

Universidad de las Ciencias Informáticas

Facultad 2



Título: “Servicios Web para la creación de aplicaciones SEO”

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autores

Yadira Estrada Melgarejo

Yasiel Jorge Martínez

Tutor

Ing. David Rodríguez Rodríguez

Ciudad de La Habana, Junio de 2009

“Año del 50 aniversario del triunfo de la Revolución”

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yadira Estrada Melgarejo
Firma de la Autora

Yasiel Jorge Martínez
Firma del Autor

Ing. David Rodríguez Rodríguez
Firma del Tutor

A Nelson, nuestro primer líder del que aprendimos mucho y le debemos la idea de la tesis.

A los muchachos del proyecto, por estar prestos a ayudarnos siempre que los necesitamos con cualquier duda o problema.

A Pedro, por el tiempo y la paciencia en ayudarnos cada vez que algo no salía bien, aun a distancia.

A nuestro tutor, por el tiempo dedicado y la disposición de ayudarnos siempre.

A nuestras parejas, por soportarnos hablar casi todo el tiempo de la tesis.

A todos nuestros amigos, siempre pendientes de los avances.

A nuestros padres, por apoyarnos en cada momento.

Yadira y Yasiel

A mi abuelo, donde quiera que esté.

A mi abuela, por el amor de siempre.

A Hubert, por el cariño de padre que siempre me ha dado.

A mi mamá, por su entrega infinita y ser la persona más especial en mi vida.

A mi familia en general, por estar siempre pendientes de mis estudios.

A mi novio Pedro, por el amor desinteresado que siempre me ha brindado.

A Yaso, por soportar mis exigencias durante todo este tiempo.

A todos mis amigos, los de ahora, los de antes, los de siempre, por estar ahí para mí.

Yadira.

A mis abuelos, que aunque no estén presente físicamente, siempre los llevaré en mi corazón.

A mis abuelas por el cariño brindado.

A mis padres, por haberme apoyado en todo momento y haber estado presentes siempre para mí.

A mi novia, por estar siempre a mi lado brindándome su infinito amor, te amo tati.

A mi compañera de tesis, por todo el esfuerzo que realizamos juntos.

A todos aquellos que de una forma u otra me apoyaron para salir siempre adelante.

Yasiel.

Resumen

Internet se ha convertido en un mar de información proveniente de todas partes. El poder de difusión con que cuenta está siendo utilizado frecuentemente para alterar o tergiversar el contenido de noticias e informaciones. Su uso se ha sumado a la campaña mediática contra Cuba, debido a esto se convierte en una prioridad posicionar de una manera eficiente en la red, los sitios que expresan la información exacta que brindan los medios cubanos.

El presente trabajo pretende facilitar el desarrollo de herramientas que ayuden a la toma de decisiones de los webmasters cubanos en el ámbito del posicionamiento web, mediante la realización de servicios web que centralicen las principales funcionalidades SEO (del inglés: Search Engine Optimization). Potenciando así aspectos importantes en este tipo de aplicaciones como la reutilización, flexibilidad, interoperabilidad, extensibilidad y asequibilidad. Para garantizar la seguridad en el intercambio de información se llevaron a cabo diferentes mecanismos como la utilización de un canal seguro al nivel de transporte.

Palabras clave: Posicionamiento Web, SEO, Servicios Web.

Índice de Contenidos

Declaración de Autoría	II
Agradecimientos.....	III
Dedicatoria	IV
Resumen	V
Índice de Contenidos	VI
Índice de Figuras.....	X
Índice de Tablas	XII
Introducción	1
Capítulo 1: Fundamentación Teórica	6
1.1 Introducción	6
1.2 Conceptos básicos del posicionamiento web.....	6
1.2.1 Internet.....	6
1.2.2 Buscadores.....	7
1.2.3 Posicionamiento Web.....	8
1.2.3.1 SEO.....	8
1.3 Servicios Web	12
1.3.1 Servicios Web en la UCI	15
1.3.2 Servicios Web en el mundo.	15
1.4 Arquitectura Orientada a Servicios.	17
1.5 Metodología de Desarrollo. RUP	19
1.6 Lenguaje de modelado. UML.....	20

1.7	Herramientas y tecnologías.....	20
1.7.1	Herramienta CASE. Visual Paradigm.....	20
1.7.2	Entorno de Desarrollo Integrado. Eclipse.....	21
1.7.3	Plataforma de desarrollo. JEE.....	21
1.7.4	Lenguaje de programación. Java.....	21
1.7.5	Servidor Web. Apache Tomcat.....	22
1.7.6	Servicios Web. Apache Axis.....	22
1.7.7	Sistema Gestor de Base de Datos. MySQL.....	23
1.8	Conclusiones.....	23
Capítulo 2: Características del Sistema.....		24
2.1	Introducción	24
2.2	Propuesta de sistema.....	24
2.3	Modelo de Dominio.....	27
2.4	Relación de los Requerimientos del sistema.....	29
2.4.1	Requerimientos funcionales.....	29
2.4.2	Requerimientos no funcionales.....	31
2.5	Modelo de Casos de Uso del Sistema.....	31
2.5.1	Definición de los actores del sistema.....	31
2.5.2	Definición de los Casos de Uso del sistema.....	32
2.5.3	Descripción de los Casos de Uso del sistema.....	34
2.6	Conclusiones.....	42
Capítulo 3: Análisis y Diseño del sistema		43

3.1	Introducción	43
3.2	Modelo de Análisis.	43
3.2.1	Diagrama de Clases del Análisis.....	43
3.3	Modelo de Diseño.	45
3.3.1	Diagrama de clases del Diseño.	45
3.3.2	Diagramas de interacción del Diseño. Diagrama de Secuencia.....	49
3.4	Diseño de la Base de Datos.	53
3.4.1	Modelo lógico de datos. Diagrama de Clases Persistentes.	53
3.4.2	Modelo físico de datos (Modelo de Datos).	54
3.5	Patrones de Diseño empleados durante el desarrollo del sistema.....	54
3.6	Tratamiento de Errores.	57
3.7	Aspectos de Seguridad en el sistema.	58
3.8	Conclusiones.....	59
Capítulo 4:	Implementación y Prueba	60
4.1	Introducción	60
4.2	Modelo de implementación del sistema.	60
4.2.1	Diagrama de Despliegue.....	60
4.2.2	Diagrama de Componentes.	61
4.3	Modelo de Pruebas.	65
4.3.1	Métodos de Prueba.	66
4.3.2	Pruebas de Caja Blanca. Caso de Prueba para el método Título.....	67
4.3.3	Pruebas de Caja Blanca. Caso de Prueba para el método palabrasPredominantesKeywords.....	70

4.4	Conclusiones.....	74
Capítulo 5: Estudio de Factibilidad		75
5.1	Introducción	75
5.2	Aplicar Método de Estimación Puntos por Casos de Uso.....	75
5.2.1	Cálculo de Puntos de Casos de Uso sin ajustar.	75
5.2.2	Ajustar los Puntos de casos de uso.....	78
5.2.3	Cálculo del Esfuerzo.....	81
5.2.4	Distribución del Esfuerzo entre las diferentes actividades de un proyecto.....	82
5.2.5	Calcular el costo de todo el proyecto y su tiempo de desarrollo.....	82
5.3	Beneficios tangibles e intangibles.	83
5.4	Análisis de costo.	84
5.5	Conclusiones.....	85
Conclusiones Generales.....		86
Recomendaciones		87
Referencias Bibliográficas.....		88
Bibliografía.....		89
Anexos		91
Glosario de Términos.....		94

Índice de Figuras

Fig. 1 Principios que deben cumplir los servicios web.	14
Fig. 2 Arquitectura SOA tradicional.	19
Fig. 3 Modelo de Dominio.	28
Fig. 4 Diagrama de Caso de Uso del Sistema.	34
Fig. 5 Diagrama de Clases del Análisis del CUS Obtener Clave Sesión.	44
Fig. 6 Diagrama de Clases del Análisis del CUS Obtener Estadígrafos Página Web.	44
Fig. 7 Diagrama de Clases del Análisis del CUS Obtener Estadígrafos Palabras Predominantes.	45
Fig. 8 Diagrama de Clases del Diseño del CUS Obtener Clave Sesión y del CUS Cambiar Propiedades Conexión.	46
Fig. 9 Diagrama de Clases del Diseño del CUS Obtener Estadígrafos Página Web.....	47
Fig. 10 Diagrama de Clases del Diseño del CUS Obtener Estadígrafos Página Web y del CUS Obtener Estadígrafos Palabras Predominantes.....	48
Fig. 11 Diagrama de Secuencia del CUS Obtener Clave Sesión.	49
Fig. 12 Diagrama de Secuencia del CUS Cambiar Propiedades Conexión.	50
Fig. 13 Diagrama de Secuencia del CUS Obtener Estadígrafos Página Web.	51
Fig. 14 Diagrama de Secuencia del CUS Obtener Estadígrafos Palabras Predominantes.	52
Fig. 15 Diagrama de Clases Persistentes	53
Fig. 16 Modelo de Datos.....	54
Fig. 17 Diagrama de Despliegue	61
Fig. 18 Diagrama de Componentes General del Sistema.	62
Fig. 19 Diagrama de Componentes del Subsistema Descargar_HTML.....	63
Fig. 20 Diagrama de Componentes del Subsistema Analizador_HTML.....	63

Fig. 21 Diagrama de Componentes del Subsistema BaseDatos	64
Fig. 22 Diagrama de Componentes del Subsistema Sesión.....	64
Fig. 23 Diagrama de Componentes del Subsistema Página_HTML.....	65
Fig. 24 Código correspondiente al método Título.....	67
Fig. 25 Grafo correspondiente al método Título.....	68
Fig. 26 Código correspondiente al método PalabrasPredominantesKeywords.....	70
Fig. 27 Grafo correspondiente al método PalabrasPredominantesKeywords.....	71
Fig. 28 Diagrama de Secuencia del CUS Obtener Estadígrafos Página Web. Escenario Cantidad de Imágenes.....	91
Fig. 29 Diagrama de Secuencia del CUS Obtener Estadígrafos Palabras Predominantes. Escenario Peso Palabras Predominantes.....	92
Fig. 30 Diagrama de Secuencia del CUS Obtener Estadígrafos Palabras Predominantes. Escenario Cantidad de Enlaces.....	93

Índice de Tablas

Tabla 1 Definición del actor del sistema.....	32
Tabla 2 Definición de los Casos de Uso del Sistema.....	32
Tabla 3 Descripción del Caso de Uso Obtener Clave Sesión.....	34
Tabla 4 Descripción del Caso de Uso Cambiar Propiedades Conexión.	36
Tabla 5 Descripción del Caso de Uso Obtener Estadígrafos Página Web.....	37
Tabla 6 Descripción del caso de Uso Obtener Estadígrafos Palabras Predominantes.	40
Tabla 7 Descripción de Errores.....	57
Tabla 8 Caso de Prueba para el Camino 1-2-7.....	68
Tabla 9 Caso de Prueba para el Camino 1-2-3-5-6-2-7.....	69
Tabla 10 Caso de Prueba para el Camino 1-2-3-4-5-6-2-7	69
Tabla 11 Caso de Prueba para Camino 1-2-8	71
Tabla 12 Caso de Prueba para el Camino 1-2-3-2-8	72
Tabla 13 Caso de Prueba para el Camino 1-2-3-4-2-8.....	72
Tabla 14 Caso de Prueba para el Camino 1-2-3-4-5-4-2-8	72
Tabla 15 Caso de Prueba para el Camino 1-2-3-4-5-6-4-2-8	73
Tabla 16 Caso de Prueba para el Camino 1-2-3-4-5-6-7-4-2-8	73

Introducción

Internet constituye una fuente de recursos de información y conocimiento compartidos a escala mundial. Cada día crece el número de usuarios que acceden a la gran red de redes en busca de información que necesitan a través de los buscadores. Lo que sucede es que no siempre encuentran exactamente la información que desean, aun existiendo, y es porque estas páginas no aparecen en los primeros resultados de búsqueda, puesto que: “más del 80% de usuarios de Internet visitan sólo los tres primeros resultados de una búsqueda”. [1]

Entonces cabe preguntarse ¿Cómo lograr posicionar las páginas web para que aparezcan en los primeros resultados de búsquedas, de los principales buscadores? Si se quiere lograr que una página sea visitada por una mayor cantidad de personas, esta tiene que aparecer en las primeras posiciones, para los diferentes criterios que puedan teclear los internautas. Para ello existen diferentes estrategias SEO, estas no son más que técnicas para la optimización de las páginas en los motores de búsqueda y esto precisamente es la base de lo que se conoce como posicionamiento web.

Los webmasters¹ en Cuba, no tienen el hábito de aplicar estrategias de posicionamiento web a sus páginas, tampoco cuentan con alguna herramienta que facilite de alguna forma la realización de este trabajo. Esto resulta un punto clave, si se pretende que sean las páginas cubanas las que aparezcan en los primeros resultados y las personas adquieran la información exacta que brindan los medios cubanos. Pues existen muchos sitios que, o no dan la información exacta o la dan tergiversada, y son estos los que regularmente aparecen en los primeros resultados de búsquedas.

Esto afecta de forma directa a Cuba, que ha sufrido a lo largo de más de 40 años, la manipulación de la información por parte de los EE.UU. y con el bloqueo impuesto, se ha tenido limitaciones para lograr

¹ Persona encargada de la gestión y mantenimiento de un servidor web, fundamentalmente desde el punto de vista técnico

integrarse a la gran red de redes y a la misma vez tener sitios web propios que aparezcan bien posicionados.

De ahí la importancia de lograr que toda página web cubana además de estar en Internet, deba estar bien posicionada.

Como expresó el Viceministro Cubano de Informática y las Comunicaciones, Jorge Luís Perdomo Di-Lella, el 23 de mayo de 2007 en una entrevista concedida:

“En nuestro caso con la utilización de Internet es posible romper el monopolio de la información y nos permite llegar a todos los rincones del mundo con la publicación de nuestra realidad y la comunicación con todas las personas e instituciones que accedan a la red”. [2]

Actualmente existen en el mundo varias empresas con sitios en Internet que brindan servicios de posicionamiento web con solo solicitarlo. También existen softwares de escritorio con este fin. Pero ninguna de estas alternativas ofrecen algunos parámetros importantes que hay que tener en cuenta a la hora de posicionar una página web, como por ejemplo: si posee o no mapa de navegación para cada página, las palabras predominantes, los enlaces rotos, la consulta formada con las palabras claves que mejor recupera el sitio a posicionar, etc. Además de que se hace necesario pagar por acceder a estos servicios. Por ejemplo el *AgentWebRanking Pro Edition*, que es un software de escritorio, tiene licencia tiempo limitado por un año, seis meses, tres meses o un mes, pagando 550, 305, 170, 60 euros respectivamente.²

La Universidad de las Ciencias Informáticas (UCI) y específicamente el Grupo de Investigación y Desarrollo de Internet (GIDI) con su Línea dedicada al tema de Posicionamiento Web, después de un tiempo de investigación, pretende realizar herramientas para la ayuda a la toma de decisiones de los webmasters en el ámbito del posicionamiento web.

Para ello se hace necesario centralizar de manera accesible las principales funcionalidades SEO que puedan ser implementadas como servicios, con el objetivo de facilitar el desarrollo de este tipo de

² Datos tomados del sitio oficial de AgentWebRanking: <http://www.agentwebranking.com/es/pricing.htm>

aplicaciones y potenciar en ellas aspectos importantes como: la reutilización, flexibilidad, interoperabilidad, extensibilidad y asequibilidad.

Además de garantizar, hacer cambios mínimos en las aplicaciones que se estén desarrollando o se hayan desarrollado, si se produce algún cambio en los algoritmos de recuperación de los buscadores, los cuales pueden perfectamente cambiar con el objetivo de optimizar sus resultados de búsqueda.

Lo antes expuesto lleva a plantearse el siguiente **problema científico**:

¿Cómo facilitar el desarrollo de aplicaciones para la ayuda a la toma de decisiones SEO, reduciendo su tiempo de desarrollo?

El **objeto de estudio** son las estrategias y procesos de gestión de información que ayuden a la implementación de servicios web, lo que define como **campo de acción** los procesos automatizados de gestión de información, para la implementación de servicios en el ámbito del posicionamiento web.

Para dar solución a este problema se plantea el siguiente **objetivo general**:

Desarrollar un repositorio de servicios web para facilitar la implementación de aplicaciones que ayuden a la toma de decisiones SEO.

Para el cumplimiento de este objetivo se han trazado las siguientes **tareas de la investigación**:

- ✓ Buscar y analizar documentación sobre las estrategias de posicionamiento web que existen en el mundo.
- ✓ Buscar y analizar documentación sobre la implementación de servicios web.
- ✓ Identificar las principales funcionalidades que por sus características pueden ser implementadas como servicios, para facilitar el desarrollo de aplicaciones para la ayuda a la toma de decisiones SEO.

La investigación estará guiada por la siguiente **Idea a Defender**: Con el desarrollo de un repositorio de servicios web, se logrará facilitar la creación de aplicaciones SEO.

Como en todo proceso investigativo se utilizaron métodos para el desarrollo de la investigación, se tuvieron en cuenta los métodos de investigación científica, que se presentan a continuación

Métodos Teóricos

Análisis y Síntesis: Por medio de este método, se realizó una investigación de los aspectos relacionados con el posicionamiento web y la implementación de servicios web, permitiendo analizar la documentación existente. Además facilitó la extracción de los elementos más importantes relacionados con el campo de acción.

Histórico Lógico: Este método permitió constatar teóricamente como ha evolucionado la utilización de servicios en el ámbito del posicionamiento web hasta estos momentos.

Modelación: Este método permitió crear modelos a través de la metodología RUP.

Métodos Empíricos

Entrevista: Este método sirvió para entrevistarse con el cliente y obtener información acerca de los principales servicios que se deben brindar.

Experimento: Este método sirvió para verificar y obtener conocimiento sobre el objeto de estudio.

El trabajo está estructurado en 5 capítulos:

En el *capítulo 1* se abordan los conceptos fundamentales que permiten entender el mundo del posicionamiento web y los servicios web. Así como la fundamentación de las herramientas y tecnologías que se emplearon en el transcurso de la investigación.

En el *capítulo 2* se describen los principales procesos involucrados en el objeto de estudio. Se describen brevemente los conceptos relacionados con el dominio del problema. Se realiza una descripción general de los requisitos funcionales y no funcionales del sistema a desarrollar. Así como las descripciones de los actores y casos de uso del sistema, y los diagramas que representan a estos últimos para un mayor entendimiento del sistema propuesto.

En el *capítulo 3* se traducen los requerimientos del sistema a una especificación que describa cómo implementar el sistema mediante la modelación del análisis y el diseño del mismo. Además de ofrecer una descripción de los patrones de diseño empleados, algunos aspectos referentes a la seguridad del sistema y el manejo de los posibles errores que puedan ocurrir.

En el *capítulo 4* se describen los temas referentes a la implementación y prueba del sistema, se hace referencia a las principales tareas que se llevan a cabo en este flujo de trabajo, así como la descripción de los artefactos que se generan en el mismo.

En el *capítulo 5* se realiza una estimación del esfuerzo, costo y tiempo de desarrollo del sistema, usando para ello el método de Análisis de Puntos de Caso de Uso. Además se analizan los resultados obtenidos, permitiendo evaluar la factibilidad o no del proyecto.

