

Universidad de las Ciencias Informáticas

Facultad 10

Agente de recuperación de información
Just-in-time (JITIR Agent) para el Proyecto
de Informatización de la Prensa

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

Autores:

Dionisio Estrada Marchena

José Ramón Fernández Pérez

Tutor:

Lic. Manuel Vázquez Acosta

La Habana
Junio de 2007

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Proyecto de Informatización de la Prensa de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Dionisio Estrada Marchena

José Ramón Fernández Pérez

Firma de los Autores

Lic. Manuel Vázquez Acosta

Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado Agente de recuperación de información Just-in-time (JITIR) para el Proyecto de Informatización de la Prensa, fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente
- Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a **<valor en MN o USD del efecto económico>**Y para que así conste, se firma la presente a los _____ días del mes de _____ del año _____.

Representante de la entidad

Firma

Cargo

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Agentes de recuperación de información Just-in-time (JITIR) para el Proyecto de Informatización de la Prensa

Autores: Dionisio Estrada Marchena y José Ramón Fernández Pérez

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes: - Independencia, - Originalidad, - Creatividad, - Laboriosidad, - Responsabilidad>

<Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de <nota>. <Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

Lic. Manuel Vázquez Acosta

Firma

Fecha

Agradecimientos

A nuestro tutor, Manuel Vázquez por la importante ayuda concedida y la dedicación que nos brindó, con el empeño de realizar este trabajo junto a nosotros.

Al profesor José Ramón Ruiz De Zárate por su ayuda en las consultas de Probabilidades.

A todos los estudiantes y profesores que nos ayudaron de una forma u otra y que condicionaron con su apoyo el resultado satisfactorio.

A nuestros amigos, que nos ayudaron tanto académica como emocionalmente.

Dedicatoria

“El esfuerzo es la clave del éxito”.

“Sólo la propia y personal experiencia hace al hombre sabio.”

Sigmund Freud

Dedico este trabajo a:

*A mis familiares y amigos que me apoyaron constantemente en este empeño
de graduarme.*

Dionisio Estrada Marchena

A la memoria de mi padre,

A mi madre Esther, a mi esposa Yarisbel y mi hijo Joseito.

A mis familiares y mis grandes amigos.

José Ramón Fernández Pérez

Resumen

Como consecuencia de la ausencia de un sistema de gestión de información dentro de los Centros de Prensa Nacional, la información puede aparecer duplicada y sin una correcta organización. A esta situación se le adiciona la ya común desconexión existente entre los artículos que permanecen guardados y que fueron publicados en algún momento con los que se publican en la actualidad. Informaciones relacionadas, con el mismo contenido, aparecen constantemente, sin embargo no aparecen vínculos a las anteriores publicaciones.

Esta desconexión entre publicaciones periodísticas condujo a la realización de esta tesis. El trabajo de diploma lleva como título: “Agente de recuperación de información Just-in-time(JITIR Agent) para el Proyecto de Informatización de la Prensa”, con el cual se pretende relacionar en un grado mucho mayor al existente, las noticias que se publican actualmente, tanto en los diarios digitales nacionales como internacionales.

La tesis estuvo basada en objetivos concretos: el estudio de la existencia de agentes recuperadores de información en las páginas Web, y más específicamente en la edición de las páginas Web en un CMS como Plone, la creación de un nuevo agente recuperador de información para este tipo de publicaciones Web, y luego de su implementación, la introducción del agente en los centros de prensa.

Para obtener un resultado tangible sobre la utilidad del nuevo agente JITIR, los autores de esta tesis aplicamos una encuesta a los usuarios finales que trabajan en algunos centros periodísticos. Un resultado positivo fue la consideración del agente como útil y fácil de emplear por parte de los periodistas, destacando la importancia que posee un recurso como este para actualizar y mejorar la actualización del lector.

Índice

| | |
|--|-----|
| LA HABANA | 1 |
| JUNIO DE 2007..... | 1 |
| DECLARACIÓN DE AUTORÍA..... | I |
| OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA..... | II |
| OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA | III |
| AGRADECIMIENTOS..... | IV |
| DEDICATORIA..... | V |
| RESUMEN | VI |
| ÍNDICE..... | VII |
| ÍNDICE DE TABLAS | X |
| ÍNDICE DE TABLAS | X |
| ÍNDICE DE FIGURAS | XI |
| ÍNDICE DE FIGURAS | XI |
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1 FUNDAMENTO TEÓRICO Y TRABAJOS RELACIONADOS ..8 | |
| INTRODUCCIÓN | 8 |
| 1.1 JUST-IN-TIME INFORMATION RETRIEVAL | 8 |
| 1.2 RECUPERACIÓN DE INFORMACIÓN EN LA WEB..... | 11 |
| 1.3 LOS AGENTES JITIR Y LA LEY DEL MÍNIMO ESFUERZO DE ZIPF | 12 |
| 1.4 MODELOS DE RECUPERACIÓN DE INFORMACIÓN | 14 |
| 1.5 LOS AGENTES JITIR EN LA WEB | 16 |
| 1.6 EVALUACIÓN DE LOS SISTEMAS RECUPERACIÓN DE INFORMACIÓN Y DE LOS AGENTES JITIR | 17 |
| 1.7 INTERFACES DE USUARIO PARA LOS AGENTES JITIR..... | 18 |
| 1.7.1 <i>Interfaz gradual</i> | 20 |
| 1.8 PLONE Y ZOPE..... | 21 |
| CONCLUSIONES | 22 |
| CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA.....23 | |
| INTRODUCCIÓN | 23 |
| 2.1 AJAX | 23 |
| 2.2 PLORA: ESTRUCTURA Y FUNCIONALIDAD..... | 25 |
| 2.3 INGENIERÍA INVERSA RELACIONADA CON ZOPE-PLONE..... | 28 |
| Diagrama de clases del diseño para Zope-Plone | 32 |
| Descripción de las clases del diseño | 32 |
| 2.4 RUP COMO METODOLOGÍA DE DESARROLLO PARA PLORA | 35 |

| | | |
|-------------------|---|-----------|
| 2.4.1 | <i>Objeto de automatización</i> | 35 |
| 2.4.2 | <i>Objetivos estratégicos del sistema</i> | 36 |
| 2.4.3 | <i>Propuesta del sistema</i> | 37 |
| 2.4.3.1 | Modelo de Dominio | 37 |
| | Representación del Modelo de Dominio..... | 38 |
| | Entidades y conceptos principales | 38 |
| 2.4.3.2 | Requerimientos de la aplicación..... | 40 |
| | Requisitos Funcionales: | 40 |
| | Requisitos No Funcionales: | 40 |
| | Definición de los actores..... | 42 |
| | Listado de Casos de Uso | 42 |
| | Diagrama de Casos de Uso del Sistema..... | 44 |
| | Descripción de los Casos de Uso del Sistema..... | 45 |
| | CONCLUSIONES | 51 |
| CAPÍTULO 3 | ANÁLISIS Y DISEÑO DEL SISTEMA | 52 |
| | INTRODUCCIÓN | 52 |
| 3.1 | ANÁLISIS..... | 52 |
| | Clases interfaz..... | 52 |
| | Diagrama de clases del análisis..... | 53 |
| | Diagrama de colaboración | 53 |
| | Flujo de sucesos-análisis para el diagrama de colaboración | 54 |
| | Requisitos especiales | 55 |
| 3.2 | DISEÑO | 55 |
| | Diagrama de clases del diseño | 56 |
| | Diagrama de clases del diseño(2)..... | 57 |
| | Descripciones de las clases del diseño | 57 |
| | Tratamiento de errores | 72 |
| | Interfaz..... | 72 |
| | Ayuda | 73 |
| | CONCLUSIONES | 73 |
| CAPÍTULO 4 | IMPLEMENTACIÓN Y PRUEBAS..... | 74 |
| | INTRODUCCIÓN | 74 |
| 4.1 | IMPLEMENTACIÓN..... | 74 |
| | Diagrama de componentes..... | 74 |
| | Diagrama de despliegue..... | 76 |
| 4.2 | PRUEBA..... | 77 |
| | Descripción de los casos de prueba para el CU: Escribir_texto | 77 |
| | Descripción de los casos de prueba para el CU: Procesar consultas | 78 |

Introducción

| | |
|--|-----------|
| Descripción de los casos de prueba para el CU: Insertar vínculo..... | 79 |
| 4.3 RESULTADOS DE LA APLICACIÓN | 79 |
| 4.3.1 Impacto del agente PloRA en el Proyecto de Prensa | 79 |
| CONCLUSIONES | 80 |
| CONCLUSIONES..... | 81 |
| RECOMENDACIONES..... | 83 |
| BIBLIOGRAFÍAS CITADAS..... | 84 |
| BIBLIOGRAFÍAS CONSULTADAS..... | 86 |
| GLOSARIO DE TÉRMINOS | 89 |

Índice de Tablas

| | |
|--|----|
| TABLA NO. 1: CLASE DE DISEÑO: KUPUEEDITOR | 33 |
| TABLA NO. 2: CLASE DE DISEÑO: LINKTOOL..... | 34 |
| TABLA NO. 3: CLASE DE DISEÑO: PORTAL_CATALOG | 34 |
| TABLA NO. 4: CLASE DE DISEÑO: EDITAR_TEXTO | 35 |
| TABLA NO. 5: ENTIDADES Y CONCEPTOS PRINCIPALES | 40 |
| TABLA NO. 6: ACTORES DEL SISTEMA | 42 |
| TABLA NO. 7: CU ESCRIBIR TEXTO | 42 |
| TABLA NO. 8: CU PROCESAR CONSULTAS | 43 |
| TABLA NO. 9: CU INSERTAR VÍNCULO..... | 44 |
| TABLA NO. 10: DESCRIPCIÓN DEL CU: ESCRIBIR TEXTO | 46 |
| TABLA NO. 11: DESCRIPCIÓN DEL CU: PROCESAR CONSULTAS..... | 49 |
| TABLA NO. 12: DESCRIPCIÓN DEL CU: INSERTAR VÍNCULO | 50 |
| TABLA NO. 13: CLASE PLORAENGINE | 59 |
| TABLA NO. 14: CLASE PLORABACKEND | 62 |
| TABLA NO. 15: CLASE PLORAQUERYSET..... | 64 |
| TABLA NO. 16: CLASE PLORACACHE | 66 |
| TABLA NO. 17: CLASE PLORAUI | 67 |
| TABLA NO. 18: CLASE EDITORADAPTER | 69 |
| TABLA NO. 19: CLASE QUERYRESPONSE | 70 |
| TABLA NO. 20: CLASE PLORATOOL | 71 |
| TABLA NO. 21: CASO DE PRUEBA PARA EL CU: ESCRIBIR_TEXTO | 78 |
| TABLA NO. 22: CASO DE PRUEBA PARA EL CU: PROCESAR CONSULTAS..... | 78 |
| TABLA NO. 23: CASOS DE PRUEBA PARA EL CU: INSERTAR VÍNCULO..... | 79 |

Índice de Figuras

| | |
|--|----|
| FIGURA NO. 1: DIAGRAMA DE CLASES DEL DISEÑO | 32 |
| FIGURA NO. 2: MODELO DE DOMINIO..... | 38 |
| FIGURA NO. 3: DIAGRAMA DE CUS | 44 |
| FIGURA NO. 4: DIAGRAMA DE CLASES DEL ANÁLISIS..... | 53 |
| FIGURA NO. 5: DIAGRAMA DE COLABORACIÓN | 53 |
| FIGURA NO. 6: DIAGRAMA DE CLASES DEL DISEÑO | 56 |
| FIGURA NO. 7: DIAGRAMA DE CLASES DEL DISEÑO(2) | 57 |
| FIGURA NO. 8: DIAGRAMA DE COMPONENTES | 76 |
| FIGURA NO. 9: DIAGRAMA DE DESPLIEGUE | 77 |
| FIGURA NO. 10: PRIMERA ENCUESTA A LOS PERIODISTAS DEL CIP..... | 90 |

Introducción

Situación del Sistema de Prensa Cubano

En la prensa nacional, los recursos informáticos forman actualmente una parte importantísima en la divulgación de noticias. Proyectos que están trabajando junto con la Batalla de Ideas, la divulgación de la verdad sobre Cuba, publicaciones provinciales y nacionales, entre otros, tienen la necesidad de estar en constante acción en la redacción, edición y publicación de argumentos que ayuden a mantener informados a los ciudadanos y el empleo de estos recursos hace posible la realización de todas las actividades digitales propias de un centro de redacción.

El atraso tecnológico que se manifiesta en los recursos para la publicación, o sea, la actualización de las páginas Web mediante transferencia FTP, la ausencia de un sistema de gestión que permita automatizar la información que se edita, y la serie de dificultades que estas carencias traen como consecuencia, son obstáculos que impiden que nuestros centros digitales hagan un mejor trabajo.

Herramientas como FrontPage y Dreamweaver son utilizadas constantemente para el diseño de los portales Web, pero además, para la gestión de la información que se publica en los sitios de los centros de la prensa nacional. Sin embargo, ninguna de ellas está destinada para realizar esta última función. Para realizar esta operación se deben emplear los sistemas de gestión de contenidos (CMS, siglas en inglés). La gestión de contenidos es el principal obstáculo que necesita ser resuelto dentro de la prensa cubana para garantizar la inmediatez y la publicación de las noticias.

Procesos como la edición, revisión y publicación, se deben hacer con rapidez porque el flujo de información que existe diariamente es amplio. Las noticias no deben perder su instantaneidad, para que

llegue el mensaje preciso que se quiere transmitir, basándose en mecanismos automatizados, sin perder de vista los estilos de los sitios que están “de cara a Internet”. Mientras no se garantice el control y la organización de la información que fluye dentro de las redacciones de los centros de prensa no estará disponible la información inmediata en los sitios nacionales de prensa.

Situación de las publicaciones en línea del Sistema de Prensa Cubano

Otra consecuencia directa del bajo nivel de automatización de nuestros medios de Prensa, detectada como problema fundamental a tratar por este trabajo de diploma, es lo que llamamos “información desconectada”. En muchas de las publicaciones periodísticas nacionales que tenemos actualmente en la red, no es común encontrar referencias entre artículos que están relacionados y que han sido publicados, permaneciendo almacenados en bases de datos sin ser utilizados apenas, incluso entre artículos realizados por un mismo periodista.

Vincular estos documentos brindaría la posibilidad de acceder a información útil, ampliaría las dimensiones de los artículos que se publican y unificaría mejor los contenidos publicados. En ocasiones, mientras se escribe un documento se hacen pausas para leer otro y es probable que se copien fragmentos como citas o se incluyan referencias. En el trabajo de Brown[1] se trata el tema de reading-while-writing, o sea, la inclusión de referencias mientras se escribe, y especifica la utilidad del Remembrance Agent[2] como artefacto que facilita la obtención de información almacenada en algún tipo de gestor de datos, carpeta personal o e-mail y ayuda además de encontrar esa información almacenada, a enlazarla para enriquecer el contenido del artículo que se va a publicar.

No obstante, muchos medios de prensa no cuentan con un sistema de gestión de contenidos, ni sistemas de recuperación de información[3] asociados, por lo que se hace muy difícil encontrar la información que se puede enlazar.

Método de investigación

Esta situación en las publicaciones nacionales fue identificada a partir del seguimiento realizado primeramente por nuestro tutor, y luego por nuestra revisión, visitando algunos sitios de importancia como: www.Granma.co.cu, www.juventudrebelde.cu, www.vanguardia.co.cu, entre otros. Al visitar estos sitios y otros más, observamos que no es común encontrar vínculos entre la información publicada y otros muchos artículos que tienen relación con dicho tema y que se presentaron en publicaciones anteriores. Estos artículos anteriores pueden servir de apoyo a la información que se muestra cada día a los lectores. Sin embargo, la ausencia de estos vínculos hace que las informaciones se encuentren en total desconexión y que no cuenten, de manera directa, con todos los argumentos que pueden estar relacionados con esta.

Just-in-time Information Retrieval (JITIR)

Incluso teniendo un sistema de gestión de contenidos, puede resultar tedioso realizar las búsquedas necesarias para encontrar los materiales relacionados. El usuario en este caso, se ve en la obligación de dejar de escribir y utilizar un buscador para obtener la información que le sea útil en su artículo.

