiedades, métodos o puede contener a su vez otros objetos.

Una de las principales ventajas de Zope/Plone es que posibilitan procesos automáticos de gestión de información, lo que posibilita la creación de observatorios proactivos que de forma automática sean capaces de obtener el conocimiento que buscan en la red sin la intervención de los humanos. Además, estos procesos no dependen de otras tecnologías. Sólo dependen de Python para lograr realizar procesamiento de la información, sin intervención del usuario como en los lenguajes clásicos para la Web, en los que el servidor sólo actúa antes las peticiones del usuario.

Después de elegida la tecnología sobre la cual se va a desarrollar la aplicación Web y definidos sus principales conceptos y ventajas, es hora de seleccionar la metodología de desarrollo que se utilizará, así como las principales herramientas con las que se trabajará en todo el proceso.

1.7.4 Fundamentación de la metodología a utilizar.

Una metodología para el desarrollo de un proceso de software es un conjunto de métodos y técnicas, procedimientos, reglas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas informáticos. Por ello hay que tener mucho cuidado a la hora de seleccionar la metodología que llevará a cabo todo el desarrollo, debido a que un error en esta selección puede traer consigo clientes insatisfechos y desarrolladores más aun insatisfechos.

Para la realización de este sistema se determinó utilizar como metodología el Proceso Unificado de Rational (Rational Unified Process, RUP) debido al potencial, características y facilidades que aporta a todo el proceso de desarrollo, Además de que esta metodología es la que adoptó la UCI para el desarrollo de sus proyectos productivos.

1.7.4.1 Proceso Unificado de Rational

El Proceso Unificado de Racional es un proceso de desarrollo de software, lo que se traduce como el conjunto de actividades necesarias para trasformar los requisitos de un usuario en un sistema de software. Además RUP es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto(RUMBAUGH 2004).

Es un proceso para el desarrollo de software orientado a objetos que utiliza el Lenguaje de Unificado de Modelado (Unified Model Language, UML) para describir todo el proceso. Está basado en componentes, lo que significa que el software en construcción esta formado por componentes que se conectan a través de interfaces bien definidas.

Sus características principales son:

1. Guiado/Manejado por casos de uso.
2. Centrado en arquitectura.

3. Iterativo e Incremental.
4. Desarrollo basado en componentes.
5. Utilización de un único lenguaje de modelación.
6. Proceso Integrado.

El proceso divide en cuatro fases el desarrollo de software.

1. **Inicio**, El Objetivo en esta etapa es determinar la visión del proyecto.

2. **Elaboración**, En esta etapa el objetivo es determinar la arquitectura óptima.

3. **Construcción**, En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

4. **Transición**, El objetivo es llegar a obtener un lanzamiento (release) del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

1.7.4.2 UML

Como se ha dicho anteriormente RUP utiliza UML para describir todo el proceso de desarrollo, es por esto que a continuación se dedica un epígrafe para describir de forma muy general que es UML.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (WIKIPEDIA 2007d).

Es importante destacar que UML es un "lenguaje" para especificar y no un método o un proceso, se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. Es el lenguaje en el que está descrito el modelo (WIKIPEDIA 2007d).

La finalidad de los artefactos es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. El modelo UML de un sistema es similar a un modelo a escala de un edificio junto con la interpretación del artista del edificio. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

Una vez seleccionada la metodología que utilizaremos para llevar a cabo el proceso de desarrollo software, se analizarán las herramientas que se utilizarán para desarrollar el proceso.

1.7.5 Herramientas de Desarrollo

A continuación se muestras una lista de todas las herramientas que se utilizarán para desarrollar el sistema.

1. ArgoUML v0.24: Es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia BSD (Berkeley Software Distribution) código abierto.

2. ArcheGenXML: Es un generador de código fuente, el cual permite generar los productos completamente funcionales de Zope/Plone basado en el framework Archetypes para modelos de UML usando los archivos XMI (.xmi, .zargo, .zumil)

3. Eclipse 3.2: Es un IDE (Del inglés Integrated Development Environment, Entorno de Desarrollo Integrado) para el desarrollo de aplicaciones. Es una plataforma de software de código abierto que emplea módulos (del inglés plug-in) para proporcionar toda su funcionalidad. Dentro de los módulos que se le pueden agregar, se encuentra el Pydev, que es que se va a utilizar para desarrollar el producto en cuestión (WIKIPEDIA 2007b).

1.8 Resumen

En este capítulo se definieron los principales conceptos y aspectos relacionados con los observatorios. Se analizaron varios tipos de observatorios donde se vieron importantes aspectos a tener en consideración a la hora de elegir cual cumple con las necesidades de esta investigación. Finalmente se propuso la tecnología y las herramientas con las que se desarrollará el observatorio de información.

1.9 Conclusiones Parciales

Después de realizado el estudio del estado del arte y analizar los aspectos importantes a tener en cuenta, cuando se va a desarrollar un observatorio de información, se concluye que:

1. Se desarrollará un observatorio de información que una vez especificado las fuentes de información, de forma activa monitoreará los campos de conocimientos que sean de interés para el usuario

2. El sistema se desarrollará con la tecnología Zope/Plone.

3. La metodología mediante la cual se guiará todo el proceso de desarrollo será RUP.

4. La herramienta para el diagramado que se utilizará será ArgoUML.

5. El IDE de desarrollo que se utiliza será Eclipse 3.2

CAPÍTULO 2: MODELO, PROPUESTA CONCRETA DE LA SOLUCIÓN

2.1 Introducción

En este capítulo se hace una descripción de la propuesta a desarrollar. Se comienza haciendo un análisis de los requisitos que se necesitan para la construcción de un observatorio de información. Luego se describe el producto que se desea desarrollar, especificando los principales conceptos a tratar y las principales responsabilidades de los módulos por los que está compuesto el producto.

Además se identifican los actores que intervienen en la aplicación y los principales casos de uso. Se hace una descripción de los casos de uso, donde se describe todo el flujo de actividades y las características de cada uno de ellos. Se muestran los principales diagramas que se diseñaron. Se describen algunos conceptos importantes a tener en cuenta cuando se desarrolla un producto para Plone.

Finalmente se describen los estándares de codificación que se adoptaron para garantizar una mejor comprensión y reutilización del código. También se describen algunas de las principales funciones que se utilizaron en la programación de los algoritmos, además de realizar el análisis de los resultados.

2.2 Propuesta a desarrollar

Después de analizar un grupo de ejemplos, expuestos en el capítulo 1, los cuales nos introdujeron en el mundo de los observatorios de información, se hace evidente que la mejor opción de acuerdo a las necesidades expuestas, es desarrollar un observatorio de información que permita lo siguiente:

- Monitorear activamente fuentes de información, de acuerdo a los intereses de cada usuario, con el objetivo de mantener un control de toda la información que generan los principales medios de información, empresas e instituciones en Internet.
- Especificar las fuentes de información que se van a monitorear. Esto implica también la gestión de estas fuentes, así como consultar las fuentes de información para descargar la información que se quiere obtener.
- Visualizar todos los resultados de sus monitoreos.

- Emitir reportes donde se plasme todo el resultado del monitoreo realizado en un período de tiempo. Este período debe ser modificable por el usuario que utilice la aplicación.

Una vez hecho un análisis de qué es lo que se necesita para desarrollar un observatorio de información, se determinaron los siguientes requerimientos funcionales.

2.3 Requerimientos funcionales

R1. Permitir adicionar fuentes de información.

R2. Permitir editar fuentes de información.

R3. Permitir eliminar fuentes de información.