Capítulo 1

Fundamentación Teórica

1.1 Introducción

En este capítulo se abordarán los conceptos fundamentales que permitirán entender el mundo del posicionamiento web y los servicios web. Así como la fundamentación de las herramientas y tecnologías que se emplearán en el transcurso de la investigación.

1.2 Conceptos básicos del posicionamiento web.

1.2.1 Internet

Internet ha supuesto una revolución sin precedentes en el mundo de la informática y de las comunicaciones. Se ha convertido en una oportunidad de difusión mundial, un mecanismo de propagación de la información y un medio de colaboración e interacción entre los individuos y sus ordenadores independientemente de su localización geográfica.

Su desarrollo y evolución gira entorno a cuatro aspectos distintos. Existe una evolución tecnológica que comienza con la primitiva investigación en conmutación de paquetes, ARPANET³ y tecnologías relacionadas. Hay aspectos de operación y gestión de una infraestructura operacional global y compleja. Existen aspectos sociales, que tuvieron como consecuencia el nacimiento de una amplia comunidad de internautas trabajando juntos para crear y hacer evolucionar la tecnología. Y finalmente, el aspecto de comercialización que desemboca en una transición enormemente efectiva hacia una infraestructura informática ampliamente desarrollada y disponible. [3]

³ Advanced Research Projects Agency Network, fue creada por encargo del Departamento de Defensa de los Estados Unidos como medio de comunicación para los diferentes organismos del país. El primer nodo se creó en la Universidad de California y fue la espina dorsal de Internet hasta 1990.

La historia es compleja y comprende muchos aspectos: tecnológico, organizacional y comunitario. Y su influencia alcanza no solamente al campo técnico de las comunicaciones computacionales sino también a toda la sociedad pues se ha convertido en una parte importante y a veces hasta indispensable en la vida de la mayoría de las personas de esta generación.

Cada vez es mayor el número de usuarios que acceden a la red en busca de información. También crece constantemente la cantidad de sitios web donde buscar. En el 2002 se contaba con 40 millones de sitios, ya en estos últimos años se ha duplicado por cada trienio, llegando a alcanzar los 160 millones en el pasado 2008. Pero la información se encuentra dispersa y se necesita para optimizar el proceso de búsqueda, la presencia de un buscador. [4]

1.2.2 Buscadores

Un buscador no es más que un *“sitio web que gestiona una base de datos confeccionada por robots. Esta base de datos almacena directamente el código de las páginas visitadas por los bots⁴ o parte de él.”* [5]

Muchos piensan que cuando hacen una consulta al buscador, este en ese momento hace la búsqueda por toda la red y después muestra los resultados de la información encontrada. Están en un error, pues si esto fuese así tardarían mucho para obtener lo que desean. Los buscadores han evolucionado a lo largo de los años, actualmente constan de tres partes fundamentales:

Robot o Araña: *“Es un programa que navega la web yendo desde cada página a todas las que esta da enlace y enviando e incorporando el código o partes de él a una base de datos de referencia”.* [5]

Base de datos documental: *“Conjunto de registros (unidades de información relevante) ordenados y clasificados para su posterior consulta, actualización o cualquier tarea de mantenimiento mediante aplicaciones específicas.”* [6]

Motor de búsqueda: *“Programa que selecciona y jerarquiza resultados entre las entradas de una base de datos en función de unos términos y criterios de búsqueda.”* [5]

⁴ Forma de referirse a los robots.

En la actualidad, los buscadores se guían por índices automáticos que les son asignados a determinadas páginas web. Cada uno tiene sus propios métodos para indexar previamente las páginas dándole una relevancia según su algoritmo, para luego almacenarlas y recuperarlas en el momento en que algún internauta realice una búsqueda. Pero para que el usuario esté satisfecho con los resultados, tiene que encontrar en las primeras posiciones, páginas relevantes que se adecuen a lo que desea realmente. Y aquí es donde entra a jugar un papel importante lo que se conoce como: posicionamiento web.

1.2.3 Posicionamiento Web.

El posicionamiento web es algo trabajoso y continuo, consiste en “...aplicar diversas técnicas tendentes a lograr que los buscadores de Internet encuadren nuestra página web en una posición y categoría deseada dentro de su página de resultados para determinados conceptos clave de búsqueda.” [7]

Existen dos tipos de posicionamiento web, en dependencia del resultado que generan los motores de búsqueda, estos se pueden clasificar en:

- ✓ Posicionamiento en enlaces patrocinados SEM (del inglés: Search Engine Marketing).
- ✓ Posicionamiento natural SEO (del inglés: Search Engine Optimization).

El primero está basado en los resultados patrocinados “(...) cuya clasificación depende del dinero que se invierta en los anuncios”. [7] El segundo tiene su cimiento en los resultados naturales u orgánicos, que “(...) están basados en el algoritmo imparcial de los buscadores (...)”. [7] y en el cual se centra el presente trabajo.

1.2.3.1 SEO

SEO, en español “Optimización para Motores de Búsqueda”, define el proceso de modificación y análisis de las páginas web para conseguir posicionarla en los puestos más altos dentro de los mejores buscadores. El análisis debe ser amplio, e incluir entre otras cosas:

✓ *Palabras predominantes.*

Son las palabras que más frecuencia tienen dentro de la página y deben estar incluidas como palabras claves del sitio. De ellas se deben tener en cuenta varios conceptos que permiten medir su importancia dentro de la página. Entre los más significativos se tienen:

Frecuencia: Es el número de veces que el término de búsqueda aparece en la página.

Recuento: Cuenta el total de palabras en una página excluyendo las etiquetas HTML (del inglés: HyperText Markup Language).

Peso: El peso es el porcentaje que el término de búsqueda representa sobre el total de palabras en la página. Dado que el término de búsqueda puede estar formado por varias palabras, la fórmula general del peso se puede definir como:

$P = n * F / R$ donde:

n= Número de palabras del término de búsqueda

R= Recuento del área

F= Frecuencia del término de búsqueda.

El peso sirve como chequeo de la proporción entre frecuencia y recuento, no tanto como una variable independiente que actúe por sí misma.

Relevancia: La relevancia mide la distancia entre el comienzo del área y el lugar donde aparece la palabra clave. Si el término de búsqueda aparece como primera palabra del área la prominencia es 100% y si es la última el 0%. Si aparece varias veces, se promedia. [5]

Tales conceptos ayudarán a conocer con más precisión cómo situar las palabras en el documento, con la frecuencia que se debe hacer, cuán separado del principio debe estar una palabra, etc. Además de que no se debe descuidar tratando de obtener esto, la calidad del contenido a publicar.

✓ *El título*

Es el factor más importante para la optimización de una página web que se desea posicionar. Es la frase con la cual se describe en pocas palabras de lo que trata una determinada página dentro de todo un sitio web. Un buen título enfocado hacia el posicionamiento en buscadores se redacta incluyendo la palabra clave a posicionar, su longitud se debe encontrar entre determinados límites y debe tener gran relación con el contenido de la página.

✓ *Las metatags*

Una gran cantidad de arañas se fijan en estas etiquetas, pues indican a los buscadores la temática de la página, así como las palabras claves que mejor definen el sitio.

✓ *Palabras claves o keywords.*

Es una palabra o combinación de estas que definen el sitio web y pueden llegar a ser las que tecleen los internautas en sus búsquedas, si se hace un buen análisis y recolección de ellas. Para que sean efectivas deben resultar comunes en el lenguaje del usuario, tener una baja competencia, estar incluidas también dentro de la URL (del inglés: Uniform Resource Locator) y por supuesto deben estar bien relacionadas con el contenido de la página web a posicionar.

✓ *Código embebido*

La presencia de código embebido puede causar problemas en el análisis de la página por parte del buscador. La razón por la cual sucede esto es que lenguajes como el Java Script y el Flash no son interpretados por los buscadores. Por lo que las páginas web que quieran estar bien posicionadas, deben disminuir el uso de este tipo de código.

✓ *Mapa de navegación en la página.*

La presencia del mapa de navegación mejora la indexación de muchos sitios. Además para el usuario es bastante cómodo, ya que facilitan la interacción con el sitio desde el punto de vista ergonómico, pues muestran cómo están estructuradas las páginas web en el sitio y permite orientar la navegación a través de esta estructura física.

✓ *Cantidad de tablas, imágenes y enlaces que posee cada página.*

Conocer la cantidad de tablas es importante ya que el buscador cuando está leyendo el contenido de la página deja para al final el análisis de estas y no lee la información dentro de ellas como lo hace un usuario. Además hay que tener en cuenta que las tablas anidadas de cierta complejidad pueden actuar como un impedimento a la hora de navegar por los sitios web.

La cantidad de imágenes es otro factor a tener en cuenta por el buscador, pues las páginas que presentan carga de imágenes, se les dificulta su indexación.

Conocer los enlaces de cada página es también importante. Se trata por un lado, de hacer un análisis de los enlaces internos de la misma y conseguir enlaces de otras páginas que enlacen a la nuestra. La finalidad es crear un circuito interno de enlaces estratégicamente ubicados y con textos previamente definidos, que apoyen a las páginas que se quieran posicionar. Por otro lado, los enlaces externos, son tenidos en cuenta por los buscadores para determinar de qué trata tu sitio web. Cuando enlazas a alguien, los buscadores van tener en cuenta prácticamente todo, texto, descripción, situación, contexto, número de enlaces, antigüedad, importancia de esa web, factores que hacen que la URL enlazada aparezca en los resultados de una u otra búsqueda.

✓ *Enlaces rotos dentro de las páginas.*

Este factor tiene gran importancia en el posicionamiento web. Poseer enlaces rotos dificulta la navegabilidad del buscador al recorrer las páginas web de un sitio y esto influye directamente en el número de visitas a la página.

✓ *Correos de contacto.*

Conocer el correo del webmaster de la página u otro contacto es esencial, ya que por esta vía se puede intercambiar links, además de contactar con este para intercambiar ideas y estrategias de posicionamiento web.

✓ *Lenguaje de la página.*

Es útil, pues permite indicarle a los motores de búsqueda el idioma en que está hecho el documento, esto puede ser muy importante para directorios que solo indexen sitios realizados en algún idioma específico.

✓ *Etiquetas relevantes de la página para el posicionamiento.*

Estas son importantes pues son empleadas con el objetivo de especificar la importancia que tiene un determinado contenido dentro de la página. Ejemplo son las etiquetas y para los textos en negrita, <u> para los subrayados, <i> para los cursivos y <h1>, <h2>, <h3>, <h4>, <h5> y <h6>, para establecer jerarquía en los contenidos.

Todos los elementos antes expuestos constituyen una guía para el webmaster a la hora de crear sus sitios web y lograr una adecuada optimización de sus páginas, en aras de obtener un buen posicionamiento web.

1.3 Servicios Web

La sociedad actual, el desarrollo de las tecnologías de la información y la comunicación, las tendencias comerciales a través de medios electrónicos y el ser humano en el siglo XXI requieren la automatización de los procesos cotidianos. Estos argumentos han sido algunos de los pilares que han hecho surgir nuevos desarrollos tecnológicos. Sobre todo en la esfera del software, creando así una nueva perspectiva sobre el desarrollo del mismo, imponiendo nuevas arquitecturas entre las que se destacan las arquitecturas web. Sobre estas últimas, se han empezado a aprovechar una gran cantidad de estándares y protocolos que facilitan la interoperabilidad de las aplicaciones sobre la red y especialmente sobre Internet naciendo así el concepto de servicio web.

Existen varias definiciones sobre los mismos, lo que muestra su complejidad a la hora de dar un adecuado concepto. Uno posible sería hablar de ellos como un *conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web*. [8]

Estos se constituyen en una capa adicional a estas aplicaciones de tal forma que pueden interaccionar entre ellas, usando para comunicarse tecnologías estándares que han sido desarrolladas en el contexto de Internet.

Ejemplo de ellas son las siguientes:

XML

XML, de sus siglas en inglés Extensible Markup Language (lenguaje de marcas extensible). Fue desarrollado por la World Wide Web Consortium (W3C) para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos puedan buscar, comparar y compartir información en la red [10] de una manera segura, fiable y fácil.

XML tiene otras aplicaciones entre las que destaca su uso como estándar para el intercambio de datos entre diversas aplicaciones o software con lenguajes privados como en el caso del SOAP.

SOAP

SOAP (del inglés: Simple Object Access Protocol) fue creado por la Microsoft y constituye el *protocolo de acceso que permite la comunicación entre aplicaciones remotas a través de Internet*. [10] De ahí que sea uno de los protocolos más importantes utilizados en los servicios web.

WSDL

Del inglés Web Services Description Language, un formato XML que se utiliza para describir servicios web.

El WSDL permite que un servicio y un cliente establezcan un acuerdo en lo que se refiere a los detalles de transporte de mensajes y su contenido, a través de un documento procesable por dispositivos. WSDL representa una especie de contrato entre el proveedor y el que solicita y especifica la sintaxis y los mecanismos de intercambio de mensajes. [8]

UDDI

Son las siglas del catálogo de negocios de Internet denominado Universal Description, Discovery, and Integration. El registro en el catálogo se hace en XML.

UDDI es un marco independiente de la plataforma para describir servicios, negocios e integrar servicios de negocios. La estructura de UDDI está basada sobre los servicios estándares de la web, lo que quiere decir que UDDI es accesible como otros servicios web. [11]

Su objetivo es ser accedido por los usuarios y clientes y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios web del catálogo de registros.

Los servicios además deben cumplir los siguientes principios:



Fig. 1 Principios que deben cumplir los servicios web.

El cumplimiento de estos principios, constituyen la ventaja de la utilización de servicios en cualquier ámbito.

Los servicios web surgieron para finalmente poder lograr la tan esperada comunicación entre diferentes plataformas. Estableciéndose rápidamente como la tecnología clave para aplicaciones integradas, de tal forma que se están convirtiendo en el despliegue de tecnología más importante de los próximos años.

1.3.1 Servicios Web en la UCI.

La UCI desde la perspectiva de Ciudad Digital está dando los primeros pasos para lograr una integración de los servicios, por los beneficios que su uso reportaría en los diferentes procesos que se llevan a cabo dentro de la misma.

Ya se cuenta con un propio UDDI, donde se pueden encontrar los servicios web para la gestión académica de las asignaturas y cursos de los estudiantes (Akademos), para la gestión de trabajadores que son plantilla de la UCI (ASSETS- Sistema de Capital Humano), para la gestión de las personas con beca en la Universidad (Registro de Residentes) entre otros.

Pero hasta el momento ninguno cuenta con funcionalidades destinadas a facilitar la implementación de aplicaciones SEO.

Por la importancia que reportaría el desarrollo de estas aplicaciones para la ayuda a la toma de decisiones de los webmasters cubanos en el ámbito del posicionamiento web, se propone brindar un servicio que ofrezca funcionalidades que puedan servir como estrategias SEO, facilitando así la implementación de este tipo de aplicaciones, tan importante para Cuba si se quiere que las páginas cubanas aparezcan en los primeros lugares de resultados de búsqueda y así lograr difundir la verdad sobre Cuba en Internet.

1.3.2 Servicios Web en el mundo.

Existen en el mundo varias empresas reconocidas, como Google y Yahoo! que brindan algunos servicios web a los que los usuarios interesados pueden acceder. Algunos de estos, pudieran servir para facilitar la implementación de aplicaciones SEO. Y entonces, cabe preguntarse: ¿Por qué no usar estos, ya

existentes, en vez de realizar nuevos servicios con este fin? Lo cierto es que existen algunas limitaciones para su uso que más adelante se tratarán.

Algunos de los servicios web que ofrece **Google**, de acuerdo con el fichero WSDL, son los siguientes:

- ✓ **doGoogleSearch** (realiza una búsqueda y recoge los resultados programáticamente)
- ✓ **doGetCachedPage** (obtiene acceso a la versión en caché de una página de la última vez que la vio Google)
- ✓ **doSpellingSuggestion** (ofrece sugerencias de deletreo de palabras que se suelen escribir mal).

Para poder usar estos servicios, primeramente se tendría que crear una cuenta en Google, para poder recibir por correo la clave de licencia de la Google API, pues esta clave se necesitará para pasarla como parámetro cada vez que invoque las funciones de búsqueda de Google.

Es necesario también descargar el kit de desarrollo de Google Web API que incluye algo de código de ejemplo en varios lenguajes de programación y lo más importante, incluye el fichero WSDL.

Además de lo antes expuesto, el uso de estos servicios tiene otras limitaciones. Google provee a cada usuario que posea una cuenta en su sitio, un límite de 1000 consultas por día al servicio, esto se debe a que esta API es solo un experimento y por ello su utilización se encuentra limitada.

La manera que tiene Google de limitar su servicio es utilizando un “Licence Key” que se debe enviar con cada consulta al servicio del sitio. Cada desarrollador puede tener solo un “Licence Key” y está prohibido crear otras cuentas para obtener más de una licencia.

Otras limitaciones con las que se cuenta al utilizar este servicio son: máximo de 10 palabras en la consulta y máximo de 10 resultados por consulta.

En el caso de **Yahoo!**, además del servicio de búsqueda tradicional, también incorpora otros servicios:

- ✓ Búsqueda de audio: permite realizar búsquedas de archivos de música

- ✓ Análisis de contenido: provee una lista de palabras significativas y frases extraídas de un amplio contenido.
- ✓ Búsqueda de imágenes: permite realizar búsquedas de archivos de imágenes.
- ✓ Búsqueda local: permite realizar búsquedas por medio de Yahoo! Local.
- ✓ MyWeb: es un motor de búsquedas personales que permite guardar páginas web, compartirlas y agregar comentarios sobre la misma.
- ✓ Búsquedas de noticias: permite realizar búsquedas de noticias en la web.
- ✓ Explorador de sitios: provee información que contiene el servicio “Site Explore” de Yahoo! acerca de páginas web.
- ✓ Búsquedas de video: permite realizar búsquedas de archivos de video.

Yahoo! también limita las peticiones de búsqueda mediante un “Application ID”, el cual es un string que se debe colocar en cada petición y solo se podrán realizar 5000 por día en el caso del servicio de búsqueda. Cada aplicación deberá tener su propio ID, siendo posible que un desarrollador obtenga múltiples IDs para sus aplicaciones.

Como se ha podido apreciar todas estas limitaciones impiden un buen uso de estos servicios, de ahí la importancia de crear nuevos servicios que no tengan en cuenta este tipo de restricciones.

1.4 Arquitectura Orientada a Servicios.

Las arquitecturas orientadas a servicios no son nuevas. Para muchas personas la primera de estas surgió en el pasado con el uso de DCOM (en español: Modelo de Objetos Componentes Distribuido)⁵ o los ORBs (en español: Agente de Petición de Objeto) de CORBA⁶.

⁵ DCOM: Facilita la distribución transparente de objetos a través de redes y de Internet.

⁶ CORBA: Common ORB Architecture, un estándar basado en ORB.

