Universidad de las Ciencias Informáticas

Facultad 1

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.

Componente para la extracción semántica de información para Orión.

Autor: Armando Pino Cárdenas

Tutores: Ing. Disnayle Jorge Chacón

Ing. José Carlos Ledesma Romero

"Año 58 de la Revolución"

Ciudad de La Habana, Cuba, junio, 2017.

Declaración de auditoría

Declaro por este medio que yo Armando Pino Cárdenas, con carné de identidad 93091134663 soy el autor principal del trabajo titulado “Componente para la extracción semántica de información para Orión” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo”. Para que así conste se firma a los __ días del mes de_____ del año _____.

Autor:

_____

Armando Pino Cárdenas

Tutores:

_____ _____

Ing. Disnayle Jorge Chacón Ing. Jose Carlos Ledesma Romero

"El tema relativo al conocimiento y la tecnología es de especial relieve en nuestra agenda, porque en él abordamos los problemas que deciden, en buena medida, el futuro de nuestros países."

Fidel Castro Ruz

Agradecimientos:

A mi mamá y papá que sin importar la situación, momento o lugar siempre han estado ahí para mí dándome todo su apoyo. Los quiero con la vida y para ustedes va dedicado este trabajo.

A mis tías, tíos y primos que también lo han dado todo por mí y se han sacrificado como si fueran mis padres para que yo pudiera avanzar y salir adelante.

A mí novia Danisleydis por estar siempre a mi lado y darme la oportunidad de compartir todas las cosas buenas y malas de la vida, esas cosas que quisiera seguir compartiendo junto a ella por el resto de mis días. Te amo mi negra.

A mi madrastra Arletys que desde que la conozco ha sido como mi segunda madre, siempre muy pendiente de mí, también te quiero mucho.

A mi suegra Miroslava con la que he compartido muy buenos momentos y desde el primer momento me acepto en su casa sin importar que yo fuera de Jovellanos y a su madre Norma abuela de mi novia, quién me abrió las puertas de su casa y a la cual quiero como la abuela que no tengo.

A mi gente de Jovellanos Nano, Lázaro Yoan, Yoan y Maikel que los quiero como hermanos y muchos me han apoyado y aconsejado en esta etapa de mi carrera y aunque ahora estamos un poco lejos ellos saben que los quiero.

A mis compañeros de aula que vienen conmigo desde los inicios aguantando mis pesadeces, gracias por siempre hacerme reír.

A Dayana, Ariadne y María que son las hermanas que me regalo esta universidad, a las tres las adoro y las quiero mucho.

A Yasiel la persona que nunca pensé que podría ser para mí lo que es hoy, mi hermano.

A Lázaro la persona que moleste incluso más que a mis tutores, gracias por siempre brindarme tu ayuda mi hermano eso te lo agradeceré hasta el final de mi vida, te quiero.
A mi gente de la facultad 2 Ángela, Elizabeth, Dianelis, Yuneisy las quiero.
A mi gente del equipo de futbol 11 de la UCI y de la facultad 1 también muchas gracias por los buenos momentos que fueron muchos.
Y por último pero no menos importante a mis tutores Jose Carlos y Disnayle que nunca perdieron la paciencia conmigo, nunca se rindieron y realmente sin su apoyo este trabajo no se hubiese desarrollado, muchas gracias por todo.

Resumen

Con el objetivo de contribuir al desarrollo de los servicios prestados a las personas cuando interactúan con el buscador Orión desarrollado en el país, se desarrolló un componente que pretende mejorar las respuestas de las consultas realizadas por los usuarios a través de un análisis semántico de la información. Este permitirá realizar un procesamiento del lenguaje natural con el objetivo de interpretar el significado semántico de las frases y palabras que se encuentran en los documentos indexados en la base de datos. Para el desarrollo e implementación de la propuesta de solución se seleccionaron como principales tecnologías: Apache Solr como mecanismo de indexación, Apache Tomcat como servidor web, Java como lenguaje de programación, Netbeans como Entorno de Desarrollo Integrado (IDE), Visual Paradigm como herramienta para el modelado y AUP en su versión para la Universidad de la Ciencias Informáticas como metodología de desarrollo. El componente implementado posee una serie de funcionalidades que posibilitan que el procesamiento la información almacenada en Solr sea más efectivo logrando una mayor exactitud en los resultados que se muestran a los usuarios.

Palabras clave: buscador web, información, lenguaje natural, procesamiento, semántico, sistema de recuperación de información.

Índice

Introducción

En la actualidad la web se ha convertido en un instrumento de gran importancia para los seres humanos. La cantidad de información que se encuentra disponible en la red es uno de los principales elementos por lo cual la misma se ha vuelto indispensable para la sociedad. Con el uso de la web se pueden realizar diferentes actividades como la búsqueda de información, la compra y venta de productos en línea y la comunicación con otras personas en el mundo a través de las redes sociales entre otras de forma cómoda, eficiente y económica. Con el rápido crecimiento que ha mostrado la web, a los consumidores de la información situada en internet se les dificulta la gestión de toda esa información, según Clay Shirky el problema fundamental reside en que se genera más información de la que se puede gestionar.

El empleo de buscadores para realizar tareas de Recuperación de Información (RI) es uno de los recursos más utilizados por las personas. Con ellos se puede encontrar casi todo lo que se busca, pero todavía muestran limitaciones, esto se debe a que la web actual presenta una serie de barreras que no permiten que el proceso de búsqueda de información para los usuarios sea realmente efectivo. Por ejemplo, la web actual fue diseñada para que sea comprendida por los humanos y no por los programas. Además es masiva ya que contiene millones de páginas y sitios web que brindan información de todo tipo a las personas; es cambiante, a cada momento se generan grandes volúmenes de datos, ya sean documentos, actualizaciones de perfiles de Facebook o comentarios en páginas y blogs. Esta web también es heterogénea ya que las organizaciones generan gran cantidad de información de forma independiente (Arenas y otros, 2015).

La propuesta para superar los desafíos que impone la web actual es la web semántica, que busca principalmente resolver los problemas de que la web esté hecha para humanos y de heterogeneidad de información, para que los contenidos puedan ser consumidos por las máquinas de forma más eficiente. La implementación de una web más avanzada constituye un gran cambio ya que tiene que transformarse una web creada en lenguaje natural en una web estructurada y organizada donde los contenidos etiquetados semánticamente serán el elemento principal (Carmen , 2009). Para disponer de una web de datos efectiva es necesario tener un lenguaje que pueda ser procesado por un computador, sea capaz de describir los recursos en la web y las relaciones entre ellos (Lapuente, 2001).

En Cuba con el desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs) a los usuarios se les dificulta el acceso a la información que desean obtener dentro del dominio cubano, la realidad es que aunque se dispone de diferentes directorios temáticos y de la plataforma Contenidos Unificados de Búsqueda Avanzada (C.U.B.A) no se cuenta con un buscador que tenga la capacidad de

realizar un correcto análisis de la información que se encuentre en sitios y páginas web para satisfacer las necesidades de información de los usuarios en el país.

En la Universidad de las Ciencias Informáticas (UCI) en la Facultad 1 se encuentra el Centro de Ideoinformática (CIDI). En dicho centro se encuentra en etapa de desarrollo el buscador Orión, un motor de búsqueda creado en sus inicios para la RI en la red de la UCI y que en estos momentos su búsqueda ya ha sido extendida sobre el dominio cubano (.cu). Este mecanismo para la recuperación de la información en la intranet todavía presenta algunas limitaciones, entre ellas se encuentra el bajo nivel de análisis de la información en formato texto ya que el sistema no posee la capacidad de realizar un análisis semántico de la información que se encuentra indexada en su base de datos para un mejor procesamiento de la misma. El buscador reconoce los documentos como un conjunto de palabras y dentro de ellas identifica los términos más relevantes, pero es incapaz de establecer una conexión entre dichas palabras e identificar el significado que puedan brindar cuando se unen para conformar una frase. Esto provoca que en ocasiones los resultados de las búsquedas realizadas al sistema no sean los esperados y a sus vez la insatisfacción de los usuarios cada vez que interactúan con este Sistema de Recuperación de Información (SRI).

Por lo planteado anteriormente surge el siguiente **problema de investigación** ¿Cómo mejorar la recuperación de información en formato texto que realiza Orión?

Del problema expuesto se identifica como **objeto de estudio**: procesamiento semántico de la información.

El **campo de acción** se enmarca en el procesamiento semántico de información para Orión.

Se define como **objetivo general**: Desarrollar un componente para la extracción semántica de información de la base de datos de indexación Apache Solr utilizando el procesamiento de lenguaje natural para mejorar la calidad de respuestas a los usuarios.

Del objetivo general se derivan los siguientes **objetivos específicos**:

- Elaborar el marco teórico conceptual y el estudio del estado del arte respecto a las tecnologías y funcionalidades de los sistemas de recuperación de información.
- Definir las tecnologías y la metodología para desarrollar la propuesta de solución.
- Realizar el análisis y diseño de la propuesta de solución.
- Desarrollar un componente que permita la extracción de información semántica del sistema de recuperación de información Orión.

- Validar el correcto funcionamiento del componente desarrollado.

Teniendo en cuenta lo anterior, se plantea la siguiente **hipótesis de investigación**: La implementación de un componente de extracción semántica de información para Orión mejorará la calidad de respuestas a las consultas de los usuarios. De la hipótesis planteada se define como **variable independiente**: Componente de extracción de información. Como **variable dependiente** se especifica: Calidad de respuesta a las consultas de los usuarios.

Tabla 1. Operacionalización de variable independiente.

Variable Independiente	Dimensión	Indicadores	Unidad de medida
Componente de extracción de información	Funcionamiento del Sistema	Integración del Sistema con Solr	Buena Mala

Tabla 2. Operacionalización de variable dependiente.

Variable Dependiente	Dimensión	Indicadores	Unidad de medida
Calidad de respuesta a las consultas de los usuarios.	Usuarios Finales	Conformidad con las funcionalidades	Alta Media Baja
		Tiempo de respuesta	

Con el fin de dar cumplimiento al objetivo general y a los objetivos específicos se plantearon las siguientes **tareas de investigación:**

- Familiarización con las principales herramientas de extracción semántica y los SRI.
- Análisis de los problemas actuales que presentan los SRI.
- Definición de las herramientas y tecnologías informáticas a utilizar para desarrollar el componente.
- Identificación de los requisitos a partir de las necesidades del cliente.
- Documentación de los requisitos utilizando el lenguaje y las plantillas definidas.
- Implementación del componente para extraer la información semántica.
- Realización de pruebas funcionales para validar el correcto funcionamiento del sistema desarrollado.

Para el desarrollo de las tareas se utilizaron los siguientes métodos científicos:

Analítico-Sintético: este método permitió la recopilación de información necesaria durante la realización del estudio del estado del arte para el desarrollo del trabajo mediante la revisión de documentos y artículos, de donde se extrajeron los elementos más significativos relacionados con el proceso de búsqueda de información. Además del análisis de las diferentes herramientas, metodologías y tecnologías a utilizar en el desarrollo del sistema.

Histórico-Lógico: fue utilizado en el análisis de la evolución de sistemas similares, de esta manera se indagó sobre los rasgos que caracterizan a estos sistemas y en aspectos para fundamentar la propuesta de solución a la problemática planteada.

Modelación: este método fue utilizado en la representación, mediante el uso de diagramas, de las características del sistema a desarrollar, relaciones entre objetos; y las actividades que intervinieron en el proceso de configuración del sistema para la recuperación de información.

Entrevista: este método de investigación fue utilizado para obtener información de forma directa con el cliente para determinar las herramientas que se utilizarían y levantamiento de requisitos funcionales y no funcionales que presenta el sistema.