R4. Realizar consultas activamente a las fuentes de información.

R5. Permitir consultar documentos descargados por las consultas realizadas.

2.4 Descripción del Producto Propuesto

Para dar cumplimiento a los objetivos planteados anteriormente y a los requerimientos funcionales, se decidió dividir el producto en dos módulos fundamentales, el recolector de conocimiento (Knowledge Collector) y el programador de consultas (Scheduler).

El módulo Knowledge Collector se encargará de todo el proceso de gestión de las fuentes de información, es decir, añadir, eliminar y editar fuentes de información. Además se encargará de una parte importante del proceso de consulta. Es el encargado de eliminar todos los rastros de las consultas anteriores y de conectarse a las fuentes de información, descargar sus documentos y hacerlos persistentes.

En el párrafo anterior se explicó abordado sobre las fuentes de información y sobre los documentos descargados, pero en ningún momento se ha definido que es una fuente de información y que es un documento descargado. Para lograr un mayor entendimiento de lo que se quiere transmitir se definirán a continuación estos conceptos importantes a la hora de desarrollar este proyecto.

Se le denomina fuente de información al contenido formado por el siguiente conjunto de atributos.

- Link
- Description
- Depth

Donde link (enlace) representa al url que se desea monitorear, por ejemplo: http://www.plone.org. Description (descripción) representa a una breve descripción del sitio que se está analizando donde se puede especificar los tópicos que aborda el sitio y otros datos de interés para el usuario. Depth (profundidad) se refiere a la profundidad con la que se va a realizar la consulta a la fuente de información, esta puede variar en un rango de 1-5 según los intereses del usuario. Cuando se realice la consulta solo se tendrán en cuenta los vínculos internos del sitio, es decir, un link que apunte fuera del sitio en cuestión no será analizado puesto que este constituiría una fuente de información que el usuario no ha especificado y perfectamente esta fuente podría tratar tópicos que el usuario no está interesado en monitorear.

Se denomina documento descargado al contenido formado por los atributos, title (título) y text (texto), donde título representa al url de la página Web que se va descargar, y texto representa el texto de la página Web que será descargado.

Una vez definidos estos conceptos se continúa con la descripción del producto. Un usuario podrá crear tantos Knowledge Collector como desee, por ejemplo: supóngase que un usuario del sistema le interesa monitorear información sobre Zope, Plone y Python. Lo que debería hacer es crear tres Knowledge Collector, uno para Zope, otro para Plone y el último para Python, dentro de los cuales se especificarían las fuentes de información a las que desea que se le realice la consulta. De esta manera se podrá tener agrupada la información una vez que se descargue.

El Knowledge Collector tiene dependencia con el módulo Scheduler debido que este último es el encargado de ejecutar las consultas a las fuentes de información.

El módulo Scheduler ejecuta todas las consultas programadas en el día. Se encargará de la manipulación de los hilos necesarios para ejecutar sus trabajos (consultas). El Scheduler constará con un hilo principal que tendrá la responsabilidad de monitorear la hora del sistema. Este hilo se creará una vez que se cree

la única instancia del Scheduler y es en ese momento donde calculará la hora que falta para las doce de la noche y se dormirá hasta que trascurra ese tiempo. Una vez que sean las doce de la noche el hilo se despertará y se crearán hilos que ejecutarán las tareas que están programadas, es decir, a cada tarea se le asociará un hilo que tendrá la responsabilidad de ejecutar la consulta. Una vez que los hilos cumplan su objetivo dejarán de existir.

Existirá sólo un Scheduler el cual controlará de forma activa todas las consultas, por ejemplo: si existen tres Knowledge Collectors existirán tres trabajos que el Scheduler deberá ejecutar a las doce de la noche, todos los días.

De acuerdo con los requerimientos del sistema y la descripción del producto se han determinado los siguientes actores del sistema.

2.5 Actores del Sistema

Actores	Justificación
Usuario Administrador	Representa a la persona que interactúa con el sistema la cual tiene los permisos adecuados, para añadir fuentes de información y visualizar el resultado de las consultas.
Reloj	Indica cuando se deben realizar las consultas de acuerdo con la hora establecida, doce de la noche.
Usuario Anónimo	Representa a cualquier persona que interactúa con el sistema pero que no tiene ningún tipo de permisos y la única acción que puede realizar es visualizar el resultado de las consultas.

Tabla 10 Actores del Sistema

A continuación se muestran los casos de uso que darán cumplimiento a los requerimientos funcionales del sistema.

CU-1	Adicionar_FI
Actor	Usuario Administrador
Descripción	El sistema le permite al usuario adicionar una fuente de información al sistema.
Referencia	R1

Tabla 11 CU: Adicionar_FI

CU-2	Editar_FI
Actor	Usuario Administrador
Descripción	El sistema le permite al usuario editar una fuente de información al sistema.
Referencia	R2

Tabla 12 CU: Editar_FI

CU-3	Eliminar_FI
Actor	Usuario Administrador
Descripción	El sistema le permite al usuario eliminar una fuente de información al sistema.
Referencia	R3

Tabla 13 CU: Eliminar_FI

CU-4	Realizar_ConsultaFI
Actor	Reloj
Descripción	El sistema activamente realizará consultas a las fuentes de información, con el objetivo de descargar sus documentos asociados.
Referencia	R4

Tabla 14 CU: Realizar_ConsultaFI

CU-5	Consultar_Documentos
Actor	Usuario Administrador, Usuario Anónimo
Descripción	El sistema le permite al Usuario consultar los documentos que se obtuvieron de la realización de una consulta.
Referencia	R5

Tabla 15 CU: Consultar_Documentos

2.6 Modelo de Casos de Uso del Sistema

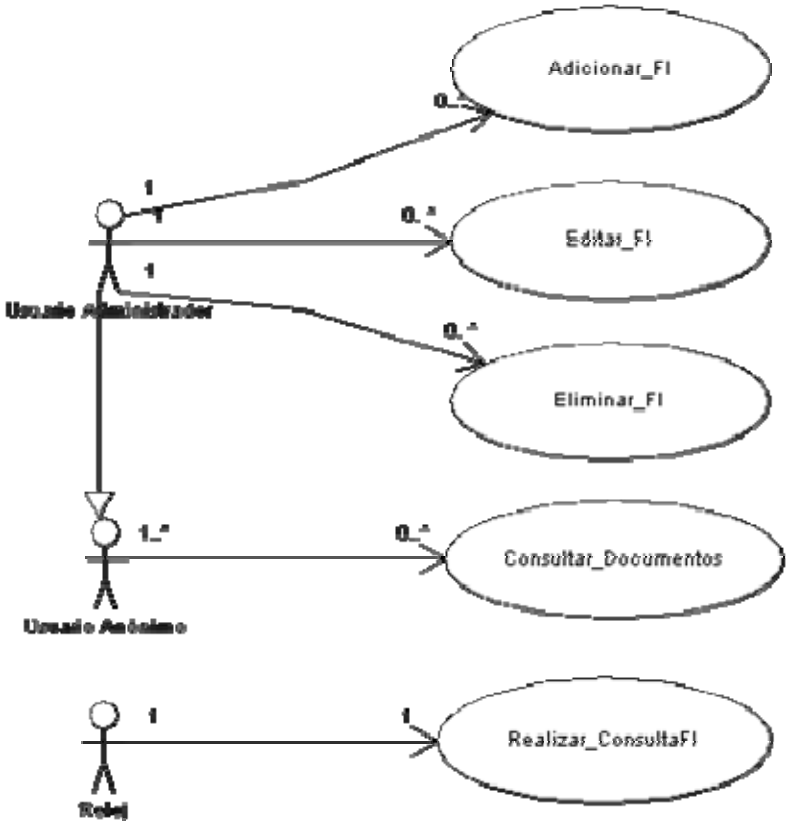

Figura 1 Modelo de Casos de Uso del Sistema

2.7 Expansión de los casos de Uso

Caso de uso:	**Adicionar_FI**
Actores :	Usuario Administrador (inicia)
Propósito:	Adicionar una fuente de información al sistema.