SOA es un modelo de componentes que interrelaciona las diferentes unidades funcionales de las aplicaciones, denominadas servicios, a través de interfaces y contratos bien definidos entre esos servicios. La interfaz se define de forma neutral, y debería ser independiente de la plataforma hardware, del sistema operativo y del lenguaje de programación utilizado. Esto permite a los servicios, contruidos sobre sistemas heterogéneos, interactuar entre ellos de una manera uniforme y universal. [12]

De aquí se infiere que SOA es una arquitectura que fomenta la creación de aplicaciones que se adaptan fácilmente a los inevitables y frecuentes cambios del negocio. La Arquitectura Orientada a Servicios es tanto un marco de trabajo como un marco de implementación para el desarrollo de los procesos de negocio de una organización. Propone desglosar las funcionalidades principales de las aplicaciones en servicios autónomos y luego orquestarlos para formar nuevas aplicaciones de negocio. Estos servicios se encuentran disponibles en la red y presentan interfaces estándares bien definidas que permiten el flujo de mensajes entre proveedores y consumidores.

La Arquitectura Orientada a Servicios (SOA), está formada por servicios de aplicación débilmente acoplados y altamente interoperables. Los servicios web se han convertido en la forma más habitual de implementarla, ya que esta tecnología posee un conjunto de características que permiten cubrir todos los principios de la orientación a servicios. La definición de la interfaz encapsula las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo. Con esta arquitectura, se pretende que los componentes de software desarrollados sean muy reusables, ya que la interfaz se define siguiendo un estándar. En la figura, se presenta la estructura básica de funcionamiento de una SOA tradicional.

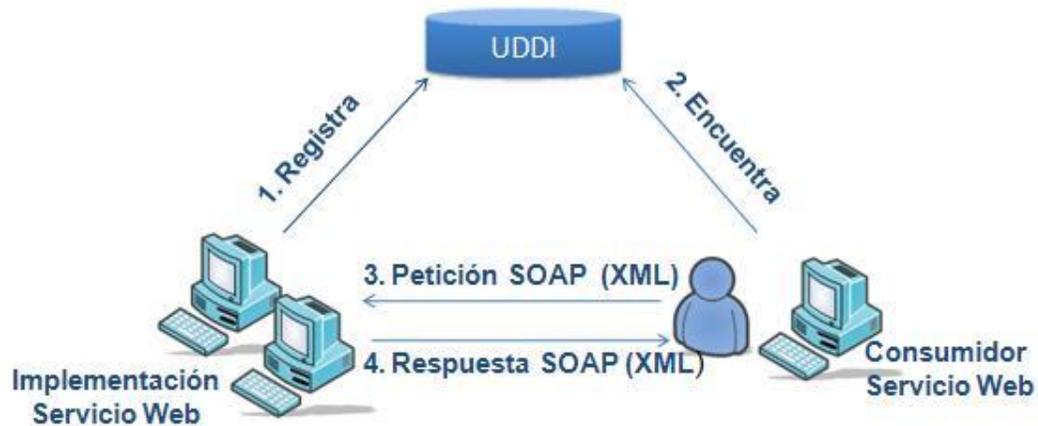


Fig. 2 Arquitectura SOA tradicional.

Se puede observar la existencia de tres roles:

Cliente del servicio: Es el que solicita la ejecución del servicio web, y por lo tanto el que lo consume.

Proveedor del servicios: Es el encargado de implementar el servicio web y ofrecerlo a los clientes a través del WSDL.

Registro del servicio (UDDI): Es un repositorio donde se almacenan las descripciones de los servicios, para que así los clientes puedan buscar el servicio web que mejor se adapte a sus necesidades.

1.5 Metodología de Desarrollo. RUP

Se ha decidido usar RUP (del inglés: Rational Unified Process) Proceso Unificado de Rational, pues junto al Lenguaje Unificado de Modelado (UML), constituye una metodología estándar de las más usadas para el análisis, implementación y documentación de sistemas orientados a objetos. Es completo y con gran nivel de organización a la hora de trabajar, puesto que se documenta todo lo que se elabora, facilitando el control de cambios y la reutilización.

Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso y son estas tres características las que distinguen precisamente a RUP de otras metodologías de desarrollo y la hacen mucho más eficaz.

1.6 Lenguaje de modelado. UML.

Se utilizará el Lenguaje Unificado de Modelado (UML) por ser el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Se define como *un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software* [13] y está destinado a los sistemas de modelado que utilizan conceptos orientados a objetos.

Es importante resaltar que UML se usa para especificar y no para describir métodos o procesos y para construir modelos, lo cual no constituye una guía al desarrollador en la forma de realizar el análisis y diseño orientado a objetos.

En otras palabras, este lenguaje estandariza los artefactos y la notación, pero no define un proceso oficial de desarrollo [13]. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

1.7 Herramientas y tecnologías.

Durante la realización de este trabajo se usarán distintas herramientas para apoyar la implementación de los servicios web. Como herramienta CASE (del inglés: Computer Aided Software Engineering), se usará Visual Paradigm, como entorno integrado de desarrollo Eclipse y el lenguaje de programación será Java.

1.7.1 Herramienta CASE. Visual Paradigm.

Se decidió utilizar Visual Paradigm (VP) en su versión 5.3 por ser una herramienta robusta, usable y portable. Además de que utiliza UML como lenguaje de modelado. Es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto, permite control de versiones y realizar ingeniería tanto directa como inversa en diferentes lenguajes, entre ellos Java.

Además se puede representar todos los tipos de diagramas UML para las distintas fases que propone RUP, como la captura de requisitos, análisis, diseño e implementación. Esta herramienta facilitará la comunicación, ya que utiliza un lenguaje estándar común a todo el equipo de desarrollo. Presenta la posibilidad de la interoperabilidad con otras aplicaciones como es el Rational Rose. Tiene disponible

distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community (que es gratuita). También se facilita licencias especiales para fines académicos.

1.7.2 Entorno de Desarrollo Integrado. Eclipse.

Eclipse Ganymede es el IDE (del inglés: Integrated Development Environment) con el que se decidió trabajar, por ser gratuito, libre y potente. Proporciona todas las herramientas y funciones necesarias para realizar el presente trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar. Incorpora un WSDL Editor, el cual permite confrontar más fácilmente los servicios web basados en XML, al proporcionar una perspectiva en tiempo de diseño del WSDL, el editor proporciona análisis sencillos, creación y expansión de los servicios web.

Si bien las funciones de Eclipse son más bien de carácter general, las características del programa se pueden ampliar y mejorar mediante el uso de plug-in lo que lo hace más extensible, además de que es multiplataforma. La utilización de Eclipse garantiza que se aumente la productividad y la calidad del software. Además brinda una potente ayuda en los aspectos de vida del desarrollo del software.

1.7.3 Plataforma de desarrollo. JEE.

La plataforma de desarrollo que se usará es JEE (del inglés: Java Enterprise Edition), pues es el estándar de la industria en soluciones empresariales. JEE aprovecha muchas de las características de la plataforma JSE (del inglés: Java Standard Edition) como la portabilidad, la API para acceso a bases de datos y un modelo de seguridad que protege los datos incluso en aplicaciones de Internet. También apoya las nuevas tecnologías de servicios web y asegura la interoperabilidad de estos. Todo con gran sencillez, portabilidad, escalabilidad e integración.

1.7.4 Lenguaje de programación. Java.

El lenguaje de programación escogido es Java, pues una de sus mejores cualidades es que es multiplataforma. Este potente lenguaje de programación ofrece además, múltiples ventajas:

- ✓ **Sencillez**, en comparación con otros lenguajes como C ó C++ y es notable la ausencia de punteros.

- ✓ **Distribuido**, proporciona las librerías y herramientas para que los programas puedan ser distribuidos, pues se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como HTTP (Hypertext Transfer Protocol). Esto ha permitido acceder a la información a través de la red con mucha facilidad.
- ✓ **Robusto** pues realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución.
- ✓ Es un lenguaje **orientado a objetos**, esto lo hace muy útil para la representación de entidades tal y como las idean las personas.
- ✓ La **seguridad** inherente a la no existencia de punteros hace el código más seguro.

1.7.5 Servidor Web. Apache Tomcat.

El servidor web que se usará es Apache Tomcat en su versión 6.0.16, pues posee un buen balance de carga, así como poca latencia. Además de poder integrarse fácilmente a muchos sistemas y servidores de forma eficiente. Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

1.7.6 Servicios Web. Apache Axis.

Para la fácil implementación de servicios web se usará Apache Axis en su versión 1.4, por ser un motor de servicios web de los más usados para proporcionar y consumir servicios. Eficaz, estable y diseñado con extensibilidad. Esta última característica le permite apoyar la creación de servicios web de manera eficiente, con la garantía de la interoperabilidad. Apache Axis, posee una activa comunidad de usuarios y son muchas las compañías que lo usan para la implementación de servicios web en sus productos. Además de que posee una fácil integración con el IDE Eclipse.

1.7.7 Sistema Gestor de Base de Datos. MySQL.

El sistema gestor de base de datos usado es MySQL en su versión 4.1, por poseer gran velocidad a la hora de realizar las operaciones, convirtiéndolo en uno de los gestores que ofrecen mayor rendimiento. Esta última característica es muy importante, cuando se desarrollan servicios web, donde la capacidad de respuesta y el bajo consumo de recursos son indispensables. Además permite conexiones entre diferentes máquinas con distintos sistemas operativos y es multihilo

Está licenciado bajo la GPL de la GNU⁷ para cualquier uso compatible con esta licencia, pero para aquellas entidades que quieran incorporarlo en productos privativos deben comprar una licencia específica que les permita este uso. El lenguaje de programación que utiliza es SQL (del inglés: Structured Query Language), utilizado de forma generalizada en las bases de datos relacionales.

1.8 Conclusiones

En el presente capítulo se han tratado los elementos necesarios relacionados con el concepto de posicionamiento web, para su mejor comprensión. Se brinda además una panorámica sobre los servicios web y los principales protocolos en que se basan, así como la arquitectura que utilizan. Además se aborda sobre las limitaciones que trae consigo, usar servicios web como los que brindan Google y Yahoo!. Se realiza un análisis de la metodología de desarrollo que se utilizará, los lenguajes de programación y todas las herramientas que se tendrán en cuenta para la implementación de los servicios web. Todos estos aspectos son de gran importancia para el desarrollo y la comprensión del sistema que se presenta.

⁷ GPL(del inglés: General Public License) es una licencia libre, sin derechos para software y otro tipo de trabajos.

Capítulo 2

Características del Sistema

2.1 *Introducción*

En el presente capítulo se describen los principales procesos involucrados en el objeto de estudio. Se describen brevemente los conceptos relacionados con el dominio del problema. Se realiza una descripción general de los requisitos funcionales y no funcionales del sistema a desarrollar. Así como, las descripciones de los actores y casos de uso del sistema, y los diagramas que representan a estos últimos para un mayor entendimiento del sistema propuesto.

2.2 *Propuesta de sistema.*

Se hace necesario lograr que los sitios web de Cuba aparezcan dentro de los primeros resultados en los buscadores para cualquier criterio de búsqueda y de esta forma hacer llegar la información exacta que brindan los medios cubanos a los internautas, pues es conocida la manipulación de la información que por parte de los EE.UU. ha sufrido Cuba, por más de 40 años.

En Cuba, es nulo el desarrollo de herramientas que ayuden a la toma de decisiones de los webmasters en el ámbito del posicionamiento web. Teniendo en cuenta la importancia de fomentar la creación de este tipo de aplicaciones y reducir además su tiempo de desarrollo, el presente trabajo se concentra en una propuesta de sistema basada en servicios web, que posibilite centralizar de manera accesible, las principales funcionalidades SEO que puedan servir para facilitar el trabajo a los desarrolladores.

A esta serie de funcionalidades se podrá acceder sin importar la plataforma o el lenguaje de programación, en la que se estén desarrollando las aplicaciones SEO. Potenciando así aspectos importantes en este tipo de aplicaciones como: la reutilización, flexibilidad, interoperabilidad, y asequibilidad.

La siguiente figura muestra el sistema que se propone, para alcanzar lo antes expuesto.

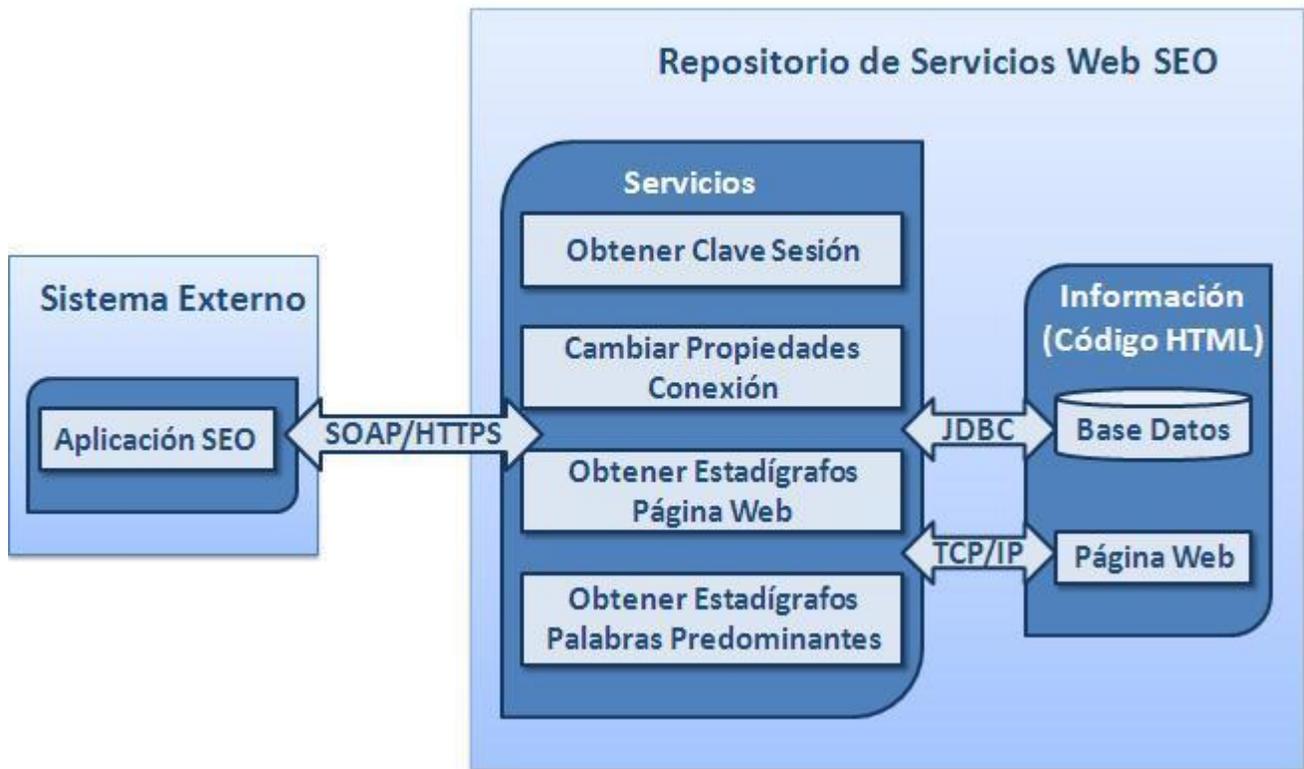


Fig. Propuesta de Sistema.

El *sistema externo* representa la aplicación SEO que consumirá los servicios que se encontrarán en el *repositorio de servicios web SEO*. En él se encuentran las siguientes funcionalidades:

- ✓ Obtener Clave Sesión.

Permitirá la autenticación del usuario con sus parámetros de conexión: usuario y contraseña, además del puerto y el proxy, en el caso que no sea una conexión directa, creándose una sesión correspondiente al mismo, devolviendo la clave que lo identifica, para que posteriormente pueda consumir los demás servicios que se brindan pasando solamente esta clave y la URL de la página. Con esto se evita tener que enviar este tipo de datos sensibles con frecuencia por el canal de comunicación. Esta clave va a ser única y contará de 28 caracteres, la misma se generará automáticamente y tendrá una vigencia igual al tiempo que el usuario esté usando el servicio web. Las sesiones van a contar con un tiempo de expiración,

marcado por la inactividad del usuario. El sistema periódicamente chequea las sesiones y cierra aquellas cuyo tiempo inactivo sea igual al de expiración.

- ✓ Cambiar Propiedades Conexión.

Permitirá cambiar las propiedades de conexión de una sesión creada previamente, dígame usuario, contraseña, puerto y proxy.

- ✓ Obtener Estadígrafos Página Web.

Representa una serie de funcionalidades como obtener cantidad de tablas, imágenes, enlaces, enlaces rotos, título, descripción, keywords, si posee mapa de navegación, correos de contacto, porcentaje de código embebido, texto en etiquetas relevantes (negrita, cursivo, subrayado y jerarquía de contenidos), lenguaje, peso entre otros estadígrafos de la página web.

- ✓ Obtener Estadígrafos Palabras Predominantes

Representa las funcionalidades referentes a las diez palabras que más se repiten en el texto de la página, dígame su peso, relevancia, densidad, frecuencia, si esta incluida en las keywords, en el título o en la URL, etc.

Todos estos estadígrafos antes mencionados, son algunos de los que se deben tener en cuenta para la adecuada optimización de las páginas web y lograr que estas sean mejor vistas por las arañas de los buscadores a la hora de indexarlas en las bases de datos de estos y por tanto tener una mejor recuperación de las mismas logrando que aparezcan dentro de los primeros resultados de búsqueda.

Para poder conocer los estadígrafos que se desean una vez pasada la URL de la página que se quiere, es necesario hacer una conexión a internet y descargar el código de esta. Con el objetivo de disminuir el número de conexiones a internet y tener un fácil acceso a la información se decidió tener una pequeña base de datos que guardará de forma persistente la información de las páginas web (URL, código HTML, fecha de última modificación), una vez que fueran descargadas.

Si se desea obtener algún estadígrafo dado una URL, se descargará la página web correspondiente por primera vez y se almacenará en la base de datos, para que en una segunda oportunidad que se quiera

hacer una petición sobre la misma URL, no sea necesario conectarse y descargar nuevamente la página. Solo habría que verificar en la base de datos la fecha de última modificación de esta, por si es necesario actualizar la información correspondiente, sino no lo es, se trabajaría con la existente.

Debido a que el envío de datos en los servicios web se hace mediante el protocolo de transporte SOAP sobre HTTP, (del inglés: HyperText Transfer Protocol), los mensajes son transmitidos en texto plano por la red, atentando contra la confidencialidad de la información. Por ello surgió la idea de utilizar HTTPS (del inglés: HyperText Transfer Protocol Secure), es la versión segura del protocolo HTTP y viene dada por el protocolo de seguridad SSL (del inglés: Secure Socket Layer) para poder asegurar la transferencia de la información.

2.3 Modelo de Dominio.

Debido a la falta de un proceso de negocio bien definido se decidió desarrollar un modelo de dominio, que contribuirá a la comprensión del sistema y permitirá a los usuarios, desarrolladores y clientes, tener un vocabulario común, lo cual facilitará una mejor comunicación entre los mismos.

A continuación se exponen los conceptos fundamentales para un mayor entendimiento del dominio del problema.

Aplicación SEO: Sistema externo que va a utilizar los servicios web que se brindan.

Servicio: Funcionalidad que es ofrecida para ser utilizada por las aplicaciones SEO.

Proveedor de Servicios: Sistema que tiene publicado diferentes servicios ha ser usados por distintas aplicaciones.

Página Web: Documento situado en la Web que contiene la información necesaria para la implementación de los servicios.

Base de Datos: Aplicación informática que maneja la información necesaria de la página web, de forma organizada y persistente.