Una alternativa a la búsqueda explícita o intencional es la utilización de Just-In-Time Information Retrieval agents (JITIR agents)[2]. Los agentes JITIR son programas de software que recuperan y presentan información de manera proactiva, basándose en el contexto local del usuario. Brindan información sin que la hayan solicitado y que esta

pueda serle útil al periodista. Los datos que el agente recupera para el usuario provienen de bases de datos de documentos (e-mail, notas, sistemas gestores de contenidos, etc.) preindexados[4].

En el documento original se describen tres ejemplos de estos agentes:

1. Remembrance Agent: agente integrado al editor Emacs. Presenta una lista de documentos que están relacionados con el documento actual que se escribe o lee.
2. Margin Notes: direcciona automáticamente páginas Web cuando se cargan. Añade una interfaz o margen dentro de la página, donde el usuario puede ver las diferentes sugerencias que se presentan a medida que va leyendo.
3. Jimminy: proporciona información basada en el entorno físico de una persona: su localización, la fecha, hora, etc.

Los agentes JITIR en el campo de la documentación, y más específicamente en el caso de la prensa cubana en la edición de noticias, se encargarían de recuperar información y proponer enlaces a documentos que podrían ser de interés para los usuarios (periodistas) y hacer vínculos a otros documentos que se han publicado en ediciones anteriores. De esta manera iría disminuyendo la “desconexión” que existe entre las múltiples publicaciones que se tienen en los centros de prensa cubanos.

El Proyecto de Informatización de la Prensa

Actualmente se está ejecutando en nuestra universidad el Proyecto de Informatización de la Prensa. Este Proyecto fomentará la integración de todos los medios de prensa cubanos en una red de información nacional, en la cual los sistemas de recuperación de

información jugarán un papel esencial.

Una de las tareas principales del Proyecto es la creación de herramientas de gestión y recuperación apropiadas para los medios de prensa. Por esta razón, esta tesis se desarrolla como parte de este Proyecto.

Tecnologías aplicadas en el Proyecto de Informatización de la Prensa

La utilización de Sistemas de Gestión de Contenidos tiene un papel relevante en el desarrollo de sitios Web. Facilitan la publicación y el manejo de los contenidos que se emplean en estos sitios. Estos contenidos publicados se pueden centralizar y así evitar que esté dispersa y duplicada. De esta manera, se puede compartir y reutilizar la documentación que se guarda. Plone[5] es el CMS que se promueve en el Proyecto de Prensa Cubana con este fin. Está basado en Zope, cuya implementación es en el lenguaje de programación Python, y establece además un sistema de publicación de documentos y una herramienta de trabajo en grupo.

Además de las tecnologías antes presentadas, debemos mencionar al editor de texto Kupu[6] que es el producto que utilizamos en el trabajo con estas tecnologías y que se va a integrar a nuestro JITIR y además es un editor de texto WYSIWYG, de código abierto.

Objetivo general del trabajo de diploma

El desarrollo de un agente JITIR para el Proyecto de Informatización de la Prensa.

Objetivos específicos

- ✓ Construir el marco teórico de la utilización de los agentes JITIR a partir del estudio del estado del arte.

Introducción

- ✓ Implementar un agente JITIR para Plone que se integre con el componente de edición Kupu.
- ✓ Realizar un experimento para evaluar la introducción del agente en los centros de prensa.

Tareas investigativas

- ✓ Estudiar en el marco internacional sobre la existencia de agentes recuperadores de información dirigidos a la Web.
- ✓ Investigar sobre el diseño y funcionamiento del CMS Plone y de Zope para realizar la ingeniería inversa correspondiente.
- ✓ Obtener a través de encuestas a los usuarios finales un determinado resultado, que posibilite próximos trabajos.

Posibles resultados

Se espera que:

La utilización continuada de este tipo de agentes promueva la creación de enlaces en nuestros periódicos en línea, y de esta forma se comparta más y mucho mejor la información dentro del Sistema de Prensa Cubano.

Estructuración del contenido

La tesis está compuesta por cuatro capítulos y las conclusiones generales, además de las secciones correspondientes a las recomendaciones, la bibliografía y las referencias bibliográficas. En el capítulo uno se amplía los temas tratados en esta introducción. Las descripciones de los agentes JITIR precedentes constituyen el punto de partida para las valoraciones necesarias a la hora de construir un nuevo agente recuperador de información. Se abordan temas como los métodos para evaluar la recuperación de información, temática

importante, que ayuda a reconocer cuál de los métodos se debe introducir en el agente JITIR a partir de sus funcionalidades.

La existencia de productos JITIR en las páginas Web también conforman un epígrafe, analizando así los antecedentes de nuestro futuro agente JITIR y poder valorar antes de construirlo si realmente es útil su creación.

El capítulo dos está orientado a documentar la propuesta de solución al problema de desconexión de las noticias. Se inicia explicando la estructura y funcionalidad del nuevo agente con el nombre de PloRA. Luego se explica la base que brinda la realización de una Ingeniería inversa sobre Zope y Plone, los cuales sostienen el nuevo producto para Plone. La representación de esta ingeniería se realiza utilizando diagramas de clases del diseño, utilizando UML como lenguaje de modelado. El próximo epígrafe desarrolla la Ingeniería directa que modela la evolución del PloRA. Este capítulo concluye con la definición del modelo de dominio que representa los conceptos involucrados en la implementación del agente JITIR.

El capítulo tres continúa con el análisis y el diseño de las clases del PloRA. Se presentan tablas descriptivas y diagramas de UML. Estos flujos de trabajo serán la base para la implementación correspondiente.

El capítulo cuatro abarca toda la implementación y las pruebas que lleva un nuevo producto de software. En la parte de implementación está representado el diagrama de los componentes que integran el agente. En la parte correspondiente a las pruebas aparecen algunas pruebas que se le aplicarán al software después de su construcción.

Capítulo 1 Fundamento teórico y trabajos relacionados

Introducción

Este capítulo trata temas referentes al estado del arte del problema planteado en la introducción (“información desconectada en las publicaciones de la prensa”) a nivel nacional. También, se abordarán temáticas sobre las tecnologías que existen en el plano internacional y que pueden solucionar dicho problema con el estudio de técnicas de recuperación de información. La integración de todas las tecnologías consultadas identifica una vía para solucionar esta dificultad existente: estudiar el uso de un agente de recuperación de información.

1.1 Just-in-time Information Retrieval

Desde principios de la década del 90 del siglo pasado, han existido diversos planteamientos que intentan definir fórmulas para ayudar a la obtención de información importante para los usuarios de la Web. Las búsquedas en la Web son realizadas con muchas finalidades: pueden ser culturales, políticas, económicas, etc., pero todas coinciden en un punto: la necesidad de encontrar información acerca de cierto tema. En estos casos, siempre es necesario emplear un artefacto que haga sugerencias de posibles documentos que el usuario puede utilizar y que estén relacionados con la temática de la actividad que esté realizando (escribiendo o leyendo) y además que sean relevantes.

Cuando una persona necesita datos acerca de un tema que está estudiando, y no los conoce entonces abandona su estudio, deja la lectura o la escritura que está realizando para buscar esa información que necesita para argumentar y fomentar sus ideas sobre dicho tema. Sin embargo, el uso de un agente de recuperación de información Just-in-time (JITIR) puede hacer de las búsquedas una tarea secundaria y así el usuario no desvía su atención de la actividad que está

Capítulo 1: Fundamentación teórica

realizando.

Los agentes JITIR son aplicaciones que pro-activamente recuperan y presentan información basada en un contexto local del usuario. En [2] se enumeran las tres características básicas de un agente JITIR:

- ✓ Proactividad: el usuario no tiene que iniciar el proceso de búsqueda, el agente está haciendo búsquedas y muestra los resultados y el usuario decide si le interesa o no.
- ✓ La información debe ser presentada de manera accesible pero no intrusiva, de manera que pueda ser ignorada por el usuario si esta no le interesa.
- ✓ La información que se presenta debe referirse a eventos que tengan lugar en el contexto local del usuario.

Los agentes JITIR tienen similitud con los motores de búsqueda en cuanto a recuperación de información, pero tienen diferencias ya que “(...) no exigen ningún tipo de acción por parte del usuario, sino que las acciones de éste sobre el entorno computacional (e-mail, página Web que esté leyendo o documento que esté escribiendo), activan la búsqueda de información que puede ser de utilidad para el usuario.” [2]

Estas tres características hacen de los agentes JITIR artefactos similares a las alarmas, aparatos de búsquedas y sistemas personalizados de ayuda, sensores de cámaras, etc. Pero a estas características se suman otros beneficios: los agentes JITIR reducen el esfuerzo mental requerido para encontrar, evaluar y acceder a la información que se necesita para incrementar el sentido de una información. Quiere decir esto que, cuando el usuario deja de escribir o leer (que sería su tarea primaria), para buscar información necesaria y completar un documento, se está produciendo un cambio en la tarea

Capítulo 1: Fundamentación teórica

principal, se abandona la idea que se lleva en el trabajo y se comienza con un proceso mental[7] de evaluar toda la información que se está buscando para encontrar la más adecuada, que tenga relación con lo que pasó a ser una tarea secundaria, la redacción o lectura iniciales. Los agentes JITIR se encargan de mantener al usuario concentrado en su tarea primaria y le ofrece a este la información que en determinado momento puede incluir a su conocimiento.

Los agentes JITIR conocidos son Remembrance Agent (RA), Margin Notes y Jimminy. Los tres fueron creados en el MIT (Massachusetts Institute of Technology) como parte de un proyecto de desarrollo. RA se integra al editor Emacs, y presenta constantemente una lista de documentos que están relacionados con el mismo material que se está leyendo o escribiendo. Margin Notes, anota automáticamente las páginas Web añadiendo hipervínculos a las informaciones relevantes según contexto de la página. Y Jimminy presenta la información en un dispositivo portátil, y se basa en el entorno físico de la persona: el lugar donde se encuentra, la hora del día, tema de conversación, etc., para realizar la búsqueda.

Estos tres agentes de recuperación de información, a pesar de tener sus características particulares (en especial la definición de contexto local es diferente en cada caso), poseen el mismo sistema de recuperación de información: Savant.

Savant consiste en dos programas fundamentales: ra-retrieve y ra-index. El programa ra-retrieve realiza la recuperación de la información basada en una consulta, y ra-index crea el índice de los archivos donde se realiza la búsqueda. En estos índices se pueden incluir recursos como colecciones de artículos o periódicos, notas, e-mails, etc. [2]. Savant actúa como un artefacto de recuperación de información que realiza sus acciones por “detrás” del agente JITIR: dada una consulta

Capítulo 1: Fundamentación teórica

se crea una lista de documentos que tienen relación con ella.

Los agentes JITIR proporcionan facilidades a la hora de encontrar contenidos de interés para los usuarios en cuanto a tiempo de devolución de la información, que se reduce cuando la búsqueda pasa de ser manual a un proceso automatizado, dado que automáticamente la velocidad de procesamiento de información es mayor, se obtiene el contenido evaluado rápidamente y el usuario ahorra tiempo y esfuerzo, y fundamentalmente no se desvincula de su actividad primaria totalmente.

1.2 Recuperación de Información en la Web

La última década del siglo pasado constituyó la fase esencial en el crecimiento y desarrollo de la World Wide Web. Fue creada para facilitar el intercambio de documentos entre investigadores de centros de desarrollo[8] pero actualmente las páginas Web forman parte importante en la publicación de información, y ya se han convertido en un enorme repositorio en la red.

La facilidad con que se publica un conocimiento en la Web tiene ventajas, pero también dificultades. Diariamente se publican nuevos artículos, noticias, páginas Web, etc., que pueden contener información importante acerca de un tema determinado, siguiendo la tendencia de digitalizar la información que circula por el mundo y todo esto gracias a esa facilidad que brinda Internet. Pero también, todo este cúmulo de datos provoca que el usuario encuentre información que no es importante o que no es confiable y así la recuperación de información pierde su principal objetivo: devolver contenido relevante para el usuario.

Los buscadores de la Web basan su funcionamiento en la comparación de palabras clave. Muchas veces sucede que los

Capítulo 1: Fundamentación teórica

usuarios no hacen formulaciones correctas de lo que desean encontrar. Generalmente se comete el error de utilizar palabras cortas, ambiguas o con una escasa representación de lo que se pretende obtener, utilizando un proceso reactivo que depende de las acciones directas del usuario, que se encarga de construir la consulta y escribirla en el buscador de la Web.

El proceso que se realiza en los agentes JITIR los hace diferentes de los buscadores Web. La proactividad es un elemento fundamental de esa diferencia y radica en la construcción automática de las consultas por parte del agente JITIR teniendo en cuenta el contexto local del usuario. Este proceso disminuye el esfuerzo utilizado por el usuario para obtener información ya que no se desvincula de la tarea primaria para conformar las consultas correctas para obtener información relevante.

1.3 Los agentes JITIR y la Ley del Mínimo Esfuerzo de Zipf

En la vida diaria, las personas siempre valoran las diversas opciones que se les presentan para resolver una situación determinada, y realizarla con el menor esfuerzo posible. Esta regla es conocida como la Ley del Mínimo Esfuerzo, la cual expresa que una persona trata de minimizar su trabajo físico o mental y así optimizar el esfuerzo en la actividad que va a realizar.[9]

La utilización de un agente JITIR tiene como objetivo reducir el esfuerzo mental que requiere un usuario para encontrar, evaluar y acceder a la información que necesita para argumentar un planteamiento, al realizarse la mayor parte de las búsquedas automáticamente. Por esta razón, en términos de esfuerzo, los agentes JITIR son herramientas que recuperan información de manera menos

Capítulo 1: Fundamentación teórica

costosa que otras. Hay que notar que aún cuando un agente JITIR reduce los costos de acceder y evaluar información, no los elimina completamente. Existe un pequeño esfuerzo que no deja de ser importante: revisar las sugerencias que brinda el JITIR y evaluar si el resultado es útil o no.

La información que devuelve un JITIR no siempre tiene la utilidad esperada para el usuario. Existen algunas categorías para valorar la utilidad de los resultados:

- ✓ Falsa positiva (inútil): en este caso la información presentada no es útil. Puede estar dada por deficiencias en el sistema de búsquedas o baja calidad en la información.
- ✓ Falsa positiva (conocida): aquí la información es útil, pero ya es conocida y está siendo usada u obviada por el usuario.
- ✓ Costo disminuido: el usuario conoce que existe esa información, pero no cree que valga la pena realizar la búsqueda. Al presentar la información directamente se disminuye el costo de acceder a esta y por tanto se espera mayor beneficio para el usuario de acuerdo al resultado obtenido.
- ✓ Incremento del beneficio esperado: la información no es útil, pero indica la existencia de otra que pueden ser de valor. Aquí el JITIR no disminuye el costo de las acciones para encontrar alguna información pero incrementa el beneficio esperado, que se fomenta con la obtención de otro contenido importante para otra búsqueda.
- ✓ Costo disminuido e incremento del beneficio esperado: la información que se proporciona es desconocida y útil. El JITIR facilita el acceso al documento completo disminuyendo el costo

Capítulo 1: Fundamentación teórica

de accesibilidad a este e incrementa el beneficio esperado de recuperar dicho documento.

Estas evaluaciones de la información devuelta por un agente JITIR permiten identificar desde los resultados relevantes para el usuario en la tarea que esté realizando, hasta aquellos que carecen de importancia y que no tienen ninguna relación con los resultados que se esperan.

Debemos tener en cuenta que todos los resultados que devuelva un JITIR estarán condicionados por el(los) modelo(s) de recuperación de información que se incluyan en el agente JITIR.

1.4 Modelos de recuperación de información

La recuperación de información (information retrieval, IR, siglas en inglés) trata acerca de obtener el contenido de la información que es relevante para las necesidades de conocimiento del usuario. Puede estar basada en recuperación de textos, imágenes, audio u otro tipo de información no textual.

Existen muchos métodos diferentes para recuperar información. Estos métodos hacen posible que se obtengan los documentos con mayor proximidad a la consulta que el usuario expresa con el contenido que desea encontrar.

- a) Modelo Espacio Vectorial: representa las consultas y los documentos como vectores, donde al indizar términos son considerados como coordenadas de un espacio de información multidimensional. Estos términos pueden ser palabras de los documentos representados en el espacio de información. La importancia del documento se representa por la distancia entre un vector consulta y un vector documento dentro de este

Capítulo 1: Fundamentación teórica

espacio vectorial.