Estructura de capítulos:

El contenido a desarrollar en el presente trabajo está estructurado en tres capítulos:

Capítulo No. 1: **Fundamentación teórica del Componente de extracción semántica de información para Orión.**

Se definen un conjunto de conceptos que permitirán obtener una panorámica del contexto de la investigación, incluyendo un estudio sobre el estado del arte del tema que se abordó con el objetivo de brindar una solución a la problemática planteada en la presente investigación. Finalmente se expondrán las herramientas y metodologías utilizadas en el desarrollo de la herramienta.

Capitulo No. 2: Análisis y diseño del Componente de extracción semántica de información para Orión.

Se exponen las características de la propuesta, definiendo los elementos técnicos de la misma: los patrones, el diseño de clases, los medios empleados, el modelo de despliegue de la arquitectura, las restricciones de diseño así como otros elementos.

Capitulo No. 3: Evaluación y aceptación de la arquitectura para el Componente de extracción semántica de información para Orión.

Se realiza una evaluación basada en una demostración que explica cómo la propuesta de solución debe dar garantías de que la solución diseñada es realizable de acuerdo con los atributos de calidad.

Capítulo 1: Fundamentación teórica del componente para extracción semántica de Información para Orión.

1.1 Introducción

En el presente capítulo se realiza un estudio del estado del arte acerca de los Sistemas de Recuperación de Información (SRI) y la Recuperación de Información (RI). También se realiza un estudio sobre la introducción de la web semántica y su relación con el procesamiento del lenguaje natural. Se profundizó en temas que requieren consulta obligatoria como son las herramientas, tecnologías, lenguajes y metodologías que se utilizaron en la investigación.

1.2 Recuperación de Información

Según el diccionario Mac Milan de tecnología de la información se considera la recuperación de información como "las técnicas empleadas para almacenar y buscar grandes cantidades de datos y ponerlos a disposición de los usuarios" (Méndez, 2004).

Después de haber consultado el concepto anterior el autor de la presente investigación determina como RI las técnicas utilizadas para mostrar y satisfacer las necesidades de información de un usuario que realiza una consulta en lenguaje natural a un SRI.

1.2.1 Operaciones de la Recuperación de Información

La recuperación de información consta de siete operaciones, las cuales están automatizadas en algún grado pero ninguno lo está en modo óptimo (Abadal, y otros, 2005).

- ✓ **Indización**: esta operación, en particular cuando se realiza en modo intelectual, se divide en realidad en otras dos:
    - **Análisis:** identificación de los temas o conceptos más relevantes del documento.
    - **Normalización**: transformación de los conceptos que expresan el contenido del documento en los términos de indización (descriptores) más adecuados. A veces, esta segunda fase recibe también el nombre de indización, obviando o dando por supuesto a la primera.

La indización puede aplicarse también a la necesidad de información. Se puede hablar, por tanto, de indización de documentos y de indización de la pregunta. En ambos casos, el resultado es un conjunto

de descriptores. En el caso de la necesidad de información, los descriptores de la pregunta pueden estar relacionados con operadores lógicos (operadores booleanos).

- ✓ **Selección:** identificación del conjunto de documentos más relevante para una necesidad de información dada. También se denomina recuperación (en este caso, debido a que es la parte más significativa del proceso, a menudo sirve para dar nombre al todo).
- ✓ **Ordenación**: determinación del orden más adecuado de presentación al usuario de los documentos seleccionados o recuperados (en caso que sean más de uno). La idea es ofrecer la lista de los documentos en orden decreciente (el más relevante primero) de probabilidad de satisfacer la necesidad de información. También se denomina ranking.
- ✓ **Interconexión:** establecimiento de relaciones hipertextuales, caminos y, en general, estructuras de navegación entre secciones del mismo documento o entre documentos distintos.
- ✓ **Categorización:** asignación de cada documento a un grupo, clase o subclase de un cuadro de clasificación, taxonomía u ontología.
- ✓ **Abstracción:** producción de resúmenes de documentos que, en algunas circunstancias, puedan sustituir la lectura del documento completo.
- ✓ **Visualización:** representación en forma gráfica de informaciones no necesariamente icónicas, así como de conceptos o procesos.

Para realizar actividades de recuperación de información es necesario disponer de herramientas que faciliten el trabajo a la hora de obtener una información, ya que la web es extensa y contiene un gran volumen de información. A continuación se presentan los sistemas encargados de realizar estas tareas.

1.2.2 Sistema de Recuperación de Información

Según Baeza-Yates y otros autores (2005) “los SRI deben de alguna manera interpretar el contenido de la información dentro de una colección de documentos y establecer con ellos, un orden de acuerdo al grado de relevancia que estos posean para las consultas de los usuarios”. Los SRI responden a una necesidad de información donde el usuario realiza una consulta por medio de una interfaz y este debe ser capaz de mostrar la mayor cantidad de resultados con el fin de satisfacer sus necesidades.

1.2.3 Arquitectura de un Sistema de Recuperación de Información

Los SRI son mecanismos informáticos, mediante los que se puede acceder a la información almacenada previamente en internet. Generalmente los sistemas de recuperación de información comparten la misma arquitectura (Herrera, 2006). A continuación se describirán dichos componentes:

Interfaz: un usuario con necesidades de información bien definidas, interactúa con la interfaz del sistema, mediante la cual introduce las consultas al mismo. La interfaz puede estar basada en una interfaz web (la más común), una interfaz de escritorio o ambas.

Sistema de Formulación de Consultas: realiza el pre-procesamiento trasladando las consultas hechas en lenguaje natural a consultas entendibles por los sistemas de información.

Mecanismo de evaluación de consultas: compara los documentos representados en el sistema de información, con la consulta pre-procesada para obtener un subconjunto de documentos relevantes que satisfagan la consulta introducida por el usuario, ordenados estos de acuerdo a un criterio de relevancia.

Mecanismo de rastreo: componente que se encarga de rastrear la web siguiendo la estructura hipertextual de la misma para almacenarlos en un lugar para su posterior análisis, en muchas ocasiones es llamado también araña o araña web.

De los componentes antes mencionados la propuesta de solución se integrará al mecanismo de evaluación de consultas ya que por medio de un criterio de búsqueda es que serán analizados los documentos.

1.2.4 Clasificación de los Sistemas de Recuperación

Existen varios tipos de SRI, atendiendo a diferentes criterios se clasifican de acuerdo a sus funcionalidades por ejemplo: el tipo de documentación que se busque. Entre las clasificaciones de los SRI se encuentran los buscadores, los metabuscadores y los directorios.

Directorios

Los directorios, o índices temáticos, son herramientas que organizan las páginas web jerárquicamente, o sea, permiten organizar la web por temas, lo que facilita la búsqueda de la información existente en un área determinada del conocimiento. Los resultados son recorridos en profundidad, lo que garantiza que al final de la jerarquía, exista una alta probabilidad de encontrar lo que realmente se necesita. Además, estos sistemas no poseen una araña u otro mecanismo automático que recorra la web en busca de nueva información como suele suceder con los motores de búsqueda, sino que es operado por humanos (KIVA, 2009).

Metabuscadores

A diferencia de los buscadores, un metabuscador no posee una base de datos propia sino que utiliza la base de datos de estos para encontrar la información solicitada por el usuario. Su único trabajo consiste en combinar las mejores páginas que ha devuelto cada buscador, logrando así un mayor abanico de resultados con mucha mayor calidad (Consoft, 2002).

Buscadores verticales

Un buscador vertical es un buscador especializado en un sector o nicho concreto, lo que le permite analizar la información con mayor profundidad que un buscador genérico, disponer de resultados más actualizados y ofrecer al usuario herramientas de búsqueda avanzadas (Ricciardi, 2009).

Motores de búsqueda

Los motores de búsqueda son programas encargados de realizar las búsquedas dentro de las bases de datos de documentos web. Actualmente se clasifican en tres categorías principales: motores de búsqueda temática, también conocidos como directorios o catálogos; motores de búsqueda por palabras claves o *crawlers* y sistemas basados en el encaminamiento de contenido (Stark, 2002).

En vista de que Orión se clasifica como un motor de búsqueda, en el trabajo solo serán especificados aspectos referentes a los motores de búsqueda.

1.3 Componentes de un Motor de Búsqueda y su funcionamiento

Un motor de búsqueda está conformado específicamente por tres elementos, donde cada uno es de gran importancia para su correcto funcionamiento: robots, índice y mecanismo de búsqueda (Camiño, 2003).

Robots

El primer componente de un motor de búsqueda es el robot, que no es más que el programa encargado de recorrer las páginas web, analizar su contenido y enlaces con otras páginas. Estas acciones se realizan periódicamente para poder tener conocimiento de posibles cambios como modificaciones en las páginas, cambios de URL, pérdidas de archivos entre otras. Los motores de búsqueda pueden estar conformados por uno o más robots.

Índice

Consiste en una base de datos que se encarga de almacenar copias de los documentos reunidos por el robot luego de su búsqueda por la web. Cuando los usuarios realizan una búsqueda, no lo hacen sobre la información actual sino sobre sus índices.

Mecanismo de búsqueda

Los mecanismos de búsquedas son visibles para los usuarios en el buscador, estos difieren de un motor a otro pero generalmente les permiten a los navegantes realizar consultas en lenguaje natural. Mediante una interfaz que puede tener uno o más cuadros de diálogo, el usuario introduce el texto sobre el tema que desea buscar con una o varias palabras. Después de haber comenzado la búsqueda el programa examina todas las páginas web contenidas en el índice, realizando una búsqueda de las palabras iguales a las que el usuario solicitó en la consulta para luego ordenarlas por el grado de relevancia y mostrárselas en pantalla a los navegantes en forma de lista.

1.4 Web Semántica

"La Web Semántica vendría a ser una extensión de la Web actual dotada de significado, esto es, un espacio donde la información tendría un significado bien definido, de manera que pudiera ser interpretada tanto por agentes humanos como por agentes computarizados" (Lapuente, 2013). Con la web semántica se busca convertir la información en conocimiento, que los datos puedan ser utilizados y comprendidos por los ordenadores sin necesidad de que un humano lo supervise. De manera que un usuario en Internet pueda encontrar respuestas a sus consultas de forma más rápida y sencilla gracias a una información mejor organizada.

Esta web semántica se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante. Las tecnologías del lenguaje humano tratan de buscar mecanismos computacionales que permitan reconocer y comprender el lenguaje natural que expresan el contenido representado en las páginas web y no puede ser procesado por los sistemas informáticos.

1.5 Procesamiento del lenguaje natural

El Procesamiento del Lenguaje Natural (PLN) surge como disciplina de la Inteligencia Artificial y la lingüística, con el objetivo de estudiar problemas derivados de la generación y comprensión automática del lenguaje natural. Esta disciplina ha realizado una serie de aportes, los mismos han mejorado una serie de tareas como el procesamiento de grandes cantidades de información en formato texto. Muestra de ello es la aplicación de estas técnicas como un componente esencial en los motores de búsqueda o herramientas de traducción automática (Cañón y otros, 2007).

1.5.1 Problemática del procesamiento del lenguaje natural

El lenguaje natural es el medio que utilizan las personas para comunicarse entre sí, este posee algunas propiedades como la variación y la ambigüedad lingüística que disminuyen la eficiencia de los motores de búsqueda. Se entiende como variación en el PLN la posibilidad que tienen las personas de utilizar varias palabras o expresiones para comunicar una misma idea y ambigüedad lingüística cuando una palabra o frase tienen más de un significado. Estas propiedades inciden en la recuperación de información de formas diferentes. La variación provoca el silencio documental, esto consiste en la omisión de documentos relevantes ya que no se utilizan en la búsqueda los mismos términos que aparecen en el documento. Por otra parte la ambigüedad lingüística provoca el ruido documental que sucede cuando se incluyen documentos que no son relevantes ya que se recuperan también documentos con los mismos términos pero con diferentes significados (Vallez y otros, 2007).