Código HTML: Lenguaje de hipertexto del cual se extrae la información necesaria para la implementación de los servicios.

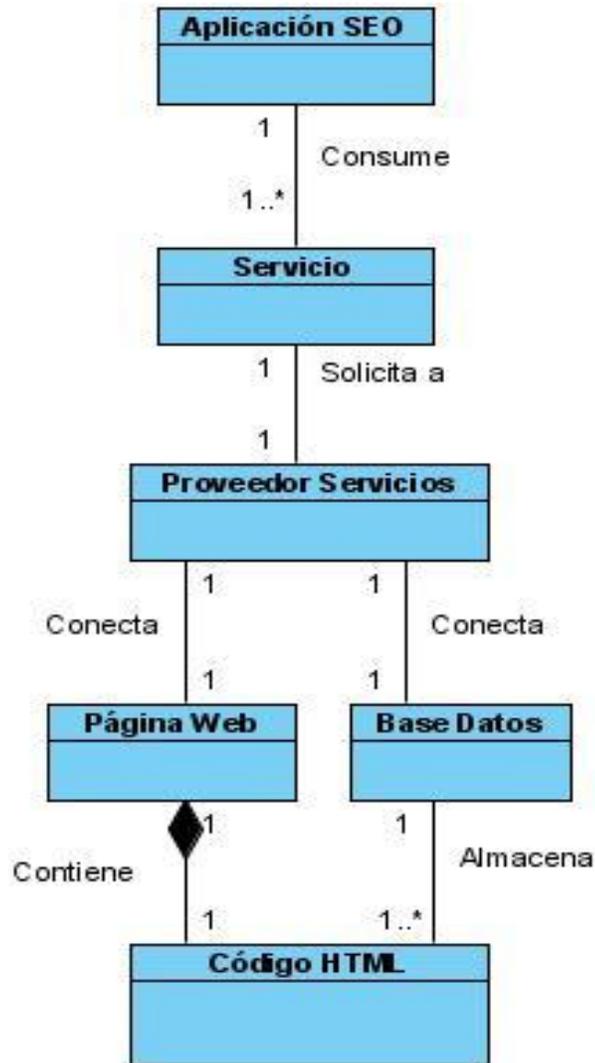


Fig. 3 Modelo de Dominio.

2.4 Relación de los Requerimientos del sistema.

La IEEE Standard Glossary of Software Engineering Terminology define requerimiento como *condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Así como la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.* [14]

A continuación se presentan los requerimientos que debe cumplir el sistema que se propone. Estos se han clasificado en funcionales y no funcionales.

2.4.1 Requerimientos funcionales.

RF1. Mostrar si posee mapa de navegación.

RF2. Mostrar cantidad de tablas de cada página.

RF3. Mostrar cantidad de imágenes de cada página.

RF4. Mostrar propiedad Alt de las imágenes.

RF5. Mostrar enlaces que posee cada página.

RF6. Mostrar enlaces rotos de cada página.

RF7. Mostrar porcentaje de código embebido que posee cada página.

RF8. Mostrar correos de contacto.

RF9. Mostrar título de la página.

RF10. Mostrar longitud del título de la página.

RF11. Mostrar keywords de la página.

RF12. Mostrar si posee keywords en la URL de la página.

RF13. Mostrar si posee keywords en el título.

RF14. Mostrar peso de la página..

RF15. Mostrar texto en negrita.

RF16. Mostrar texto en cursiva

RF17. Mostrar texto en subrayado.

RF18. Mostrar jerarquía de contenidos.

RF19. Mostrar descripción de la página

RF20. Mostrar lenguaje de la página.

RF21. Mostrar palabras que más predominan.

RF21.1 *Mostrar peso.*

RF21.2 *Mostrar relevancia.*

RF21.3 *Mostrar frecuencia.*

RF21.4 *Mostrar densidad.*

RF21.5 *Mostrar si está posicionada en el título.*

RF21.6 *Mostrar si están en la URL.*

RF21.7 *Mostrar si son keywords.*

RF22. Obtener clave de sesión.

RF23. Cambiar propiedades de la conexión.

2.4.2 Requerimientos no funcionales

Rendimiento: Por las características del sistema, se ha definido que el tiempo límite de demora que un consumidor tiene para solicitar un servicio antes de que expire su sesión es de 15 minutos, pasado ese tiempo tendría que volver a obtener una nueva sesión.

Seguridad: Para mantener una mayor confiabilidad se necesita conocer la identidad del usuario (clave de sesión) que consume los servicios, cada vez que realice una petición. Se le debe incorporar la opción de poder usar certificados digitales, permitiendo enviar los datos vía HTTPS. Además se debe garantizar la disponibilidad de los servicios en cualquier momento para los desarrolladores.

Hardware: Mínimo de Memoria RAM: se recomienda 512 MB.

Software: Se requiere como servidor web Apache Tomcat, para el despliegue del Servicio Web, así como la maquina virtual de java JRE1.6.0 y el JDK1.6.0.

Portabilidad: Independientemente de la plataforma de desarrollo en que fueron implementados los servicios, las aplicaciones que lo consuman pueden hacerlo sin tener en cuenta esta, desde cualquier plataforma o lenguaje de programación.

Soporte: Se requiere una documentación apropiada que describa todas las funcionalidades del sistema desarrollado así como una guía para su uso.

2.5 Modelo de Casos de Uso del Sistema.

En esta sección se identifican los actores y casos de uso del sistema, quedando determinado el Diagrama de Casos de Uso del Sistema. Se describen además cada uno de estos casos de uso.

2.5.1 Definición de los actores del sistema.

Los actores del sistema intercambian información con él, aunque no forman parte de él. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. A continuación se muestra el actor y la justificación que tiene en el sistema.

Tabla 1 Definición del actor del sistema.

Actores del Sistema	Sistema Externo
Justificación	Es la aplicación o sistema que se conecta para consumir los servicios web.

2.5.2 Definición de los Casos de Uso del sistema.

Los casos de uso son fragmentos de funcionalidades que agrupan los requisitos que debe cumplir el sistema. El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario. En el sistema que se modela se han definido los siguientes casos de uso:

Tabla 2 Definición de los Casos de Uso del Sistema.

CU_1	Obtener Clave Sesión
Actor	Sistema Externo
Descripción	Es el caso de uso donde se obtiene la clave de sesión, la cual es requerida para consumir las demás funcionalidades que brinda el servicio web.
Referencia	RF22

CU_2	Cambiar Propiedades Conexión
Actor	Sistema Externo
Descripción	Es el caso de uso que da la posibilidad de poder modificar una vez que haya obtenido la clave de sesión, los valores de las propiedades de la conexión, dígame: usuario, contraseña, proxy y puerto.
Referencia	RF23

CU_3	Obtener Estadígrafos Página Web
Actor	Sistema Externo
Descripción	Es el caso de uso que agrupa los servicios que brindan los estadígrafos de una página web dada, entiéndase por ejemplo: cantidad de imágenes, tablas, etiquetas relevantes, enlaces, si posee mapa de navegación o no, etc.
Referencia	RF1, RF2, RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20.

CU_4	Obtener Estadígrafos Palabras Predominantes
Actor	Sistema Externo
Descripción	Es el caso de uso que agrupa los servicios que brindan los estadígrafos de las palabras predominantes de una página web dada, entiéndase por ejemplo: relevancia, peso, frecuencia y si esta forma parte de las keywords o en de la URL.
Referencia	RF21, RF21.1, RF21.2, RF21.3, RF21.4, RF21.5, RF21.6, RF21.7.

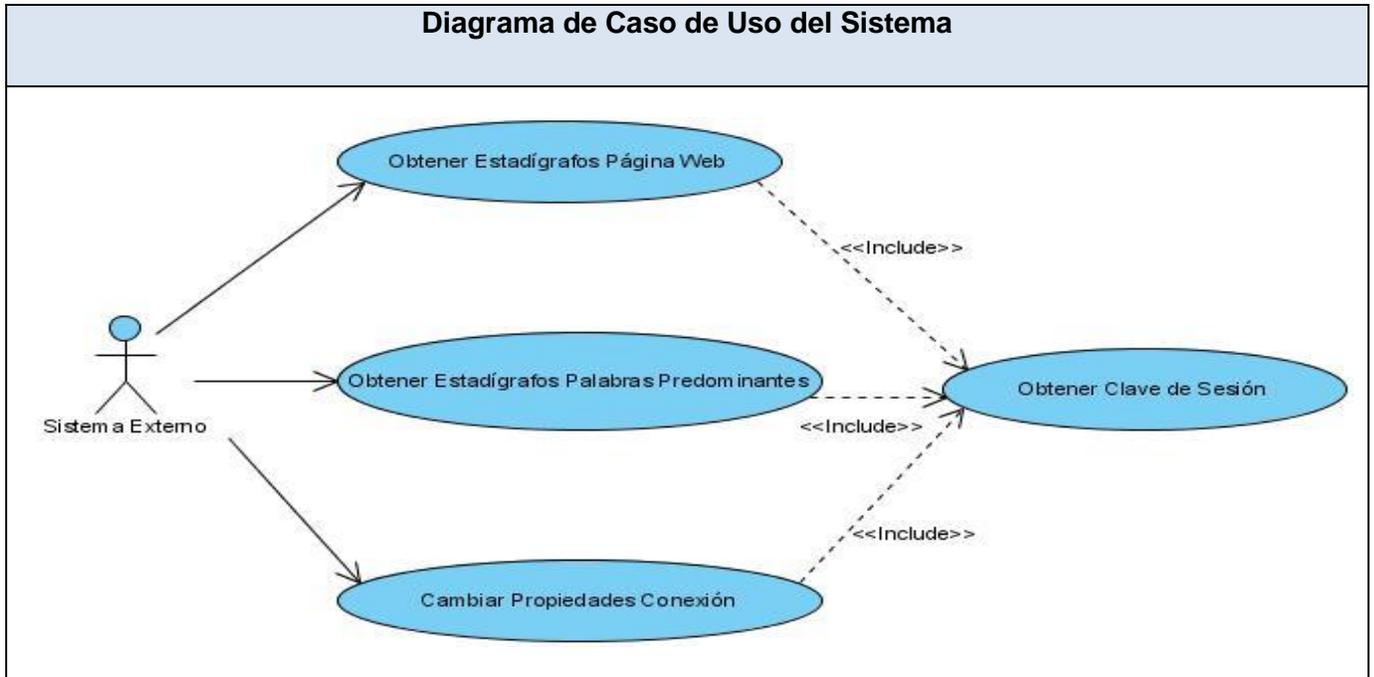


Fig. 4 Diagrama de Caso de Uso del Sistema.

2.5.3 Descripción de los Casos de Uso del sistema.

Tabla 3 Descripción del Caso de Uso Obtener Clave Sesión.

Caso de Uso	Obtener Clave Sesión
Actor	Sistema Externo
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir el servicio para obtener la clave de sesión y poder ejecutar los demás servicios que se brindan. Para ello se introducen los datos necesarios para el procesamiento del servicio a través de la aplicación, el servicio solicitado es ejecutado y se brinda la respuesta correspondiente.

Precondiciones	
Referencias	RF22
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El sistema externo realiza la petición del servicio para obtener la clave de sesión, para ello introduce los datos necesarios (usuario, contraseña, proxy, puerto).	2. Verifica validez de los datos. 3. Crea sesión y devuelve la clave de la misma.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	2. Si los datos no son correctos se envía mensaje de error: "Ha ocurrido un error. Compruebe que tiene conexión a internet y que los datos entrados son correctos"
Poscondiciones	Se obtiene clave de sesión.

Tabla 4 Descripción del Caso de Uso Cambiar Propiedades Conexión.

Caso de Uso	Cambiar Propiedades Conexión	
Actor	Sistema Externo	
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir el servicio para poder cambiar las propiedades de su conexión, dígame: usuario, contraseña, proxy y puerto. Para ello se introducen los parámetros que desea modificar y la clave de sesión que identifica los mismos. El servicio solicitado es ejecutado y por último, el sistema actualiza las propiedades de conexión correspondientes.	
Precondiciones	El sistema externo debe haber obtenido una clave de sesión.	
Referencias	RF23	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	

<p>1. El sistema externo realiza la petición del servicio para cambiar las propiedades de conexión, para ello introduce los datos necesarios (usuario, contraseña, proxy, puerto, clave de sesión).</p>	<p>2. Verifica validez de los datos. 3. Modifica las propiedades de conexión que se desean cambiar.</p>
<p>Flujo Alterno</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>2. Si los datos no son correctos el sistema envía mensaje de error: "Parámetros incorrectos"</p>
<p>Poscondiciones</p>	<p>Las propiedades de conexión son modificadas.</p>

Tabla 5 Descripción del Caso de Uso Obtener Estadígrafos Página Web.

<p>Caso de Uso</p>	<p>Obtener Estadígrafos Página Web</p>
<p>Actor</p>	<p>Sistema Externo</p>
<p>Resumen</p>	<p>El caso de uso se inicia cuando un sistema externo desea consumir algún servicio para obtener algunos de los estadígrafos de una página web. Para ello se introducen los datos necesarios para el procesamiento de los servicios a través de la aplicación, los servicios solicitados son ejecutados y se brinda una respuesta en correspondencia con el mismo.</p>

Precondiciones	El sistema externo debe haber obtenido una clave de sesión.	
Referencias	RF1, RF2, RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20.	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El sistema externo realiza la petición de uno de los siguientes servicios, pasando la clave de sesión y la URL de la página:</p> <ul style="list-style-type: none"> a. <i>Mostrar si posee mapa de navegación.</i> b. <i>Mostrar cantidad de tablas de cada página.</i> c. <i>Mostrar cantidad de imágenes de cada página.</i> d. <i>Mostrar propiedad Alt de las imágenes.</i> e. <i>Mostrar enlaces que posee cada página.</i> f. <i>Mostrar enlaces rotos de cada página.</i> g. <i>Mostrar porcentaje de código embebido que posee cada página.</i> 	<p>2. Verifica validez de los datos.</p> <p>3. Ejecuta la petición y envía respuesta.</p>	

<p><i>h. Mostrar correo de contacto.</i></p> <p><i>i. Mostrar título de la página.</i></p> <p><i>j. Mostrar longitud del título de la página..</i></p> <p><i>k. Mostrar keywords de la página.</i></p> <p><i>l. Mostrar si posee keywords en la URL de la página.</i></p> <p><i>m. Mostrar si posee keywords en el título de la página.</i></p> <p><i>n. Mostrar peso de la página.</i></p> <p><i>o. Mostrar lenguaje de la página.</i></p> <p><i>p. Mostrar descripción de la página.</i></p> <p><i>q. Mostrar texto en negrita.</i></p> <p><i>r. Mostrar texto en cursiva.</i></p> <p><i>s. Mostrar texto en subrayado.</i></p> <p><i>t. Mostrar texto en jerarquía de contenidos.</i></p>	
<p>Flujo Alterno</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>

	2. Si los datos entrados no son correctos se envía mensaje de error: "Parámetros incorrectos"
Poscondiciones	Se obtiene respuesta del sistema.

Tabla 6 Descripción del caso de Uso Obtener Estadígrafos Palabras Predominantes.

Caso de Uso	Obtener Estadígrafos Palabras Predominantes.
Actor	Sistema Externo
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir algún servicio para obtener algunos de los estadígrafos de las palabras que más predominan en una página web. Para ello se introducen los datos necesarios para el procesamiento de los servicios a través de la aplicación, los servicios solicitados son ejecutados y se brinda una respuesta en correspondencia con el mismo.
Precondiciones	El sistema externo debe haber obtenido una clave de sesión.
Referencias	RF21, RF21.1, RF21.2, RF21.3, RF21.4, RF21.5, RF21.6, RF21.7.
Prioridad	Crítico
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
<p>1. El sistema externo realiza la petición de uno de los siguientes servicios, pasando la clave de sesión y la URL de la página:</p> <ul style="list-style-type: none"> a. <i>Mostrar palabras que más predominan.</i> b. <i>Mostrar frecuencia</i> c. <i>Mostrar peso.</i> d. <i>Mostrar relevancia.</i> e. <i>Mostrar densidad.</i> f. <i>Mostrar si esta posicionada en el título.</i> g. <i>Mostrar si están en la URL.</i> h. <i>Mostrar si son keywords.</i> 	<p>2. Verifica validez de los datos.</p> <p>3. Ejecuta la petición y envía respuesta.</p>
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	<p>2. Si los datos no son correctos se envía mensaje de error: "Parámetros incorrectos".</p>
Poscondiciones	Se obtiene respuesta del sistema.

2.6 Conclusiones

En este capítulo se realiza la propuesta del sistema a desarrollar. Se estructura el Modelo de Dominio, describiendo los conceptos fundamentales que en él se exponen, pues no existen procesos de negocio bien definidos. Se listan los requerimientos funcionales y no funcionales que debe cumplir el software para satisfacer las necesidades del cliente. Además se definen los actores y casos de uso del sistema, así como la descripción textual de cada uno de ellos, lo cual constituye de gran importancia para las próximas iteraciones.

Capítulo 3

Análisis y Diseño del sistema

3.1 Introducción

Una vez definidos los requerimientos, se está en condiciones de traducirlos a una especificación que describa cómo implementar el sistema. Precisamente lograr esto mediante la modelación del análisis y el diseño del sistema, es el objetivo del presente capítulo. Además de ofrecer una descripción de los patrones de diseño empleados, algunos aspectos referentes a la seguridad del sistema y el manejo de los posibles errores que puedan ocurrir.

3.2 Modelo de Análisis.

El Modelo de Análisis proporciona una vista interna del sistema. Contiene clases del análisis y sus objetos organizados en paquetes que colaboran.

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que solo se interesa por los requisitos funcionales [15]. El Modelo de Análisis sirve como una primera aproximación del diseño que se realizará posteriormente.

3.2.1 Diagrama de Clases del Análisis.

En un diagrama de Clases del Análisis se muestran las clases que están asociadas con el dominio del problema y las relaciones entre ellas. A continuación se muestran los diagramas de clases de análisis por cada caso de uso.

Diagrama de Clases del Análisis del CUS Obtener Clave Sesión y el CUS Cambiar Propiedades Conexión.

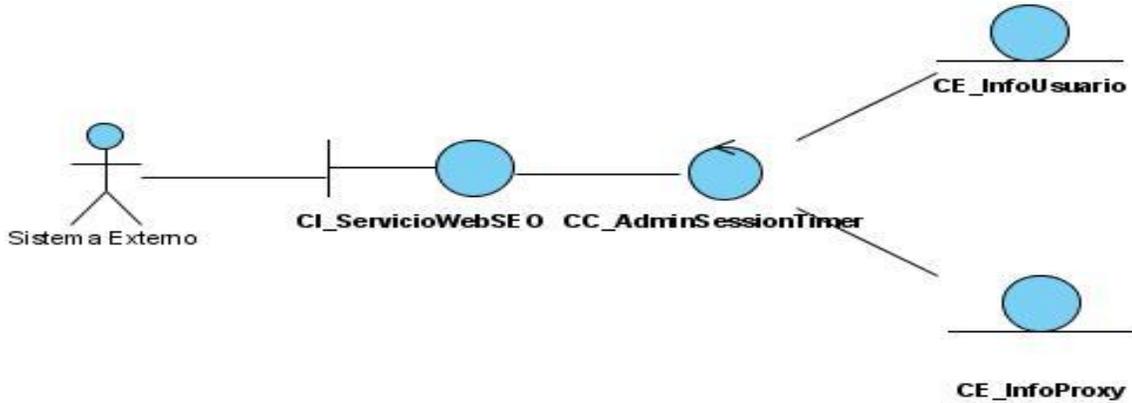


Fig. 5 Diagrama de Clases del Análisis del CUS Obtener Clave Sesión.