<table>
<tr><td colspan="5">Resumen:

El CUS (caso de uso del sistema) se inicia cuando el actor desea adicionar una fuente de información, el actor selecciona la opción adicionar fuente de información y llena los campos obligatorios, seguidamente el sistema se encarga de hacer persistente esa información.</td></tr>
<tr><td colspan="5">Referencias:

RF 1</td></tr>
<tr><td colspan="5">Precondiciones:

El actor haya entrado al sistema</td></tr>
<tr><td colspan="5">Poscondiciones:

La instancia del caso de uso termina cuando se haya adicionado una fuente de información.</td></tr>
<tr><td colspan="5">Curso normal de los eventos:</td></tr>
<tr><td colspan="2">Acción del actor:</td><td colspan="3">Respuesta del proceso del Sistema:</td></tr>
<tr><td>1</td><td>El actor selecciona la opción adicionar fuente de información.</td><td>1.1</td><td colspan="2">El sistema muestra el formulario a llenar por el actor.</td></tr>
<tr><td>2</td><td>El usuario introduce los datos de la nueva fuente información y presiona el botón guardar.</td><td>2.1</td><td colspan="2">El sistema verifica la integridad de los datos y se adiciona esta al sistema.</td></tr>
<tr><td></td><td></td><td>2.2</td><td colspan="2">El sistema muestra la fuente de información adicionada.</td></tr>
<tr><td colspan="2">Prioridad :</td><td colspan="3">Crítico</td></tr>
</table>

Tabla 16 Expansión CU: Adicionar_FI

Caso de uso:	**Editar_FI**		
Actores :	Usuario Administrador (inicia)		
Propósito:	Editar una fuente de información al sistema.		
Resumen: El CUS se inicia cuando el actor desea editar una fuente de información, el actor selecciona la fuente de información que desea editar, selecciona la pestaña editar y modifica los campos que considere necesarios, seguidamente el sistema se encarga de guardar los cambios realizados.			
Referencias: RF 2			
Precondiciones: El actor haya entrado al sistema			
Poscondiciones: La instancia del caso de uso termina cuando se haya editado una fuente de información.			
Curso normal de los eventos:			
Acción del actor:		**Respuesta del proceso del Sistema:**	
1	El actor selecciona la fuente de información que desea editar.	**1.1**	El sistema muestra los datos de la fuente de información seleccionada.

2	El actor selecciona la pestaña editar.	**2.1**	El sistema muestra un formulario con todos los datos de la fuente de información.
3	El actor modifica los datos y presiona el botón guardar.	**3.1**	El sistema actualiza los nuevos datos y muestra los datos de la fuente de información actualizados.
Prioridad :	Crítico		

Tabla 17 Expansión CU: Editar_FI

Caso de uso:	**Eliminar_FI**
Actores :	Usuario Administrador (inicia)
Propósito:	Eliminar una fuente de información al sistema.
Resumen: El CUS se inicia cuando el actor desea eliminar una fuente de información, el actor selecciona la fuente de información que desea eliminar, selecciona la pestaña acción/eliminar, el sistema pregunta si esta seguro de realizar la acción, si el usuario selecciona la opción eliminar, el sistema elimina la fuente de información.	
Referencias: RF 3	
Precondiciones: El actor haya entrado al sistema	

Poscondiciones: La instancia del caso de uso termina cuando se haya eliminado una fuente de información.			
Curso normal de los eventos:			
Acción del actor:		**Respuesta del proceso del Sistema:**	
1	El actor selecciona la fuente de información que desea eliminar.	**1.1**	El sistema muestra los datos de la fuente de información seleccionada.
2	El actor selecciona la pestaña acción/eliminar.	**2.1**	El sistema pregusta si esta de acuerdo en eliminar la fuente de información
3	El actor asegura que desea eliminar la fuente de información.	**3.1**	El sistema elimina la fuente de información.
Prioridad :	Crítico		

Tabla 18 Expansión CU: Eliminar_FI

Caso de uso:	**Realizar_ConsultaFI**
Actores :	Reloj (inicia)
Propósito:	Consultar las fuentes de información y extraerle su contenido.
Resumen: El CUS se inicia todos los días a las doce de la noche cuando el actor activa el proceso de realizar una consulta. En ese momento el sistema elimina todos los rastros que existen de la consulta anterior, luego hace una recuperación de todas las fuentes de información que hasta el momento tiene guardadas, se conecta a cada una de ellas y descarga su documento asociado. Estos documentos se hacen persistentes teniendo en cuenta el url, y	

<table>
<tr><td colspan="4">el texto del url descargado, es decir, conforman una pareja url, texto.</td></tr>
<tr><td colspan="4">Referencias:

RF 4</td></tr>
<tr><td colspan="4">Precondiciones:

Existencia de un Knowledge Collector y una fuente de información.</td></tr>
<tr><td colspan="4">Poscondiciones:

La instancia del caso de uso termina cuando los documentos han sido descargados de las fuentes de información y hechos persistentes en la base de datos de Zope.</td></tr>
<tr><td colspan="4">Curso normal de los eventos:</td></tr>
<tr><td colspan="2">Acción del actor:</td><td colspan="2">Respuesta del proceso del Sistema:</td></tr>
<tr><td>1</td><td>El actor activa el proceso de realizar consulta</td><td>1.1</td><td>El sistema elimina los rastros de la consulta anterior.</td></tr>
<tr><td></td><td></td><td>1.2</td><td>El sistema se conecta a las fuentes de información.</td></tr>
<tr><td></td><td></td><td>1.3</td><td>El sistema descarga los documentos de las fuentes de información.</td></tr>
<tr><td></td><td></td><td>1.4</td><td>El sistema guarda en la base de datos de Zope los documentos descargados</td></tr>
<tr><td colspan="2">Prioridad :</td><td colspan="2">Crítico</td></tr>
</table>

Tabla 19 Expansión CU: Realizar_ConsultaFI

<table>
<tr><th colspan="2">Caso de uso:</th><th colspan="3">Consultar_Documentos</th></tr>
<tr><td colspan="2">Actores :</td><td colspan="3">Usuario Administrador, Usuario Anónimo</td></tr>
<tr><td colspan="2">Propósito:</td><td colspan="3">Permite al los usuarios de la aplicación consultar los documentos descargados previamente por el caso de uso Realizar_ConsultaFI.</td></tr>
<tr><td colspan="5">Resumen:

El CUS se inicia cuando el actor decide consultar un documento, el actor selecciona el documento que desea visualizar y el sistema muestra el contenido del documento consultado.</td></tr>
<tr><td colspan="5">Referencias:

RF 5</td></tr>
<tr><td colspan="5">Precondiciones:

Existencia de un documento para consultar.</td></tr>
<tr><td colspan="5">Poscondiciones:

La instancia del caso de uso termina cuando el documento ha sido consultado.</td></tr>
<tr><td colspan="5">Curso normal de los eventos:</td></tr>
<tr><td colspan="2">Acción del actor:</td><td colspan="3">Respuesta del proceso del Sistema:</td></tr>
<tr><td>1</td><td>El actor selecciona el documento que desea consultar.</td><td>1.1</td><td colspan="2">El sistema muestra los datos del documento seleccionado.</td></tr>
<tr><td colspan="2">Prioridad :</td><td colspan="3">Crítico</td></tr>
</table>

Tabla 20 Expansión CU: Consultar_Documentos

2.8 Diagrama de Clases del Diseño

Como se explicó al final del capítulo 1 se utilizará UML para modelar los diagramas de clases que se generen. La herramienta de diagramado que se utilizará será ArgoUML por las facilidades de integración que tiene con ArchGenXML la cual permite crear productos para Zope/Plone una vez proporcionado el .zargo que se obtiene de la modelación en el ArgoUML.