- b) **Modelo Probabilístico:** muestra la recuperación de la información como el esfuerzo de ordenar los documentos por la probabilidad para que, dada una consulta, dicho documento sea útil. Y apoyándose en la retroalimentación, este modelo lista documentos que ya han sido señalados como relevantes o no relevantes a dicha consulta.
- c) **Acercamiento a Bases de Conocimientos:** muchas veces el conocimiento acerca de un tema particular puede usarse como consulta. Aquí se puede esperar un buen resultado a partir del conocimiento en una base de datos para emitir un criterio específico.
- d) **Data Fusion:** es una técnica con varios algoritmos, métodos de indizado y de búsqueda que son utilizados para producir juegos de documentos importantes. Estos resultados se combinan y se obtienen los documentos útiles (el sistema Savant es un ejemplo de sistema de recuperación de información Data Fusion).

El modelo de espacio vectorial[10] es implementado en la mayoría de los motores de búsqueda de la Web. Sus respuestas están en correspondencia a la coincidencia entre los términos de las consultas que hacen los usuarios y los documentos que se analizan en las bases de datos. Este método al utilizarse en un JITIR posibilita que se obtengan artículos que estén relacionados con el ambiente donde esté trabajando el usuario, de manera proactiva, y dichos archivos se representen en un espacio multidimensional de información.

El método de espacio vectorial se emplea en las búsquedas que se realizan en Plone para obtener los objetos que se guardan en las

bases de datos de Zope.

1.5 Los agentes JITIR en la Web

Margin Notes es un agente de recuperación de información Just-in-time utilizado en la Web. Este agente muestra automáticamente, en una interfaz en un lateral de las páginas Web que el usuario visita, un listado de páginas Web que se relacionan con el contexto que el usuario está visualizando o leyendo en la Web visitada cuando esta es cargada. Su utilidad se limita a listar un grupo de sugerencias de documentos de otras páginas relacionadas con el contexto que se está analizando. Sin embargo, estas características no expanden su utilidad a los usuarios que editan las páginas Web. O sea, es un agente JITIR que trabaja en páginas Web pero que no funciona a la hora de editar una Web, y ni siquiera utilizando un CMS como Plone para gestionar el contenido que se publican donde los documentos están guardados en el mismo servidor sobre el cual funciona el Plone. Quiere esto decir que cuando un usuario, fundamentalmente periodistas, redactores de noticias, etc., escribe un artículo la mayoría de las veces necesita consultar documentos que ya se han publicado. Pero como no cuenta con una herramienta que le busque esa información, en el mejor de los casos en su propio ambiente PC; aunque podría ser en un cúmulo de informaciones dentro de una base de datos, entonces se ve obligado a dejar la redacción de su noticia para buscar la información que necesita.

Para mayor comodidad en las actividades de los redactores, sería de gran ayuda un artefacto que haga el trabajo de búsqueda mientras ellos se concentran en la escritura de sus futuras publicaciones. Este agente sería capaz de recuperar y mostrar el contenido relacionado con lo escrito hasta el momento por el usuario y con cierto valor de relevancia sugerido por el propio agente recuperador. Esta última

Capítulo 1: Fundamentación teórica

característica le brinda la posibilidad al cliente de hacer valoraciones en el momento de vincular el documento redactado con la(as) publicación(es) anterior(es) que es(son) sugerida(s) en la interfaz del agente JITIR, o simplemente desecharlas por no interesarle el resultado obtenido de la búsqueda realizada.

1.6 Evaluación de los sistemas recuperación de información y de los agentes JITIR

Los métodos de recuperación de información son evaluados en términos de relevancia dada una consulta. La relevancia de un documento con respecto a una consulta es el grado de importancia que posee dicho documento. Este grado de importancia se calcula utilizando métodos matemáticos como el modelo de espacio vectorial, que en el caso de Zope, devuelve los resultados más importantes identificados por un valor numérico asignado al resultado de una búsqueda en una base de datos, donde los documentos devueltos responden con las necesidades de información del usuario que hizo la consulta en un nivel muy exacto.

Dentro de los sistemas de recuperación de información se manejan dos características fundamentales que definen el resultado que se obtiene de las búsquedas de información importante para los usuarios. Estos términos son: precision y recall. Recall es la medida de que todos los documentos relevantes a la consulta sean sugeridos al usuario. En este caso se muestran todos los documentos pero sin especificar cuales sugerencias tienen mayor relevancia, esto traería como consecuencia que el usuario desvíe su atención de la tarea que está realizando para distinguir aquellos documentos que tengan la importancia requerida para ser enlazados con el contexto, entre todos los que fueron devueltos por el agente JITIR.

Capítulo 1: Fundamentación teórica

Sin embargo, precisión asegura que los documentos devueltos sean los que mayor relevancia tengan de acuerdo a la consulta obtenida del entorno de trabajo del usuario. Esta condición hace posible que el usuario mantenga su atención en la redacción del artículo. Sólo cambiará la tarea que está realizando en el momento que haga una valoración de las sugerencias que se muestran en la interfaz del agente, para seleccionar cual de ellas vinculará con el cuerpo del documento que está redactando. Es importante señalar que este desvío de atención que realiza el usuario es necesario, pues en ese instante él necesita escoger la sugerencia que más relevancia él considere que posee.

Los agentes JITIR normalmente devuelven pocas sugerencias, y estas con los mayores grados de relevancia. Si se ofrecen muchos documentos como sugerencias, conduce a demasiada distracción para el trabajo del usuario. Esta es la razón fundamental por la que se escogen sistemas con mayor precisión que recall en la implementación de los agentes JITIR.

1.7 Interfaces de usuario para los agentes JITIR

En el diseño de interfaces que componen a un agente JITIR es necesario valorar una característica propia de estos artefactos. La no intrusividad que debe cumplir un agente recuperador de información es esencial para que el usuario mantenga su atención en su tarea primaria y no se distraiga con mensajes innecesarios o mucha información que tenga que analizar para incluir lo necesario en el artículo que esté redactando.

Los agentes JITIR funcionan en conjunto con ambientes de trabajo de los cuales obtienen las consultas necesarias para sugerir documentos que puedan argumentar un análisis que se realice acerca

Capítulo 1: Fundamentación teórica

de un tema determinado. Estas sugerencias que brinda el JITIR y que son mostradas en la interfaz predeterminada de este artefacto, constituyen la tarea secundaria que el usuario tiene disponible para valorar la información brindada. Si la interfaz de usuario no es discreta y cada vez que se genera una sugerencia produce acciones que lo distraigan, se corre el riesgo de que éste pierda concentración de manera involuntaria de lo que está realizando y comience a analizar innecesariamente las sugerencias, dejando su trabajo a un lado.

Este cambio de actividad no debe producirse como consecuencia de acciones producidas por la interfaz del agente JITIR. Los cambios entre los tipos de actividades deben ocurrir de manera conciente por parte del usuario ya que la necesidad de incrementar el contenido de un documento siempre está presente, y por tanto, la interfaz será discreta pero no invisible, para que el usuario conozca que tiene un apoyo en el trabajo que está realizando.

Al contrario de las personas que utilizan los buscadores Web para encontrar información que les pueda ser útil, los usuarios de los agentes JITIR no buscan información de manera espontánea, el agente JITIR obtiene las consultas del entorno donde trabaja el usuario y genera sugerencias de documentos que le pueden interesar, de ahí es el carácter proactivo del agente.

Existen razones más importantes a considerar a la hora de crear interfaces para un agente JITIR. Puede que el texto sugerido se conozca, el usuario no desee ser distraído o simplemente no desee ninguna información nueva. Por esto, la interfaz debe ser no intrusiva, o sea, que no distraiga la atención del usuario, y así se logra que enfoque su atención en la tarea primaria sin atender a otros estímulos. Pero tampoco debe ser “invisible”, o sea, debe ser accesible para

Capítulo 1: Fundamentación teórica

alternar entre la tarea primaria y la información proporcionada.

En este caso el usuario manifiesta lo que se conoce como atención dividida: que es la habilidad de intercambiar entre la tarea principal (redacción o lectura de documentos) y la secundaria (valoración de los documentos devueltos en la interfaz del agente JITIR) sin mucho esfuerzo. Con esta condición se garantiza que el usuario utilice el JITIR para obtener información que tiene determinada relevancia con respecto a la tarea primaria que está realizando. Un diseño de interfaz del JITIR es apropiado cuando le permite al usuario atender su tarea principal y dividir su atención cuando lo desee. En el caso de los agentes JITIR, la atención dividida es aún más fácil por el hecho de que la información proporcionada está relacionada con el contexto actual.

1.7.1 Interfaz gradual

Sobre las sugerencias mostradas por un JITIR se pueden hacer tres afirmaciones: primero, nunca serán útiles el ciento por ciento del tiempo, incluso con la recuperación perfecta de datos. Hay momentos en que un usuario no desea más información, y se ve afectado por una sobrecarga de información y no desea seguir distrayéndose. En segundo lugar, el usuario puede determinar si una sugerencia le será útil, si asume que brinda alguna información sobre el contenido del contexto donde se encuentra. Y la tercera, la propia acción de determinar si una sugerencia es útil es una distracción y una carga mental para el usuario ya que se desvía de la tarea primaria para realizar esta valoración.

Teniendo en cuenta estas características es importante mostrar las sugerencias en una interfaz discreta que progresivamente actualice la información acerca del tema que el usuario esté tratando. La

Capítulo 1: Fundamentación teórica

información que se presenta en fases progresivas, o sea, que se incrementa de una fase a otra, se conoce como interfaz gradual. Cada fase de una interfaz gradual proporciona un poco más de información, con el único costo de requerir en las primeras fases menos atención que en las últimas para leer y entender la información ofrecida en la interfaz. La intención de una interfaz gradual con nuevas interfaces de usuario, es aumentarle a ésta el contenido informativo que mostró en la primera fase. Estas fases transportan dos tipos de información: contenido que pueden ser documentos, artículos, resúmenes, etc. e información sobre este contenido, que puede ser fechas relacionadas con los documentos, los títulos y autores de los artículos, descripciones y líneas resúmenes, entre otras informaciones. La técnica de interfaz gradual reduce el costo de los falsos positivos y reduce el esfuerzo requerido para recuperar la información sugerida, o sea, obtener el cuerpo íntegro del documento sugerido. Estas interfaces graduales son diseñadas para que los usuarios puedan decidir más fácilmente cual información es más útil para el trabajo que esté realizando.

1.8 Plone y Zope

Plone es un Sistema de Gestión de Contenidos o CMS, por sus siglas en inglés (Content Management System), de código abierto, basado en Zope y programado en Python. Puede utilizarse como servidor intranet o Internet, como un sistema de publicación de documentos y como una herramienta de trabajo en grupo para colaborar entre entidades distantes. Es un sistema donde los objetos tienen muchos atributos.

Zope es un servidor de aplicaciones orientado a objetos, programado en Python y de código abierto. Constituye un espacio de trabajo para la creación y publicación de aplicaciones Web. En Zope todos los elementos son objetos, desde las páginas que se crean hasta

Capítulo 1: Fundamentación teórica

una simple imagen, y todos se almacenan en una base de datos[11]. Esto es fundamental a la hora de permitir hacer búsquedas complejas y muy potentes de recursos dentro de una página Web en Zope. Al tener registrados todos los objetos en la base de datos, se simplifica mucho al realizar interfaces de búsquedas, donde las páginas HTML son ficheros almacenados en disco.

Conclusiones

Toda la teoría hasta aquí citada es la base para la construcción de una aplicación que puede brindar apoyo al trabajo que se realiza todos los días en los centros de prensa nacional. Un aporte al desarrollo de estos institutos, hacen de algunos de nuestros medios de información una fuente importante para conocer cuanta noticia sea editada.

Capítulo 2 Características del sistema

Introducción

La teoría relacionada con las generalidades para diseñar un agente JITIR fue el principal objetivo del capítulo anterior. En este capítulo, nos apoyaremos en dicha teoría, y con las condiciones que debe cumplir un sistema automatizado, construiremos un producto para vincularlo al CMS Plone.

Este producto debe trabajar en conjunto con el editor de texto kupu. De esta integración se obtiene el texto que escribe el usuario y que formará las consultas para realizar las búsquedas en el servidor de Zope a través del JITIR. Las sugerencias devueltas por el agente JITIR se producirán mediante el uso de tecnologías AJAX. En el servidor se realizan las búsquedas y son mostradas en la interfaz de usuario del agente recuperador de información. De esta forma el usuario puede determinar en que parte del documento quiere insertar un vínculo con otro artículo ya editado seleccionando la sugerencia que más útil le sea.

2.1 AJAX

El espacio de las 3W(World Wide Web) está encaminado a la realización de pedidos y respuestas, desde una página cliente a una servidora. Con el desarrollo alcanzado en la publicación de contenidos en la red y con el incremento del acceso a estos, también se han incrementado estas peticiones, y por supuesto, las respuestas. Por esta razón, los encargados de estos servicios, tratan de mantenerlos confiables, actualizados, para mejorar el acceso de los clientes. Pero también, se trata siempre de desarrollarlos para ampliar el flujo de datos que pueda existir entre las peticiones y las respuestas y optimizar este flujo.

Capítulo 2: Características del sistema

Para disminuir la sobrecarga de peticiones al servidor con datos innecesarios, que se repetían constantemente, se introdujo una tecnología llamada AJAX, con el objetivo de disminuir esos efectos. AJAX por el momento hace posible que las interacciones entre cliente y servidor se hagan transportando solo los datos útiles e involucrados en la petición realizada por el cliente, con el objetivo de no sobrecargar el servidor con datos innecesarios. AJAX (Asynchronous JavaScript And XML) es una terminología nueva para agrupar una serie de tecnologías ya existentes como son:

- ✓ Exhibición e interacción dinámicas usando el Document Object Model.
- ✓ Intercambio y manipulación de datos usando XML.

AJAX es también una técnica de desarrollo Web para crear aplicaciones interactivas que se ejecutan en el cliente, es decir, en el navegador del usuario, manipulando la página actual dentro de sus navegadores usando métodos de Document Object Model. Los scripts y las rutinas son dirigidas al servidor buscando los datos que son usados para actualizar la página, eliminar registros, extender formularios Web, devolver peticiones simples de búsqueda, o editar árboles de categorías y mantener comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre el sistema y en la misma página sin necesidad de recargarla.

En aplicaciones que utilizan AJAX se pueden enviar peticiones al servidor Web para obtener únicamente la información necesaria empleando algún lenguaje para servicios Web basado en XML, y usando javascript en el cliente para procesar la respuesta del servidor Web. Esto redundará en una mayor interacción gracias a la reducción de información intercambiada entre servidor y cliente y que parte del

Capítulo 2: Características del sistema

proceso de la información lo hace el propio cliente, liberando al servidor de ese trabajo[12].

El uso de AJAX en la realización de otras aplicaciones es opcional o de poca utilización, sin embargo, en nuestro caso es el apoyo tecnológico ya que la interacción entre cliente y servidor se debe mantener constantemente sin que el usuario vea entorpecido el trabajo que realiza debido a que la página esté actualizando datos innecesarios. A medida que el redactor edita sus noticias, el agente JITIR interactúa con el servidor mediante la utilización de AJAX y brinda sugerencias a cerca del contexto en el que se está escribiendo en el caso de que existan noticias publicadas que se relacionen con las peticiones realizadas al servidor.

2.2 PloRA: estructura y funcionalidad.

Con este trabajo de diploma se pretende realizar una aplicación que asemeje su funcionalidad a la de un Remembrance Agent para Emacs, pero en nuestro caso que se relacione con el CMS Plone. De esta mezcla es que surge el término PloRA: Plone Remembrance Agent. Este nuevo agente JITIR debe unirse a las páginas de edición en Plone, para escribir una nueva noticia o página Web, dependiendo siempre de las actividades de los trabajadores de la prensa. El kupu es el editor con el que debe interactuar el PloRA.

El agente PloRA, al vincularse con el editor kupu de las páginas del Plone, debe comportarse como una aplicación de fácil manejo para un usuario común, o sea, en el caso de los periodistas(que por el momento, son los usuarios finales del PloRA) no necesitan tener conocimientos amplios de informática para trabajar con su interfaz. Solo es necesario conocer la actividad fundamental que se puede realizar con este agente: insertar enlaces a documentos relacionados a

Capítulo 2: Características del sistema

partir de las sugerencias devueltas.