1.5.2 El procesamiento del lenguaje natural en la recuperación de información textual

La complejidad del trabajo con lenguaje natural cobra gran importancia cuando se necesita recuperar información textual para satisfacer la necesidad de información de un usuario. Las técnicas de PLN son muy utilizadas en la recuperación de información textual tanto para representar la consulta formulada por el usuario como para facilitar la descripción del contenido de los documentos. No existe una técnica de PLN que permita la extracción del significado de un documento de forma inequívoca. Existen dos aproximaciones para el PLN, por una parte está el procesamiento estadístico del lenguaje natural y por otra el procesamiento lingüístico del lenguaje natural (Baeza-Yates, 2004).

1.6 Procesamiento lingüístico del lenguaje natural

Esta aproximación se basa en la aplicación de diferentes reglas y técnicas para codificar de forma explícita el conocimiento lingüístico. En este modelo los documentos son analizados a partir de diferentes niveles lingüísticos, por herramientas que incorporan al texto las anotaciones propias de cada nivel.
Para realizar un análisis lingüístico es necesario que se apliquen una serie de pasos que se muestran a continuación:

- Realizar el análisis morfológico donde los etiquetadores (*taggers*) asignan a cada palabra su categoría gramatical a partir de los rasgos morfológicos identificados.
- Después de identificar las palabras del texto se observa cómo es que estas se relacionan entre sí para formar unidades superiores (Sintagmas y frases).

- En este punto se aplican formalismos descriptivos del lenguaje (*parsers*) que tienen como objetivo fijar la estructura sintáctica del texto.
- A partir de la estructura sintáctica del texto el siguiente objetivo es obtener el significado de las frases que lo componen. Se trata de identificar la representación semántica de las frases.

1.7 Ontologías

"Una ontología es una especificación de una conceptualización, esto es, un marco común o una estructura conceptual sistematizada y de consenso no sólo para almacenar la información, sino también para poder buscarla y recuperarla. Una ontología define los términos y las relaciones básicas para la compresión de un área del conocimiento, así como las reglas para poder combinar los términos para definir las extensiones de este tipo de vocabulario controlado" (Lapuente, 2013). Al utilizar ontologías se busca referenciar los datos dentro de las páginas web por medio de los metadatos bajo un esquema común sobre algún dominio del conocimiento.

El uso de ontologías va a proporcionar una forma de representar y compartir el conocimiento utilizando un vocabulario común por lo que habrá un protocolo específico de comunicación. Además permitirá usar un formato de intercambio de conocimiento y la reutilización del mismo.

1.7.1 Componentes de una Ontología

Conceptos: son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento u otros.

Relaciones: representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a.

Funciones: son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden aparecer funciones como categorizar-clase, asignar-fecha.

Instancias: se utilizan para representar objetos determinados de un concepto.

Axiomas: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología.

1.8 Estudio de Sistemas Homólogos

En la actualidad existen muchos sistemas utilizados para la búsqueda de información en formato texto. A continuación se realizará un estudio acerca del comportamiento que pueden presentar estas aplicaciones web en el análisis semántico de la información.

1.8.1 Internacionales

Google

Google es un buscador privado de los más conocidos y utilizados en Internet. Usa varias arañas con el objetivo de recolectar y ordenar la información que el usuario desea consultar. En 2012 Google lanzó *Knowledge Graph* con el objetivo de facilitar la labor de entendimiento de una consulta. Con *Knowledge Graph* en el proceso de búsqueda de información por parte de un usuario si éste utiliza el lenguaje natural, el buscador va a tener la capacidad de interpretar estos términos y presentar los resultados de información acorde con su motivación de búsqueda.

Bing

El buscador Bing fue creado por la empresa Microsoft con el fin de remplazar el Live Search. Bing ayuda a identificar los resultados de búsqueda relevantes a través de funciones como Best Match (El mejor resultado), que identifica y destaca la mejor respuesta posible. Además permite realizar previsualización de videos, búsqueda avanzada y filtrado de contenido lo que permite una mayor calidad de respuesta para el usuario. En la actualidad al buscador Bing se le ha incorporado el motor de búsqueda Powerset. Powerset es un motor de búsqueda que en un principio tenía como objetivo el de procesar el lenguaje natural, entender la búsqueda completa y no enfocarse solo en palabras clave.

1.8.2 Nacionales

C.U.B.A

Contenidos Unificados para Búsqueda Avanzada (C.U.B.A.) es una plataforma que integra los servicios web disponibles en la red cubana. Redcuba como también se le puede llamar está basada en la arquitectura del motor de búsqueda Orión, desarrollado por la Universidad de las Ciencias Informáticas (UCI).

Surge como respuesta a la necesidad de mostrar a los usuarios la información existente en el dominio cubano y brinda la posibilidad de acceder a sitios de interés cultural, informativos e investigativos. Esta plataforma facilita al usuario la búsqueda de contenidos en varios formatos digitales (páginas web, imágenes y documentos), proporciona referencias concretas de un tema alojado en varias fuentes.

Se realizaron una serie de consultas a los diferentes buscadores analizados por medio de los criterios de búsqueda semántica. Donde se califican con una puntuación de 0 si no se cumple, 1 si se cumple pero con limitaciones y 2 se cumple del todo. A continuación se muestran las consultas realizadas por cada criterio:

- ✓ Objetos y atributos
  - Perro negro gato blanco
  - Gato negro perro blanco
  - Presión alta dolor cabeza
- ✓ Palabras polisémicas
  - Armar: ensamblar algo o construirlo.
  - Armar: conseguir más armas.
  - Bomba: artículo para bombear agua o aire.
  - Bomba: explosivo.
  - Cura: sacerdote.
  - Cura: medicina.
- ✓ Expansión por sinónimos
  - Perro gigante-enorme perro
  - Agua fría-agua helada
- ✓ Semántica temporal
  - Real Madrid gana la liga de 2017.
  - Matanza quedó segundo lugar entre 2013 y 2017.
  - XIII aniversario de la UCI.
- ✓ Operadores sobre predicados
  - Perro no blanco.
  - Gato no negro.
  - Rico y oloroso.

Criterios de búsqueda	Google	Bing	C.U.B.A
Objetos y atributos	1	1	0
Palabras polisémicas	1	1	0
Expansión por sinónimos	1	0	0
Semántica temporal	1	0	0

Figura 1. Comparación de los buscadores nacionales e internacionales por los criterios de búsqueda semántica. Fuente: Elaboración propia.

Conclusiones del estudio de homólogos

Después de un análisis realizado a los sistemas homólogos se llega a la conclusión de que el SRI cubano no cumple con ninguno de los requerimientos que debe tener la búsqueda semántica y si más bien los buscadores Google y Bing cumplen en gran parte con los criterios de búsqueda definidos en el estudio por el autor, el mismo determina realizar un componente propio ya que no es posible utilizar las funcionalidades que presentan los buscadores internacionales utilizados debido a que estos son de carácter privativo.

1.9 Selección del entorno de desarrollo para la construcción de la solución

Para la implementación de la solución del presente trabajo es necesario el estudio de las metodologías, tecnologías y herramientas utilizadas para el desarrollo en el proyecto Orión.

1.9.1 Metodología, herramientas y técnicas utilizadas para el desarrollo de la solución.

Solr

Apache Solr es una plataforma de búsquedas basada en Apache Lucene, que funciona como un "servidor de indexación". Sus principales características incluyen búsquedas de texto completo, resaltado de

resultados, agrupamiento dinámico, y manejo de documentos ricos (como Word y PDF). Solr es escalable, permitiendo realizar búsquedas distribuidas y replicación de índices, y actualmente se está usando en muchos de los sitios más grandes de internet.

La principal característica de Solr (o al menos la más útil) es su API estilo REST, ya que en vez de usar drivers o APIs programáticas para la comunicación con Solr es posible hacer peticiones HTTP y obtener resultados en XML o JSON. Solr presenta un esquema de datos configurables, utiliza varios caches para agilizar las búsquedas y realiza navegación de resultados por facetas, es decir, explora la información desde diferentes perspectivas (Ramos, 2012).

Metodología AUP versión para la UCI

El Proceso Unificado Ágil *de Scott Ambler o Agile Unified Process* (AUP) es una versión simplificada del Proceso Unificado de *Rational* (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles conceptos que aún se mantienen válidos en RUP. AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo.

En la Universidad de la Ciencias Informáticas (UCI) se desarrolló una versión de esta metodología ágil con el objetivo de crear un estándar en el proceso productivo de manera que se adapte al ciclo de vida de los proyectos en la UCI. En esta variación de AUP para la UCI se mantiene la primera fase de inicio, la segunda fase de ejecución en esta variación para la UCI resume las fases restantes de elaboración, construcción y transición. Finalmente se encuentra la fase de cierre donde se analiza los resultados del proyecto y se realizan las actividades formales de cierre de proyecto (Sánchez, 2015).

Lenguajes de programación

Los lenguajes de programación son las herramientas que permiten la interacción entre el ser humano y las máquinas. Existen diversos lenguajes de programación y se pueden clasificar como, lenguajes de programación del lado de servidor y lenguajes de programación del lado del cliente.

Java

Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Para la implementación del código de la aplicación se utilizó java como lenguaje de programación. Un elemento también importante para la selección de este lenguaje se debe a que la herramienta sobre la cual se implementará la propuesta de solución está desarrollada con el mismo lenguaje de programación (Ramírez, 2013).

A continuación se muestran una serie de características donde se describe de forma más detallada las propiedades y funcionalidades que brinda este lenguaje de programación:

Orientado a objetos: implementa la tecnología de C++ y soporta las tres características del paradigma orientado a objetos. Encapsulamiento: Implementa información oculta. Polimorfismo: El mismo mensaje se envía a diferentes objetos, resultando en comportamientos que dependen de la naturaleza del objeto que recibió el mensaje. Herencia: Puede definir nuevas clases y comportamientos (métodos) basados en clases existentes.

Distribuido: presenta extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Por si sólo no es distribuido, pero proporciona herramientas para que nuestros programas puedan serlo.

Simple: ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de estos. Elimina muchas de las características de otros lenguajes como C++, para mantener reducida la especificación del lenguaje.

Robusto: realiza verificaciones en busca de problemas, tanto en tiempo de compilación, como de ejecución. La comprobación de tipos ayuda a detectar errores. Obliga a la declaración explícita de los métodos.

Seguro: la seguridad tiene dos facetas: Se eliminan características como los apuntadores y el casting implícito para prevenir el acceso ilegal a la memoria El código Java pasa por muchas verificaciones antes de ser ejecutado en una máquina mediante el classloade[1].

Herramientas CASE

Las herramientas *Computer Aided Software Engineering* (CASE por sus siglas en ingles), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, compilación automática, documentación o detección de errores, entre otras.

Visual Paradigm 8.0

Para el modelado del software se utiliza como herramienta CASE, Visual Paradigm que utiliza UML como lenguaje de modelado. Es fácil de usar y soporta todo el ciclo de vida del desarrollo de software: análisis,

[1] Classloade: es una clase encargada de cargar otras clases en memoria según la necesidad que tenga la máquina virtual java (Caules, 2013).

diseño orientado a objetos, construcción, prueba y despliegue. Permite realizar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación para facilitar una mayor comprensión de la implementación. Adicionalmente es sencilla de instalar, fácil de utilizar y actualizar. Además es multiplataforma y cuenta con una abundante documentación, como son: tutoriales, demostraciones interactivas, entre otros recursos.

NetBeans8.0

Para la propuesta de desarrollo se define utilizar IDE Netbeans en su versión 8.0. Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. Netbeans IDE2 es un producto libre y gratuito sin restricciones de uso. Permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de Netbeans y un archivo especial (*manifest file*) que lo identifica como módulo (Salazar y otros, 2011).

Netbeans propone un esqueleto para organizar el código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS. Además posee un sistema para examinar todos los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.

Stanford Parser

El Stanford Parser es un analizador sintáctico probabilístico desarrollado por la Universidad de Stanford que se basa en seleccionar el mejor análisis según aquel que sea más probable a partir de un conjunto de ejemplos analizados correctamente por lingüistas.