2.8.1 Los productos en Plone

Cuando se crea un producto en ArgoUML y se genera el código con el ArchGenXML, se obtiene un producto con una estructura muy bien definida, la cual es indispensable para que un producto pueda ser reconocido e instalado por el Plone. Todos los productos una vez que son generados por ArchGenXML están compuestos por grupo de paquetes y módulos. Los principales módulos que se crean son el ***__init__.py*** que se utiliza para inicializar el producto de modo que cuando Zope lea este archivo en el arranque, termine la estalación del producto. El módulo ***config.py*** se usa para almacenar todas las variables de configuración del producto. Además se crea un módulo para cada contenido o herramienta que se genere. Los principales paquetes que se crean son el **Skins** donde se almacenarán todas las vistas que se utilizarán en el producto y el módulo **Extention** donde se guarda el módulo ***install.py*** en el cual se almacenará los métodos que permiten instalar y desinstalar el producto en Plone.

Es necesario analizar algunos conceptos importantes de Plone así como sus usos, con el objetivo de que se entienda mejor porque la utilización de cada uno de estos cuando se desarrolla con la tecnología Zope/Plone. Los conceptos son:

2.8.2 Conceptos de Plone

- CMF(Content Management Framework, Framework de administración de contenidos)
- Archetype (Arquetipo)
- Content Types (Tipos de Contenidos)

- Tool (Herramienta)

2.8.2.1 CMF

CMF es un framework de administración de contenidos de Zope que proporciona un sistema servicios y de objetos contenidos útiles para construir portales dinámicos. Plone obtiene estas características y mejora muchas de ellas para proporcionar al usuario un producto de alta calidad.(MCKAY 2005)

2.8.2.2 ArcheType

Archetype es un framework para desarrollar tipos de contenidos para proyectos de Plone. Una vez que la descripción de un contenido ha sido escrita en python, Archetype manipula casi todo lo demás, incluyendo la creación de vistas y los formularios de edición para los desarrolladores por lo que no es necesario escribir nada de este código. Esto es muy importante porque con su ayuda se puede desarrollar muy rápido y con una pequeña cantidad de código. Es decir, en un proyecto desarrollado con otra tecnología, por ejemplo, asp.net, se tienen que crear vistas, formularios, validaciones, etc., para cada una de las interfaces de usuario, lo que consume mucho tiempo y genera mucho código, lo que representa un esfuerzo mayor a la hora de desarrollar. Con Plone todo este trabajo se puede ahorrar cuando usamos Archetype.(MCKAY 2005)

2.8.2.3 Content Types

Cuando se crea una instancia de un sitio de Plone y se trabaja en él, se observa que puede agregar nuevas carpetas, imágenes, documentos, etc. Todos estos objetos para Plone son contenidos. Plone trae consigo un conjunto de contenidos los cuales se pueden adicionar para conformar un sistema de gestión de contenidos, una intranet o un portal de cualquier clase. De ahí que un contenido es un objeto que puede ser adicionado y editado a través de la interfaz de Plone por un usuario el cual tiene que tener los permisos adecuados para poder realizar estas operaciones. Ahora, muchas veces es necesario tener un tipo de contenido que no viene incluido en el Plone, esto está dado según las necesidades de cada desarrollador. En ese momento es donde es necesario crear contenidos personalizados, los cuales responderán a una determinada responsabilidad. Un contenido básicamente es un componente que está

compuesto por esquemas archetypes, una implementación persistente simple que se corresponde con atributos y métodos, unas directivas de configuración y nada de html.

2.8.2.4 Tool

Los tools son servicios que residen dentro del sitio de Plone. Sólo se pueden instanciar una vez y pueden ser accedidos por todos los objetos del sitio, existen muchas herramientas en una instancia de plone. Las herramientas pueden almacenar atributos y métodos como lo hacen los contenidos. Generalmente ellas almacenas datos de configuración y métodos útiles para que el resto de los productos usen.

Una vez esclarecidos estos conceptos fundamentales se analizarán los usos de los principales estereotipos que se utilizarán para modelar los diagramas de clases en el ArgoUML.

2.8.3 Estereotipos Utilizados

- <<subsystem>>: Se utilizará para definir productos independientes.
- <<package>>: Se utilizará para definir paquetes auxiliares que se utilicen en la implementación.
- <<content_class>>: Se utiliza para generar tipos de contenidos.
- <<python_class>>: Se utiliza para generar clases de python que no contengan archetypes.
- <<portal_tool>>: La clase que se genera con este estereotipo es una herramienta.

Después de hacer un análisis de los requerimientos del sistema, se decidió desarrollar dos productos para Plone que en su conjunto formarán el Observatorio de Información, los cuales se agruparán en dos subsistemas. El primer producto Recolector de Conocimiento (Knowledge Collector) será el subsistema principal que dependerá del subsistema secundario Programador de Consultas (Portal Scheduler Tool) que será una herramienta de Plone, la cual permitirá chequear las tareas que se deben ejecutar en el día.

2.8.4 Definición de subsistemas de diseño

De acuerdo a lo antes expuesto se propone los siguientes modelos de subsistemas.

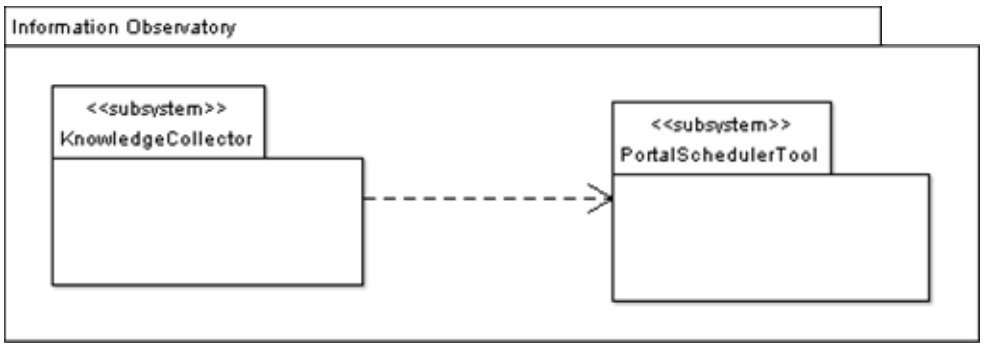

Figura 2 Subsistema Observatorio de Información

El subsistema Knowledge Collector se compone por los módulos y paquetes que genera automáticamente ArgoUML los cuales se mencionaron en el epígrafe 2.8.2. Además incluye el paquete Beatifullsoup, que es un analizador sintáctico (parser) para html el cual entre sus funciones principales incluye, la extracción de los url del documento que analiza, de ahí que se utilice en este subsistema.