Para que PloRA pueda brindar sugerencias a partir del contexto local del usuario, es necesario que se vincule con dicho contexto. Pero para posibilitar su adaptación a la interfaz de cualquier editor de texto, que no sea solamente el kupu, se hace necesario incluir un adaptador[13] que permita relacionarlos con el agente PloRA. Además del adaptador, se necesita sustituir la capa(layer) correspondiente al WYSIWYG del kupu por una capa con el mismo nombre que contenga las modificaciones que incluyen la interfaz del PloRA. De esta manera el Plone al cargar todas las capas que tiene definidas, reconoce el WYSIWYG modificado antes que el del kupu original.

Al vincularse el PloRA con el editor, el adaptador que media entre ellos se inicializa cuando se carga totalmente la página de edición, y obtiene el identificador que posee el editor de texto, para de esta forma poder aplicar las funciones definidas en el agente a dicho editor. Después de inicializado el agente, comienza a “observar” el contenido que posee el editor, si el usuario escribe, el agente JITIR obtiene el texto que se escribió. El texto obtenido debe estar situado en un elemento de bloque[14] en el HTML de la página, si esto no ocurre entonces se busca el elemento padre de ese elemento y devuelve el texto del elemento de bloque en su totalidad.

El texto que se obtiene es generalmente un párrafo. Éste debe sufrir un proceso de transformaciones para ejecutar las búsquedas correctas en el servidor. Primeramente el párrafo se divide en oraciones, pero como las oraciones pueden ser extensas en ocasiones, es necesario hacer divisiones más pequeñas, por ejemplo, tomando otros signos de puntuación dentro de la oración. Luego de esta fragmentación se eliminan los conjuntos de palabras menos significativas(con tres o menos caracteres: pronombres, artículos, preposiciones, etc.) que

Capítulo 2: Características del sistema

aparezcan en la redacción y se seleccionan cinco palabras. Con las palabras de la oración filtrada se conforman las consultas que se enviarán al servidor.

Para lograr una actualización de las consultas que se harán al servidor, se comparan los párrafos en los cuales se mueve el usuario dentro del editor con la finalidad de no mezclar las consultas extraídas de un párrafo con las de otro y mostrar las respuestas del servidor de acuerdo con el párrafo donde se encuentre el cursor.

Conformado el grupo de consultas, se envían al servidor con el objetivo de buscar documentos relacionados con éstas y devolverlos. Para este proceso utilizamos la herramienta PloRATool que se relaciona con la herramienta para Plone, el portal_catalog, enviándole una consulta.

En la construcción de nuestro agente JITIR está presente un mecanismo de búsqueda e indización de documentos. A este motor se le conoce como portal_catalog y cumple funciones similares al motor de búsqueda de los agentes anteriores; conocido como Savant. El método que utiliza el portal_catalog para buscar e indizar los documentos que están almacenados dada una consulta es el modelo espacio vectorial teniendo en cuenta el algoritmo Okapi con su versión TF/iDF[15](Frecuencia de un Término y Frecuencia inversa de un Documento, o Term Frequency y inverse Document Frequency, en inglés). Este método se encarga de asignar un “peso” o relevancia(score) al par consulta-documento que sea encontrado por el portal_catalog.

La variable relevancia puede considerarse perteneciente a un conjunto difuso. No existen mediciones exactas para determinar cuán relevante es un documento u otro, o sea, que puede considerarse

Capítulo 2: Características del sistema

irrelevante, poco relevante o relevante el mismo documento por diferentes consideraciones personales. Pero con la utilización del algoritmo Okapi, dada una consulta se puede recuperar información relevante de una base de datos partiendo de la frecuencia con que aparezca la consulta en cada documento que se encuentren en esta. La relevancia está representada por el score que le asigna automáticamente el algoritmo de recuperación a un documento, dado por un cálculo matemático. Una vez obtenida la respuesta del servidor, la herramienta propia del PloRA(PloRATool) genera un XML.

El documento XML generado por el PloRATool es procesado para conformar una lista de objetos con los datos: título del documento, autor(es), descripción, etc.). Esta información tomará una estructura que pueda mostrarse al usuario a través de una interfaz. Pero la verdadera relevancia de los documentos mostrados al usuario, será la que éste último decida al apoyándose en la información que se brinda de cada sugerencia.

2.3 Ingeniería inversa relacionada con Zope-Plone

La Ingeniería inversa de software es el proceso general de análisis a un software específico para determinar cómo está diseñado o cómo funciona. Se encarga de crear y mantener aplicaciones de software en las cuales se aplican tecnologías y prácticas de manejo de proyectos e ingeniería, etc. Además, está relacionada con el costo que se aplica a un software[16]. Y se denomina Ingeniería inversa ya que se desarrolla en sentido contrario a las tareas comunes de la ingeniería.

La Ingeniería inversa[17] está unida a un conjunto de técnicas y herramientas que ayudan al entendimiento de cómo está formada una aplicación (en el caso en que se analice un software) que se encuentra en manos de los usuarios. Ayuda además a identificar los

Capítulo 2: Características del sistema

componentes del sistema, sus interrelaciones y a crear las representaciones dentro del sistema en un nivel de mayor abstracción[18]. En el análisis moderno de software se emplea con frecuencia la Ingeniería inversa, y destacan propósitos como los siguientes:

- 1) Encontrar códigos maliciosos y patrones que componen la implementación de virus.
- 2) Descubrimiento inesperado de fallas en aplicaciones.
- 3) Permite el estudio de las aplicaciones y obtener el conocimiento almacenado en código

Existen dos categorías de las aplicaciones de Ingeniería inversa: seguridad relacionada y desarrollo de software relacionado.

En el caso del desarrollo de software relacionado, que es el caso particular que vamos a aplicar en nuestro trabajo, se emplea para analizar aplicaciones sin documentación publicada, que aunque es software libre, no presentan publicaciones acerca de este tema con suficiente argumentos. Nuestro propósito fundamental al introducir la Ingeniería inversa en el trabajo de diploma se debe a que se emplea un software(CMS Plone y Zope, en su conjunto), con el cual nuestra aplicación debe relacionarse y es necesario el conocimiento previo de cómo funciona para identificar directamente las partes con las que se debe relacionar el nuevo agente JITIR.

La documentación perteneciente a la implementación del CMS Plone y de Zope no se encuentra disponible y por esta sencilla razón es esencial incluir el modelado de este sistema. No se trata la aplicación de manera íntegra ya que sería una investigación muy amplia (además se necesitaría utilizar herramientas que realicen

Capítulo 2: Características del sistema

Ingeniería inversa). Sin embargo, documentar las partes con las cuales se enlaza el agente JITIR es importante. Es la manera más eficiente de dejar el camino preparado para aquellos que deseen continuar con el trabajo que se está realizando en esta tesis.

Debemos analizar los elementos que componen a Plone y Zope y con los cuales se va a relacionar el agente JITIR que se desea implementar. De esta forma quedará claro qué recursos de los que existen se tienen en cuenta para realizar el resto de las actividades de programación para PloRA.

Una herramienta esencial en la búsqueda de información que posee Plone es el `portal_catalog`. Esta herramienta es una versión extendida de la herramienta ZCatalog[19], que a su vez constituye un artefacto de indizado y búsqueda de todo tipo de objetos que contenga el Zope. El catálogo que emplea Plone ayuda a indizar contenido y proporciona una interfaz para examinar rápidamente el contenido del sitio o los sitios que tenga creado. El `portal_catalog` proporciona un método para buscar contenido con mucha precisión (Okapi BM25 Rank). Al realizarse una búsqueda en Plone se está solicitando información directamente del índice con que cuenta, y el catálogo devuelve los resultados que concuerden con la consulta realizada[19].

El `portal_catalog` tiene habilitados muchos índices para buscar cualquier objeto definido en el Zope, entre ellos podemos encontrar: `Date`, `Creator`, `Description`, `SearchableText`, `Title`, etc. El índice `SearchableText` es el que contiene una descripción, un título y el cuerpo del texto. `ZCTextIndex` es el tipo correspondiente al índice `SearchableText` que busca eficientemente los textos almacenados en la base de datos. Y este es el índice que ofrece la precisión necesaria en las respuestas que un agente JITIR debe mostrar a los usuarios.

Capítulo 2: Características del sistema

De manera general, Zope soporta contenidos, presentaciones y componentes lógicos. Las plantillas de las páginas constituyen componentes de presentación y se utilizan para mostrar contenidos. Pero estas páginas tienen un mecanismo para su organización. El CMF(Content Management Framework) es un sistema que sostiene y se ocupa de gestionar las plantillas del Plone, además de los skins o pieles y de objetos de contenidos[20]. Plone se comporta como una capa encima del CMF y este a su vez es una aplicación que se ejecuta encima de Zope.

Los CMF y la propiedad Acquisition permiten a los diferentes contenidos de Plone utilizar funciones y características de otros objetos. El término Acquisition define la propiedad que obtienen los objetos que están contenidos en otros, adquiriendo el comportamiento de estos últimos, o sea, toman las características de sus contenedores.

Por las posibilidades que brindan estas dos propiedades de Plone, y por la ventaja de la reutilización del código fuente de este, utilizamos la plantilla WYSIWYG del kupu, donde se encuentra la definición HTML y las funciones para ser reconocida por el CMF y por Zope, y le incluimos un pequeño código HTML que estructura la interfaz de usuario donde el PloRA debe mostrar las sugerencias. Con la sustitución de esta plantilla no se afecta el comportamiento del editor ni sus funciones, solo que ahora incluye una nueva interfaz.

Capítulo 2: Características del sistema

Diagrama de clases del diseño para Zope-Plone

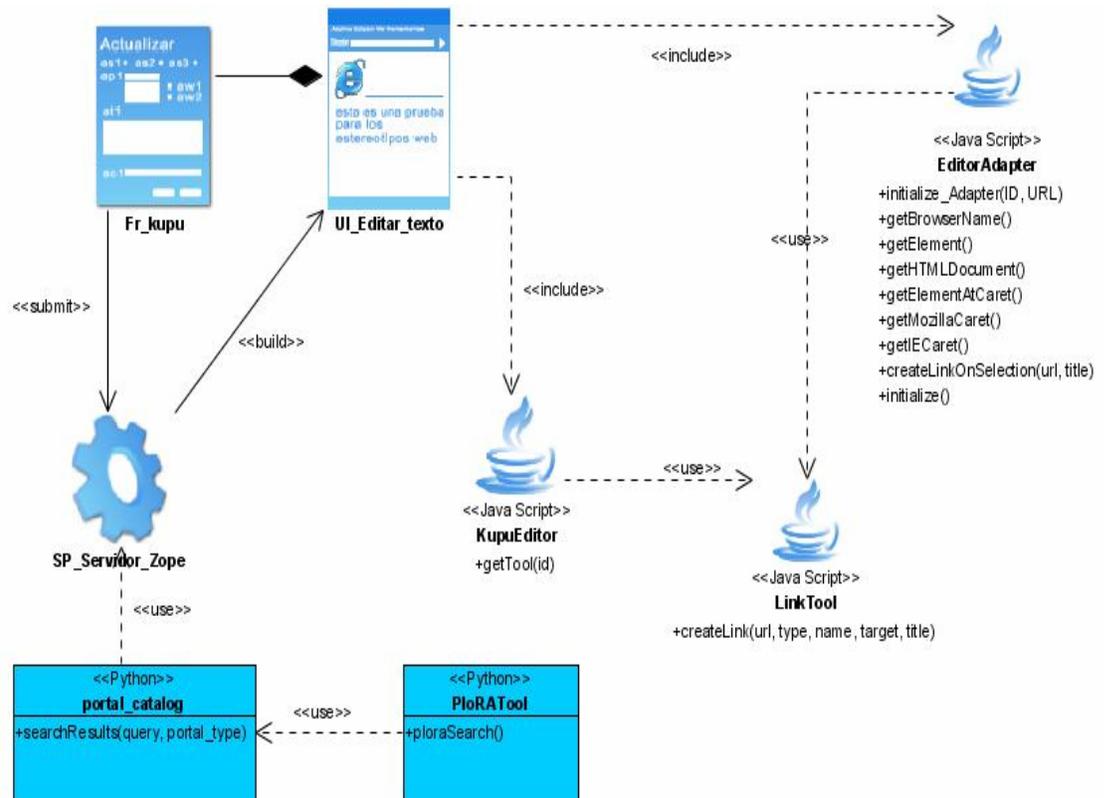


Figura No. 1: Diagrama de clases del diseño

Descripción de las clases del diseño

| | |
|-----------------------------|------|
| Nombre: KupuEditor | |
| Tipo de clase: controladora | |
| Atributos | Tipo |

Capítulo 2: Características del sistema

| | |
|--------------------|--|
| filters | array |
| tools | array |
| logger | object |
| config | object |
| Para cada función: | |
| Nombre: | getTool(id) |
| Descripción | Mediante el identificador de una herramienta se puede acceder a todas sus funcionalidades. |

Tabla No. 1: Clase de diseño: KupuEditor

| | |
|-----------------------------|--|
| Nombre: LinkTool | |
| Tipo de clase: controladora | |
| Atributos | Tipo |
| | |
| Para cada función: | |
| Nombre: | createLink(url, type, name, target, title) |

Capítulo 2: Características del sistema

| | |
|-------------|--|
| Descripción | Crear un vínculo en el texto seleccionado en el editor de texto. |
|-------------|--|

Tabla No. 2: Clase de diseño: LinkTool

| | |
|-----------------------------|---|
| Nombre: portal_catalog | |
| Tipo de clase: controladora | |
| Atributos | Tipo |
| | |
| Para cada función: | |
| Nombre: | searchResults(consulta, portal_type) |
| Descripción | Devuelve el resultado de una búsqueda en el servidor dada una consulta. |

Tabla No. 3: Clase de diseño: portal_catalog

| | |
|-------------------------|------|
| Nombre: Editar_texto | |
| Tipo de clase: interfaz | |
| Atributos | Tipo |

Capítulo 2: Características del sistema

| | |
|--------------------|---|
| | |
| Para cada función: | |
| Nombre: | save() |
| Descripción: | Guarda el documento escrito. |
| Nombre: | cancel() |
| Descripción: | Cancela las operaciones realizadas anteriormente. |

Tabla No. 4: Clase de diseño: Editar_texto

2.4 RUP como metodología de desarrollo para PloRA

“El Proceso Unificado de Rational[21] (RUP, el original inglés *Rational Unified Process*) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos”. La documentación que se le hace a las diferentes aplicaciones es la base para futuras modificaciones y mejoras que las hagan más prácticas y manejables.

2.4.1 Objeto de automatización

Como se ha explicado anteriormente, la prensa necesita de aplicaciones que aceleren el proceso de gestión y publicación de noticias y para eso está trabajando el Proyecto de Informatización de la Prensa, pero además, la utilización de un agente de recuperación de

Capítulo 2: Características del sistema

información Just-in-time puede ser de gran utilidad para: primero, evitar que los redactores de noticias interrumpen su trabajo para buscar información en otros contextos que no sea en el que están trabajando, y segundo, se disminuye la cantidad de información desconectada que existe en las actuales publicaciones nacionales.

Esencialmente, en la estructura del PloRA existen varios elementos que deben funcionar de manera automática. Uno de ellos es la creación de consultas, el cual se produce sin que el usuario intervenga directamente, ya que no hace las búsquedas explícitamente, pero al escribir en el editor en que se encuentre se toman automáticamente partes de ese contexto para elaborar consultas que luego generen sugerencias de otros documentos relacionados con ese tema y se puedan vincular al contexto actual. Otro elemento a automatizar es la creación de vínculos en el área de edición. En este caso es donde se conectan los diferentes documentos que se obtienen en las sugerencias y que el usuario estima que tienen una relevancia alta para su contexto local.

Con esta aplicación se quiere automatizar el proceso de insertar vínculos y procesar sugerencias en el contexto local del usuario en los editores que utiliza el plone.

2.4.2 Objetivos estratégicos del sistema

- ✓ Realizar una aplicación de fácil acceso y con una interfaz cómoda y no intrusiva para el usuario del sistema.
- ✓ Contribuir al desarrollo tecnológico de los centros de prensa.
- ✓ Realizar un sistema eficiente que responda a las necesidades de los trabajadores de los centros de redacción.

Capítulo 2: Características del sistema

2.4.3 Propuesta del sistema

El agente JITIR PloRA debe funcionar como un artefacto recuperador de información, o sea, que brinde información relevante para el usuario, pero además que le posibilite a dicho usuario enlazar esta información con el texto que está escribiendo o leyendo mediante la funcionalidad que debe poseer (crear vínculos).