Diagrama de Clases del Análisis del CUS Obtener Estadígrafos Página Web.

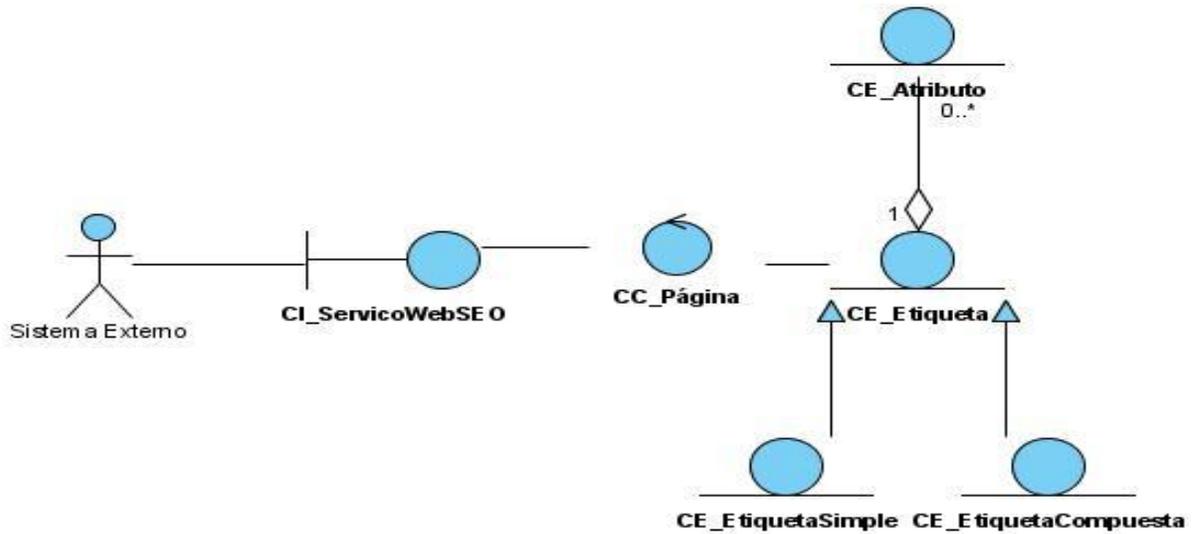


Fig. 6 Diagrama de Clases del Análisis del CUS Obtener Estadígrafos Página Web.

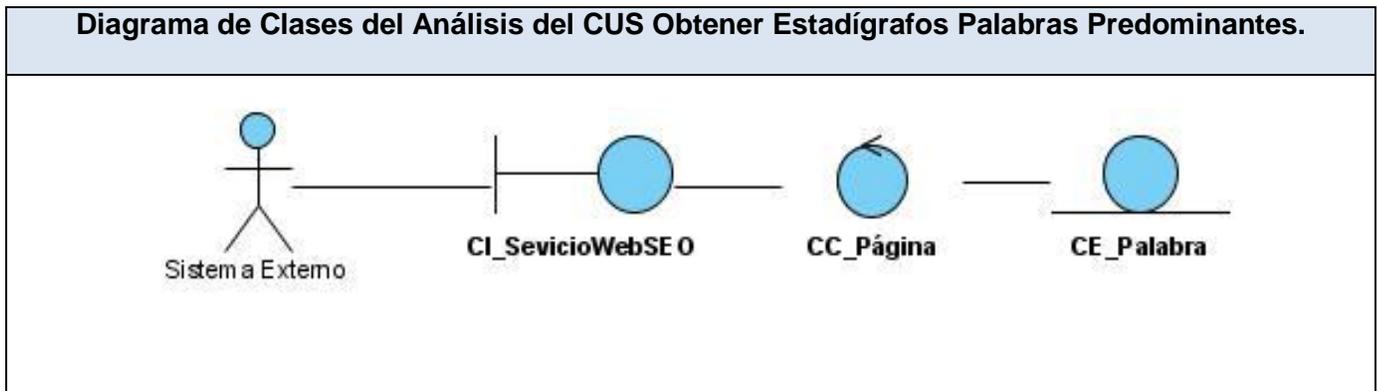


Fig. 7 Diagrama de Clases del Análisis del CUS Obtener Estadígrafos Palabras Predominantes.

3.3 Modelo de Diseño.

El Modelo de Diseño es un modelo de objetos que describe la realización de los casos de uso, y sirve como una abstracción del modelo de implementación. Está compuesto por clases del diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos.

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos [15]. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

3.3.1 Diagrama de clases del Diseño.

Un diagrama de Clases del Diseño es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Las clases de diseño se especifican utilizando la sintaxis del lenguaje de programación elegido. A continuación se muestran los diagramas de clases del diseño para cada uno de los casos de uso.

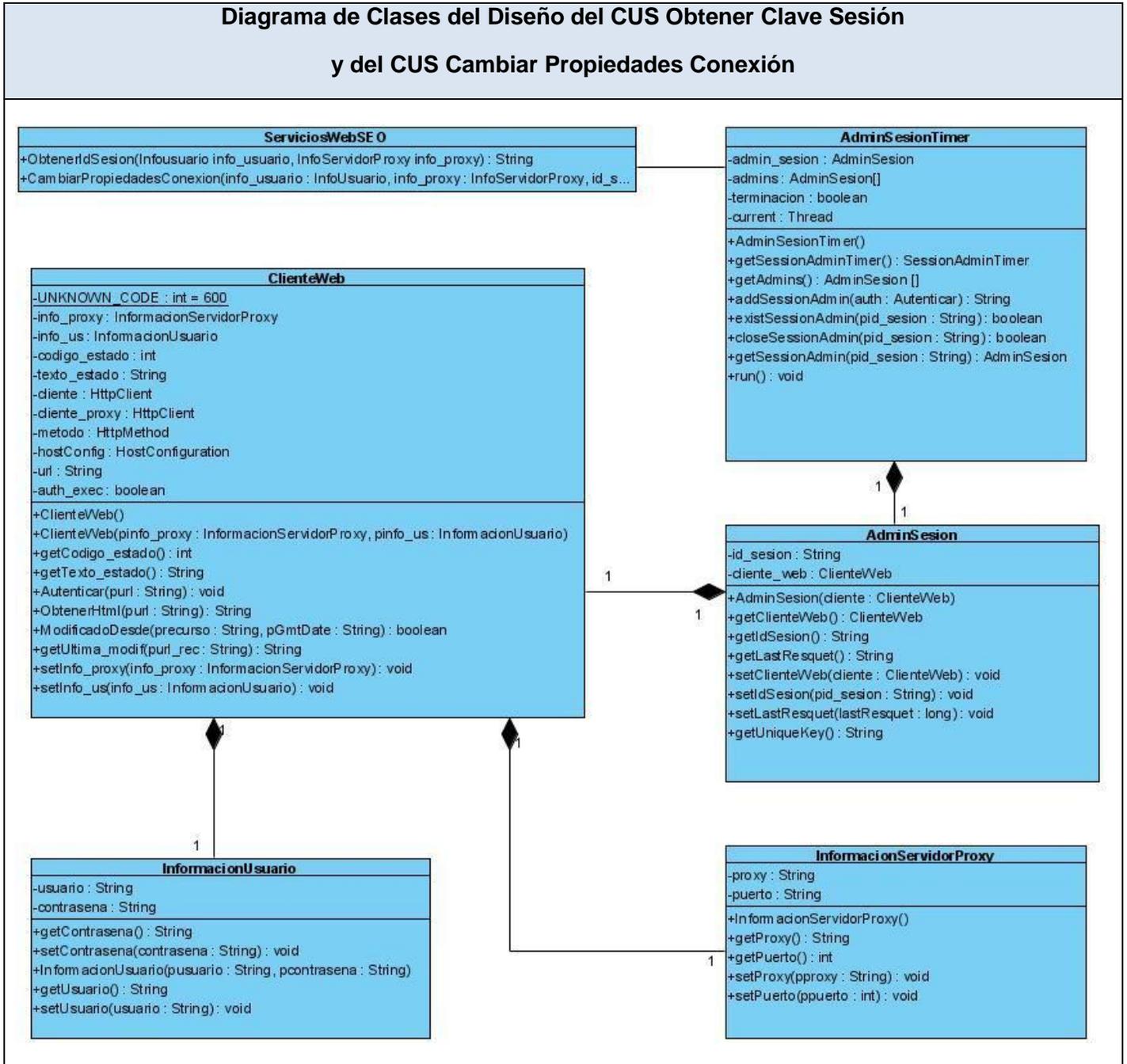


Fig. 8 Diagrama de Clases del Diseño del CUS Obtener Clave Sesión y del CUS Cambiar Propiedades Conexión.

3.3.2 Diagramas de interacción del Diseño. Diagrama de Secuencia.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Están formados por un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. A continuación se muestran los diagramas de secuencia correspondiente a algunos de los escenarios reflejados en cada caso de uso (Ver más en Anexos). Estos a diferencia de los de colaboración, destacan el orden temporal de los mensajes.

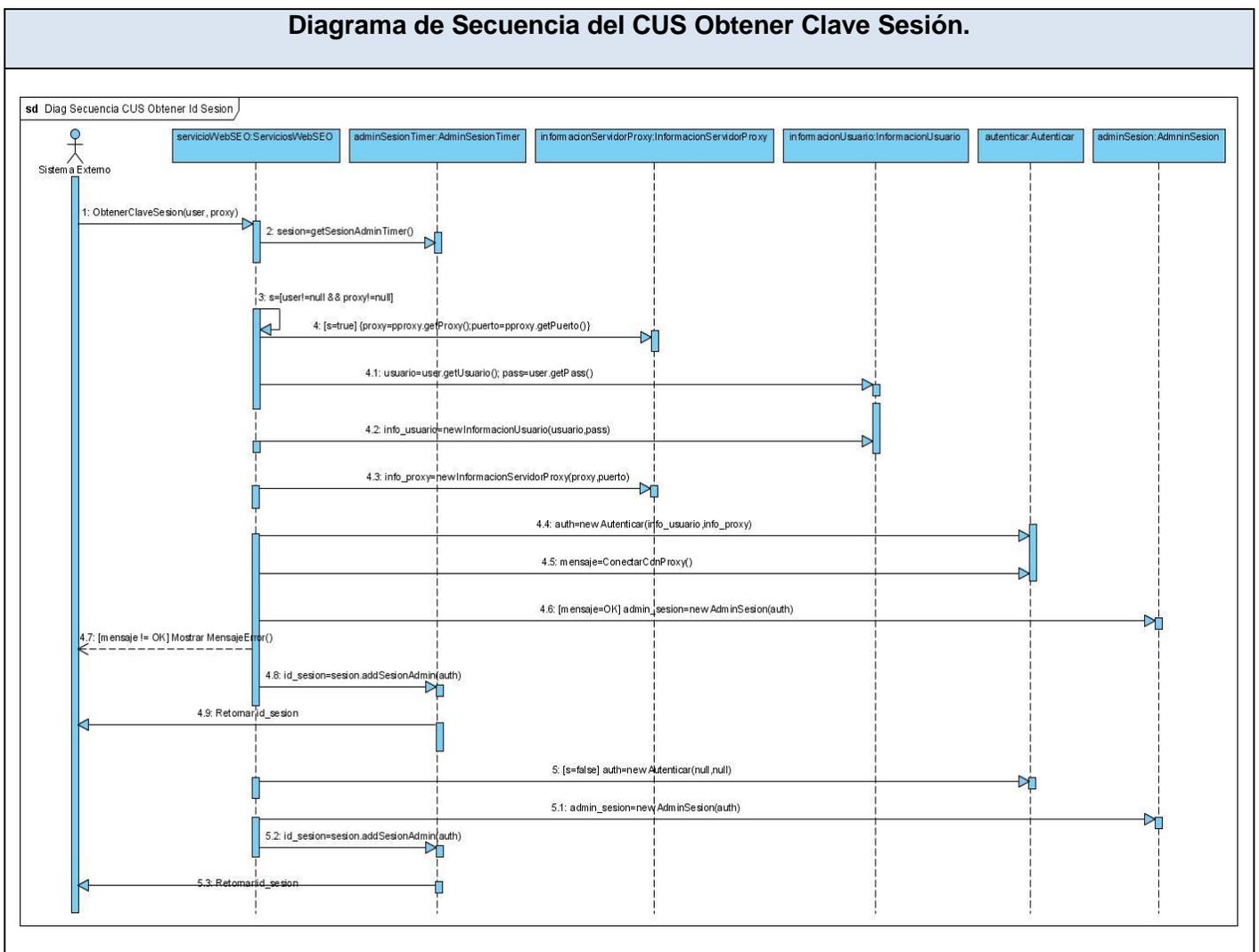


Fig. 11 Diagrama de Secuencia del CUS Obtener Clave Sesión.

Diagrama de Secuencia del CUS Cambiar Propiedades Conexión.

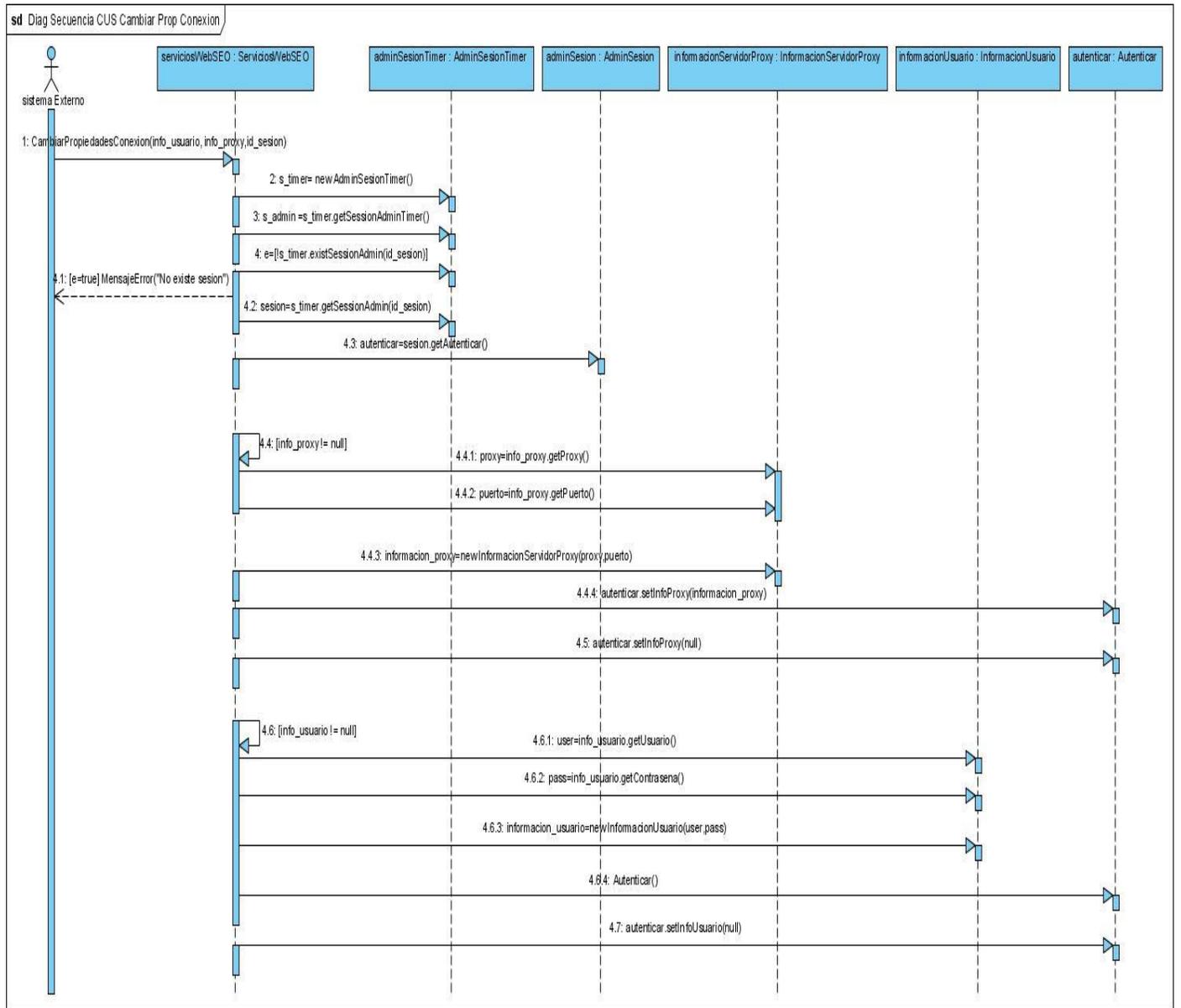


Fig. 12 Diagrama de Secuencia del CUS Cambiar Propiedades Conexión.

3.4 Diseño de la Base de Datos.

Las bases de datos desempeñan un papel crucial en casi todas las áreas de aplicaciones de las computadoras. Su diseño es una de las tareas más importantes en la construcción de un sistema que requiera manejar la persistencia de la información.

El sistema que se propone, hace uso de una base de datos muy sencilla, pues no interesa almacenar gran volumen de información. Solo se necesita guardar de manera persistente, la información obtenida de la página web una vez que se descargue (URL, código HTML y fecha de última modificación). Con esto se garantizaría almacenar la información de la página en forma de caché en la base de datos, para acceder a ella de forma más sencilla, evitando tener que hacer la conexión a internet y descargar la página cada vez que el usuario introduzca una misma URL. De esta manera se consultaría de forma local en la base de datos si la URL existe y si es así se chequearía la fecha de última modificación, por si es necesario actualizarla. En caso de que no se encuentre almacenada, se descargaría por primera vez guardando la información que se necesita para posteriores operaciones.

3.4.1 Modelo lógico de datos. Diagrama de Clases Persistentes.

El modelo lógico de datos representa la información que maneja el sistema. Es una fuente de información para el modelo físico. A continuación se muestra el Modelo lógico de datos que no es más que el diagrama de clases persistentes.