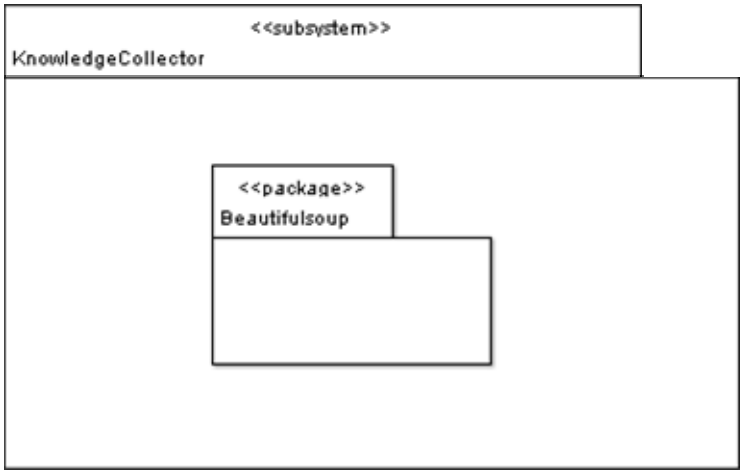

Figura 3 Subsistema Knowledge Collector

El subsistema Portal Scheduler Tool no se analizará puesto que sólo cuenta con los módulos y paquetes que se crean por defecto cuando se genera el producto por el ArchGenXML.

2.8.5 Diagrama de clases por subsistemas de diseño

A continuación se muestra el diagrama de clases del subsistema Knowledge Collector.

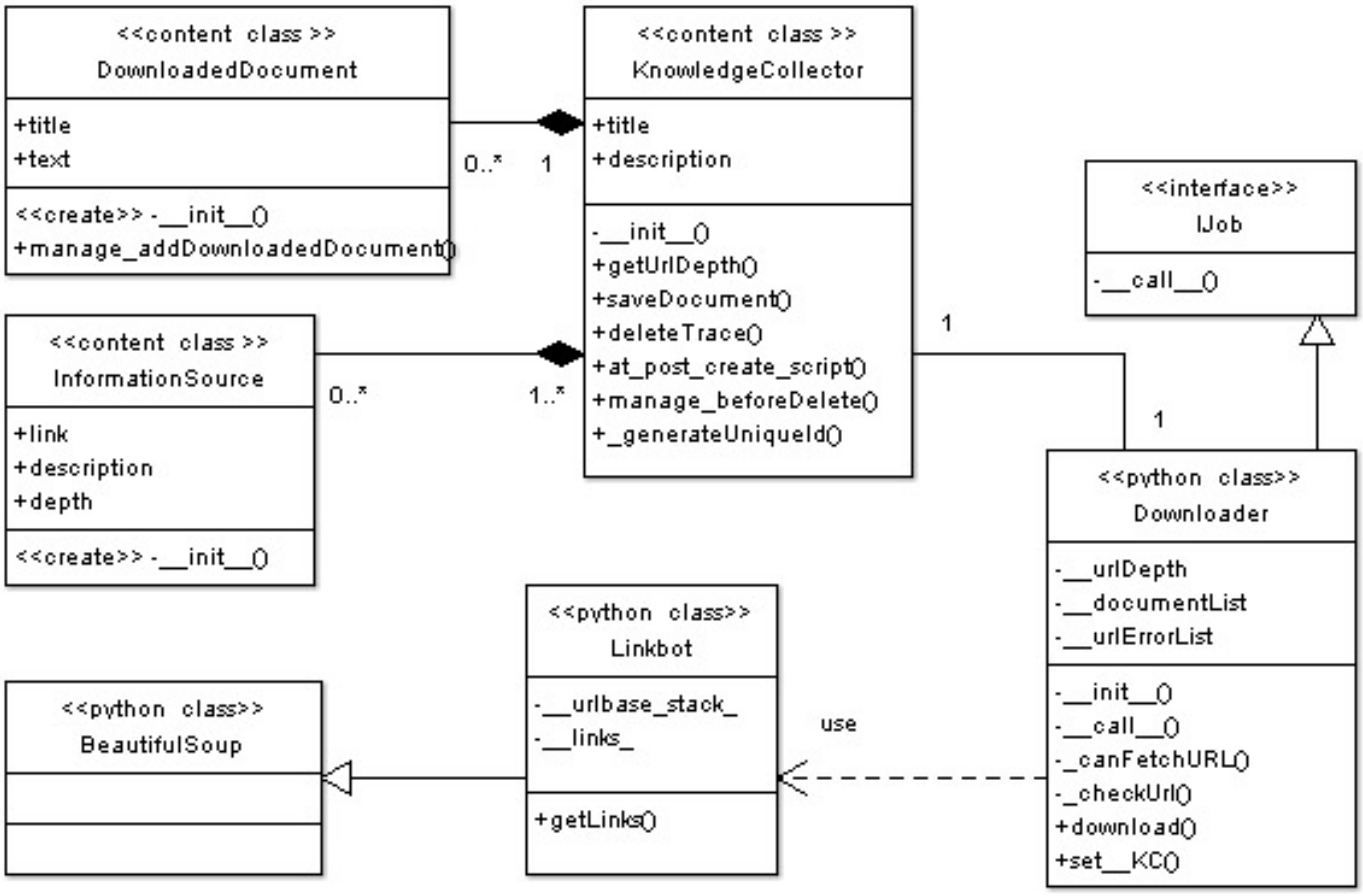

Figura 4 Diagrama de Clases Subsistema Knowledge Collector

Los diagramas de clases bien realizados son la base de un buen funcionamiento del producto a desarrollar, diseñando buenos productos se puede garantizar una implementación de código mucho más clara y más usable. A continuación se mostrarán algunos aspectos que se tuvieron en cuenta a la hora de implementar el producto.

A continuación se muestra el diagrama de clases del Portal Scheduler Tool.