2.4.3.1 Modelo de Dominio

El redactor de noticias se desarrolla en el ambiente del editor de texto Kupu. Mientras el usuario escribe su noticia, el agente PloRA debe ser capaz de recuperar lo que se escribe en el editor y fragmentarlo, en esta primera versión, utilizando los signos de puntuación que existan en el texto obtenido. De esa fragmentación se obtienen consultas. Las consultas son enviadas al servidor con el objetivo de buscar documentos, noticias, artículos, utilizando tecnologías AJAX para este envío. En caso de que el servidor tenga almacenados documentos relacionados con las peticiones hechas, entonces se obtienen los que tengan mayor relevancia, ya que la relevancia debe estar dada por la frecuencia de aparición de las consultas en los documentos según la definición del algoritmo Okapi empleado por la herramienta portal_catalog que utiliza Plone, y estos documentos obtenidos son enviados a la interfaz de usuario del agente. Las respuestas del servidor son igualmente entregadas por la utilización de tecnologías AJAX. Para que las respuestas sean satisfactorias y cumplan su objetivo deben mostrar en la interfaz de usuario una estructura organizada que le permita al usuario determinar el título del artículo que fue devuelto, el autor, la descripción correspondiente a la publicación, y otros datos de interés. Estos datos, al organizarse de acuerdo a la estructura predefinida, se convierten en sugerencias que el usuario puede o no utilizar. En caso de usarlas, el

Capítulo 2: Características del sistema

usuario puede seleccionar una palabra o frase en el editor y hacer un enlace con el documento devuelto en la sugerencia haciendo clic en la sugerencia que estime conveniente.

Representación del Modelo de Dominio

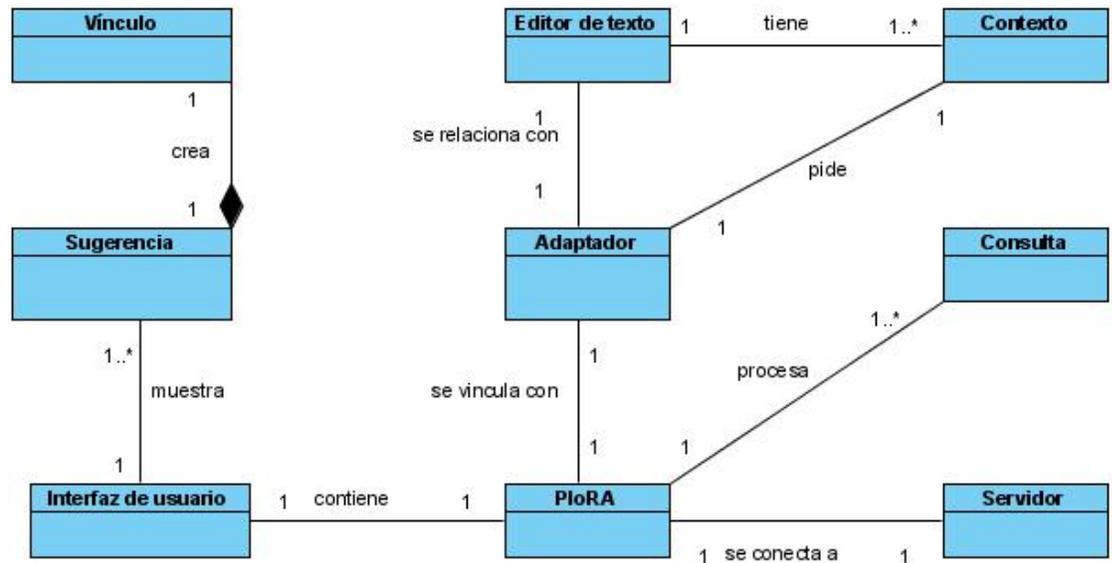


Figura No. 2: Modelo de Dominio

Entidades y conceptos principales

| Concepto | Descripción |
|-----------------|---|
| Editor de texto | Producto perteneciente a Plone que permite escribir información a publicar en una página Web. En este caso se hace referencia al editor kupu. |
| Contexto | Fragmento de texto donde se encuentra situado el mouse. |

Capítulo 2: Características del sistema

| | |
|---------------------|--|
| Adaptador | Patrón de diseño utilizado para unir interfaces diferentes y que intercambian información. |
| Consulta | Pregunta que se realiza a un servidor con el objetivo de obtener algún contenido almacenado en éste. |
| PloRA | Agente recuperador de información definido para procesar consultas contando con funciones definidas internamente. |
| Servidor | Aplicación que almacena datos útiles y que brinda información acerca de estos datos en caso de realizársele alguna consulta. |
| Interfaz de usuario | Interfaz que muestra información importante para el usuario y le permite interactuar con nuevas funciones de la aplicación. |
| Sugerencia | Contenido relevante que se muestra al usuario con el objetivo de que lo utilice si lo desea. |

Capítulo 2: Características del sistema

| | |
|---------|--|
| Vínculo | Enlace que une a documentos que están total o parcialmente “desconectados” entre sí. |
|---------|--|

Tabla No. 5: Entidades y conceptos principales

2.4.3.2 Requerimientos de la aplicación

Al identificar los requisitos de la aplicación, se van identificando las funcionalidades y cualidades con las que debe contar la aplicación. De esta forma se llega a un mejor entendimiento entre los usuarios y el trabajo de los desarrolladores. Además, una buena elección de los requerimientos hace posible que el mantenimiento futuro de la aplicación lleve un orden y un seguimiento detallado.

Requisitos Funcionales:

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Las condiciones que debe cumplir la aplicación son las siguientes:

- ✓ R1 – Mostrar sugerencias
- ✓ R2 – Crear vínculo

Requisitos No Funcionales:

Los 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 y confiable.

Las propiedades que debe tener el producto son:

Apariencia o interfaz externa:

Capítulo 2: Características del sistema

- ✓ El diseño debe ser discreto, sencillo, que no desvíe la atención del usuario.

Usabilidad:

- ✓ La aplicación puede ser utilizada por cualquier persona que tenga conocimientos básicos de computación.
- ✓ El software tendrá un manual de usuario o ayuda disponible para hacer las consultas necesarias y usar todas las funcionalidades disponibles.

Rendimiento:

- ✓ Los tiempos de respuesta serán rápidos, con un margen de hasta 8 segundos para obtener las sugerencias que espera el cliente.

Soporte:

- ✓ En la parte del cliente se necesita un navegador capaz de interpretar JavaScript.
- ✓ Alta velocidad de respuesta a peticiones hechas al servidor

Capítulo 2: Características del sistema

Definición de los actores

| Actores | Justificación |
|----------------------|---|
| Redactor de noticias | Persona que se encarga de iniciar los casos de uso del sistema y está en constante interacción con la aplicación. |
| Reloj | Sistema que interactúa con la aplicación que se desarrolla, además, ejecuta determinadas funciones en intervalos de tiempo que condicionan la actividad de funciones del sistema PloRA. |

Tabla No. 6: Actores del sistema

Listado de Casos de Uso

| | |
|-------------|---|
| CU-1 | Escribir texto |
| Actores | Redactor de noticias |
| Descripción | En este caso de uso el usuario inicia las acciones escribiendo en el editor de texto del Plone. |
| Referencias | R1 |

Tabla No. 7: CU Escribir texto

Capítulo 2: Características del sistema

| | |
|-------------|---|
| CU-2 | Procesar consultas |
| Actores | Reloj |
| Descripción | Se inicia cuando el intervalo de tiempo indicado en la aplicación se cumple. Se va a obtener un párrafo, donde se encuentre el cursor. Las consultas se construirán realizando un filtrado de palabras donde se eliminen aquellas que pueden tener menor relevancia en una búsqueda (artículo, preposición, pronombre) en el servidor Zope. |
| Referencias | R1 |

Tabla No. 8: CU Procesar consultas

| | |
|---------|----------------------|
| CU-3 | Insertar vínculo |
| Actores | Redactor de noticias |

Capítulo 2: Características del sistema

| | |
|-------------|---|
| Descripción | En este caso de uso el usuario selecciona en qué parte del documento desea incluir el vínculo seleccionando el fragmento donde lo quiere colocar, y hace clic en la sugerencia devuelta que le sea más relevante, y así relacionar el documento devuelto. |
| Referencias | R2 |

Tabla No. 9: CU Insertar vínculo

Diagrama de Casos de Uso del Sistema

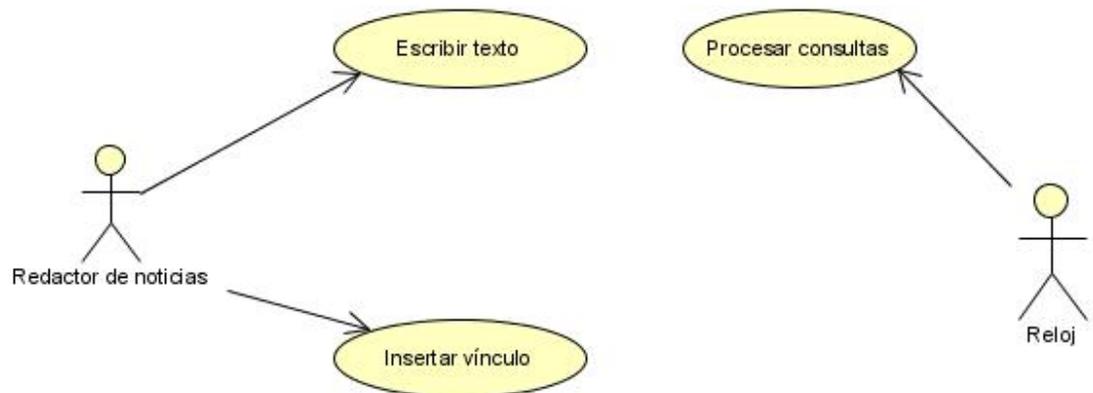


Figura No. 3: Diagrama de CUS

Capítulo 2: Características del sistema

Descripción de los Casos de Uso del Sistema

| | |
|-------------------------|---|
| CU –1 | Escribir texto |
| Propósito | Escribir información relacionada con determinada noticia que se desea publicar por parte del redactor de noticias. |
| Actor | Redactor de noticias |
| Resumen | El caso de uso se inicia cuando el redactor de noticias comienza a escribir en el editor de texto de la página de edición de Plone. |
| Referencias | R1 |
| Precondiciones | El actor haya escrito en el editor de texto. |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |

Capítulo 2: Características del sistema

| | |
|--|---|
| 1. El redactor escribe en el editor de texto. | 1.1 El adaptador que posee la aplicación pide el identificador (ID) del editor y así obtiene el párrafo donde está el cursor dentro del contexto. |
| Flujo Alternativo | |
| Acción del actor | Respuesta del sistema |
| 1. El redactor no escribe en el editor de texto. | 1.1 El sistema obtiene el contexto donde se encuentra el cursor pero este se encuentra vacío, o sea, no contiene palabras. |
| Prioridad | Critico. |

Tabla No. 10: Descripción del CU: Escribir texto

| | |
|-----------|---|
| CU -2 | Procesar consultas |
| Propósito | Iniciar un proceso de creación de consultas derivadas de las oraciones definidas en el contexto obtenido. |
| Actor | Reloj |

Capítulo 2: Características del sistema

| | | |
|-------------------------|---|--|
| Resumen | <p>Este caso de uso es la secuencia automática de la acción que inicia el usuario al escribir en el editor de texto y que continúa después con las búsquedas realizadas al servidor. De éste se obtienen los documentos relacionados con las consultas hechas y con un alto nivel de relevancia. De los resultados que se extraen del servidor, se conforman las sugerencias que se van a mostrar al usuario.</p> | |
| Referencias | R1 | |
| Precondiciones | El usuario escribe en el editor de texto. | |
| Flujo Normal de Eventos | | |
| Acción del Actor | Respuesta del Sistema | |

Capítulo 2: Características del sistema

| | |
|---|---|
| <p>1. El reloj del sistema evalúa el intervalo señalado en la aplicación.</p> | <p>1.1 El adaptador inicia un proceso de obtención de consultas a partir de la identificación de oraciones.</p> <p>1.2 Estas oraciones son filtradas, eliminando las palabras más cortas que tienden a ser menos relevantes en una búsqueda.</p> <p>1.3 Se construyen las consultas después del filtrado.</p> <p>1.4 Las consultas son enviadas al PloRA para almacenarlas y controlar sus respectivas respuestas. Este proceso se realiza si no están guardadas en la cache del PloRA.</p> <p>1.5 Se mandan al servidor todas las consultas correspondientes al contexto actual del cursor.</p> <p>1.6 En caso de que el servidor encuentre documentos relacionados con la consulta realizada, los devuelve.</p> |
|---|---|

Capítulo 2: Características del sistema

| | |
|-------------------|---|
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| | <p>1.4 Si están almacenadas en la cache, se obtienen las respuestas correspondientes.</p> <p>1.6 Si no encuentra documentos relacionados en la base de datos, no muestra sugerencia alguna.</p> |
| Prioridad | Critico. |

Tabla No. 11: Descripción del CU: Procesar consultas

| | |
|-----------|--|
| CU -3 | Insertar vínculo |
| Propósito | Introducir uno o más enlaces en el texto que se está escribiendo en el editor. Estos enlaces unen a los documentos que fueron devueltos por el servidor y que contribuyen a incrementar el contenido informativo que se va a publicar. |
| Actor | Redactor de noticias |

Capítulo 2: Características del sistema

| | | |
|---|--|--|
| Resumen | El caso de uso se inicia cuando el usuario lee las sugerencias y escoge cuál de ellas va a vincular con el texto escrito. | |
| Referencias | R2 | |
| Precondiciones | El usuario lee las sugerencias para escoger la que le es más representativa. | |
| Flujo Normal de Eventos | | |
| Acción del Actor | Respuesta del Sistema | |
| 1. El redactor lee las sugerencias, escoge una para incluir un enlace y hace clic en la seleccionada. | 1.1 Se inserta un enlace en la palabra seleccionada en el cuerpo del texto escrito. Este enlace une el contenido del documento devuelto con lo que se está redactando. | |
| Flujo alternativo | | |
| Acción del actor | Respuesta del sistema | |
| Prioridad | Critico. | |

Tabla No. 12: Descripción del CU: Insertar vínculo

Capítulo 2: Características del sistema

Conclusiones

Comprender el tema de Ingeniería inversa tratado en el epígrafe 2.4 es fundamental para saber cómo funciona el ambiente donde se desarrollará el agente JITIR. En el siguiente epígrafe, se inició el proceso incremental de RUP relacionado con el modelo de dominio y las características propias del nuevo producto para Plone, y quedó definido de esta forma por tratarse del desarrollo de un subsistema que se unirá a la parte descrita en la Ingeniería inversa.

Capítulo 3 Análisis y diseño del sistema

Introducción

Los flujos de trabajo de análisis y diseño son los encargados de modelar con precisión cada paso de la aplicación que se llevará a la implementación. Este capítulo desarrollará estos temas para explicar y representar cada paso llevado a cabo en la construcción de los respectivos modelos.

3.1 Análisis

Las clases que se identifican en el flujo de trabajo de análisis y diseño, fundamentalmente en el análisis, están asociadas con el contexto del dominio del problema. Por esta razón representan conceptos y relaciones. En el análisis se logra una abstracción del sistema, constituyendo una entrada fundamental a la hora de modelar la aplicación.

Clases interfaz

Estas clases aparecen al modelar las interacciones entre el sistema y su(s) autor(es), y pueden identificarse a partir de estas situaciones:

- ✓ Por cada interacción actor-caso de uso aunque aparezcan más de una ventana en la solución.
- ✓ Una clase para cada sistema externo que será el responsable de la relación del sistema con cada uno de los usuarios.
- ✓ Una clase para cada actor que represente un dispositivo sobre el cual el sistema actúa o recibe información.

Las clases de control coordinan el trabajo de uno o varios casos de uso, relacionando las actividades de los objetos que implementan la funcionalidad de los casos de uso, por lo que definen el flujo de

Capítulo 3: Análisis y Diseño del sistema

control y las transacciones dentro de un caso de uso asignando el trabajo a otros objetos.