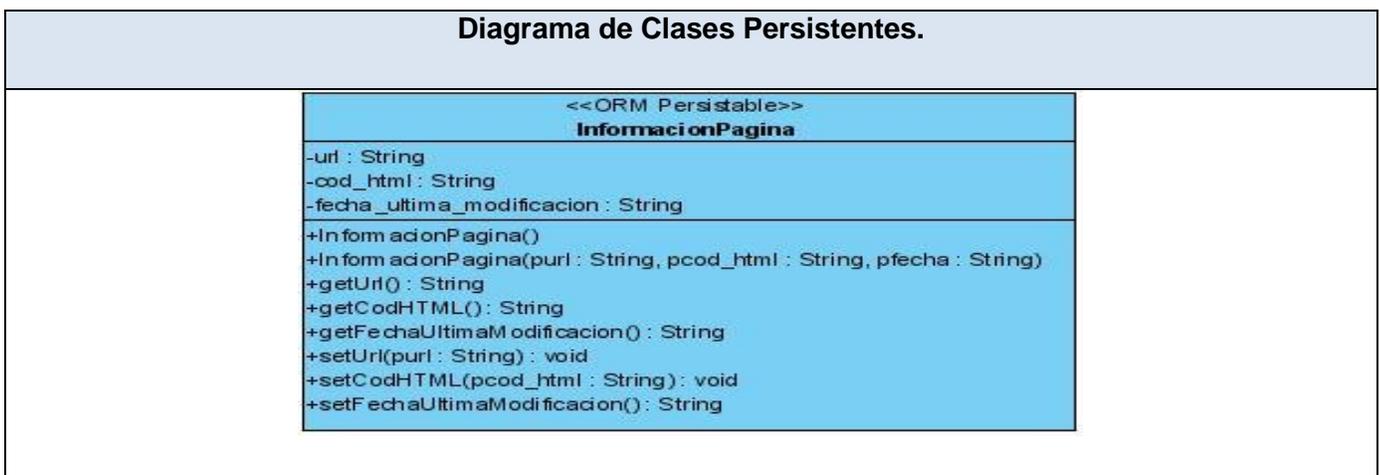


Fig. 15 Diagrama de Clases Persistentes

3.4.2 Modelo físico de datos (Modelo de Datos).

El Modelo físico de datos es creado a partir de las clases persistentes, este proporciona conceptos que describen los detalles de cómo se almacenan los datos en el ordenador. A continuación se muestra el Modelo físico de datos que no es más que el Modelo de Datos del sistema.

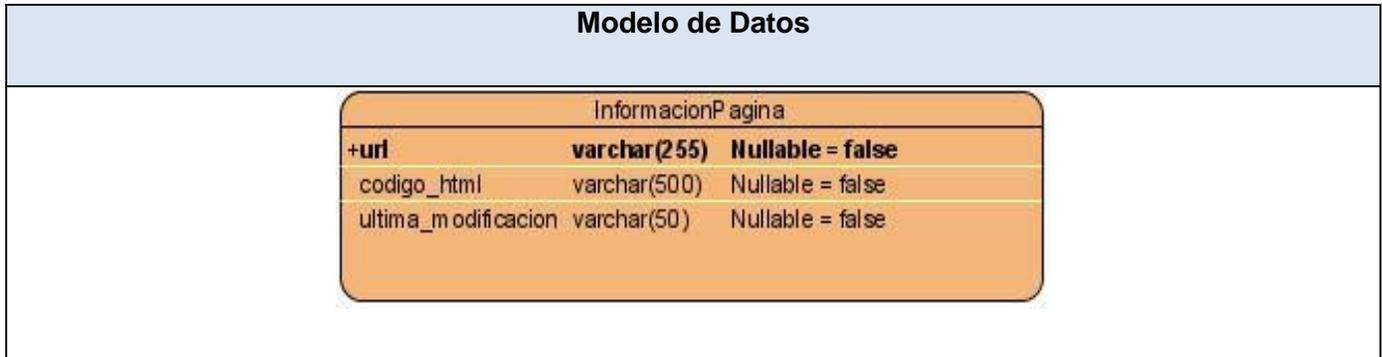


Fig. 16 Modelo de Datos

3.5 Patrones de Diseño empleados durante el desarrollo del sistema.

La habilidad más importante en el análisis y el diseño orientados a objetos es asignar eficientemente las responsabilidades a los componentes del software, por ser una actividad que repercute en la solidez, capacidad de mantenimiento y en la reutilización de los componentes de software.

Los diseñadores expertos en orientación a objetos y otros diseñadores de software, van formando un amplio repertorio de principios generales y de expresiones que los guían al crear software. A algunos se les puede asignar el nombre de patrones, si se codifican en un formato estructurado que describe el problema y su solución y se les asigna un nombre.

En palabras más simples un patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas.

Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería de software. Solo intentan codificar el conocimiento, las expresiones y los principios ya existentes. En consecuencia, no introducen ideas nuevas, son una mera codificación de los principios básicos más usados.

Durante el desarrollo del sistema se usaron los patrones GRASP (del inglés: General Responsibility Assignment Software Patterns) los cuales *describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones* [13]. A continuación se hace una breve descripción de los mismos.

Experto: Este patrón se basa en asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cualquier responsabilidad.

Con su puesta en práctica se logra que se conserve el encapsulamiento, pues los objetos se valen de su propia información para hacer lo que se les pide. Las clases se hacen más fáciles de mantener y comprender. Además soporta un bajo acoplamiento, lo que favorece que el sistema sea mucho más robusto y fácil de comprender.

Creador: Con este patrón se resuelve el problema de saber quién debería ser responsable de crear una nueva instancia de alguna clase, pues se encarga de asignarle la responsabilidad a una clase de crear instancias de otra, ya sea agregando, utilizando, conteniendo, registrando o creando objetos de la otra clase.

Su uso permite un bajo acoplamiento y mejores oportunidades de reutilización.

Alta Cohesión: Con el uso de este patrón, se resuelve el problema de saber cómo mantener la complejidad dentro de límites manejables, pues es el encargado de asignar una responsabilidad de modo que la cohesión siga siendo alta.

Su puesta en práctica permite lograr un diseño claro y fácil de comprender. Además de que aumenta la capacidad de reutilización.

Bajo Acoplamiento: Este patrón resuelve la problemática de saber cómo dar soporte a una dependencia escasa y a un aumento de la reutilización, mediante la asignación de responsabilidades a las clases de manera que la comunicación entre ellas sea la menor posible

Su uso permite, aumentar la reutilización, que el diseño sea fácil de entender, además de que no se afecte el diseño por cambios en otros componentes.

También se hizo uso de los patrones GOF (del inglés: Gang of Four). A continuación se describen los mismos:

Observer: Este patrón es de comportamiento, define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y se actualicen automáticamente todos los objetos que dependen de él.

Su uso permite lograr un acoplamiento mínimo entre sujetos y observadores, pueden pertenecer a diferentes capas de abstracción de un sistema. Además permite tener una capacidad de comunicación mediante difusión, es decir, libertad de añadir y quitar observadores en cualquier momento, lo cual constituye una ventaja.

Fachada: Este patrón es estructural y es el encargado de proporcionar una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar.

Su puesta en práctica permite ocultar a los clientes, los componentes del subsistema, reduciendo así el número de objetos con los que tratan los clientes y haciendo que el subsistema sea más fácil de usar. Además promueve un débil acoplamiento entre el subsistema y sus clientes, lo cual permite modificar los componentes de un subsistema sin que sus clientes se vean afectados. Las fachadas ayudan a estructurar en capas un sistema y las dependencias entre los objetos. Pueden eliminar dependencias complejas o circulares (consecuencia importante cuando el cliente y el subsistema se implementan por separado).

Una vez vistos estos patrones, cabe destacar también que en la implementación de servicios web se emplean además otros patrones propios de este tipo de tecnología. A continuación se describe uno de los usados en la presente investigación.

Procesador de documentos: Este patrón resuelve el problema de saber cómo se crea un contrato bien definido y fácil de utilizar que sea compatible con los principios de diseño orientado a servicios. Para ello se define un esquema XML para representar mensajes de solicitud y respuesta para el servicio. El contrato se define antes del desarrollo del servicio y se puede utilizar posteriormente para generar el

código del servicio. La adopción de un enfoque de "contrato como primer paso" ayuda a asegurar una elevada interoperabilidad del servicio.

Los patrones de diseño posibilitan la reutilización y se utilizan como guía para el desarrollo de software, pues garantizan un producto de calidad. Por estas razones, es recomendado que se apliquen con frecuencia durante el desarrollo de cualquier sistema informático que se este desarrollando.

3.6 Tratamiento de Errores.

Durante el desarrollo del sistema, se han definido una serie de acciones que pueden provocar un mal funcionamiento del mismo, para cada una de ellas se definió un tratamiento específico, para asegurar una respuesta adecuada que garantice cierto grado de estabilidad. Los mensajes de error llegarán a los usuarios en un lenguaje común, con el objetivo de facilitar su comprensión, pero sin relevar información propia del sistema que pueda atentar contra su seguridad. En la siguiente tabla se describen las posibles acciones, que pueden producir errores.

Tabla 7 Descripción de Errores.

Error	Causa	Respuesta del sistema
compruebe _propiedades_conexión	El usuario intenta obtener clave sesión y las propiedades de conexión no son válidas.	Notificar al usuario que las propiedades de conexión introducidas no son válidas.
sesión_no_existe	El usuario intenta consumir un servicio web pasando una clave de sesión no válida.	Notificar al usuario que la clave de sesión pasada no existe.

error_conexion_con_url	El sistema intenta descargar el código html de una url y existe algún problema de conexión a la página solicitada.	Notificar al usuario que no fue válido la conexión a la url especificada por algún tipo de problema.
propiedades_conexión_duplicadas	El usuario intenta obtener una clave de sesión con propiedades de conexión que ya se encuentran en uso por otra sesión.	Notificar al usuario que las propiedades de conexión especificadas se encuentran en uso por otra sesión.

3.7 Aspectos de Seguridad en el sistema.

En aras de garantizar la seguridad del sistema, y pretendiendo lograr la confiabilidad, integridad y disponibilidad de los datos, se han tomado una serie de medidas.

Primeramente se ha trabajado con las sesiones de usuarios. Una sesión es creada, cuando se desea consumir alguna de las funcionalidades del servicio web. Para ello previamente, el usuario debe obtener la clave de sesión a través de la funcionalidad ObtenerIdSesion, donde se autentica el usuario con los parámetros de conexión entrados (usuario, contraseña, puerto y proxy) y es creada la sesión.

Esta clave va a ser única y contará de 28 caracteres combinando números y letras, la misma se generará automáticamente y tendrá una vigencia igual al tiempo que el usuario esté usando el servicio web. Después de esto, solo se comprobará su identidad a través de su clave de sesión, evitando que datos sensibles como el usuario y la contraseña viajen por el canal de comunicación frecuentemente.

Las sesiones van a contar con un tiempo de expiración, marcado por la inactividad del usuario, con el objetivo de prevenir que una aplicación que se haya quedado abierta, no sea utilizada por personas no autorizadas. El sistema periódicamente chequea las sesiones y cierra aquellas cuyo tiempo inactivo sea igual al de expiración.

Por otra parte, se ha tenido en cuenta utilizar HTTPS, es la versión segura del protocolo HTTP y viene dada por el protocolo de seguridad SSL, para ellos se utilizó la opción de certificado autofirmado, y poder asegurar la transferencia de la información, pues el envío de datos en los servicios web se hace mediante SOAP/HTTP, viajando los mensajes en texto plano por la red, atentando contra la confidencialidad de la información.

3.8 Conclusiones.

Este capítulo se centró fundamentalmente en la realización de los diagramas de clases de análisis y diseño del sistema, así como los diagramas de interacción, correspondientes a cada caso de uso. Lo cual constituye una importante entrada para el próximo flujo de trabajo, la implementación. Se hizo además una descripción de los patrones de diseño usados, se llevaron a cabo acciones para manejar los posibles errores que puedan ocurrir, así como la descripción de algunos aspectos que se tuvieron en cuenta para garantizar la seguridad del sistema.

Capítulo 4

Implementación y Prueba

4.1 Introducción

En el presente capítulo se describen los temas referentes a la implementación y prueba del sistema, se hace referencia a las principales tareas que se llevan a cabo en este flujo de trabajo, así como la descripción de los artefactos que se generan en el mismo.

4.2 Modelo de implementación del sistema.

En la implementación se describió cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue, partiendo de los resultados obtenidos en el diseño.

El modelo de implementación esta compuesto por los diagramas de despliegue y componentes, artefactos generados en este flujo de trabajo, al describir los componentes a construir, su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

4.2.1 Diagrama de Despliegue.

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación, que describe la distribución, entrega e instalación de las partes que configuran el sistema físico. Muestra las relaciones físicas de los nodos que participan en la ejecución del sistema, describiendo la arquitectura física del mismo en términos de: procesadores, dispositivos y componentes de software.

A continuación se muestra el diagrama de despliegue realizado de acuerdo a los requerimientos del sistema propuesto con anterioridad.

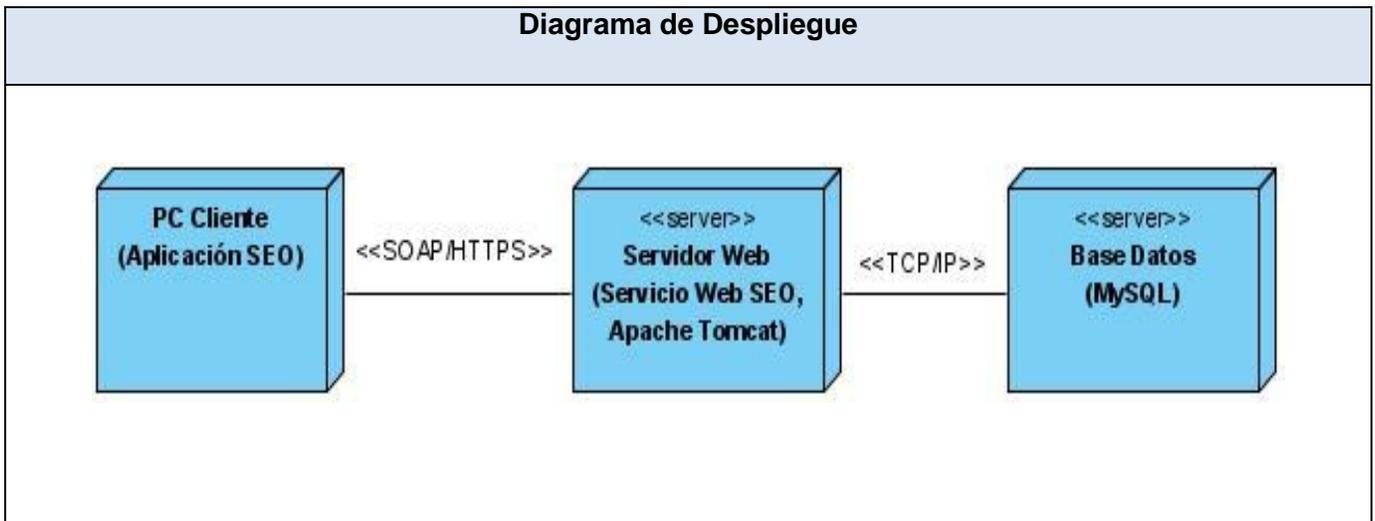


Fig. 17 Diagrama de Despliegue

4.2.2 Diagrama de Componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación especificado [16]. Es otra forma de representar una vista estática del sistema, que representa la organización y dependencia entre los componentes físicos que se necesitan para ejecutar la aplicación, sean estos componentes de código fuente, librerías, binarios o ejecutables.

A continuación se muestran los componentes creados en la solución propuesta, a través del diagrama de componentes.

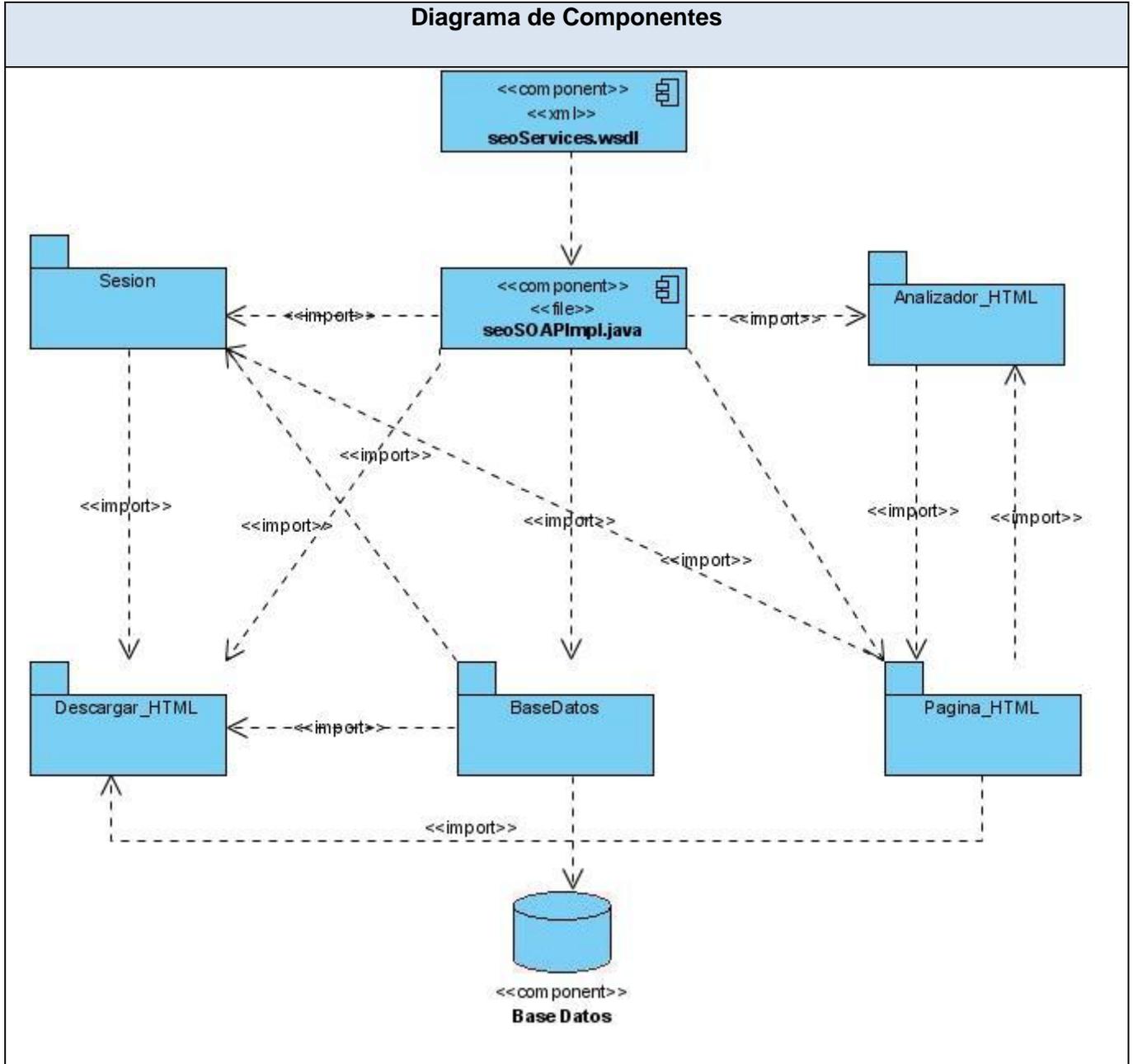


Fig. 18 Diagrama de Componentes General del Sistema.