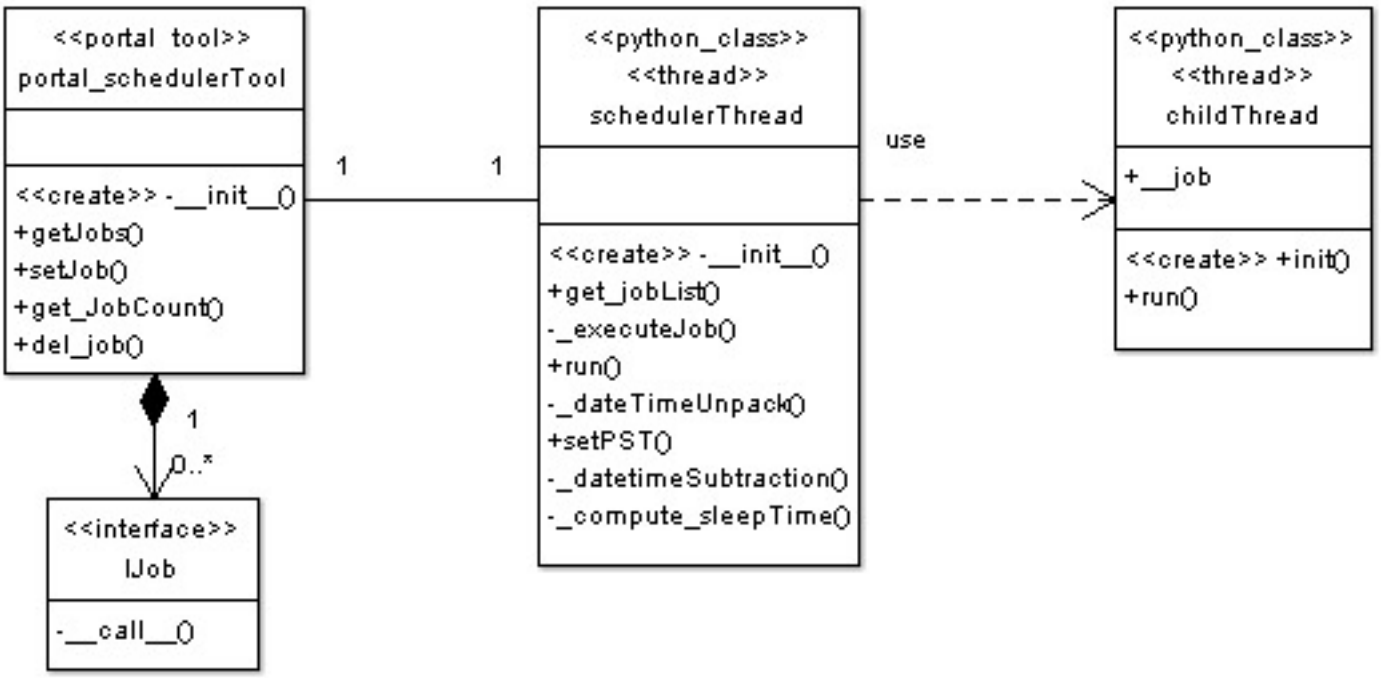

Figura 5 Diagrama de Clases Subsistema Portal Scheduler Tool

2.9 Implementación

Estándares de codificación que se establecen para lograr una codificación eficiente y reutilizable.

2.9.1 Estándares de codificación

Desde hace mucho tiempo se aboga porque los programadores programen sus códigos con un lenguaje que pueda ser entendido por cualquier otro colega, que en algún momento obtenga ese código para reutilizarlo en uno de sus proyectos. Una de las filosofías de la comunidad de software libre es la reutilización de código, es decir, nadie es dueño del código, todo el mundo puede obtenerlo y mejorarlo o adaptarlo según sus necesidades. Por lo que para lograr un mayor entendimiento del código a la hora de reutilizar algún componente y para lograr una mayor claridad cuando se programe se utilizarán los siguientes estándares de codificación.

2.9.1.1 Sangría

Se usarán cuatro espacios para cada nivel de sangría. Para códigos antiguos que se quieran perder, se puede mantener una sangría de 8 espacios para cada nivel.

2.9.1.2 Tabs y Espacios

La separación del código Python se realizará a través de espacios, tabs o combinaciones de espacios o tabs.

2.9.1.3 Importaciones:

Las importaciones usualmente deben estar en líneas diferentes. Las importaciones siempre se ponen en la parte superior del archivo, justo después de cualquier comentario de módulo y antes de cualquier módulo global y constantes. Siempre se utilizará el camino absoluto del paquete para todas las importaciones, ejemplo.

```
from AccessControl import ClassSecurityInfo
from Products.Archetypes.atapi import *
```

2.9.1.4 Espacios en blanco en expresiones y declaraciones:

Evitar el espacio blanco en las situaciones siguientes:

- Inmediatamente después del paréntesis, corchetes o llaves.

```
Si: spam(ham[1], {eggs: 2})
No:  spam( ham[ 1 ],
)
```

- Inmediatamente antes de coma, punto y coma, dos puntos:

```
Si: if x == 4: print x, y; x, y = y, x
 No:  if x == 4 : print x , y ; x , y = y , x
```

- Inmediatamente antes de la apertura de paréntesis que inicializan la lista de argumentos de una llamada de función:

```
No: spam(1)
```

```
No:  spam (1)
```

- Inmediatamente antes de la apertura de paréntesis que inician un índice o slicing:

```
Si: dict['key'] = list[index]
No:  dict ['key'] = list [index]
```

- Más de un espacio alrededor de una asignación(o otras) de operador para alinearlo con otro:

```
Si:
      x = 1
      y = 2
      long_variable = 3
No:
      x             = 1
      y             = 2
      long_variable = 3
```

2.9.1.5 Comentarios:

Los comentarios que contradicen el código son peores que ningún comentario. Siempre se debe hacer una prioridad de mantener los comentarios actualizados cuando el código es cambiando.

Los comentarios deben ser oraciones completas. Si el comentario es una frase o una oración, su primera palabra debe estar en mayúscula, a menos que sea un identificador que empiece con una letra minúscula (nunca se debe alterar la forma de un identificador)

Si un comentario es corto, el punto al final puede ser omitido. Los bloques de comentarios generalmente consisten en uno o más párrafos construidos con completas oraciones, y cada una de estas oraciones debe terminar en un punto.

Se deben usar dos espacios después del punto final de la oración. Los comentarios de bloque se aplican generalmente a cierto (o todo el) código que los siga, y están indentados al mismo nivel que ese código. Cada línea del bloque de comentario empieza con el signo # y un espacio simple (a menos que sea indentado dentro del comentario). Los párrafos dentro de un bloque de comentario están separados por una línea conteniendo un simple #.

2.9.1.5.1 Las cadenas de documentación:

Se deben escribir cadenas de documentación para todos los módulos públicos, funciones, clases y métodos, pero también se debe tener un comentario que describa que es lo que hace un método, este comentario debe aparecer después de la línea “def”, por ejemplo:

```
"""Return a foobang

      Optional plotz says to frobnicate the bizbaz first.

      """
```

2.9.1.6 Control de versiones

Cuando se usa un controlador del versiones como Subversion, CVS, o RCS se escribe lo siguiente en el archivo de código.

```
__version__ = "$Revision: 43264 $"
# $Source$
```

Estas líneas deben ser incluidas después de las cadenas de documentación de los módulos y antes de cualquier otro código, separados con líneas en blanco tanto encima como debajo.

2.9.1.7 Estilos de nombres recomendados:

Existen muchos y diferentes estilos de nombramiento, a continuación se detallarán cada uno de los estilos que se utilizarán. Estos estilos cumplen los estándares internacionales de codificación para el lenguaje de programación Python.

2.9.1.7.1 Nombres de módulos:

Los módulos deben tener nombres cortos y en minúsculas, sin subrayado (underscores).

2.9.1.7.2 Nombres de Clases

Para nombrar las clases se debe usar el siguiente formato: La primera letra de cada palabra debe comenzar con una letra mayúscula. Si el nombre de la clase es una combinación de palabras no se debe usar subrayado (underscore) para separar estas, por ejemplo:

```
Si:
      class MyClass
No:
      Class My_Class
```

2.9.1.7.3 Nombres de funciones:

Los nombres de funciones deben ser en minúsculas, con palabras separadas por subrayado como sea necesario para mejorar la legibilidad.

2.9.1.7.4 Nombres de Variables:

Para evitar nombres que choquen con subclases se utiliza el subrayado, se utilizan dos subrayados al inicio de la variable para evitar este problema.

2.9.1.8 Nombramientos a evitar

Nunca se debe utilizar la letra minúscula “l” u “o”, u “O” como nombres simples de variables, en algunas fuentes de letras estas letras no se distinguen de los números 1 y 0, cuando se tenga que utilizar la letra “l” minúscula, se debe sustituir por la “L”.

2.9.2 Implementaciones de métodos

Después de definidos los principales estándares que se utilizaron cuando se desarrollo el código del producto, a continuación se muestran los principales métodos que se implementaron.