Diagrama de clases del análisis

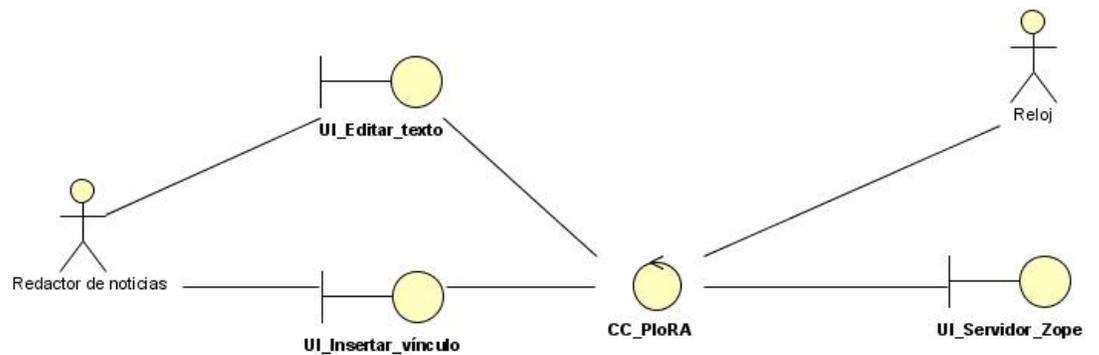


Figura No. 4: Diagrama de clases del análisis

Diagrama de colaboración

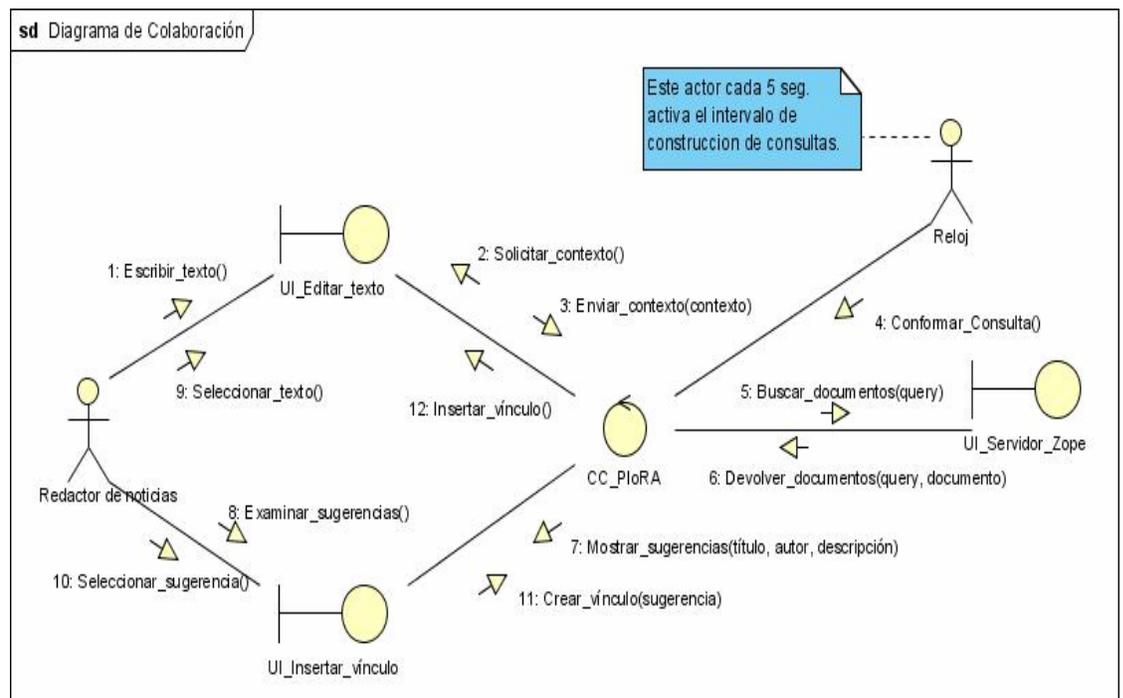


Figura No. 5: Diagrama de colaboración

Los diagramas de colaboración, y todos en general, pueden ser difíciles de leer e interpretar por sí mismos. En estos casos puede ser

Capítulo 3: Análisis y Diseño del sistema

útil un documento adicional que los explique. Este texto debe obviar los atributos, responsabilidades, etc., de los objetos que se manejan en él. Este documento se conoce como:

Flujo de sucesos-análisis para el diagrama de colaboración

El redactor de noticias escribe en la UI_Editar_texto el texto(1) que va a conformar la noticia. Al inicializarse el editor por parte del Plone, se inicializa también el agente que funcionará como clase controladora. La CC_PloRA solicita el contexto(2) en el cual se encuentra el cursor, la UI_Editar_texto envía el contexto solicitado(3), y la clase controladora realiza un conjunto de operaciones como: dividir el contexto en oraciones, filtrar dichas oraciones, etc. El intervalo de tiempo que condiciona la conformación de las consultas a partir del contexto obtenido se activa por la acción directa del actor Reloj y CC_PloRA señala el cumplimiento del intervalo y se comienza a crear las consultas(4) dada la cantidad de oraciones del contexto obtenido. La CC_PloRA solicita al servidor los documentos a través de las consultas conformadas (5). El servidor utiliza su herramienta portal_catalog para buscar los documentos que se relacionen con las consultas y los devuelve(6) a la clase controladora. CC_PloRA conforma una lista de datos conocida como sugerencia(está formada por el título del artículo, el autor, una breve descripción y un valor de relevancia o score) y la muestra(7) al usuario empleando la UI_Insertar_vínculo. Estas sugerencias constantemente se muestran al usuario.

El redactor si desea introducir un enlace a partir de las sugerencias dadas en la interfaz de usuario, examina(8) cual de ellas tiene mayor relevancia a su consideración. De encontrar alguna de su interés, selecciona el texto(9) donde desee insertar el enlace. El redactor selecciona una sugerencia(10) mediante la UI_Insertar_vínculo

Capítulo 3: Análisis y Diseño del sistema

haciendo clic en ella. Esta interfaz solicita la creación de un vínculo(11). La controladora CC_PloRA crea el vínculo en el editor de texto(12) después de procesar la sugerencia marcada por el usuario.

Requisitos especiales

Los requisitos especiales son descripciones textuales que recogen todos los requisitos no funcionales[22].

Cuando el usuario escribe en la interfaz del editor(UI_Editar_texto), las respuestas que debe mostrar la interfaz UI_Insertar_vínculo no deben demorar más de cinco segundos. UI_Insertar_vínculo tendrá además un color que permita identificar su ubicación en la estructura de la página. Sin embargo, este color no debe llamar demasiado la atención del usuario mientras trabaja para que no pierda su concentración.

3.2 Diseño

El diseño debe ser capaz de soportar los requisitos funcionales y no funcionales, además de describir cómo se implementará el sistema; o sea, la definición de un plano para construir la aplicación. También, debe ejecutar las funciones descritas en los casos de uso y sobre todo ser flexible a cambios. Hay que tener en cuenta que el diseño contempla clases estructuradas y subsistemas con interfaces definidas (componentes). Además de especificar el lenguaje de programación a utilizar en el flujo de trabajo de implementación.

En nuestra aplicación desarrollaremos la mayor parte del código de implementación en lenguaje JavaScript, con algunas líneas de código en lenguaje Python. Javascript es un lenguaje de programación diseñado para ejecutarse a través de un intérprete utilizado fundamentalmente en páginas Web. Este lenguaje está basado en

Capítulo 3: Análisis y Diseño del sistema

prototipos y aunque no posee herencia entre las clases que lo componen, propia de los lenguajes orientados a objetos, se puede considerar un lenguaje en el que se puede programar orientado a objetos, donde los objetos que representan las clases son clonados y sus funcionalidades son reutilizadas.

Diagrama de clases del diseño

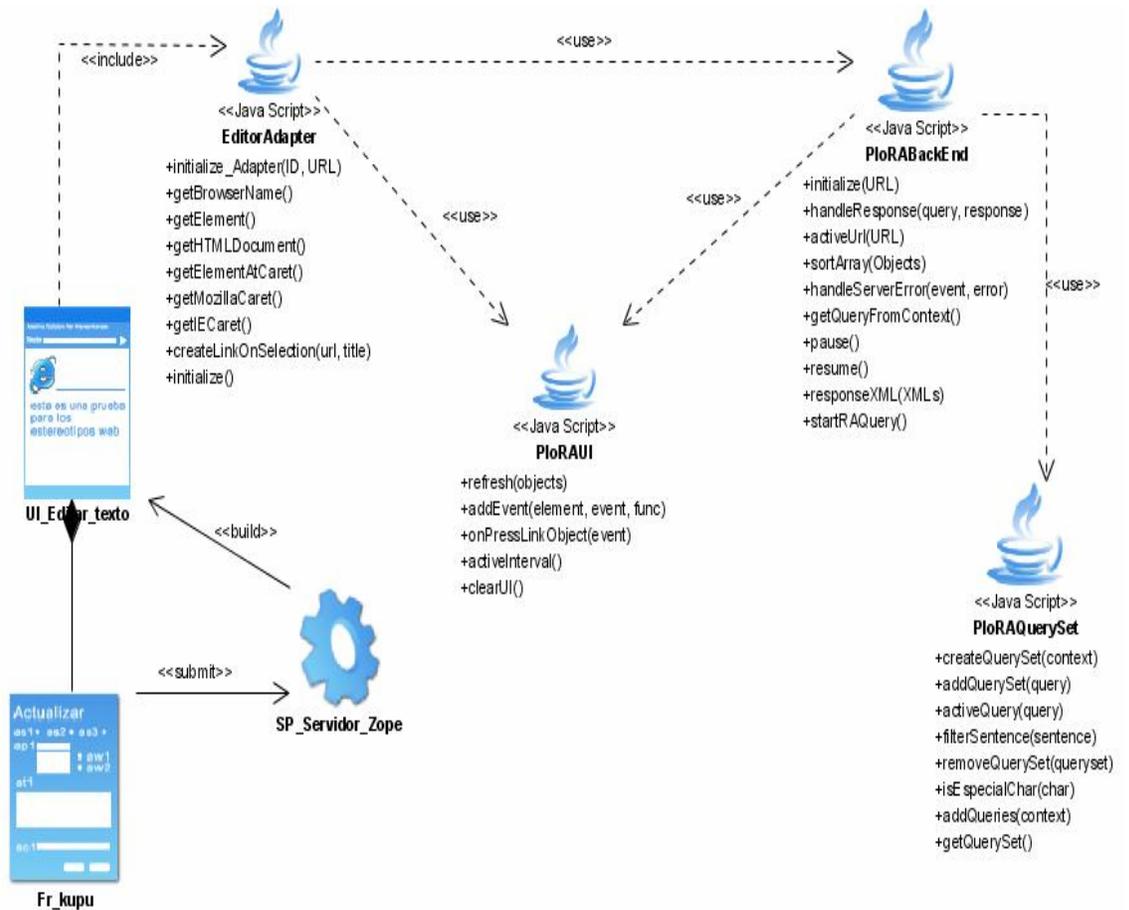


Figura No. 6: Diagrama de clases del diseño

Capítulo 3: Análisis y Diseño del sistema

Diagrama de clases del diseño(2)

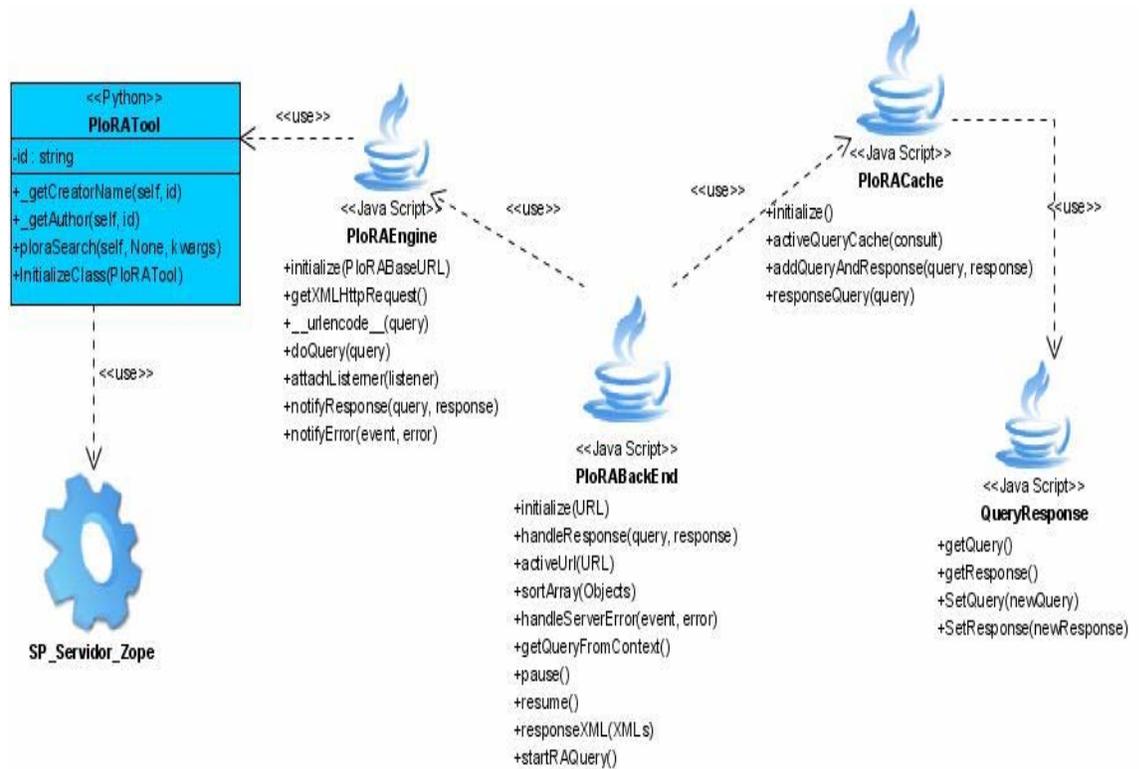


Figura No. 7: Diagrama de clases del Diseño(2)

Descripciones de las clases del diseño

| | |
|-----------------------------|-------|
| Nombre: PloRAEngine | |
| Tipo de clase: controladora | |
| Atributo: | Tipo: |

Capítulo 3: Análisis y Diseño del sistema

| | |
|----------------------------|--|
| HTTP_RESPONSE_OK | int |
| AJAX_READYSTATE_1 | int |
| AJAX_READYSTATE_OK | int |
| Para cada responsabilidad: | |
| Nombre: | initialize(PloRABaseURL) |
| Descripción: | Inicializar y conformar la url del agente que va a utilizar el objeto AJAX para realizar las búsquedas en el servidor. Además de obtener la dirección donde se guarda la instancia del sitio Plone activo. |
| Nombre: | getXMLHttpRequest() |
| Descripción: | Función que crea el objeto HttpRequest para lograr el funcionamiento de AJAX en los navegadores utilizados por el usuario. |
| Nombre: | __urlencode__(str) |
| Descripción: | Devuelve la consulta codificada para que sea interpretada correctamente por el navegador. |
| Nombre: | doQuery(consulta) |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|--|
| Descripción: | Interactúa con el servidor enviando la consulta y recibiendo la respuesta de dicha consulta, todo este proceso se realiza mediante AJAX. |
| Nombre: | attachListener(listener) |
| Descripción: | Coloca un listener u observador para notificar la ocurrencia de algún error. |
| Nombre: | notifyResponse(consulta, respuesta) |
| Descripción: | Maneja las respuestas del servidor y sus respectivas consultas, y las envía a la clase PloRABackEnd. |
| Nombre: | notifyError(event, error) |
| Descripción: | Notifica los errores en las respuestas devueltas por el servidor. |

Tabla No. 13: Clase PloRAEngine

| | |
|-----------------------------|------|
| Nombre: PloRABackEnd | |
| Tipo de clase: controladora | |
| Atributo | Tipo |