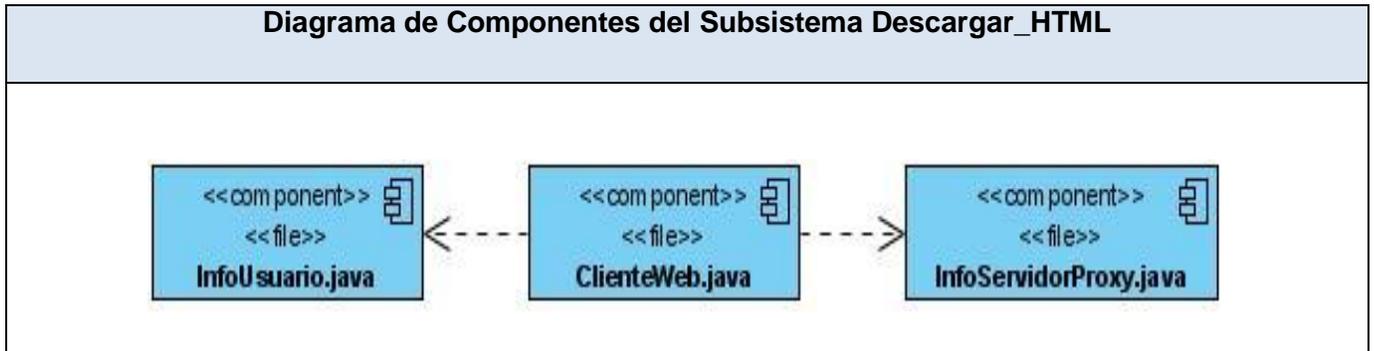


Fig. 19 Diagrama de Componentes del Subsistema Descargar_HTML

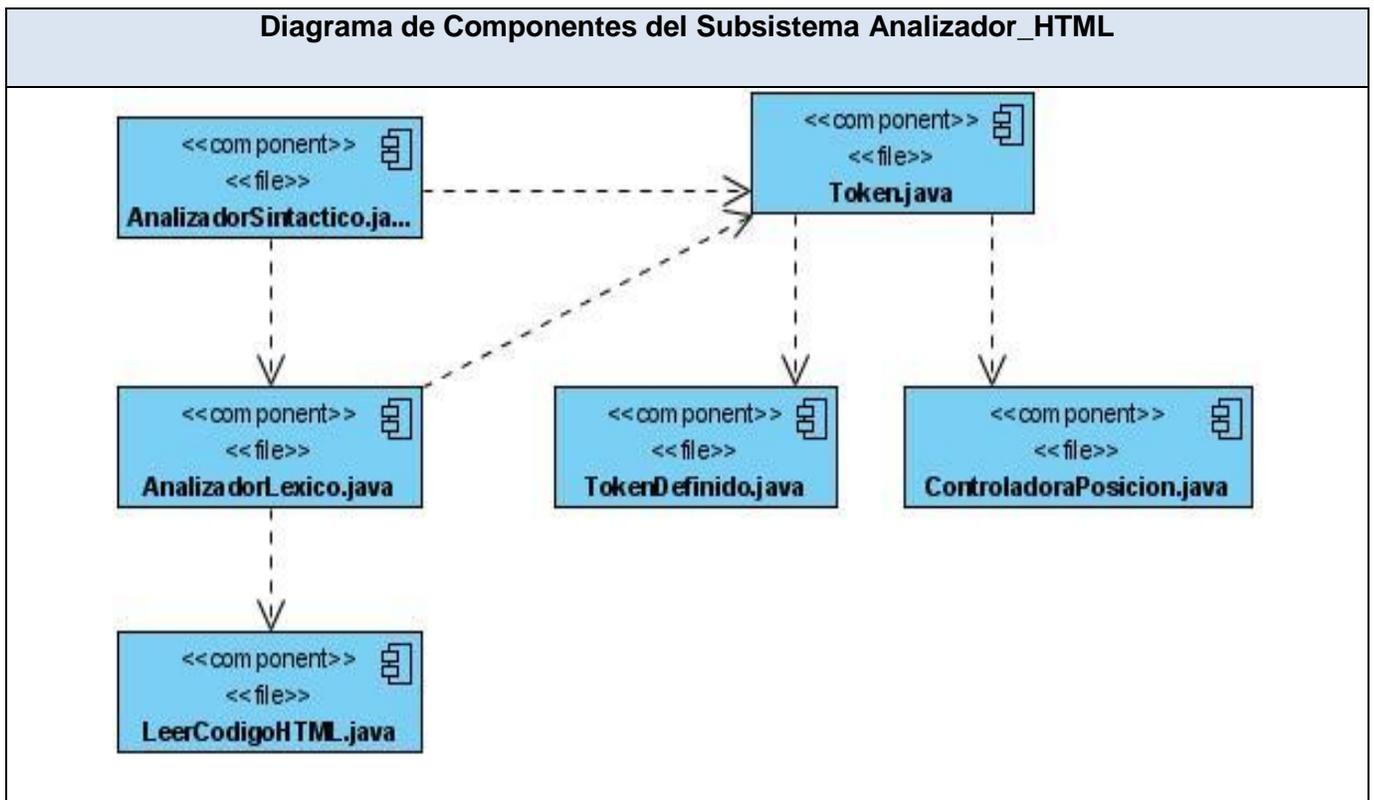


Fig. 20 Diagrama de Componentes del Subsistema Analizador_HTML

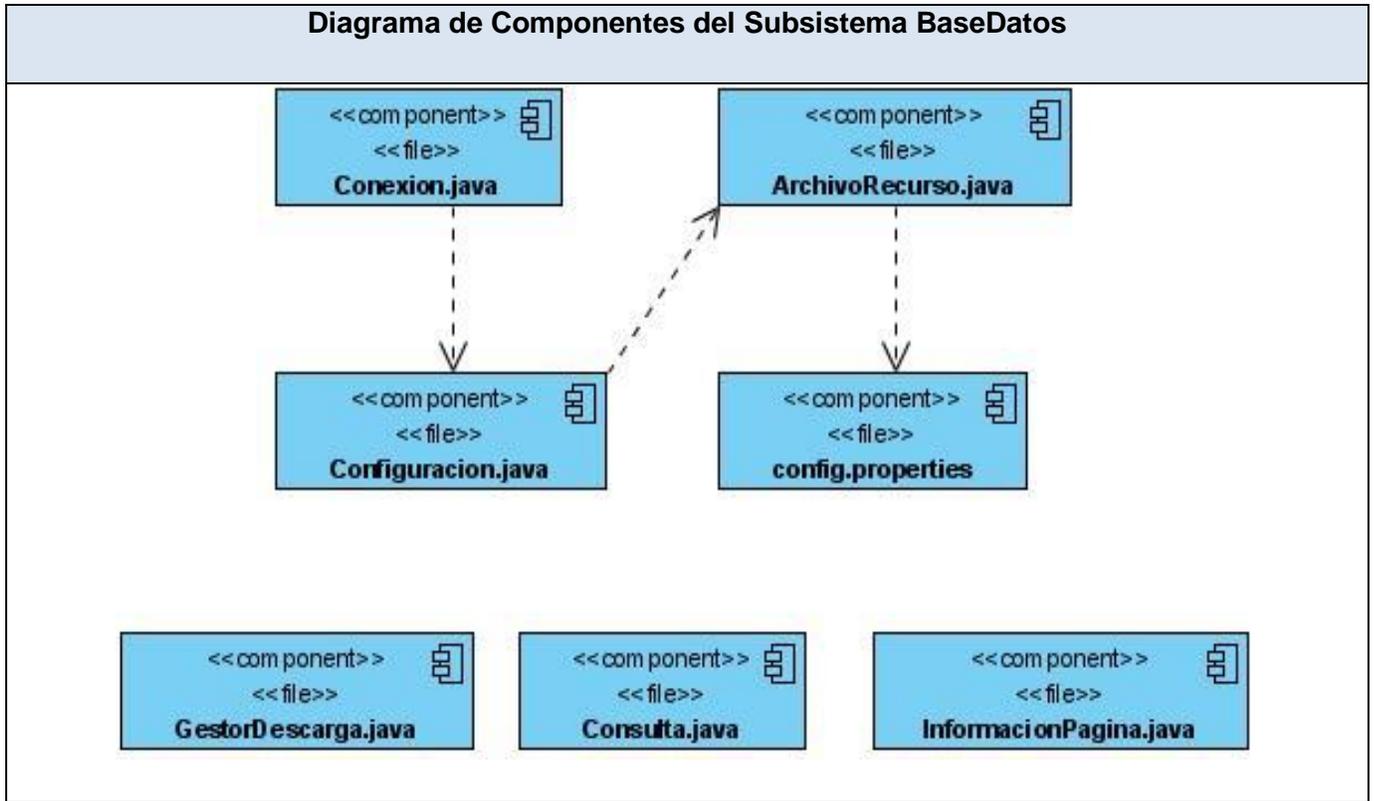


Fig. 21 Diagrama de Componentes del Subsistema BaseDatos

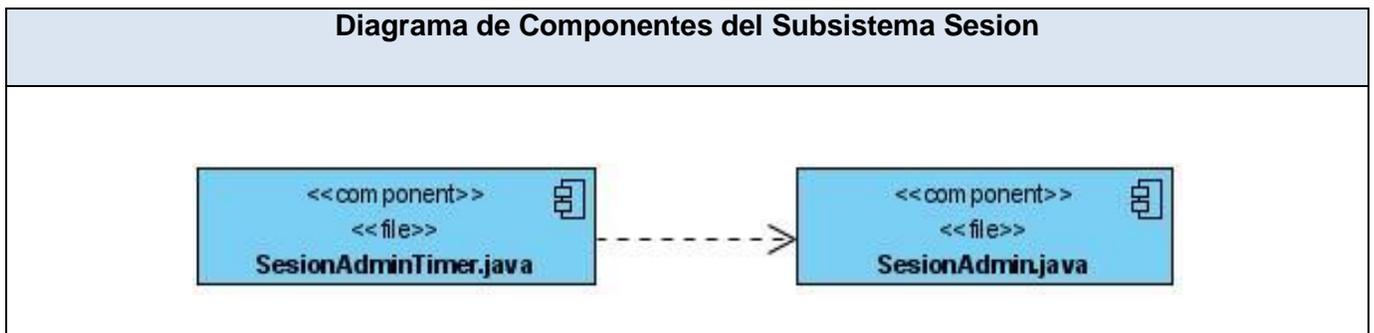


Fig. 22 Diagrama de Componentes del Subsistema Sesión

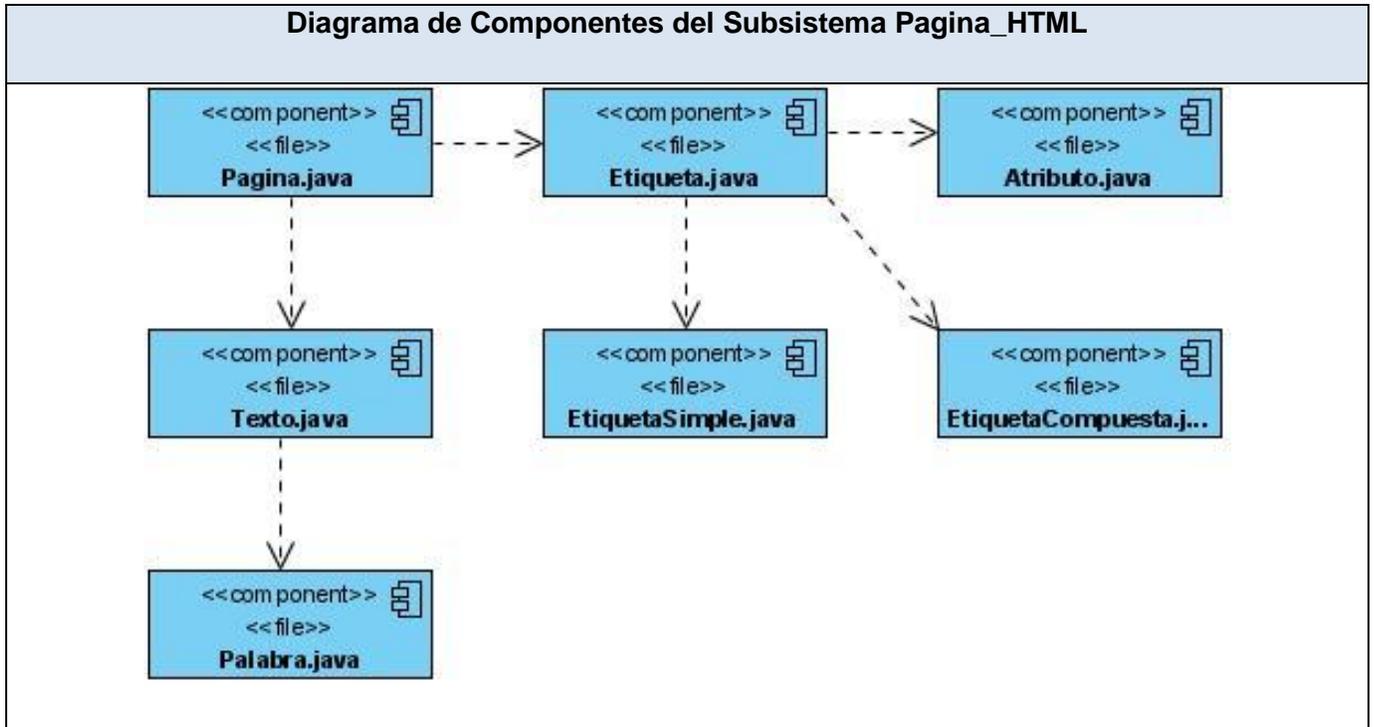


Fig. 23 Diagrama de Componentes del Subsistema Pagina_HTML.

4.3 Modelo de Pruebas.

El desarrollo del software implica en sí, errores que pueden empezar a darse desde el primer momento del proceso en el que los requerimientos pueden estar especificados de forma errónea, así como en los posteriores pasos del diseño e implementación. Es por esto, que se hace necesario contar con una actividad que garantice la calidad.

Las pruebas es la actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. [17]

4.3.1 Métodos de Prueba.

Para la realización de pruebas al sistema se emplean diferentes métodos. Los Métodos de Prueba de Software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. A continuación se presentan, dos de los más conocidos: *caja negra* y *caja blanca*.

El primero se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene [17]. Estas pruebas verifican las especificaciones funcionales y no consideran la estructura interna del programa. Son hechas sin el conocimiento interno del producto y no validan funciones ocultas.

Con el segundo método se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado [17]. Estas requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas de diseño o el código.

Las pruebas desarrolladas en el sistema, han sido con el método de caja blanca. El tipo de prueba usado dentro de este método fue el de Caminos Básicos, este consiste en probar todos los caminos independientes, donde se comprueba la estructura de control con el fin de asegurar que todas las sentencias se ejecuten por lo menos una vez.

En este tipo de pruebas se calcula la complejidad ciclomática del algoritmo a partir de un grafo de flujos, esta constituye la medida cuantitativa que define la complejidad de un programa, y permite conocer el número de caminos independientes que tiene el mismo.

La complejidad ciclomática da un límite superior que define la cantidad de casos de pruebas que se tendrían que hacer para asegurar que cada línea/sentencia se ejecute al menos una vez. A continuación se describen algunos de los casos de prueba realizados al sistema.

4.3.2 Pruebas de Caja Blanca. Caso de Prueba para el método Título.

```

public java.lang.String titulo(java.lang.String idSesion,
    java.lang.String url_pagina) throws java.rmi.RemoteException {

    InformacionPagina info_pagina = null; ①
    String titulo = " ";

    try {
        info_pagina = GestorDescarga.ObtenerHtml(url_pagina, idSesion);
    } catch (RemoteException e1) {
        e1.printStackTrace();
    }

    try {
        Analizador_Lexico lex = new Analizador_Lexico(new Leer_Codigo_Html(
            info_pagina.getCodigo_html()));
        try {
            Analizador_Sintactico sint = new Analizador_Sintactico(lex);
            Pagina pagina = new Pagina(sint, info_pagina.getCodigo_html());

            LinkedList<Etiqueta> list = new LinkedList<Etiqueta>();
            pagina.ObtenerEtiquetasDadoTipo("title", pagina
                .getLista_etiquetas(), list);

            for (int i = 0; i < ((Etiqueta_Compuesta) (list.get(0))) ②
                .getLista_etiqueta().size(); i++) {

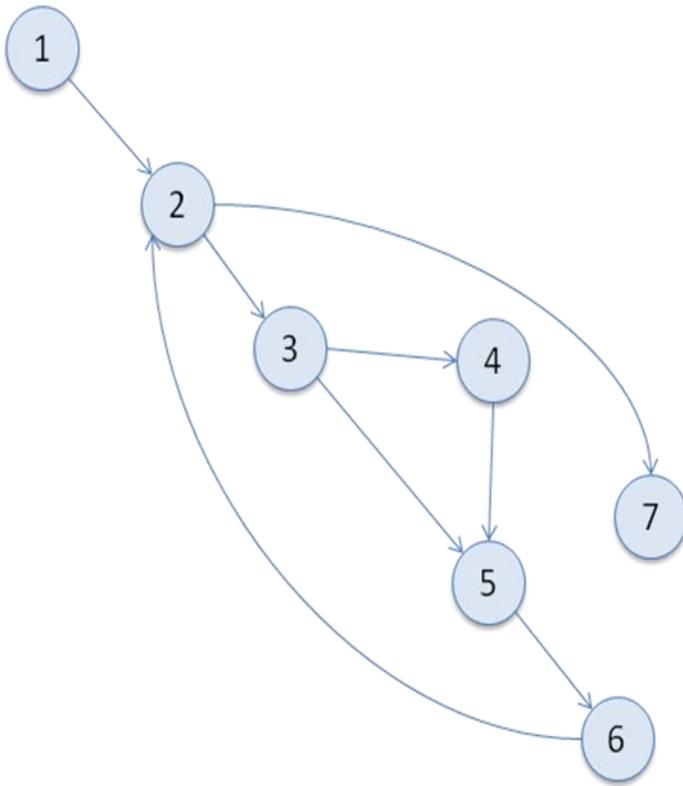
                if (((Etiqueta_Compuesta) (list.get(0))) ③
                    .getLista_etiqueta().get(i).getNombre()
                    .equalsIgnoreCase("")) {

                        titulo = ((Etiqueta_Compuesta) (list.get(0))) ④
                            .getLista_etiqueta().get(i).getTexto(); ④
                    } ⑤
                } ⑥
            } catch (Exception e) {
                e.printStackTrace();
            }

            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        titulo = Texto.PalabraCorrecta(titulo); ⑦
        return titulo;
    }
}

```

Fig. 24 Código correspondiente al método Título.



Complejidad Ciclomática.

$$V(G) = A - N + 2 = 8 - 7 + 2 = 3$$

Caminos Independientes.

1-2-7

1-2-3-5-6-2-7

1-2-3-4-5-6-2-7

Fig. 25 Grafo correspondiente al método Título.

Tabla 8 Caso de Prueba para el Camino 1-2-7

Entrada	Resultados	Condiciones
Se entra la URL de la página y el id_sesión para consumir el servicio que ofrecerá el Título de la página, pero la lista de etiquetas de dicha página se encuentra vacía.	La variable titulo donde se almacena la información a retornar, permanece vacía al igual que la lista de etiquetas.	Los parámetros entrados (URL, id_sesión) sean correctos.

Tabla 9 Caso de Prueba para el Camino 1-2-3-5-6-2-7

Entrada	Resultados	Condiciones
Se entra la URL de la página y el id_sesión para consumir el servicio que ofrecerá el Título de la página, pero el nombre de la etiqueta es diferente de vacío.	La variable titulo donde se almacena la información a retornar, permanece vacía al igual que la lista de etiquetas.	Los parámetros entrados (URL, id_sesión) sean correctos.