El Stanford Parser trabaja con la dependencia del software CoreNPL de la Universidad de Stanford. Este elemento proporciona un conjunto de herramientas de análisis lingüístico. La incorporación de esta dependencia al Stanford Parser proporciona, entre otras, las anotaciones que soporta cada lenguaje humano. Al ser una herramienta orientada principalmente al inglés, las anotaciones para otros lenguajes no están del todo desarrolladas, siendo el español el que menos anotaciones incorpora después del árabe.

Conclusiones Parciales

En el presente capítulo mediante un estudio del estado del arte se analizaron los principales elementos que se consideran imprescindibles para sustentar la propuesta de solución y se llega a las siguientes conclusiones:

- El estudio de los principales conceptos sobre los SRI, RI, ontologías y otros aspectos relacionados con estos como el procesamiento del lenguaje natural permitió un mayor entendimiento de la presente investigación.
- Con el estudio de los sistemas homólogos se identificaron las deficiencias que presentan los buscadores nacionales, de ahí la importancia de mejorar su funcionamiento.
- Con la selección de herramientas y tecnologías basadas en software libre se concretó una base tecnológica para el desarrollo de la propuesta de solución.

Capítulo 2: Análisis y diseño del Componente de extracción semántica de información para Orión.

2.1 Introducción

En el presente capítulo se exponen los principales aspectos relacionados con el diseño del sistema. Para definir las funcionalidades que tendrá dicha aplicación se generaron los artefactos relacionados con la metodología que se utilizará para el desarrollo de la solución como son la especificación de los requisitos funcionales y no funcionales con los que contará el software, así como la especificación de las historias de usuarios que se generaron para cada requisito funcional que haya sido definido para el sistema. Para el diseño de la aplicación se definirán estilos y patrones de arquitectura y diseño que se emplearan para lograr buenas prácticas de diseño y programación.

2.2 Propuesta de solución

Dada la situación problemática del presente trabajo de diploma se tiene como propuesta de solución un componente encargado de procesar semánticamente la información que se recupera en internet mediante la implementación de un plugin en Solr. Se propone la arquitectura cliente servidor para el desarrollo del sistema.

El procesamiento de la información comenzara desde el momento en que se extraigan los documentos almacenados en Solr. Luego de tener toda la documentación en el componente se pasará a la lectura de archivos y auto seguido a la separación de todas las oraciones del texto. Después se realizará el proceso de *tokenizer* de las palabras del texto y posteriormente al etiquetado gramatical de las mismas. Una vez que se han etiquetado todas las palabras en dependencia del papel que juegan en la oración se identifican todas las entidades nombradas existentes en el documento utilizando la librería Stanford CoreNPL, la cual solo reconoce en el lenguaje español a las personas, las organizaciones y los lugares. Culminado este paso se procede a extraer las tripletas (sujeto + predicado + objeto).

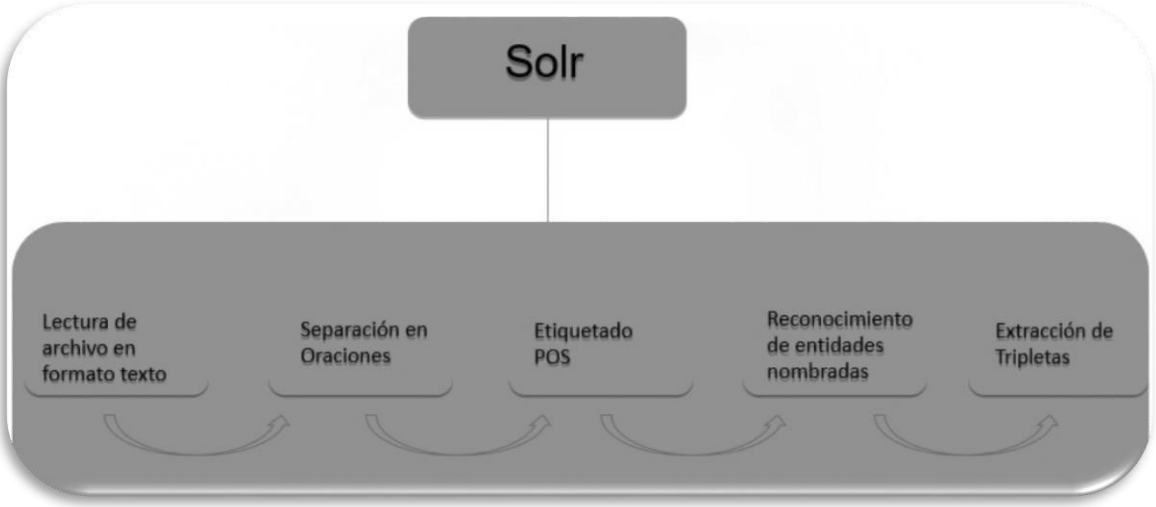


Figura 2. Propuesta de solución. Fuente: Elaboración propia.

2.3 Modelo de dominio

En el modelo de dominio es donde se representan todas las clases conceptuales que intervienen en un determinado proceso, mediante un diagrama de clases donde no se realiza ninguna operación. El modelo de dominio está conformado por las clases conceptuales, las relaciones entre las clases y los atributos de cada una de las clases conceptuales. A continuación se describen cada una de las clases que intervienen en el modelo de dominio:

Orión: Sistema de recuperación o motor de búsqueda.

Solr: Base de datos de indexación de Orión.

Documento: Recurso que son publicados en la web.

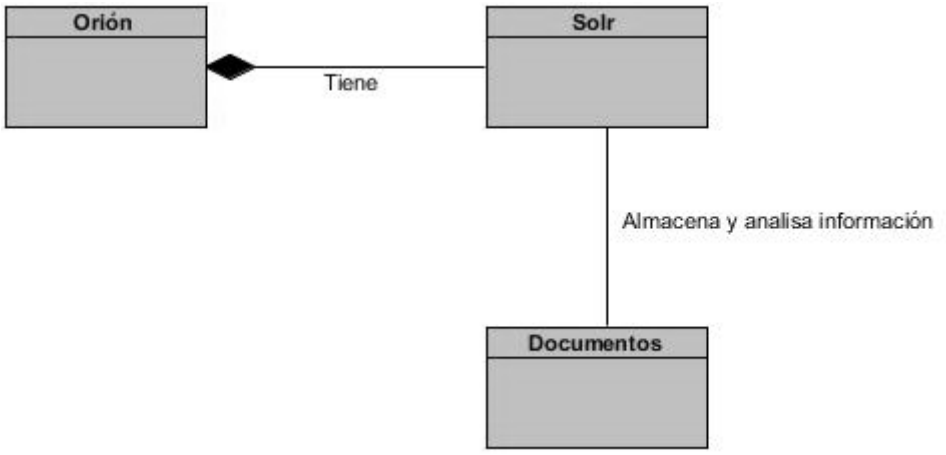


Figura 3. Diagrama de modelo de dominio. Fuente: Elaboración propia.

2.4 Requerimientos del Sistema

Una vez implementado el sistema que se tiene como propuesta de solución deberá cumplir con ciertos requisitos definidos previamente por el equipo de desarrollo y el cliente.

Los requerimientos son condiciones o capacidades que el sistema tiene que tener para satisfacer algún documento formal, estos describen a su vez todo lo que el sistema debe hacer o tener, también deben ser especificados por escrito como acuerdo del contrato.

Una de sus definiciones los describe como “especificación de lo que debe ser implementado. Estos son descripciones de cómo el sistema se debe comportar, de las propiedades y atributos del mismo. Deben ser una restricción del proceso de desarrollo del sistema”.

2.4.1 Requisitos funcionales

Los requerimientos funcionales definen qué hace el sistema, describen entradas y salidas, es decir, las funciones del sistema. A continuación se presentan los requerimientos funcionales identificados:

Tabla 3.Requisitos funcionales.

Requisitos	Descripción
RF.1 Identificar sector específico o contexto en el que se basa el contenido.	Después de haber analizado el contenido de un documento el sistema debe ser capaz de identificar el tema del cual se trata en dicho documento.
RF.2 Identificar grado de similitud entre un documento y los demás.	A través de un documento ya analizado el sistema debe ser capaz de buscar documentos ya almacenados que traten de temas similares al documento que se analizó.
RF.3 Identificar valores específicos de objetos de una entidad	El sistema debe de identificar valores específicos y entidades de cualquier tipo.
RF.4 Identificar tripletas en los contenidos.	El sistema debe ser capaz de extraer de la frase Sujeto+Predicado+Objeto.

2.4.2 Requisitos no funcionales

Los requerimientos no funcionales, son un tipo de requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. A continuación se muestran los requisitos no funcionales definidos para el sistema.

Disponibilidad

RNF.1 El sistema debe estar disponible cada vez que el sistema requiera de su servicio.

Usabilidad

RNF.2 Se pretende que el sistema sea utilizado ´por personas con conocimientos mínimos de informática.

Rendimiento

- ✓ Hardware

RNF.3 PC: 4 GB RAM.
RNF.4 CPU: Intel 4 núcleos.
RNF.5 Disco duro: 80 GB disponibles.

- ✓ Software

RNF.6 Sistema Operativo: Ubuntu.
RNF.7 Servidor para Solr: Tomcat 7.
RNF.8 Servidor Web: Apache 2.2.

2.5 Historias de Usuario (HU)

Las HU son técnicas empleadas en el cuarto escenario de la metodología AUP-UCI para especificar los requisitos del software. Se tratan de tarjetas donde el cliente realiza una breve descripción de las características que debe poseer el software. Las estimaciones de esfuerzo asociado a cada HU las realiza el desarrollador utilizando como medida el punto, donde cada punto equivale a una semana real de programación.

Tabla 4.Historia de usuario #1.

Historia de Usuario	
Número: HU_1	**Usuario:** Sistema.
Nombre de historia: Identificar sector específico o contexto en el que se basa el contenido.	
Prioridad en negocio: Medio	**Riesgo en desarrollo:** Medio
Puntos estimados: 2	**Iteraciones asignadas:**1
Programador responsable: Armando Pino Cárdenas	
Descripción: Después de haber analizado el contenido de un documento el sistema debe ser capaz de identificar el tema del cual se trata en dicho documento.	
Observaciones:	

Tabla 5.Historia de usuario #2.

Historia de Usuario	
Número: HU_2	**Usuario:** Sistema.
Nombre de historia: 2. Identificar grado de similitud entre un documento y los demás.	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 3	**Iteraciones asignadas:**2
Programador responsable: Armando Pino Cárdenas	
Descripción A través de un documento ya analizado el sistema debe ser capaz de buscar documentos ya almacenados que traten de temas similares al documento que se analizó.	
Observaciones:	

Tabla 6.Historia de usuario #3.

Historia de Usuario	
Número: HU_3	**Usuario:** Sistema.
Nombre de historia: Identificar valores específicos de objetos de una entidad.	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteraciones asignadas:**3
Programador responsable: Armando Pino Cárdenas	
Descripción: El sistema debe de identificar valores específicos y entidades de cualquier tipo.	
Observaciones:	

Tabla 7.Historia de usuario #4.

Historia de Usuario	
Número: HU_4	**Usuario:** Sistema.
Nombre de historia: Identificar tripletas en los contenidos.	
Prioridad en negocio: Media	**Riesgo en desarrollo:** Alto

Puntos estimados: 3	**Iteraciones asignadas:**4
Programador responsable: Armando Pino Cárdenas	
Descripción: El sistema debe ser capaz de extraer de todas las oraciones que conforman el documento Tripletas de Sujeto+Predicado+Objeto.	
Observaciones:	

2.6 Diseño

El diseño de software es un proceso que cuenta con varios pasos, pero en realidad se puede ver como uno solo. Este proceso se refiere a establecimiento de estructuras de datos, construcción de la arquitectura general del software, las representaciones de interfaz y algoritmos.