Capítulo 3: Análisis y Diseño del sistema

| | |
|---------------------------|--|
| __interval__ | int |
| arrayObjects | array |
| arrayXML | array |
| Para cada responsabilidad | |
| Nombre: | initialize(URL) |
| Descripción: | Inicializa los parámetros necesarios de esta clase. |
| Nombre: | handleResponse(consulta, respuesta) |
| Descripción: | Las consultas y las respuestas correspondientes se almacenan en el PloRACache si no están activas en éste. Y se guarda en un arreglo el XML que corresponde con el conjunto de consultas formadas del contexto actual. |
| Nombre: | responseXML(XMLs) |
| Descripción: | De los XMLs almacenados por la función handleResponse() se conforma la estructura de las sugerencias que se van a mostrar al usuario. Antes de mostrarlas, se ordena con refresh(). |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|---|
| Nombre: | activeURL(URL) |
| Descripción: | Comprueba la existen de una url en el arreglo de objetos para economizar espacio y evitar datos duplicados. |
| Nombre: | sortArray(Objects) |
| Descripción: | Ordena los objetos que se van a mostrar mediante el valor del dato "score" que poseen. |
| Nombre: | handleServerError(event, error) |
| Descripción: | Controla los errores que se producen en las respuestas del servidor. |
| Nombre | startRAQuery() |
| Descripción | Obtiene las consultas, comprueba que estén almacenadas en la cache del agente, si no están, envía una petición al servidor y si están obtiene su respuesta que aparece en la cache. |
| Nombre: | getQueryFromContext() |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|---|
| Descripción: | Obtiene el contenido del editor de texto donde se encuentra situado el cursor y se crea el contenedor de consultas con la función <code>createQuerySet()</code> , y devuelve lo que contiene el contenedor. |
| Nombre: | <code>pause()</code> |
| Descripción: | Detiene las peticiones que se producen al servidor, para que el usuario pueda realizar otras acciones. |
| Nombre: | <code>resume()</code> |
| Descripción: | Inicia las peticiones al servidor cuando el usuario comienza a escribir en el editor de texto. |

Tabla No. 14: Clase PloRABackEnd

| | |
|-----------------------------|------|
| Nombre: PloRAQuerySet | |
| Tipo de clase: controladora | |
| Atributo | Tipo |

Capítulo 3: Análisis y Diseño del sistema

| | |
|---------------------------|--|
| querySet | array |
| especialChar | array |
| cantReal | int |
| Para cada responsabilidad | |
| Nombre: | createQuerySet(context) |
| Descripción: | Comprueba que la primera oración del contexto obtenido es exactamente igual a la primera oración del próximo contexto que el usuario elija, para de esta forma eliminar las consultas guardadas en el querySet actual y conformar otras consultas. |
| Nombre: | addQuerySet(consulta) |
| Descripción: | Verifica si la consulta obtenida no está en el arreglo de consultas activas y si no está la adiciona al principio del mismo. |
| Nombre: | activeQuery (consulta) |
| Descripción: | Verifica si la consulta está activa para no repetirla en caso de que se encuentre. |
| Nombre: | filterSentence (sentence) |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|---|
| Descripción: | Escoge las palabras que tienen más de 3 caracteres y los caracteres como: " ", ",", "!", son desechados para formar las consultas. |
| Nombre: | removeQuerySet(queryset) |
| Descripción: | Cada vez que el usuario cambie de contexto se deben cambiar las consultas obtenidas de ese contexto, con el objetivo de no repetir los resultados mostrados al usuario. |
| Nombre: | isEspecialChar(char) |
| Descripción: | Verifica si es un caracter especial contenido en el arreglo especialChar para así eliminarlo. |
| Nombre: | getQuerySet() |
| Descripción: | Devuelve el arreglo de consultas que van a ser guardadas en la cache, con las respuestas correspondientes. |
| Nombre: | addQueries(context) |
| Descripción: | Separa las oraciones que conforman el párrafo que brinda el contexto donde se encuentra el cursor. |

Tabla No. 15: Clase PloRAQuerySet

Capítulo 3: Análisis y Diseño del sistema

| | |
|-----------------------------|---|
| Nombre: PloRACache | |
| Tipo de clase: controladora | |
| Atributo | Tipo |
| ArrayCache | array(30) |
| Para cada responsabilidad | |
| Nombre: | initialize() |
| Descripción: | Crea un arreglo para almacenar objetos en la cache. |
| Nombre: | activeQueryCache(consulta) |
| Descripción: | Comprueba que la consulta está activa en la cache del agente. |
| Nombre: | addQueryAndResponse(consulta, respuesta) |
| Descripción: | Adiciona dada una consulta la respuesta correspondiente a ella en el arreglo, si no está en la cache. |
| Nombre: | responseQuery(consulta) |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|--|
| Descripción: | Obtiene la respuesta de una consulta dada. |
|--------------|--|

Tabla No. 16: Clase PloRACache

| | |
|---------------------------|--|
| Nombre: PloRAUI | |
| Tipo de clase: interfaz | |
| Atributo | Tipo |
| | |
| Para cada responsabilidad | |
| Nombre: | refresh(objects) |
| Descripción: | Muestra las sugerencias que aparecen en el arreglo de objetos. |
| Nombre: | addEvent(element, event, func) |
| Descripción: | Activa según el navegador que se emplea el evento que debe producirse dentro del editor. |
| Nombre: | onPressLinkObject(event) |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|---|
| Descripción: | Verifica el evento que se produce y de acuerdo al navegador envía los parámetros correspondientes a la función createLinkOnSelection(). |
| Nombre: | activeInterval() |
| Descripción: | Activa la función resume(). |
| Nombre: | clearUI() |
| Descripción: | Elimina las sugerencias que aparecen en la interfaz de usuario del PloRA. |

Tabla No. 17: Clase PloRAUI

| Nombre: EditorAdapter | |
|-----------------------------|--------|
| Tipo de clase: controladora | |
| Atributo | Tipo |
| ID_Kupu | string |
| browser | string |
| selection | string |
| Para cada responsabilidad: | |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|---|
| Nombre: | Initialize_Adapter(ID, URL) |
| Descripción: | Inicializa las funciones existentes en clases como PloRAEngine, PloRACache, así comienza el funcionamiento de agente JITIR. |
| Nombre: | getBrowserName() |
| Descripción: | Obtiene el nombre del navegador que se está utilizando. |
| Nombre: | getElement() |
| Descripción: | Controla el objeto iframe del editor de texto. |
| Nombre: | getHTMLDocument () |
| Descripción: | Obtiene el documento que está contenido en el objeto frame del editor. |
| Nombre: | getElementAtCaret() |
| Descripción: | De acuerdo al navegador activa las funciones de cada uno para obtener el texto escrito en el editor. |
| Nombre: | getMozillaCaret () |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|---|
| Descripción: | Obtiene el texto donde se encuentra el cursor para el navegador Mozilla. |
| Nombre: | getIECaret () |
| Descripción: | Obtiene el texto seleccionado por el usuario en el Internet Explorer. |
| Nombre: | createLinkOnSelection(url, título) |
| Descripción: | Utiliza la herramienta del kupu(linktool) para crear vínculos mediante la función createLink(url, type, null, target , título). |

Tabla No. 18: Clase EditorAdapter

| Nombre: QueryResponse | |
|-----------------------------|--------|
| Tipo de clase: controladora | |
| Atributo | Tipo |
| consulta | string |
| respuesta | string |
| Para cada responsabilidad: | |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|--|
| Nombre: | getQuery() |
| Descripción: | Obtiene una consulta. |
| Nombre: | getResponse() |
| Descripción: | Obtiene la respuesta de la consulta. |
| Nombre: | setQuery() |
| Descripción: | Cambia la consulta existente por una nueva. |
| Nombre: | setResponse() |
| Descripción: | Cambia la respuesta existente por una nueva. |

Tabla No. 19: Clase QueryResponse

| | |
|-----------------------------|--------|
| Nombre: PloRATool | |
| Tipo de clase: controladora | |
| Atributo | Tipo |
| id | string |
| Para cada responsabilidad: | |

Capítulo 3: Análisis y Diseño del sistema

| | |
|--------------|--|
| Nombre: | <code>_getCreatorName(self, id)</code> |
| Descripción: | Obtiene utilizando la herramienta 'portal_membership' que definida en Zope el nombre de los usuarios del sitio Plone activo con derechos de edición sobre las páginas. |
| Nombre: | <code>_getAuthorURL(self, id)</code> |
| Descripción: | Devuelve los archivos pertenecientes al usuario autenticado en el formulario de autenticación del Plone. |
| Nombre: | <code>ploraSearch(self, REQUEST=None, **kwargs)</code> |
| Descripción: | Realiza búsquedas en las bases de datos del Zope y devuelve su resultado en un formato XML. |
| Nombre: | <code>InitializeClass(PloRATool)</code> |
| Descripción: | Activa la clase para que sea reconocida como una herramienta en el ambiente de administración de Zope. |

Tabla No. 20: Clase PloRATool

Con el modelo de clases del diseño realizado y con las descripciones que corresponden a cada una de las clases, es muy fácil llegar a la etapa de la codificación de la aplicación: la implementación. Este flujo de trabajo abarca lo que corresponde a la escritura de

Capítulo 3: Análisis y Diseño del sistema

código fuente que ayuda completar prácticamente el trabajo teórico argumentado en los anteriores flujos de trabajo.

Tratamiento de errores

El tratamiento de errores en las aplicaciones, de manera general ayuda a validar las posibles respuestas que pueden darse en determinado momento por el sistema. Además, evita la revelación de datos sensibles de este o la ocurrencia de errores en tiempo de ejecución que afecten la imagen real del sistema.

En el caso particular del producto PloRA no es necesario controlar los errores que se produzcan en la interfaz del Plone(particularmente en el editor kupu). La funcionalidad de este agente JITIR es recuperar información del texto escrito por el usuario sin revisar si está escrito correctamente o no.

En la interfaz de usuario del PloRA no se introducen errores puesto que está diseñada para mostrar información sin que sea necesaria introducciones de valores que se presten a producir errores.

Interfaz

En epígrafes anteriores se explicó el significado de la utilización de una interfaz gradual(ramping interfaces, en inglés) en los agentes JITIR. Sin embargo, en nuestra tesis debemos aclarar que el empleo de una interfaz gradual propiamente dicha, está marcada para una versión superior del producto. Nos encontramos en la fase inicial, y a partir de esta se le brinda al usuario una información que aunque mínima, satisface las funciones elementales(muestra sugerencias y permite incluir estas como un enlace en un fragmento de texto del editor) del PloRA como agente recuperador de información.

Capítulo 3: Análisis y Diseño del sistema

Ayuda

El nuevo producto para Plone cuenta con una ayuda de orientación al usuario. Está constituida por una página FAQ online, y una interfaz incorporada a la interfaz de usuario del PloRA. Está destinada para orientar al usuario a la hora de seguir los pasos de utilización, para asegurarle un correcto funcionamiento. Además, en los aspectos de utilización quedan claros los métodos de empleo, teniendo en cuenta que es un producto que cumple con funcionalidades muy elementales.

Conclusiones

El análisis y el diseño realizados en este capítulo tuvieron como objetivo modelar y describir las clases que, primero, integran la estructura del producto Zope–Plone como base para nuestra propia ingeniería y luego esta propiamente dicha. Con estos modelos y descripciones se obtienen artefactos con datos suficientes para pasar a la implementación del producto.

Capítulo 4: Implementación, Pruebas y Resultados

Capítulo 4 Implementación y Pruebas

Introducción

En este capítulo se efectuarán los flujos de trabajo de implementación y prueba. Con las definiciones correspondientes a estos se pretende obtener el código fuente de la aplicación y en la parte perteneciente a las pruebas se realizarán una serie de pruebas que condicionen las funcionalidades con las que contará la aplicación.

4.1 Implementación

La implementación se inicia con el resultado del modelado del diseño. En esta etapa se implementa el sistema en términos de componentes(código fuente, scripts, ficheros de código binario, ejecutables, etc.).

Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean estos 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. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

Normalmente los diagramas de componentes se utilizan para modelar código fuente, versiones ejecutables, bases de datos físicas,

Capítulo 4: Implementación, Pruebas y Resultados

entre otros:

- ✓ Código fuente: En el modelado de código fuentes se suelen utilizar para representar las dependencias entre los ficheros de código fuente, o para modelar las diferentes versiones de estos ficheros. Para ello se deben identificar el conjunto de archivos de código fuente de interés y estereotiparlos como archivos file, agruparlos en paquetes, utilizar valores etiquetados para la información de versiones y modelar las dependencias de compilación.
- ✓ Código ejecutable: Se utiliza para modelar la distribución de una nueva versión a los usuarios. Para tal propósito se identifican el conjunto de componentes ejecutables que intervienen, se utilizan estereotipos para los diferentes tipos de componentes (ejecutables, bibliotecas, tablas, archivos, documentos, etc.), se consideran las relaciones entre dichos componentes que la mayoría de las veces incluirán interfaces que son exportadas (realizadas) por ciertos componentes e importadas (utilizadas) por otros.
- ✓ Bases de datos física: UML permite el modelado de bases de datos físicas así como de los esquemas lógicos de bases de datos.

Se pueden utilizar estereotipos como <<link>> o <<compile>> para distinguir la distinta naturaleza de las dependencias. Igualmente se pueden definir estereotipos para las distintas clases de componentes. UML proporciona algunos estereotipos predefinidos como: <<file>>, <<library>>, <<executable>>, <<table>>, <<scripts>> y <<document>>.

Capítulo 4: Implementación, Pruebas y Resultados

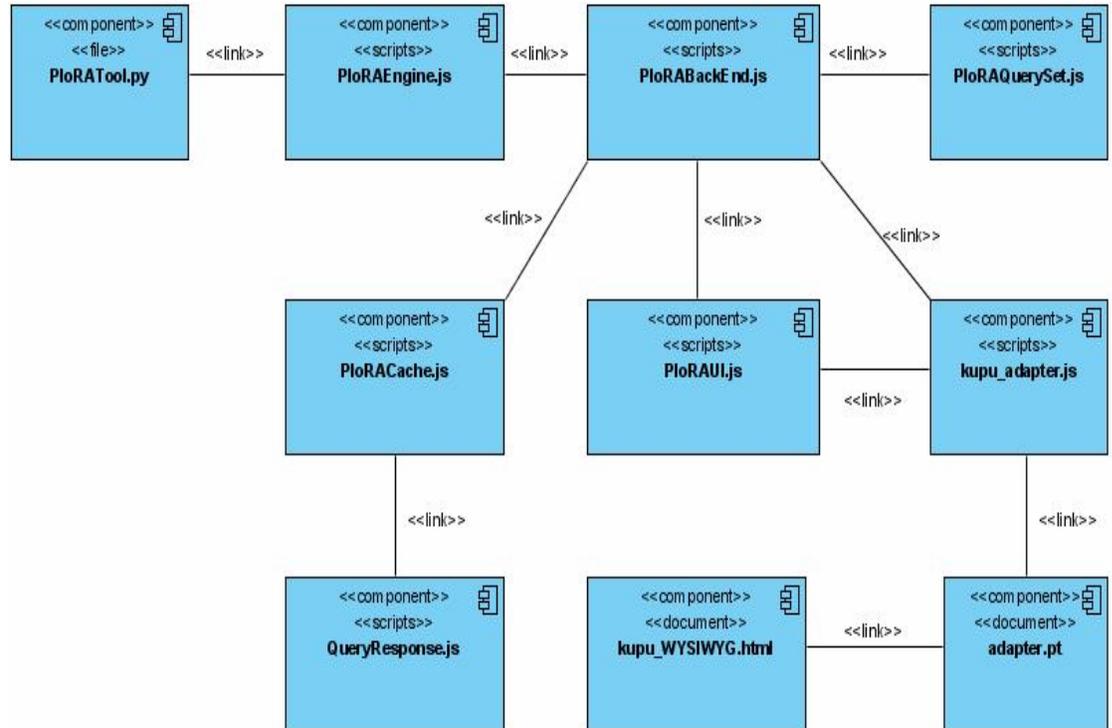


Figura No. 8: Diagrama de componentes

Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes).

Capítulo 4: Implementación, Pruebas y Resultados



Figura No. 9: Diagrama de despliegue

4.2 Prueba

Las pruebas que se le realizan a un software no aseguran la ausencia de errores en este, sin embargo, pueden demostrar que existen deficiencias en la implementación.

El modelo de pruebas describe cómo se prueban los componentes ejecutables en el modelo de implementación con pruebas de integración y de sistema. Además, es la colección de casos de pruebas, procedimientos de prueba y componentes de prueba.

Los casos de prueba especifican una forma de probar el sistema, incluyendo la entrada, resultados y condiciones bajo las que ha de probarse.