2.9.2.1 Método KnowledgeCollector._saveDocument()

Este método pertenece al contenido knowledgeCollector. Su responsabilidad es hacer persistentes los documentos descargados suministrados por las consultas. Este método utiliza la función

manage_addDownloadedDocument para crear contenidos del tipo downloadedDocument que le son suministrados a través del parámetro downloadedDocumentList el cual es una lista de listas ([[url,text],[url1,text1],[url2,text2]]).

```
security.declarePrivate('_saveDocument')
    def _saveDocument(self,downloadedDocumentList):
        """
          Para guardar el documento descargados en la base de datos de
          Zope.
        """
        # Se crean los contenidos descargados dinamicamente.
        for url, html in downloadedDocumentList:
            #crear el contenido
            id=self._generateUniqueId("DownloadedDocument")
            manage_addDownloadedDocument(self,id,url,html)
            # Update navigation tree and search index data
            self[id].reindexObject()
```

Tabla 21 Método _saveDocument

2.9.2.2 Método Downloader.download()

Este método se utiliza para descargar los documentos de las fuentes de información. Entre las responsabilidades que tiene se encuentran, conectarse a las fuentes de información, descargar el documento de la página Web, extraer los links asociados a la página Web que se está analizando y guardarlos, si no existen en la lista de fuentes de información. El método utiliza el analizador sintáctico Linkbot para obtener de un documento descargado los links asociados, _cantFetchURL es un método que se utiliza para verificar si la sintaxis del url que se analizará es correcta, de manera que se pueda proseguir con el proceso, _checkUrl que se utiliza para establecer la conexión con el url.

```
def download(self,):
        """
         Metodo para descargar todos los documentos de las fuentes de
         Informacinon.
        """

        # Se obtienen los elementos de la lista (url, depth).
        for url, depth in self.__urlDepth:
          # Se conecta al sitio y se descarga todo el html.
```

```
        # De la pagina y se guarda en la lista.
        if self._canFetchURL(url):

            fh = self._checkUrl(url)
            if fh:
                try:
                    html=fh.read()
                except AttributeError:
                    self.__attributeError = self.__attributeError + 1
                    self.__urlErrorList.append(url)
                    print  "AttributeError: %s: no se pudo descargar \
                        el contenido de la pagina" % (url)

                # Se guarda el documento en el la lista.
                self.__documentList.append([url,html])
                self.__documentCont = self.__documentCont + 1

                # Se extraen los links referenciados en el url.
                linkbot = Linkbot(url,html)
                links = linkbot.getLinks()

                # Se guarda los link asociados a al url.
                # Si la profundida es == 0, no intereza guardarla.
                if depth > 1:
                    for urlnew in links:
                        # Se chequea si existe la nueva
                        # url en la lista.
                       if not urlnew in self.__urlDepth :
                           #si no existe se adiciona el url y
                           #se le pone la profundida del padre menos 1
                           if  url in urlnew:
                               self.__urlDepth.append([urlnew,\
                                depth-1])
                               print 'Se a adicionado el url: %s'\
                                     % urlnew
                               self.__urlCont = self.__urlCont + 1

    print 'Se adicionaron %d urls' % self.__urlCont
    print 'se crearon %d documentos'  % self.__documentCont
    print 'Ocurrieron %d urllib2.HTTPError' % self.__httpError
    print 'Ocurrieron %d urllib2.URLError' % self.__urlError
    print 'Ocurrieron %d Unexpecte Error' % self.__unexpectedError
    print 'Ocurrieron %d AttributeError' % self.__attributeError
    return self.__documentList
```

Tabla 22 Método download

2.9.2.3 Método SchedulerThread.run()

La clase SchedulerThread hereda de la clase Thread de manera que ella es un hilo, su objetivo es monitorear la hora del sistema para que a la hora indicada (12 PM) ejecute los trabajos que tiene pendientes (downloaders).

Este método esta formado por varios métodos auxiliares. El método auxiliar _date_timeUnpack recibe como atributo un objeto datetime y devuelve una lista formada por la hora, minuto, segundo y milisegundo los cuales son necesarios cuando se quiere comparar fechas y calcular tiempo que se debe esperar para que se ejecute la consulta. Otro método auxiliar que se utiliza es _compute_sleepTime el cual se utiliza para calcular el tiempo que hay que dormir el hilo hasta que se ejecute la próxima consulta. Las constantes START_TIME1 y START_TIME2 representan las horas 00:00:00:00000 (12 pm) y 01:00:00:000000 (1 AM) respectivamente. El método _executeJob() recibe como parámetro un trabajo, este en su implementación, lo que hace es crear una instancia de un hilo y asignarle ese trabajo. A continuación se muestra el código del método para un acercamiento a lo que se quiere transmitir.

```
def run(self):
        """
        Run Scheduler jobs
        """
        cont = 0
        while True:
            localTime = datetime.datetime.now()
            if self._date_timeUnpack(str(localTime)) >\
               self._date_timeUnpack(START_TIME1) and \
               self._date_timeUnpack(str(localTime)) <\
               self._date_timeUnpack(START_TIME2):
                for job in self.__jobList:
                    self._executeJob(job)
                    print 'Ejecutando hilo%d' %(cont)
                    cont += 1
            sleepTime = self._compute_sleepTime \
                (self._datetimeSubtraction(START_TIME1,str(localTime)))
            time.sleep(sleepTime)
```

Tabla 23 Método run

2.10 Evaluación de los resultados

La evaluación de los resultados de un software es muy importante porque permite conocer como se va a comportar este cuando haga su lanzamiento. Para este software en específico, se hizo un caso de prueba donde se medió la efectividad del software descargando los contenidos de los sitios Web que visita.

Las principales variables que se tuvieron en cuenta en el caso de prueba fueron las siguientes:

- Url Descargados: representan al número de url que se descargaron del sitio Web cuando se ejecutó el proceso de descarga.
- Documentos Descargados: representan al número de documentos que se descargaron del sitio Web cuando se ejecutó el proceso de descarga.
- HttpError: representan a los errores ocurridos porque no se pudo descargar el contenido del la página Web.
- UrlError: representan errores ocurridos porque no se pudo establecer la conexión con el url especificado.
- % Efectividad: representa la efectividad de descarga que tiene el algoritmo. La efectividad se calculará como sigue:

$$\%Efectividad = (100 * DocumentosDesc\arg ados) / UrlDesc\arg ados$$

Para realizar el caso de prueba se tomaron una muestra de sitios Web de la universidad, que fueron a su vez las fuentes de información que utilizaron los Knowledge Collectors para descargar sus contenidos. A continuación se muestran los sitios Web seleccionados.

- http://10.32.30.6:5901/trac/gugle
- http://tesis.uci.cu/news.php
- http://ujc.uci.cu/

- http://teleformacion.uci.cu
- http://feu.uci.cu/

Para lograr un mayor esclarecimiento en los resultados y no crear confusión cuando se analice la efectividad del software. Se creó para cada fuente de información un Knowledge Collector que tiene la responsabilidad de descargar todos los contenidos del sitio Web. Se corrió el software cinco veces para cada Knowledge Collector creado, donde en cada corrida se varió la profundidad de descarga (Depth) empezando por uno y terminando por cinco, que es la máxima profundidad que se puede especificar. El objetivo de esta estrategia fue conocer el valor de las variables que se analizaron para cada valor de profundidad que se especifique. De manera que, conociendo la cantidad de url y documentos que se descargaron así como la cantidad de errores que ocurrieron en el proceso, se podrá calcular el por ciento de efectividad del software, lo que dará una idea de cuan bueno es el algoritmo que se implementó y se está utilizando en el proceso de descarga, que es el principal proceso que ocurre en el sistema.

Después de realizar las corridas de cada Knowledge Collector que se creó se obtuvieron los siguientes resultados, los cuales están organizados en tablas por sitio Web analizado.

En la Tabla 24 se muestran los resultados obtenidos después de correr el algoritmo en el sitio http://10.32.30.6:5901/trac/gugle.