2.6.1 Estilo arquitectónico del sistema

Para el desarrollo de la propuesta de solución se seleccionó la arquitectura n-capas. Esta es una arquitectura cliente-servidor donde el principal objetivo es separar la capa de la lógica de negocios de la capa de diseño. La principal ventaja de esta arquitectura es que el desarrollo se puede llevar a varios niveles, en caso de que se realice algún cambio solo se ataca el nivel requerido (Llorente, 2010). A continuación se muestran las los diferentes niveles que presenta dicha arquitectura para la propuesta de solución.

Equipo y servidor web: Capa de datos.

Servidor Solr y Plugin: Capa de Negocio.

Esta arquitectura se divide en dos partes:

Índice: Sistema de ficheros que almacena la información. Contiene la configuración de Solr y la definición de su estructura de datos.

Servidor: Proporciona el acceso a los índices y las características adicionales. Admite la integración de plugin para añadir funcionalidades.

Figura 4.Arquitectura del Sistema. Fuente: Elaboración propia.

2.6.3 Modelo de Despliegue

Un diagrama de despliegue es utilizado para mostrar la estructura física del sistema, incluyendo las relaciones mediante protocolos entre el hardware y el software. Como se muestra en la figura, la distribución física del sistema en tiempo de ejecución consta de dos nodos. El primer nodo ¨Servidor apache web¨ es el encargado de atender y responder todas las peticiones realizadas por los usuarios. El segundo nodo representa el servidor de Solr donde estará incluido el plugins de Solr.

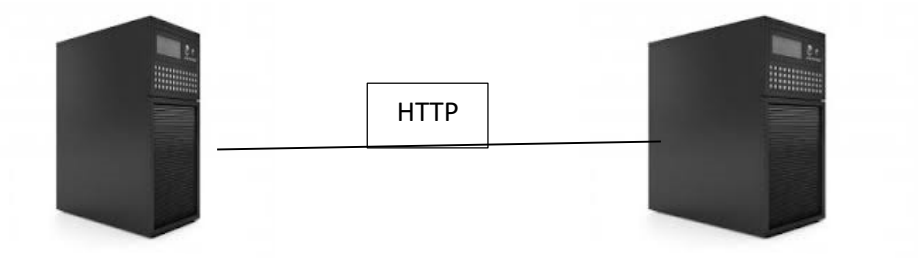


Figura 5. Modelo de despliegue. Fuente: Elaboración propia.

2.7 Patrones de diseño

Los patrones de diseño son soluciones simples que se le dan a problemas comunes del diseño orientado a objetos. Estas soluciones se basan en la experiencia y se ha demostrado que funcionan. Con el

desarrollo de múltiples diseños hay problemas que se repiten, es decir, responder a cierto patrón (Larman, 2004).

Para el diseño de la propuesta de solución se tuvieron en cuenta los patrones Generales de Software para Asignación de Responsabilidades (GRASP[2]). Entre ellos se encuentran experto en información, bajo acoplamiento, creador, alta cohesión, controlador.

Experto en información

Este patrón plantea que se debe asignar una responsabilidad al experto en información, en otras palabras, a la clase que cuenta con los datos necesarios para cumplir la responsabilidad. De esta forma, se conserva el encapsulamiento de la información, puesto que los objetos ejecutan las tareas que le corresponden de acuerdo a la información que poseen, lo que da lugar a sistemas más robustos y fáciles de mantener (Larman, 2004). Este patrón se encuentra representado en la claseTripletas.java.

A continuación se muestra un fragmento de código:

```
package ClasificationWord;

public class Tripletas {

    private String subject;
    private String verb;
    private String predicate;

    public Tripletas(String subjectTree, String predicateTree, String objectTree) {
        this.subject = subjectTree;
        this.verb = predicateTree;
        this.predicate = objectTree;
    }

    public String getSubject() {
        return subject;
    }

    public void setSubject(String subject) {
        this.subject = subject;
    }

    public String getVerb() {
        return verb;
```

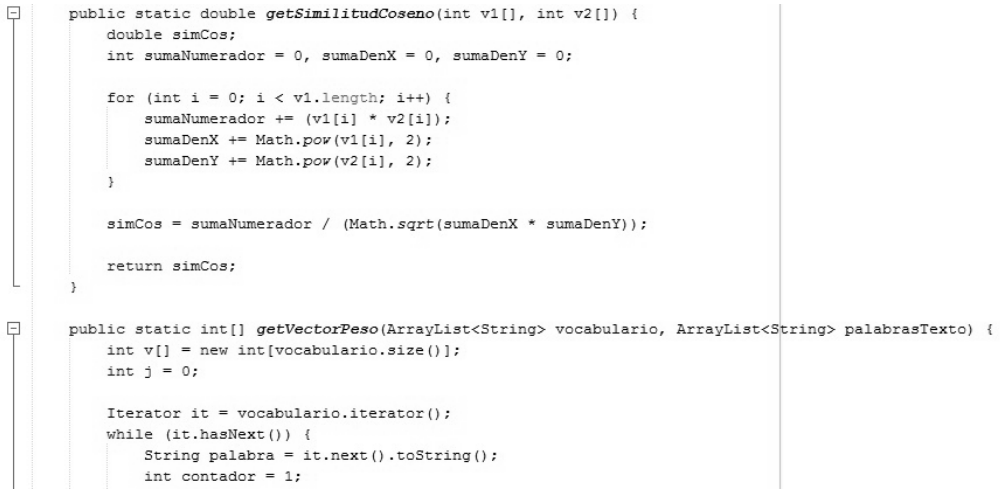
Figura 6.Código donde se evidencia el patrón de experto. Fuente: Elaboración propia.

Bajo Acoplamiento

El patrón bajo acoplamiento impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados negativos que puede producir un

[2] GRASP: General *Responsibility Assignment Software Patterns.*

acoplamiento alto (Larman, 2004). Este patrón se encuentra presente en la clase Cliente_Solr.java. A continuación se muestra un fragmento de código donde este se encuentra representado.
Este patrón quedó evidenciado en las clases SimilarDocuments.java, Similar.java y ValueComparator.java donde estas no llaman a ningún método realizar sus funcionalidades, por lo tanto no dependen de ninguna otra clase. A continuación se muestra un fragmento de código de patrón evidenciado en la clase Similar.Documents.java:



```
public static double getSimilitudCoseno(int v1[], int v2[]) {
    double simCos;
    int sumaNumerador = 0, sumaDenX = 0, sumaDenY = 0;

    for (int i = 0; i < v1.length; i++) {
        sumaNumerador += (v1[i] * v2[i]);
        sumaDenX += Math.pow(v1[i], 2);
        sumaDenY += Math.pow(v2[i], 2);
    }

    simCos = sumaNumerador / (Math.sqrt(sumaDenX * sumaDenY));

    return simCos;
}

public static int[] getVectorPeso(ArrayList<String> vocabulario, ArrayList<String> palabrasTexto) {
    int v[] = new int[vocabulario.size()];
    int j = 0;

    Iterator it = vocabulario.iterator();
    while (it.hasNext()) {
        String palabra = it.next().toString();
        int contador = 1;
```

Figura 7.Código donde se evidencia el patrón Bajo acoplamiento Fuente: Elaboración propia.

Alta Cohesión

Este patrón plantea que se debe asignar una responsabilidad de modo que la cohesión siga siendo alta (una clase tiene responsabilidades moderadas). Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que no realicen un trabajo enorme. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. El patrón de alta cohesión se evidencia en las clases Tripletas y CategoryWord, donde estas se hacen responsable de los documentos y el análisis respectivamente.

Creador

Este patrón plantea que se debe asignar a una clase X la responsabilidad de crear una instancia de una clase Y. La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. Este patrón es el encargado de guiar la asignación de responsabilidades relacionadas con la creación de objetos. Este patrón se pone de manifiesto en la clase CategoryWor*d*, ya que es la encargada de crear la estructura del árbol sintáctico.

Diagrama de clases del diseño

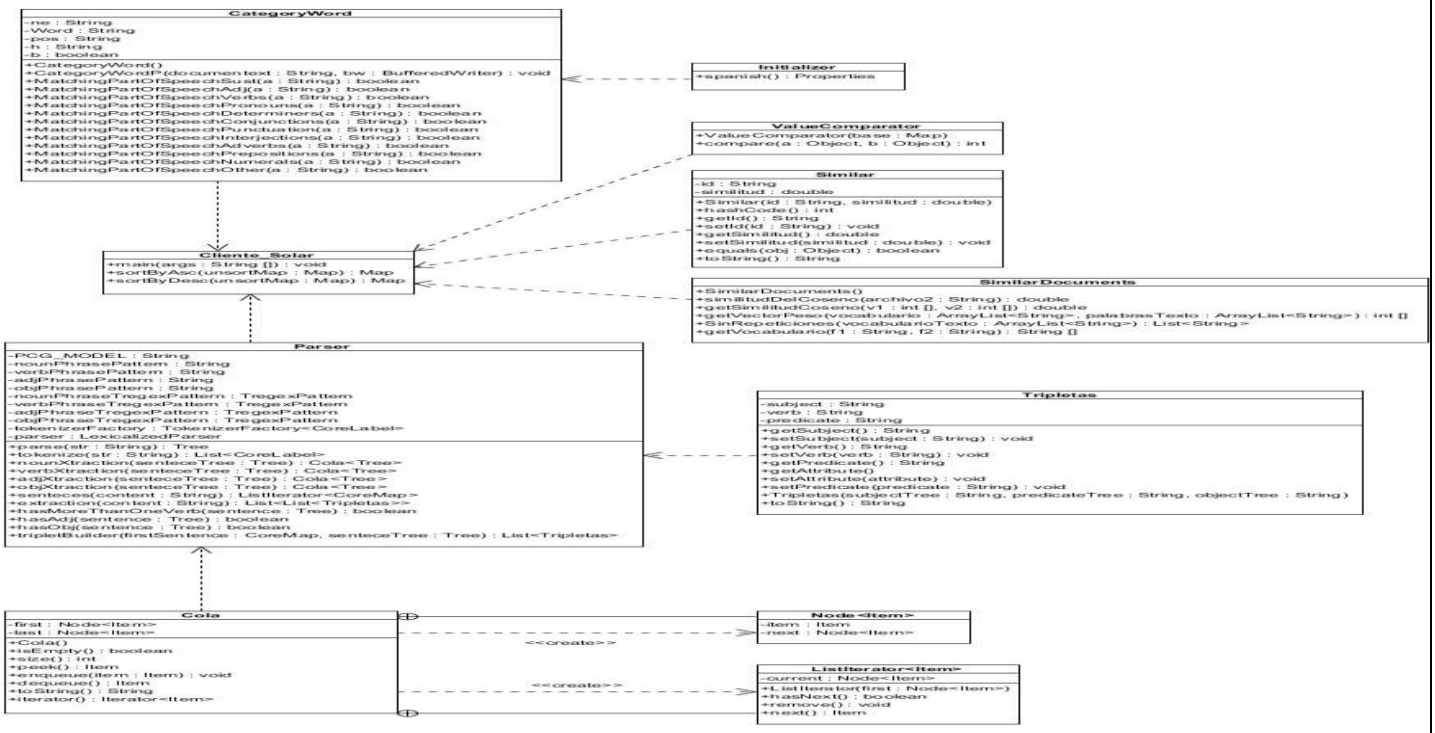


Figura 8. Diagrama de clases del diseño. Fuente: Elaboración propia.

Conclusiones Parciales

En el presente capítulo se realizó el análisis y diseño de la aplicación, después de abordar una serie de aspectos se llega a las siguientes conclusiones:

- Mediante la identificación de la arquitectura de la propuesta de solución se pudo lograr una mayor organización a los elementos que darán forma a la aplicación.

- El Modelo de Dominio realizado a partir de los procesos identificados permitió conocer todos los términos y conceptos presentes en el entorno.