Descripción de los casos de prueba para el CU: Escribir_texto

| Entrada | Resultado |
|---|--|
| El redactor de noticias no escribe en el editor kupu. | PloRA no recupera información del servidor y por tanto no muestra sugerencias. |

Capítulo 4: Implementación, Pruebas y Resultados

| | |
|--|---|
| El redactor de noticias escribe en el editor kupu. | La interfaz del PloRA muestra sugerencias obtenidas del servidor correspondientes a las consultas realizadas y los documentos relacionados con estas. |
| El redactor de noticias escribe en el editor kupu. | La interfaz del PloRA no muestra sugerencias del texto escrito puesto que no se encontraron documentos relacionados en el servidor. |

Tabla No. 21: Caso de prueba para el CU: Escribir_texto

Descripción de los casos de prueba para el CU: Procesar consultas

| Entrada | Resultado |
|--|---|
| Se cumple el intervalo de reloj cada 5 segundos y se activa la función startQuery(). | Se obtiene el contexto donde está el cursor para Procesar consultas. |
| Dentro del intervalo de 5 segundos. | No solicita un nuevo contexto para mantener las sugerencias dadas hasta que se inicie el intervalo de tiempo. |

Tabla No. 22: Caso de prueba para el CU: Procesar consultas

Capítulo 4: Implementación, Pruebas y Resultados

Descripción de los casos de prueba para el CU: Insertar vínculo

| Entrada | Resultado |
|---|---|
| El usuario hace clic en una sugerencia pero no ha seleccionado un fragmento texto en el editor. | Se inserta un vínculo en la posición del cursor con el mismo nombre con que aparece la sugerencia en la interfaz. |
| El usuario selecciona un fragmento del texto y hace clic en una sugerencia. | Se inserta un vínculo correspondiente a la sugerencia en el texto seleccionado. |

Tabla No. 23: Casos de prueba para el CU: Insertar vínculo

4.3 Resultados de la aplicación

Cuando se terminan las fases de implementación y prueba de un producto se pasa a entregarle al usuario la aplicación para que la comience a emplear. Este usuario siempre espera un resultado favorable del funcionamiento de la nueva aplicación. De este tema tratará este epígrafe. Los resultados que el agente PloRA ha provocado en los usuarios para los que fue creado.

4.3.1 Impacto del agente PloRA en el Proyecto de Prensa

Como un primer acercamiento a los resultados esperados de la nueva aplicación para Plone, se conformó y aplicó una encuesta a algunos trabajadores de periódicos del Centro de Prensa Nacional, con el objetivo de recoger estados de opiniones sobre el nuevo producto. La encuesta no cuenta con una metodología estadística definida, pero considera los criterios de los usuarios para describir la utilidad del

Capítulo 4: Implementación, Pruebas y Resultados

producto.

Los resultados arrojados por la encuesta dan una primera idea del trabajo realizado por cada uno de los encuestados que participó en la interacción con el nuevo producto de Plone. Aunque se encuestaron solo 9 periodistas, con sus respuestas se logró dar por cumplidos muchos de los objetivos y características propias del producto y que solo se obtendrían con el análisis y uso del usuario final. Entre los criterios relevantes se haya la utilidad de encontrar materiales importantes que se puedan relacionar con el contexto editado. También se evidenció el ahorro de tiempo en obtener información relevante con respecto al empleado en encontrarla utilizando un buscador Web, y esto en el mejor de los casos, ya que podría suceder que no se encuentre tal información. Además, se logró que la interfaz fuera de fácil manejo y no intrusiva o que provocara distracciones innecesarias al usuario mientras realizaba su labor, a pesar de encontrarse en la primera fase del producto(Anexo 1).

Con estos criterios consideramos que la introducción del PloRA en el Centro de Prensa Nacional, y específicamente en los periódicos Juventud Rebelde, Trabajadores y Granma Internacional surtió un efecto positivo entre sus usuarios, y lo más importante constituye la reducción de la información desconectada entre las publicaciones periodísticas existentes.

Conclusiones

Concluido este capítulo queda implementada la aplicación que se planteó en los objetivos de nuestro trabajo. Además, cuenta con la documentación pertinente con la inclusión de los respectivos diagramas representativos.

Conclusiones

En este documento hemos plasmado y cumplido los objetivos trazados en el trabajo de diploma para optar por el título de ingenieros en ciencias informáticas. Nos propusimos desarrollar una aplicación vinculada al proyecto de informatización de la Prensa Nacional, que ayudara a sus redactores a enlazar todas aquellas noticias que se vincularan y que estuvieran desconectadas. Este objetivo se cumplió eficazmente con la elaboración de un producto, que aunque se encuentra en su primera versión, brinda los servicios para los que fue creado.

También se trazaron objetivos específicos, a los cuales se le dio cumplimiento en el transcurso del desarrollo de la aplicación. La construcción del marco teórico de la utilización de los agentes JITIR a partir del estudio del estado del arte, se materializó con la consulta a múltiples bibliografías que ampliaron los conocimientos relacionados con el tema de la utilidad los JITIR en el mundo, y específicamente en las páginas Web.

Se logró implementar la aplicación con el nombre PloRA: Plone Remembrance Agent. La codificación se realizó con el empleo de JavaScript como lenguaje de programación, aunque algunos ficheros fueron implementados en Python para lograr que el producto se vincule con el CMS Plone y con Zope.

Con la introducción del nuevo producto para Plone en el Centro de Prensa Nacional, se logró un primer acercamiento al resultado esperado por parte de los autores de este trabajo. El impacto alcanzado con la utilización del PloRA fue renovador para los periodistas. Las consideraciones recogidas de parte de los periodistas directamente, nos brindaron la gratitud por la incorporación a su labor

Conclusiones

diaria, de un artefacto que les ahorra tiempo y esfuerzos para encontrar información importante.

Con la realización de este trabajo, los autores aprendimos un tema completamente desconocido y es el de “información desconectada” en las publicaciones periodísticas. Profundizamos en la ingeniería del CMS Plone y Zope, que aunque nos habíamos iniciado en el trabajo con estos, desconocíamos los detalles de muchas partes de la implementación. El empleo del JavaScript y el Python como lenguajes de programación para implementar una aplicación que debía integrarse a páginas de edición Web.

Recomendaciones

Se recomienda a los seguidores de esta idea renovadora en la universidad que continúen con el estudio de las tendencias de los agentes JITIR en el mundo, y a la vez con el desarrollo de esta herramienta creada en el marco de esta tesis. La introducción de agentes recuperadores de información en otros sectores laborales puede ser de significativa importancia.

En el caso del desarrollo de esta aplicación, recomendamos que se amplíe el uso del PloRA con la inclusión del editor FCKEditor, y otros editores de texto de la Web, como parte de próximas versiones de este producto. Además, la intención es incrementar las facilidades de uso para los usuarios y que puedan emplear el editor que más agradable les sea.

Bibliografías citadas

1. BROWN, P. J. y BROWN, H. *Integrating Reading and Writing of Documents.* Disponible en: <http://jodi.ecs.soton.ac.uk/Articles/v05/i01/Brown/?printable=1>.
2. B.J.RHODES, P. M. *Just-in-time information retrieval agents.* IBM SYSTEM, 2000, vol. 39, p. 690.
3. CHRISTOPHER D. MANNING, P. R., HINRICH SCHÜTZE. *The Information Retrieval Book.* July 8, 2006. p. 5.
4. RAQUEL BLANCH CLAVER, R. M. G. V., DANIEL LAGUNAS MONTAÑANA, VANESA MILLA ASTORGA, JOSEPH ANTONI NÀCHER CARDO. *Just-in-time Information Retrieval Agents.* Disponible en: http://personales.upv.es/ccarrasc/doc/2004-2005/JosepANacheretal/proyecto_SRP.htm#_Toc104949942.
5. FOUNDATION, P. *What is Plone?* Disponible en: <http://plone.org/about/plone>.
6. OSCOM. *OSCOM Kupu.* 2001-2006, Disponible en: <http://kupu.oscom.org/>.
7. POSNER, M. *Foundations of Cognitive Science.* 1989,
8. FOUNDATION, W. *World Wide Web.* 2007, Disponible en: <http://es.wikipedia.org/wiki/Web>.
9. ZIPF, G. K. *Human Behavior and the Principle of Least Effort.* 1949. p. 39-40.
10. MÉNDEZ, F. J. M. *Modelo Espacio Vectorial.* marzo 22, 2005, Disponible en: <http://irsweb.blogspot.com/2005/03/el-modelo-del-espacio-vectorial-i.html>.
11. JIMÉNEZ, J. A. A. *Sistema de gestión de contenidos web, distribuido con herramientas síncronas de colaboración: BABEL.* Julio, 2004, p 62.
12. FOUNDATION, W. *Ajax.* 2007, Disponible en: <http://es.wikipedia.org/wiki/AJAX>.
13. ---. *Adapter pattern* Disponible en: http://en.wikipedia.org/wiki/Adapter_pattern.
14. POZO, J. R. *La estructura de un documento HTML.* junio 21,

Bibliografías citadas

- 2001, Disponible en: <http://html.conclase.net/tutorial/html/2/4>.
15. B.J.RHODES, P. M. *Just-in-time information retrieval agents*. 2000. p. 73.
 16. FOUNDATION, W. *Ingeniería de Software*. marzo 24, 2007, Disponible en: [http://es.wikipedia.org/wiki/Ingeniería de software](http://es.wikipedia.org/wiki/Ingeniería_de_software).
 17. ---. *Ingeniería Inversa*. abril 13, 2007, Disponible en: [http://es.wikipedia.org/wiki/Ingeniería inversa](http://es.wikipedia.org/wiki/Ingeniería_inversa).
 18. EILAM, A. *Reversing: Secrets of Reverse Engineering*. Wiley Publishing, Inc., 2006, p 11.
 19. ENFOLDSYSTEMS. *The Definitive Guide to Plone*. First edition ed. may 16th, 2005. p. 314.
 20. AMOS LATTEIER, M. P., CHRIS MCDONOUGH, PETER SABAINI. *The Zope Book 2.6*. p. 158.
 21. FOUNDATION, W. *Proceso Unificado de Rational* Disponible en: [http://es.wikipedia.org/wiki/Proceso Unificado de Rational](http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational).
 22. RUMBAUGH, I. J. G. B. J. *El proceso unificado de desarrollo de software*. 2004. vol. I, p. 180.

Bibliografías consultadas

B.J.RHODES, P. M. Just-in-time information retrieval agents. *IBM SYSTEM*, 2000, vol. 39, p. 690.

Just-in-time information retrieval agents. 2000. p. 73.

CHRISTOPHER D. MANNING, P. R., HINRICH SCHÜTZE. *The Information Retrieval Book*. July 8, 2006. p. 5.

FOUNDATION, W. *Ajax*. 2007, Disponible en: <http://es.wikipedia.org/wiki/AJAX>.

World Wide Web. 2007, Disponible en: <http://es.wikipedia.org/wiki/Web>.

MÉNDEZ, F. J. M. *Modelo Espacio Vectorial*. marzo 22, 2005, Disponible en: <http://irsweb.blogspot.com/2005/03/el-modelo-del-espacio-vectorial-i.html>.

OSCOM. OSCOM Kupu. 2001-2006, Disponible en: <http://kupu.oscom.org/>.

POSNER, M. *Foundations of Cognitive Science*. 1989,

RAQUEL BLANCH CLAVER, R. M. G. V., DANIEL LAGUNAS MONTAÑANA, VANESA MILLA ASTORGA, JOSEPH ANTONI NÀCHER CARDO. *Just-in-time Information Retrieval Agents*. Disponible en: http://personales.upv.es/ccarrasc/doc/2004-2005/JosepANacheretal/proyecto_SRP.htm#_Toc104949942.

ZIPF, G. K. *Human Behavior and the Principle of Least Effort*. 1949. p. 39-40.

EILAM, A. *Reversing: Secrets of Reverse Engineering*. Wiley

Bibliografías consultadas

Publishing, Inc., 2006, p 11.

ENFOLDSYSTEMS. *The Definitive Guide to Plone*. First edition ed. may 16th, 2005. p. 314.

FOUNDATION, W. *Adapter pattern* Disponible en: http://en.wikipedia.org/wiki/Adapter_pattern.

Ingeniería Inversa Disponible en: http://es.wikipedia.org/wiki/Ingeniería_inversa.

Proceso Unificado de Rational Disponible en: http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational.

Ingeniería de Software. marzo 24, 2007, Disponible en: http://es.wikipedia.org/wiki/Ingeniería_de_software.

JIMÉNEZ, J. A. A. *Sistema de gestión de contenidos web, distribuido con herramientas síncronas de colaboración: BABEL*. Julio, 2004, p 62.

POZO, J. R. *La estructura de un documento HTML*. junio 21, 2001, Disponible en: <http://html.conclase.net/tutorial/html/2/4>.

RUMBAUGH, I. J. G. B. J. *El proceso unificado de desarrollo de software*. 2004. vol. I, p. 180.

WIKIPEDIA. *Archetype*. febrero 12, 2007, Disponible en: http://en.wikipedia.org/wiki/Archetype_%28information_science%29.

AMOS LATTEIER, M. P., CHRIS MCDONOUGH, PETER SABAINI. *The Zope Book 2.6*. p. 158.

FOUNDATION, P. *What is Plone?* Disponible en: <http://plone.org/about/plone>.

Bibliografías consultadas

ENFOLDSYSTEMS. *The Definitive Guide to Plone*. First edition ed. may 16th, 2005. p. 45, 60, 75

LARMAN, C. *UML y Patrones*. vol. I, II, p. 89, 124

PRESSMAN, R. S. *Ingeniería de Software: un enfoque práctico*. vol. I, II, p. 15, 300

RAQUEL BLANCH CLAVER, R. M. G. V., DANIEL LAGUNAS MONTAÑANA, VANESA MILLA ASTORGA, JOSEPH ANTONI NÀCHER CARDÓ. *Just-in-time Information Retrieval Agents*. Disponible en: http://personales.upv.es/ccarrasc/doc/2004-2005/JosepANacheretal/proyecto_SRP.htm#_Toc104949942.

RUMBAUGH, I. J. G. B. J. *El proceso unificado de desarrollo de software*. 2004. vol. I, II, p. 37, 55, 91

Glosario de términos

Adaptador: Los adaptadores son patrones de diseño, en los cuales se representan soluciones generales a problemas comunes en el diseño de software.

AJAX: (Asynchronous JavaScript And XML) tecnologías que posibilitan interacciones entre cliente y servidor sin manejo de datos innecesarios en el transporte de peticiones y respuestas.

Cache: tipo de memoria temporal utilizada para almacenar consultas y respuestas relacionadas con un contexto determinado.

CMS: Content Management Systems, Sistema de Gestión de Contenidos. Estos sistemas se encargan de organizar, almacenar, crear o eliminar contenidos de información.

Información Desconectada: terminología referida a la total desvinculación existente entre publicaciones periodísticas digitales que tratan acerca de un mismo tema.

Information Retrieval: IR, recuperación de información. Trata de obtener el contenido de la información que puede ser relevante para los usuarios.

Intérprete: Dígase intérprete al programa capaz de analizar y ejecutar programas escritos en un lenguaje de alto nivel.

JITIR Agent: agente de recuperación de información Just-in-time. Estos agentes son capaces de obtener información del contexto local del usuario y mostrar información relacionada con este, que se encuentra almacenada en bases de datos, e-mails, carpetas personales, etc.

Anexo1

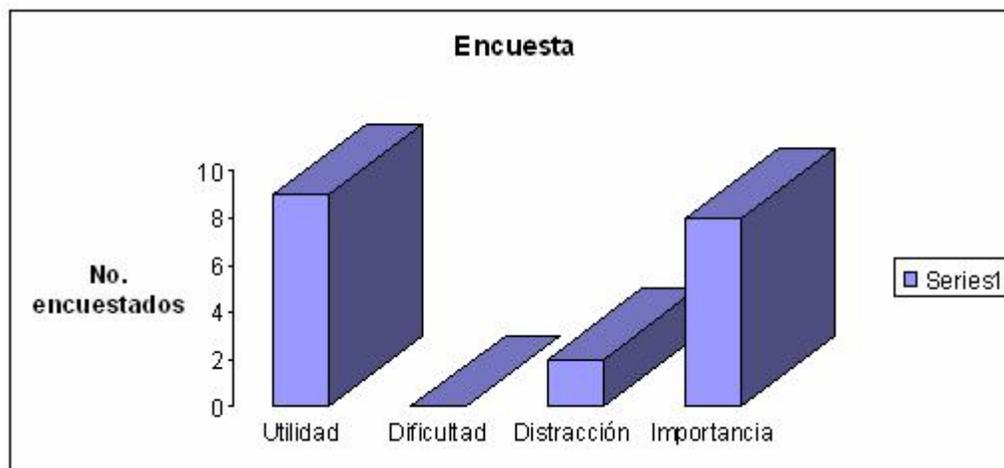


Figura No. 10: Primera encuesta a los periodistas del CIP