Depth	Url Descargados	Documentos Descargados	Http Error	Url Error	% Efectividad
1	0	1	0	0	100%
2	29	24	5	0	82.75%
3	79	74	5	0	93.7%
4	144	139	5	0	96.5%
5	266	261	5	0	98.5%

Tabla 24 Sitio Web: http://10.32.30.6:5901/trac/gugle

En la Tabla 25 se muestran los resultados obtenidos después de correr el algoritmo en el sitio http://tesis.uci.cu/news.php.

Depth	Url Descargados	Documentos Descargados	Http Error	Url Error	% Efectividad
1	0	1	0	0	100%
2	77	67	10	0	87.0%
3	117	107	10	0	91.4%
4	137	119	10	8	86.8%
5	137	119	10	8	86.8%

Tabla 25 Sitio Web: http://tesis.uci.cu/news.php

En la Tabla 26 se muestran los resultados obtenidos después de correr el algoritmo en el sitio http://ujc.uci.cu/.

Depth	Url Descargados	Documentos Descargados	Http Error	Url Error	% Efectividad
1	0	1	0	0	100%
2	36	36	0	0	100%
3	166	166	0	0	100%
4	349	349	0	0	100%
5	447	447	0	0	100%

Tabla 26 Sitio Web: http://ujc.uci.cu/

En la Tabla 27 se muestran los resultados obtenidos después de correr el algoritmo en el sitio http://teleformacion.uci.cu.

Depth	Url Descargados	Documentos Descargados	Http Error	Url Error	% Efectividad
1	0	1	0	0	100%
2	27	26	1	0	96.3%
3	58	57	1	0	98.2%
4	58	57	1	0	98.2%
5	58	57	1	0	98.2%

Tabla 27 Sitio Web: http://teleformacion.uci.cu

En la Tabla 28 se muestran los resultados obtenidos después de correr el algoritmo en el sitio http://feu.uci.cu/.

Depth	Url Descargados	Documentos Descargados	Http Error	Url Error	% Efectividad
1	0	1	0	0	100%
2	79	79	0	0	100%
3	143	143	0	0	100%
4	172	172	0	0	100%
5	172	172	0	0	100%

Tabla 28 Sitio Web: http://feu.uci.cu/

Una vez obtenidos los resultados de todas las corridas se puede concluir que el algoritmo implementado tiene una efectividad promedio de 95.7%, los valores del por ciento de efectividad oscilaron entre 82% y 100% aproximadamente lo que es muy bueno porque garantiza una alta calidad en el monitoreo. La mayoría de los errores que se obtuvieron fueron todos del tipo HttpError, lo que significa que la página Web que se trato de descargar en ese momento no estaba disponible.

2.11 Resumen

En este capítulo se hizo un análisis de los principales requerimientos que se necesitan para desarrollar un observatorio de información, además de definir actores y casos de usos. Se describieron los casos de uso del sistema, así como los diagramas de clases de los productos que se desarrollaron y los principales conceptos que hay que tener en cuanta cuando se desarrolla un producto para Plone. Finalmente se abordaron los estándares de codificación que se utilizaron en la implementación, se mostraron los principales métodos desarrollados y el caso de prueba analizado en el epígrafe evaluación de los resultados.

2.12 Conclusiones Parciales

En el presente capítulo después de hacer un análisis de qué se necesita para desarrollar un observatorio de información se obtuvieron los siguientes artefactos:

- Los requerimientos funcionales necesarios para implementar el sistema.
- El diagrama de casos de uso del sistema.
- La descripción de casos de uso del sistema.
- Los diagramas de subsistemas y clases necesarios para llevar a cabo la implementación.
- La implementación del sistema mostrando los principales métodos desarrollados.
- El caso de prueba donde se analizó la efectividad del sistema en el proceso de descarga de documentos.

CONCLUSIONES

- En la bibliografía consultada durante la revisión de la literatura no se encontró información sobre un observatorio cubano del tipo del observatorio desarrollado en este trabajo de diploma.

- Se realizó un estudio de los principales sistemas automatizados que existen en el mundo, lo que permitió definir las principales características con las que cuenta el observatorio de información actualmente.

- Se obtuvieron artefactos, tales como requerimientos funcionales, diagramas de casos de uso, diagramas de subsistemas y clases.

- Con la implementación de este observatorio de información se han creado las bases para el desarrollo de un observatorio nacional que puede ser utilizado por todos los centros de investigaciones del país, etc.

RECOMENDACIONES

- Continuar refinando la aplicación para que pueda convertirse algún día en un observatorio del nivel de Google News, Yahoo News.
- Continuar con el estudio del modelo de concurrencia de Zope, que permite la ejecución de todo el mecanismo de hilos usando la base de datos de Zope.
- Chequear que antes de guardar la información en la base de datos de Zope esta no exista, con el objetivo de guardar solo información actualizada.
- Desarrollar un módulo que permita configurar la conexión del Proxy
- Implementación de un Scheduler (programador de tareas) que permita al usuario especificar la hora en que se van a ejecutar los procesos de consultas.

BIBLIOGRAFÍA

BALLBUG http://www.ballbug.com/, 2007.

BLOGLINES. *http://www.bloglines.com/*, 2007.

DIGG. *http://digg.com/*, 2007a.

---. *http://digg.com/about*, 2007b.

FEEDSHOW. *http://www.feedshow.com/*, 2007.

FEILEX, O. http://kiza.kcore.de/software/snownews/, 2005.

FINDORY http://findory.com/, 2007.

GERMÁN, R. *Ver desde la Ciudadanía*

Observatorios y Veedurías de Medios de Comunicación en América Latina., 2003.

GIBSON, O. *Google to appeal, as court rules news site is illegal*, 2006.

GONZÁLEZ, F. *Google News vs Yahoo News*, 2004.

GOOGLE http://news.google.com/intl/es_es/about_google_news.html, 2007a.

---. *http://reader.google.com/*, 2007b.

GOOGLE, N. *http://news.google.com/*, 2007c.

LICENSE, G. G. P. http://www.gnu.org/copyleft/gpl.html.

MCKAY, A. *The Definitive Guide to Plone*, Apress, 2005.

MEMEORANDUM http://www.memeorandum.com/, 2007.

MICROSOFT http://uk.newsbot.msn.com/, 2007.

RAGGLE. *http://www.raggle.org/*, 2007.

ROJO. *http://www.rojo.com/*, 2007.

RUMBAUGH, I. J. G. B. J. *El proceso unificado de desarrollo de software*. Félix Varela, 2004. p.

SITE, P. P. W. http://plone.org/, 2007a.

SITE, Z. P. W. http://www.zope.org/, 2007b.

SUSANA, H. D. *Tipología de los observatorios de medios en Latinoamérica*, 2005. 6-19.

TAILRANK http://tech.tailrank.com, 2007.

TECHMEME. *http://www.techmeme.com/*, 2007.

VIVERO. *Definición del Observatorio Tecnológico Regional*, 2005.

WESMIRCH http://www.wesmirch.com/, 2007.

WIKIPEDIA. *Agregador*, 2007a.

--- Eclipse 2007b.

---. *Observatorio Aragonés de la Sociedad de la Información*, 2007c.

--- UML, 2007d.

WINRSS. *http://www.brindys.com/winrss/*, 2007.

YAHOO. *http://news.yahoo.com/* 2007.

ZAKON, R. H. *Hobbes' Internet Timeline v8.2*, 2006.

ZOPE, P. L. Z. V. http://www.zope.org/Resources/ZPL.