- Con la especificación del conjunto de requisitos funcionales y no funcionales con lo que deberá cumplir la propuesta de solución se logró una mayor comprensión de la aplicación que se pretende realizar.
- El diseño del diagrama de clases del diseño permitió definir las relaciones entre las clases del componente, permitiendo visualizar la relación entre las mismas, así como las funcionalidades y atributos que deben presentar cada una de estas.

- La elaboración del diagrama de despliegue permitió identificar la disposición física de los componentes que intervienen en el sistema en tiempo de ejecución.

Capítulo No. 3: Implementación y validación del Componente para la extracción semántica de información para Orión.

3.1 Introducción

La fase e implementación de un sistema es de las más importantes para el desarrollo de un software. Esta fase materializa en forma de código la arquitectura, todos los artefactos y descripciones definidos en la anterior etapa de análisis y diseño con el objetivo de realizar el producto final que desea obtener el cliente. Al realizar pruebas a un software se garantiza de que el mismo cumpla con los requerimientos y funcionalidades que pide que el cliente. Estas pruebas se llevan a cabo en la etapa de validación donde se realizan un conjunto de pruebas, cada una con un objetivo específico.

3.2 Modelo de Componentes

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (Hernández, 2013).

Diagrama de componentes

Un diagrama de componentes proporciona una visión física de la construcción del sistema de información. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos. El diagrama de componente da la posibilidad al equipo de desarrollo de entender un diseño existente o crear uno nuevo después de analizar el diseño según sus bloques principales.

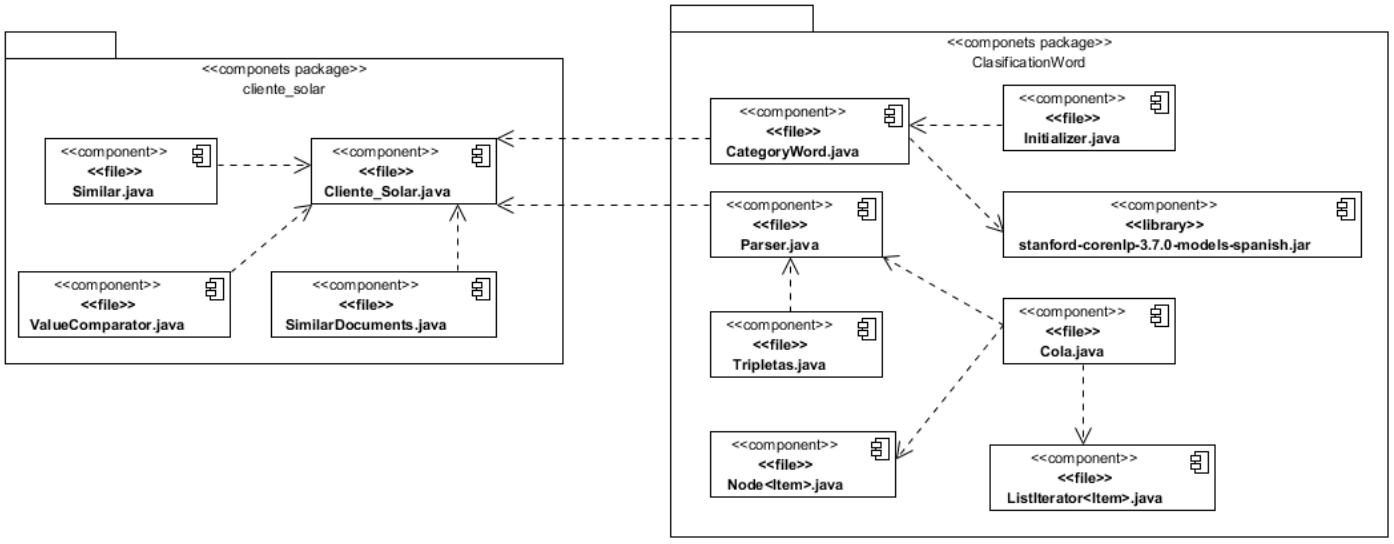


Figura 9. Diagrama de Componentes. Fuente: Elaboración propia.

3.3 Estándares de codificación utilizados

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Seguir un determinado estilo de codificación permite a los programadores revisar, mantener y actualizar el código de una manera sencilla y ordenada, evitando que incurran en errores y malas prácticas que dificultan la comprensión de las líneas de código. A continuación se muestran algunos de los estándares utilizados en el desarrollo de la aplicación.

Líneas en blanco

Una de las mejores prácticas de codificación que ayuda a garantizar la capacidad de mantenimiento de los sistemas es la inclusión de líneas en blanco entre secciones de código, funciones, clases, sentencias, declaraciones y comentarios. En este trabajo se ha definido el uso de líneas en blanco para separar funciones de una misma clase, así como secciones de código dentro de una misma función. Es prudente señalar que este estilo agrega más líneas de código al programa, pero a su vez se gana en legibilidad y limpieza en el código.

```
public class Similar {

    String id;
    Double similitud;

    public Similar(String id, Double similit
        this.id = id;
        this.similitud = similitud;
    }
```

Figura 10. Líneas en blanco. Fuente: Elaboración propia.

Identificadores

Para la definición del nombre de las clases, funciones, variables y constantes se tuvo en cuenta el estilo *lowerCamelCase y UpperCamelCase*. El primer estilo establece que la separación entre palabras internas de los identificadores deberá realizarse escribiendo la letra inicial en minúscula, a excepción de la primera palabra. Además, no deberá colocarse ningún carácter especial entre palabras de los identificadores. En el segundo caso del estilo *UpperCamelCase* se utiliza para las clases, es cuando la primera letra de cada una de las palabras es mayúscula.

```
public class CategoryWord {
```

Figura 11. UpperCamelCase. Fuente: Elaboración propia.

Comentarios

Los comentarios en el código representan la documentación interna más precisa de un software. Estos garantizan el entendimiento de lo que realmente realiza un determinado bloque de código, evitando confusiones y agilizando considerablemente las tareas de revisión y mantenimiento.

```
//Este metodo es para extraer las palabras del texto y devuelve un arreglo de palabras.
public static String[] getVocabulario(String f1, String f2) throws FileNotFoundException, IOException {

    String lineaTexto = "", palabra;
    List<String> vocabulario = new ArrayList<>();
    palabrasTexto1 = new ArrayList<>();
    palabrasTexto2 = new ArrayList<>();
    Set<String> set = new HashSet<>();//Se utiliza como lista sin repeticiones
```

Figura 12. Comentarios. Fuente: Elaboración propia.

Identación

Una de las prácticas más recomendadas para la implementación consiste en la indentación del código. Esta costumbre enfatiza en comenzar a escribir cada línea de código a diferentes distancias desde el

borde izquierdo del área de edición. La distancia deberá regirse por la jerarquía que se forma al introducir sentencias dentro de bloques de estructuras. Gracias al uso de Netbeans como IDE de desarrollo, los espacios de indentación son ajustados automáticamente, permitiendo a los programadores enfocarse en otras funciones de mayor importancia. Por lo tanto, la unidad de indentación de bloques de sentencias es de dos espacios.

```
Annotation annotation;
annotation = new Annotation(documentext);
pipeline.annotate(annotation);
List<CoreMap> sentences = annotation.get(SentencesAnnotation.class);

for (CoreMap sentence : sentences) {

    for (CoreLabel token : sentence.get(TokensAnnotation.class)) {

         //Sustantivos
        if (MatchingPartOfSpeechSust(token.tag())) {
              if (MatchingPartOfSpeechSust(token.get(PartOfSpeechAnnotation.class))) {
            Word = token.get(CoreAnnotations.TextAnnotation.class);
            pos = token.get(PartOfSpeechAnnotation.class);
            ne = token.get(NamedEntityTagAnnotation.class);
              String j = token.docID();
              h = token.tag();
            h = "Sustantivo";
```

Figura 13. Identación. Fuente: Elaboración propia.

3.4 Validación del componente para la extracción semántica de información para Orión

A continuación se detallan las pruebas de software realizadas al componente implementado. El principal objetivo que se busca con la realización de estas pruebas es la determinación de las no conformidades respecto a las funcionalidades que debe cumplir el software, las vulnerabilidades que atentan contra la seguridad de la información que se manipula, así como la correcta integración de los diferentes componentes de la arquitectura del sistema.

Pruebas Funcionales

Las pruebas funcionales son aquellas que se le realizan al software para validar que las funcionalidades implementadas funcionen de acuerdo a las especificaciones de los requisitos que se definieron en anteriores capítulos. Para la aplicación de pruebas funcionales en la presente investigación se utilizó el método de caja negra, diseñando casos de pruebas basados en casos de uso para verificar la correcta entrada y salida de datos del sistema.

A continuación, se muestran fragmentos de algunos casos de pruebas elaborados para las historias de usuario “Identificar valores específicos de objetos de una entidad” y “Identificar tripletas en los contenidos”.

Escenario	Descripción	Variable1	Variable2	Respuesta del sistema
Identificar valores específicos de objetos de una entidad	Reconocer entidades de cualquier tipo	V	V	El sistema devuelve un conjunto de datos con las entidades presentes en el documento analizado. Pueden ser persona (PERS), organización (ORG), lugar (LUG) y otros (OTROS). [palabra: Donald, entidad: PERS] [palabra: Trump, entidad: PERS] [palabra: Teatro, entidad: LUG] [palabra: Manuel, entidad: PERS] [palabra: Artime, entidad: PERS] [palabra: Miami, entidad: LUG]
		https://dragones.uci.cu/2017/06/del-gran-teatro-de-la-habana-al-teatro-manuel-artime-de-miami/	Han pasado poco más de diez días desde que el presidente Donald Trump emitiera su discurso en el Teatro Manuel Artime de Miami	

Figura 14. Muestra del escenario: Identificar valores específicos de objetos de una entidad.

Fuente: Elaboración propia.

En la tabla anterior, se muestran los valores que deben tomar las variables que intervienen en el proceso de identificar valores específicos de objetos de una entidad. Además de esto, se evidencia la respuesta correcta del sistema para cada uno de los juegos de datos de entrada. Las variables que intervienen en este proceso son:

Variable1: Representa la ubicación del documento que es analizado.

Variable2: Representa la estructura de la frase analizada.

La siguiente tabla contiene un caso de prueba para el escenario relacionado con identificar tripletas en los contenidos. En este, la Variable1 representa la ubicación del documento que es analizado y la Variable2 representa la estructura de la frase analizada.

Escenario	Descripción	Variable1	Variable2	Respuesta del sistema
Identificar tripletas en los contenidos	Extraer de las frases el sujeto, predicado y objeto.	V	V	El sistema devuelve un conjunto de datos relacionados con el análisis sintáctico de las frases, siendo el sujeto el primer sustantivo presente en la oración, el predicado el primer verbo presente después del sujeto y objeto lo que resta de la frase. [Tripleta: sujeto: relación, predicado: consiste, objeto: consiste en respeto, comprensión, confianza y preocupación]
		https://dragones.uci.cu/2017/06/a-quien-se-parecen-los-adolescentes-y-jovenes/	Una buena relación entre abuelos, padres e hijos consiste en respeto, comprensión, confianza y preocupación	

Figura 15. Muestra del escenario: Identificar tripletas en los contenidos. Fuente: Elaboración propia.

Como se muestra en el siguiente gráfico se realizaron dos iteraciones de pruebas funcionales al sistema. En la primera se detectaron 3 no conformidades, relacionadas fundamentalmente con: determinar el grado de similitud entre los documentos, la obtención de tripletas de las frases que presentaban omisión del sujeto y la obtención de la categoría gramatical de los términos de las frases analizadas en los

documentos. Estas no conformidades fueron resueltas en su totalidad y en una segunda iteración de pruebas no se detectaron no conformidades. Esto muestra, que el sistema se ajusta a las necesidades del cliente y cumple con los requisitos funcionales definidos.

Figura 16. No conformidades de las pruebas funcionales. Fuente: Elaboración propia.

Pruebas de Integración

Luego de haber concluido con la realización de pruebas funcionales al sistema se procedió a realizar las pruebas de integración al componente con el objetivo principal de verificar la correcta conexión y operación conjunta con la base de datos de indexación Solr.

"El proceso de integración de un sistema implica construir este a partir de sus componentes y probar el sistema resultante para encontrar problemas que pueden surgir debido a la integración de los componentes" (Sommerville, 2005). Los componentes que se integran pueden ser comerciales, reutilizables o componentes nuevos que se hallan desarrollados.

Durante la realización de las pruebas de integración entre el componente desarrollado y Solr se realizaron 2 iteraciones:

Iteración # 1:

- No se podía realizar la conexión entre el componente y el core general que se encontraba en Solr.

- El componente no indexaba todos los campos que se le pedían en la consulta a Solr.
- Problemas en la extracción de documentos desde el core general.
- No se mostraban los nuevos campos añadidos a Solr.

Iteración # 2:

No se encontraron no conformidades para la segunda iteración.

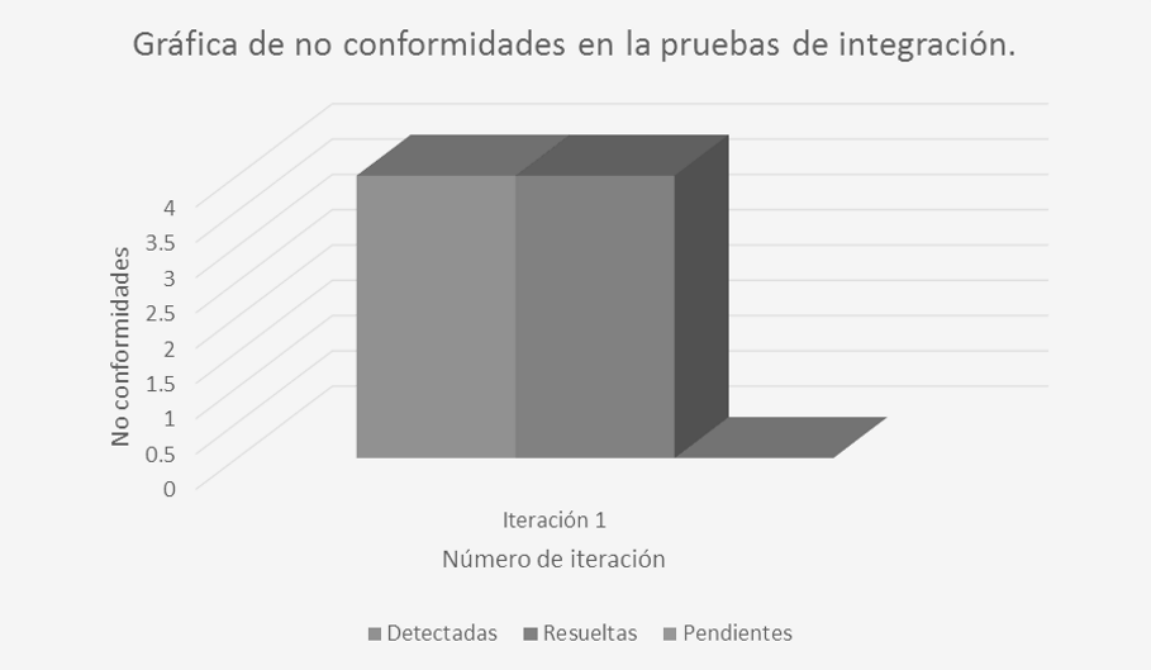


Figura 15. No conformidades de pruebas de integración. Fuente: Elaboración propia.

3.5 Validación de la Hipótesis

Criterio de Experto-Escalamiento de Likert

Para la validación de la hipótesis de investigación en el presente trabajo se escogió el criterio de expertos escalamiento de Likert. La valoración basada en este juicio de expertos permite obtener valoraciones sobre temas relacionados con la propuesta de solución.

Para el desarrollo de la validación se realizó la selección de 5 expertos relacionados con el área de la informática y la lingüística, temas que están relacionados con la propuesta de solución. Para la selección de los expertos se tuvo en cuenta dominio del tema que se encuesta y categoría docente donde deben tener el grado de ingenieros o master en ciencias.

Los expertos expresan sus valoraciones mediante los siguientes indicadores:

Tabla 8.Parámetros para la validación de la Hipótesis utilizando Escalamiento de Likert

5	4	3	2	1
Totalmente de acuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	Totalmente desacuerdo

De esta forma se calculan los por cientos de concordancia de los expertos en las respuestas dadas a las preguntas en la encuesta. Luego se calcula el índice porcentual (IP) que integra en un valor la aceptación de cada planteamiento evaluado mediante la siguiente fórmula:

IP = 5(%)+4(%)+3(%)+2(%)+1(%) / 5

Tabla 9. Valoración de expertos

	Escala					
Preguntas	IP	TD	DA	NI	ED	TA
Pregunta 1	88	40	60	0	0	0
Pregunta 2	88	40	60	0	0	0
Pregunta 3	96	80	20	0	0	0
Pregunta 4	92	60	40	0	0	0
Pregunta 5	96	80	20	0	0	0

Conclusiones Parciales

En este capítulo mediante la implementación y validación del componente para extraer información semántica desde el buscador Orión se llega a las siguientes conclusiones:

- El diseño del diagrama de componentes permitió observar con mayor claridad la estructura general del sistema diseñado la relación que va a existir entre cada uno de sus componentes.
- La utilización de estándares de codificación garantizó una estructura base para la organización lógica del código fuente, permitiendo de esta manera que se puedan realizar futuras modificaciones a la aplicación y el acceso al código tenga un bajo impacto para los desarrolladores.

- Con la realización de pruebas al componente desarrollado se detectaron varias vulnerabilidades, lo cual permitió darle solución con el objetivo de presentarle al usuario una aplicación con mayor calidad.

Conclusiones Generales

Una vez terminado el presente trabajo de investigación se llegan a las siguientes conclusiones:

- A través del estudio del estado del arte realizado y de los fundamentos teóricos relacionados con la recuperación de información, el procesamiento del lenguaje natural y la web semántica se determinó que existen varios SRI que presentan algunas funcionalidades de gran importancia en la recuperación de información semántica, definiéndose una propuesta de solución propia del autor ya que los sistemas homólogos internacionales son de carácter privado y los nacionales no cumplen con ninguna de las condiciones establecidas.
- El enfoque ágil que propone la metodología AUP en su versión para la UCI fue la guía a seguir para describir todos los procesos y sub-procesos que se debían de ejecutar, además de la selección y el uso de herramientas para el desarrollo de la solución.
- Después de haber estudiado todos los elementos que intervienen que el proceso de recuperación de información fue posible realizar el diseño de la propuesta de solución.
- En la implementación de la solución para la recuperación de información mediante la utilización de las herramientas y tecnologías seleccionadas se solucionó la problemática planteada en la presente investigación.
- La utilización del Stanford Parser como herramienta para el procesamiento del lenguaje natural contribuyo en gran parte al desarrollo de la propuesta de solución.
- La ejecución de las pruebas de software realizadas permitió erradicar las insuficiencias detectadas en la herramienta desarrollada logrando así un producto más seguro y funcional conforme a las necesidades de los usuarios finales.
- La aplicación del criterio de expertos escalamiento de Likert permitió validar de forma correcta la hipótesis planteada en la investigación.

Recomendaciones

Luego de haber finalizado la investigación y desarrollo de la propuesta de solución del presente trabajo se recomienda:

Realizar en próximas investigaciones un componente que realice un completo procesamiento de la información utilizando la librería de código abierto Freeling.

Bibliografía

Abadal, Ernest.; Codina, Lluís. Bases de datos documentales: Características, funciones y métodos. Madrid: s.n., 2005.

Acosta, D. Rational. Rational. 2011. [En línea] 3 de Marzo de 2011. [Citado el: 4 de Diciembre de 2016.] http://www.rational.com.ar/herramienta/roseenterprise.html.

Ambysoft. Agilemodeling. 2014. [En línea] 2014. [Citado el: 3 de diciembre de 2015.] http://www.agilemodeling.com/essays/simpleTools.htm#SelectingCASE.

Apache Solr. Apache Software Foundation. 2015. [En línea] 2015. [Citado el: 4 de Diciembre de 2016.] http://httpd.apache.org/...

Applications. International Semantic Web Conference (ISWC), Collected Posters. Sardinia (Italy), June 2002.

Arenas, M.; Buil Aranda, Carlos. La web semántica: Herramientas para la publicación y extracción efectiva de información en la web. 2015. https://www.coursera.org/learn/web-semantica/lecture/P1aI0/video-de-bienvenida.

McIlraith, S.; Narayanan, M.; Paolucci, T.; Payne, R.; Sycara, K. DAML-S: Web Service Description for the Semantic Web. The First International Semantic Web Conference (ISWC), June 2002.

Baisley, B. Unified Modeling Language Infrastructure. s.l.: Pearson, 2006.

Baeza-Yates, Ricardo, y otros. Crawling a country: Better strategies than breadth-first for web page ordering. 2005. [En línea] 2005. [Citado el: 12 de Octubre de 2016.] https://udesantiago.pure.elsevier.com/en/publications/crawling-a-country-better-strategies-than-breadth-first-for-web-p.

Baeza-Yates, R. Challenges in the Interaction of Information Retrieval and Natural Language Processing. In Proc. 5 th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2004), Seoul, Corea. Lecture Notes in Computer Science vol. 2945, pages 445-456, Springer.

Barski, C. The enigmatic art of knowledge representation.

Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. 5, mayo de 2001, Scientific American, Vol. 284.

Bonilla, D.; Tramullas Saz, J. 2005. Directorios temáticos especializados: definición, características y perspectivas de desarrollo. 1, Madrid: s.n., 2005, Vol. 28. 0210-0614.

Brachman, R.; Schmolze, J. An overview of the KL-ONE Knowledge Representation System, Cognitive Science, Vol. 9, No. 2, pp. 171-216. 1985.

Brito Acuña, K. Selección de metodologías de desarrollo para aplicaciones web en la facultad de informática de la Universidad de Cienfuegos. 2009. [En línea] 2009. [Citado el: 29 de Noviembre de 2016.] http://www.eumed.net.

Camiño, R. Motores de búsqueda sobre salud en Internet. La Habana: s.n., Septiembre de 2003, Scielo. ISSN 1024-9435.

Cañón, P.; Correa, S. PROCESAMIENTO DEL LENGUAJE NATURAL EN LA RECUPERACIÓN DE INFORMACIÓN. 2007.

Cardelas, C. ¿Qué es un motor de búsqueda? ¿Qué es un motor de búsqueda? 2007. [En línea] 7 de Enero de 2013. [Citado el: 29 de Noviembre de 2016.] http://mcclaudiamari.blogspot.com.

Castells, P. La Web Semántica. Madrid: s.n.

Caules, C. 2013. El concepto de ClassLoader - Arquitectura Java. El concepto de ClassLoader - Arquitectura Java. [En línea] Octubre de 16 de 2013. [Citado el: 2 de diciembre de 2016.] http://www.arquitecturajava.com/el-concepto-de-classloader.

Cobo, Á y otros. PHP y MySQL: Tecnologías para el desarrollo de aplicaciones web. España: Ediciones Díaz de Santos, 2005. 84-7978-706-6.

Communications of the ACM. The Intuitive Beauty of Computer Human Interaction. Special issue on Programming by Demonstration, 43, 3, March 2000.

Deerwester S.; Dumais S. T.; Furnas G. W.; Landauer T. K.; Harshman R. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science. Vol. 41. No. 6. PP. 391-407. 1990.

Delugach H.; Towards, S. Conceptual Structures Interoperability Using Common Logic Computer. Science Department Univ. of Alabama in Huntsville. Third Conceptual Structures Tool Interoperability Workshop. 2008.

Fernández, J. El codigok. El codigok. 2011. [En línea] 14 de Mayo de 2011. [Citado el: 4 de Diciembre de 2016.] http://www.elcodigok.com.ar/2010/09/caracteristicas-de-un-excelente-entorno-de-desarrollo-integrado...

Fernández, O. Introducción al lenguaje de. 2005.

Gelbukh A.; Sidorov, G. Procesamiento automático del español con enfoque en recursos léxicos grandes. Centro de Investigación en Computación, Instituto Politécnico Nacional, México, 2006.

Giginio, R. El Desarrollo Científico Tecnológico. El Desarrollo Científico Tecnológico. . 2010 [En línea] Rubén Higinio Rivera Aguilera, julio de 2010. http://www.monografias.com/.../desarrollo-cientifico-tecnologico.

González, L. Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información. 2012. [En línea] 2012. [Citado el: 23 de Febrero de 2017.] https://www.redib.org/recursos/Record/oai_articulo983403-extensión-visual-paradigm-uml-desarrollo-dirigido-modelos-aplicaciónes-gestión-información.

Guevara, Y. Diario de la juventud cubana. Diario de la juventud cubana. Guzmán, Paola Lizbeth. Motores de búsqueda; soluciones con aplicaciones de Google. 2011.

Helbig, H. Knowledge Representation and the Semantics of Natural Language. Lecture Notes in Computer Science Publisher Springer-Verlag Berlin Heidelberg 2006.

Hensman, S. Construction of Conceptual Graph representation of texts. Department of Computer Science, University College Dublin. Belfield, Dublin 4. Proceedings of Student Research Workshop at HLTNAACL, 2004.

HERNÁNDEZ, L. Modelo de implementación. [En línea] Modelo de implementación, 2013. [Citado el: 15 de Marzo de 2017]. Disponible en: http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html.

Hernández, C. 2011. Buscadores y Metabuscadores. [En línea] 18 de Noviembre de 2011. [Citado el: 25 de Febrero de 2017.]

Hernández, M. Generador de los grafos conceptuales a partir del texto en español. Tesis de Maestría. Instituto Politécnico Nacional. Centro de Investigación en computación. 2007.

Herrera, A. Modelos de Sistemas de Recuperación de Información Lingüística Difusa.2006.

Kamaruddin, S.S.; Bakar, A.A.; Hamdan, A.R.; Nor, F.M. Conceptual graph formalism for financial text representation, Information Technology, 2008. ITSim 2008. International Symposium, Vol.3, Aug. 2008., pp.1-6.

Kipper, K.; Korhonen A.; Ryant N.; Palme, M. Extending VerbNet with Novel Verb Classes. Fifth International Conference on Language Resources and Evaluation (LREC 2006). Genoa, Italy. June, 2006.

Klabbankoh, B.; Pinngern Q. Applied Genetic Algorithms in Information Retrieval. Faculty of Information Technology, King Mongkut´s Institute of Techology Ladkrabang, 2000.

Knublauch, H y otros. The Semantic Web. s.l.: Department of Computer Science, University of Toronto, 2004. 978-3-540-23798-3.

Lapuente, M.J. Ontologías. Madrid: s.n., 2013.

Larman, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design. 2004. Larman. 2013. Introducción al análisis y diseño orientado a objetos. México: Prentice Hal.ISBN:970-1 7-0261-1, 2013.

Laureano, Prof. Dra. Ana Lilia. El lenguaje de programación Java. El lenguaje de programación Java. 2004.

Last, M.; Maimon, O. A compact and Accurate Model for Classification, IEEE Transactions on Knowledge and Data Engineering. Vol. 16. No. 2. PP. 203-215. 2004.

Leyva, P.R; Viltres, H.; Pons Flores, L.A. Componentes y funcionalidades de un sistema de recuperación de la información. La Habana: s.n., Mayo de 2016, Revista Cubana de Ciencias Informáticas, págs. 150-162. ISSN: 2227-1899.

Leyva, P.R. 2016. Componentes y funcionalidades de un sistema de recuperación de la información. 2016.

Luna, J.A.; López Bonilla, M.; Durley Torres, I. Metodologías y métodos para la construcción de Ontologías. 50, Medellín: s.n., 2012. 0122-1701.

Maganto, A.S. Normas sobre metadatos. 2008.

Manning, C.; Schütze H. Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: Mayo 1999.

McIlraith, S.; Narayanan, M.; Paolucci, T. R. Payne.; Sycara, K. DAML-S: Web Service Description for the Semantic Web. The First International Semantic Web Conference (ISWC), June 2002.

Mendez, F.J. Martínez. Recuperación de información: Modelos, Sistemas y Evaluación. Murcia: s.n., 2004.

Miller, M. 2012. ISBN 978-0-7897-4365-7.

Mineau, G. W.; Stumme, G.; Wille, R. Conceptual Structures Represented by Conceptual Graphs and Formal Concept Analysis, International Conference on Conceptual Structures, 1999.

Montes-y-Gómez, M. Minería de texto: Un nuevo reto computacional. 3er Taller Internacional de Minería de Datos MINDAT-2001, Universidad Panamericana, Ciudad de México, Octubre 2001.

Netbeans. Netbeans IDE. Netbeans IDE. 2015 [En línea] 2015. [Citado el: 4 de Diciembre de 2016.] http://netbeans.org...

N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, & M. A. Musen. Creating Semantic Web Contents with Protege-2000. IEEE Intelligent Systems 16(2), pp. 60-71, 2001.

Ochando, Prof. Dr. Manuel Blázquez. Técnicas avanzadas de recuperación de información. Técnicas avanzadas de recuperación de información. 2012 [En línea] 11 de Octubre de 2012. [Citado el: 29 de Noviembre de 2016.] http://ccdoc-tecnicasrecuperacioninformacion.blogspot.com.

Pedraza-Jiménez, R.; Codina, Ll.; Rovira, C. Web semántica y ontologías en el procesamiento de la. Noviembre de 2007, El profesional de la
información, págs. 569-578.

Pressman, R. Ingeniería del software: un enfoque práctico. s.l.: 5 ed, 2005.

Castells, P.; Macías, J. A. Context-Sensitive User Interface Support for Ontology-Based Web.

Ramos, L. M Estrada. APACHE SOLR, UN MOTOR 11, 2012, Vol. 13. 1067-6079.

Ramírez, I. ¿Qué es java? [En línea] 10 de Abril de 2013. [Citado el: 29 de Noviembre de 2016.] https://articulos.softonic.com/que-es-java.

Rautenstrauch, R. Opciones avanzadas en las búsquedas de Google y Bing. Opciones avanzadas en las búsquedas de Google y Bing. 2010 [En línea] 28 de Octubre de 2010. [Citado el: 6 de Diciembre de 2016.] http://www.apasionadosdelmarketing.es.

Redondo, E.P y otros. Ontologías, metadatos y agentes: recuperación "semántica" de la. Ontologías, metadatos y agentes: recuperación "semántica" de la. Departamento de Biblioteconomía y Documentación. Granada: Jornadas de Tratamiento y Recuperación de la Información, Ontologías, metadatos y agentes: recuperación "semántica" de la información.

Rijsbergen Van, C.J. Information Retrieval. Department of Computing Science, University of Glasgow Second edition. 1.979.

Romero, A. R. Introducción a XML en castellano. 2000.

Salazar, O. A.; Medina Aguirre, F. A.; Chaves Osorio, J.A. Herramientas para el desarrollo rápido de aplicaciones web. 47, 2011, Vol. 1. 0122-1701.

Salinas, S. O. Traducción automática de textos a grafos conceptuales. Universidad Nacional de Colombia. 2011.

Salton, G.; Lesk, M. E. The SMART automatic document retrieval systems and illustration Common. ACM, 1965.

Sánchez, T. R. 2015. Metodología de desarrollo para la Actividad productiva de la UCI. 2015.

Schenker A.; Bunke Horst, M. L. A. K. Graph-theoretic techniques for Web content mining. World Scientific Publishing, 2005.

Ávila, Y.; Llanes, N. Sistema automatizado para la gestión de información en rehabilitación. 2008.

Shannon, C. A mathematical theory of communication. The Bell System Technical Journal, vol. 27, pp.379–423, October 1948.

Shehata, S.; Karray, F.; Kamel, M. Enhancing Text Retrieval Performance using Conceptual Ontological Graph. Data Mining Workshops, International Conference, pp. 39-44, Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06), 2006.

Sowa, J. F. Conceptual Graph Standard. Committee on Information Interchange and Interpretation, 2000

Sowa, J. F. Conceptual Graphs. Handbook of Knowledge Representation. Foundations of Artificial Intelligence, Vol. 3, Harmelen F.V., Lifschitz V., Porter B., (Ed.), Elsevier, 2008, pp 213-237.

Gruber, T. R. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), pp. 199-220, 1993.

Wainerman, E. 2001. Motores de búsqueda en Internet. 2001.

Zaninotto, F.; Potencier, F. Symfony. 2008.

Anexos

Anexo # 1.Encuesta realizada a los expertos.

Tabla 10. Cuestionario de actitudes realizado al experto #1

Afirmación	**Alternativas de respuestas**				
	1	**2**	**3**	**4**	**5**
El componente maneja de forma correcta la extracción y el envío de datos hacia Solr.				X	
El componente maneja de forma correcta el indicador tiempo de respuesta.					X
El componente maneja de forma correcta la variable calidad de respuesta.				X	
El componente cumple de forma correcta con todas las funcionalidades					X
Todos los elementos del componente se integraron y trabajan de forma correcta.					X

Tabla 11.Cuestionario de actitudes realizado al experto #2

Afirmación	**Alternativas de respuestas**				
	1	**2**	**3**	**4**	**5**
El componente maneja de forma correcta la extracción y el envío de datos hacia Solr.					X
El componente maneja de forma correcta el indicador tiempo de respuesta.				X	
El componente maneja de forma correcta la variable calidad de respuesta.					X
El componente cumple de forma correcta con todas las funcionalidades				X	
Todos los elementos del componente se integraron y trabajan de forma correcta.					X

Tabla 12.Cuestionario de actitudes realizado al experto #3

Afirmación	**Alternativas de respuestas**				
	1	**2**	**3**	**4**	**5**
El componente maneja de forma correcta la extracción y el envío de datos hacia Solr.				X	
El componente maneja de forma correcta el indicador tiempo de respuesta.				X	
El componente maneja de forma correcta la variable calidad de respuesta.					X
El componente cumple de forma correcta con todas las funcionalidades				X	
Todos los elementos del componente se integraron y trabajan de forma correcta.				X	

Tabla 13.Cuestionario de actitudes realizado al experto #4

Afirmación	**Alternativas de respuestas**				
	1	**2**	**3**	**4**	**5**
El componente maneja de forma correcta la extracción y el envío de datos hacia Solr.					X
El componente maneja de forma correcta el indicador tiempo de respuesta.					X
El componente maneja de forma correcta la variable calidad de respuesta.					X
El componente cumple de forma correcta con todas las funcionalidades					X
Todos los elementos del componente se integraron y trabajan de forma correcta.					X

Tabla 14. Cuestionario de actitudes realizado al experto #5

Afirmación	**Alternativas de respuestas**				
	1	**2**	**3**	**4**	**5**
El componente maneja de forma correcta la extracción y el envío de datos hacia Solr.				X	
El componente maneja de forma correcta el indicador tiempo de respuesta.				X	
El componente maneja de forma correcta la variable calidad de respuesta.					X
El componente cumple de forma correcta con todas las funcionalidades					X
Todos los elementos del componente se integraron y trabajan de forma correcta.					X

Tabla 12.Resultados del cuestionario de actitudes.

	Indicadores (Preguntas)				
Expertos	P1	P2	P3	P4	P5
E1	4	5	4	5	5
E2	5	4	5	4	5
E3	4	4	5	4	4
E4	5	5	5	5	5
E5	4	4	5	5	5