Tabla 10 Caso de Prueba para el Camino 1-2-3-4-5-6-2-7

Entrada	Resultados	Condiciones
Se entra la URL de la página y el id_sesión para consumir el servicio que ofrecerá el Título de la página, el nombre de la etiqueta es vacío.	La variable titulo donde se almacena la información a retornar, cambia su valor por el título de la página de la URL dada.	Los parámetros entrados (URL, id_sesión) sean correctos.

4.3.3 Pruebas de Caja Blanca. Caso de Prueba para el método palabrasPredominantesKeywords

```

public java.lang.String[] palabrasPredominantesKeywords(
    java.lang.String idSesion, java.lang.String url_pagina)
    throws java.rmi.RemoteException {

    String keyw[] = keywords(idSesion, url_pagina); ①
    String palabras_pred[] = palabrasPredominantes(idSesion, url_pagina);
    String result_palabras[] = new String[10];
    int d = 0;

    for (int i = 0; i < keyw.length; i++) ②
    {
        if (keyw[i] != null) ③
        {
            for (int j = 0; j < palabras_pred.length; j++) ④
            {
                if (palabras_pred[j] != null) ⑤
                {
                    if (keyw[i].toLowerCase().trim().equalsIgnoreCase( ⑥
                        palabras_pred[j].toLowerCase().trim()))
                    {
                        result_palabras[d] = palabras_pred[j]; ⑦
                        d++;
                    }
                }
            }
        }
    }
    return result_palabras; ⑧
}

```

Fig. 26 Código correspondiente al método PalabrasPredominantesKeywords.

Complejidad Ciclomática.

$$V(G) = A - N + 2 = 12 - 8 + 2 = 6$$

Caminos Independientes.

1-2-8

1-2-3-2-8

1-2-3-4-2-8

1-2-3-4-5-4-2-8

1-2-3-4-5-6-4-2-8

1-2-3-4-5-6-7-4-2-8

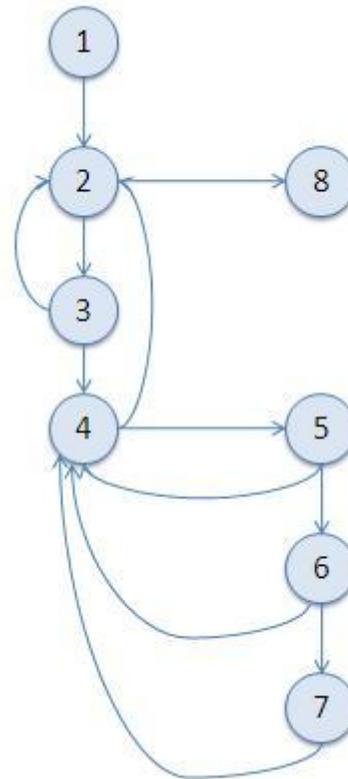


Fig. 27 Grafo correspondiente al método PalabrasPredominantesKeywords.

Tabla 11 Caso de Prueba para Camino 1-2-8

Entrada	Resultados	Condiciones
Se entra la URL de la página y el id_sesión para consumir el servicio que ofrecerá las palabras predominantes que se encuentran en las keywords de la página especificada, pero esta no tiene keywords.	El arreglo result_palabras donde se almacena la información a retornar, permanece vacío.	Los parámetros entrados (URL, id_sesión) sean correctos.

Tabla 12 Caso de Prueba para el Camino 1-2-3-2-8

Entrada	Resultados	Condiciones
Se entra la URL de la página y el id_sesión para consumir el servicio que ofrecerá las palabras predominantes que se encuentran en las keywords de la página especificada, pero el arreglo de las keywords es null en determinada posición.	Se retorna el arreglo result_palabras, con la información almacenada hasta el momento.	Los parámetros entrados (URL, id_sesión) sean correctos.

Tabla 13 Caso de Prueba para el Camino 1-2-3-4-2-8

Entrada	Resultados	Condiciones
Se entra la URL de la página y el id_sesión para consumir el servicio que ofrecerá las palabras predominantes que se encuentran en las keywords de la página especificada, pero esta no tiene palabras predominantes.	Se retorna el arreglo result_palabras, con la información almacenada hasta el momento.	Los parámetros entrados (URL, id_sesión) sean correctos.

Tabla 14 Caso de Prueba para el Camino 1-2-3-4-5-4-2-8

Entrada	Resultados	Condiciones
Se entra la URL de la página y el id_sesión para consumir el servicio que ofrecerá las palabras predominantes que se encuentran en las keywords de	Se retorna el arreglo result_palabras, con la información almacenada hasta el momento.	Los parámetros entrados (URL, id_sesión) sean correctos.

la página especificada, pero el arreglo de palabras predominantes es null en determinada posición.		
--	--	--

Tabla 15 Caso de Prueba para el Camino 1-2-3-4-5-6-4-2-8

Entrada	Resultados	Condiciones
Se entra la URL de la página y el id_sesión para consumir el servicio que ofrecerá las palabras predominantes que se encuentran en las keywords de la página especificada, pero la palabra predominante no es igual a la keyword en esa posición.	Se retorna el arreglo result_palabras, con la información almacenada hasta el momento.	Los parámetros entrados (URL, id_sesión) sean correctos.

Tabla 16 Caso de Prueba para el Camino 1-2-3-4-5-6-7-4-2-8

Entrada	Resultados	Condiciones
Se entra la URL de la página y el id_sesión para consumir el servicio que ofrecerá las palabras predominantes que se encuentran en las keywords de la página especificada, pero se cumple que la palabra predominante es igual a la keyword en esa posición.	Se retorna el arreglo result_palabras, con la información almacenada hasta el momento.	Los parámetros entrados (URL, id_sesión) sean correctos.

4.4 Conclusiones

Durante el capítulo se describieron los principales artefactos de los flujos de trabajo de implementación y prueba, es decir los diagramas de despliegue y componentes, así como dos ejemplos de casos de pruebas realizadas al sistema, mediante el método de caja blanca.

Capítulo 5

Estudio de Factibilidad

5.1 Introducción

En el presente capítulo, se realiza una estimación del esfuerzo, costo y tiempo de desarrollo del sistema, usando para ello el método de Puntos de Caso de Uso. Además se analizan los resultados obtenidos, permitiendo evaluar la factibilidad o no del proyecto.

5.2 Aplicar Método de Estimación Puntos por Casos de Uso

Existen varios métodos que permiten hacer una estimación inicial del proyecto, dando la oportunidad de evaluar la factibilidad del mismo. Entre ellos se encuentran: el Análisis de Puntos de Función, Cocomo II y Análisis de Puntos de Caso de Uso.

Este último, es el método usado en la siguiente investigación, por resultar muy efectivo para estimar el esfuerzo requerido en el desarrollo de los primeros casos de uso de un sistema, si se sigue una aproximación iterativa como el Proceso Unificado de Rational. En este tipo de aproximación, los primeros casos de uso a desarrollar son los que ejercitan la mayor parte de la arquitectura del software y los que a su vez ayudan a mitigar los riesgos más significativos. [18]

Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

5.2.1 Cálculo de Puntos de Casos de Uso sin ajustar.

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor, se calcula a partir de la siguiente ecuación:

UUCP = UAW + UUCW

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Para calcular UAW.

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Los criterios se muestran en la siguiente tabla:

Tipo	Descripción	Peso	Cant*Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	1*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	0*3
Total			2

Para calcular UUCW.

Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.

Los criterios se muestran en la siguiente tabla:

No.	Nombre de Caso de Uso	No. transacciones	Tipo
1	Obtener Clave Sesión	2	Simple
2	Cambiar Propiedades de Conexión	2	Simple
3	Obtener Estadígrafos Página Web	21	Complejo
4	Obtener Estadígrafos Palabras Predominantes	9	Complejo

Tipo	Descripción	Peso	Cant*Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	2*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	0*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	2*15
Total		40	

Luego: $UUCP = 2+40$

$UUCP = 42$

5.2.2 Ajustar los Puntos de casos de uso

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

Donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Para Calcular TCF.

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5. A continuación se muestra el significado de estos valores:

0: No presente o sin influencia,

1: Influencia incidental o presencia incidental

2: Influencia moderada o presencia moderada

3: Influencia media o presencia media

4: Influencia significativa o presencia significativa

5: Fuerte influencia o fuerte presencia

En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Factor	Descripción	Peso	Valor	Σ (Pesoi * Valori)
T1	Sistema distribuido	2	3	6
T2	Objetivos de performance o tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	2	2
T4	Procesamiento interno complejo	1	4	4
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	1	0.5
T7	Facilidad de uso	0.5	3	1.5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	5	5
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	4	4
T13	Se requieren facilidades especiales de entrenamiento a los usuarios	1	1	1
Total				51

Luego: $TCF = 0.6 + 0.01 * \Sigma$ (Pesoi * Valori) (Donde Valor es un número del 0 al 5)

$$TCF = 0.6 + 0.01 * 51$$

$$TCF = 1.11$$

Para Calcular EF

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en la estimación de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

En la siguiente tabla se muestra el valor asignado y el peso de cada uno de estos factores.

Factor	Descripción	Peso	Valor	Σ (Pesoi * Valori)
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4	6
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en orientación a objetos	1	5	5
E4	Capacidad del analista líder	0.5	4	2
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	4	8
E7	Personal part-time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	3	-3
Total				24.5

$$EF = 1.4 - 0.03 * \Sigma (\text{Pesoi} * \text{Valori}) \text{ (Donde Valor es un número del 0 al 5)}$$

$$EF = 1.4 - 0.03 * 24.5$$

$$EF = 0.665$$

Luego: $UCP = 42 * 1.11 * 0.665$

$$UCP = 31.0023$$

5.2.3 Cálculo del Esfuerzo

Realizados ya, los pasos anteriores se puede proceder al cálculo del esfuerzo. Se debe tener en cuenta que este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de las funcionalidades especificadas en los casos de uso.

Para ellos se usó la siguiente ecuación:

$$E = UCP * CF$$

Donde:

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

Para calcular CF

CF = 20 horas-hombre (si Total EF \leq 2)

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF \geq 5)

Total EF = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

Como Total EF = 2 + 0

Total EF = 2

CF = 20 horas-hombre (porque Total EF \leq 2)

Luego E = 31.0023 * 20 horas-hombre

E = 620.046 horas-hombre

5.2.4 Distribución del Esfuerzo entre las diferentes actividades de un proyecto.

Finalmente, para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. La siguiente tabla muestra el porcentaje asignado a cada una de estas actividades. Con este criterio, y tomando como entrada la estimación de tiempo calculada a partir de los Puntos de Casos de Uso, se pueden calcular las demás estimaciones para obtener la duración total del proyecto.

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	155.012 horas-hombre
Diseño	20%	310.023 horas-hombre
Implementación	40%	620.046 horas-hombre
Prueba	15%	232.517 horas-hombre
Sobrecarga	15%	232.517 horas-hombre
Total	100%	1550.115 horas-hombre

El Esfuerzo Total sería 1550.115 horas-hombre, si se estima teniendo en cuenta las condiciones que presentan los desarrolladores de este software, que un mes tiene 176 horas laborables, pues se trabajan 8 horas diarias 22 días al mes, entonces el Esfuerzo Total en mes-hombre sería 8.807 mes-hombre.

5.2.5 Calcular el costo de todo el proyecto y su tiempo de desarrollo.

$$\text{Costo} = \text{CHM} * \text{ET} / \text{CH}$$

Donde:

CH: Cantidad de hombres.

CHM: Costo Hombre - Mes

ET: Esfuerzo Total

Si la Cantidad de hombres es 2 y se tiene un Salario Promedio mensual igual a \$100.00.

Entonces $CHM = CH * \text{Salario Promedio}$

$$CHM = 2 * 100$$

$$CHM = 200.00$$

Luego $\text{Costo} = CHM * ET / CH$

$$\text{Costo} = 200.00 * 8.807 / 2$$

$$\text{Costo} = \$ 880.7$$

Tiempo = E_T / CH

$$\text{Tiempo} = 8.807 / 2$$

$$\text{Tiempo} = 4.4035 \approx 4 \text{ meses}$$

De los resultados obtenidos se puede estimar, que 2 hombres trabajando en el proyecto pueden llegar a desarrollarlo en aproximadamente 4 meses y su costo total se estima sería de \$880.7.

5.3 Beneficios tangibles e intangibles.

Como se ha expuesto en los capítulos anteriores, lograr el posicionamiento de las páginas cubanas en los principales buscadores del mundo, es de vital importancia. Se necesita que los internautas adquieran la información exacta que brindan los medios cubanos y no a través de terceros, pues en muchas ocasiones es fruto de tergiversación.

Para ello se hace necesario contar con herramientas que ayuden a los webmasters cubanos, en la toma de decisiones en el ámbito del posicionamiento web, y Cuba no cuenta con ninguna hasta el momento.

El desarrollo de un servicio web con las principales funcionalidades para un análisis SEO, como el que se propone en la siguiente investigación, es el primer paso para fomentar la creación de este tipo de aplicaciones tan importantes para Cuba, disminuyendo considerablemente su tiempo de desarrollo.

Por tanto, resumiendo los beneficios inmediatos, se podría decir que son generalmente intangibles:

- ✓ Fomento de la creación de aplicaciones SEO, pues se hace mucho más fácil su implementación con el uso de servicios web.
- ✓ Disminución del tiempo de desarrollo de las aplicaciones SEO que se desarrollen, dado por la reutilización que se deriva del uso de los servicios web.
- ✓ Aumento de la interoperabilidad, flexibilidad y asequibilidad, en las aplicaciones SEO que se desarrollen, gracias al uso de servicios web.
- ✓ Facilidad al cambio, pues si es necesario modificar la implementación de alguna funcionalidad del servicio web, esta se hace sin que la aplicación que lo consume se vea afectada.
- ✓ Aumento la confidencialidad de la información sensible, pues los datos se envían utilizando HTTPS.

5.4 *Análisis de costo.*

En el mundo existen gran cantidad de herramientas SEO, así como compañías dedicadas a ofrecer servicios sobre posicionamiento web. Entre estas últimas se tienen por solo citar algunas:

*AltamiraWeb.net*⁸, la cual ofrece servicios de optimización de códigos HTML y keywords por 30 euros cada página, un servicio de optimización básica por 500 euros al mes, que solo incluye la identificación de las palabras claves del sitio para conseguir más visitas y dar de alta manualmente al sitio en 200 directorios, entre otros servicios.

⁸ En línea en: <http://altamiraweb.net/posicionamiento.htm>

*Soy Digital*⁹, la cual ofrece un servicio básico por 90 euros, uno profesional por 155 y uno corporativo por 195 euros al mes.

*XeoWeb*¹⁰, ofrece alta en directorios por un precio que va desde los 80 euros hasta los 900 euros, en dependencia del número de directorios en que incluyan el sitio web. Cuenta además con servicios de informes de posicionamiento a un precio de 120 euros, todo ello sin incluir en los precios el IVA¹¹.

Lo antes expuesto permite hacer un análisis de la factibilidad de realización del sistema que se propone, pues se reducirían considerablemente los costos. Teniendo en cuenta además que la tecnología utilizada para su desarrollo es totalmente libre, por tanto no hay que incurrir en gastos de licencias de uso. Solo se incurriría en gastos como el del salario a los desarrolladores. Pero analizando estos contra los múltiples beneficios que se reportarían detallados con anterioridad y comparando el valor comercial de productos similares con el costo estimado del servicio web propuesto, que es de \$880.7, se puede concluir que su implementación es realmente factible.

5.5 Conclusiones

En el capítulo se hizo una estimación del esfuerzo, costo y tiempo de desarrollo del sistema, mediante el método de Análisis de Puntos de Caso de Uso. Así como un análisis, para determinar la factibilidad de llevar a cabo el sistema, haciendo un estudio de los beneficios que ofrece contra los costes de desarrollo, donde se llegó a la conclusión de que era factible su realización.

⁹ En línea en: <http://www.soydigital.com/index.php?optMenu=PosicionamientoWeb>

¹⁰ En línea en: <http://www.seoperu.com/>

¹¹ Impuesto al valor agregado.

Conclusiones Generales

Después de culminar la investigación se puede concluir que se cumplió el objetivo propuesto de desarrollar un repositorio de servicios web para facilitar la implementación de aplicaciones que ayuden a la toma de decisiones SEO, reduciendo su tiempo de desarrollo.

Para su cumplimiento se desarrolló un servicio web que cuenta con las principales funcionalidades que pueden servir para facilitar el trabajo a los desarrolladores de este tipo de aplicaciones. Potenciando así aspectos importantes en ellas como: la reutilización, flexibilidad, interoperabilidad, y asequibilidad.

A modo de prueba se desarrolló una aplicación web, que funcionara como cliente del servicio web desarrollado y permitiera comprobar el funcionamiento del mismo. Se incorporan además, aspectos de seguridad en el sistema, pues el envío de datos se hace por un canal seguro usando para ello, el protocolo HTTPS para asegurar la transferencia de la información a nivel de transporte.

Para lograr una solución que cumpliera con el objetivo planteado, se realizó un análisis de las estrategias de posicionamiento web que existen en el mundo, donde a partir de ahí se pudo identificar las principales funcionalidades que por sus características pudieron ser implementadas como servicios. También se realizó un análisis sobre algunos servicios web que existen en el mundo que pudieran servir con el fin que se pretende, de los cuales se pudo concluir que poseen algunas limitaciones para su uso. De ahí la importancia de la presente investigación, sobre todo por constituir el primer paso para el fomento de la creación de aplicaciones SEO para Cuba, tan necesarias si se pretende que sean las páginas cubanas las que aparezcan en los primeros lugares de búsqueda de los buscadores. Además de una estimación del esfuerzo, costo y tiempo de desarrollo del sistema, permitiendo evaluar la factibilidad del mismo.

Para la realización de lo antes expuesto, se empleó como metodología de desarrollo RUP así como herramientas y tecnologías, multiplataforma según las políticas de la universidad.

Recomendaciones

Con el objetivo de mejorar y seguir perfeccionando el servicio web creado se recomienda:

- ✓ Mantener un seguimiento y estudio sobre los cambios que puedan ocurrir en las técnicas de posicionamiento web y los algoritmos de recuperación de los buscadores, pues estos pueden cambiar en aras de optimizar aun más los resultados de las búsquedas, con el propósito de agregar nuevas funcionalidades al servicio o actualizar alguna existente.
- ✓ En aras de mejorar aspectos en la seguridad de los servicios, incluir la seguridad a nivel de mensajes.
- ✓ Comenzar a usarlo en las herramientas SEO que se desarrollarán en la Línea de Posicionamiento Web del Proyecto GIDI.

Referencias Bibliográficas

1. **Campos, Sergio García.** Algoritmos de posicionamiento de Google, Yahoo y MSN. *Recuperacion y organizacion de la informacion*. [En línea] Abril de 2006. <http://posicionamientoweb.tripod.com/>.
2. **del Sol Reyes, Aday.** CubaSí.cu. *Con Internet se rompe el monopolio de la información*. [En línea] 18 de Mayo de 2007. <http://www.cubasi.cu/DesktopDefault.aspx?SPK=160&CLK=156146&LK=1&CK=80621&SPKA=35>.
3. **Leiner, Barry M., y otros.** Una breve historia de Internet. [En línea] 1 de Diciembre de 1999. <http://www.ati.es/DOCS/internet/histint/histint1.html>.
4. TuFunción. *El crecimiento de Internet*. [En línea] 8 de Julio de 2008 . <http://tufuncion.com/crecimiento-internet>.
5. **de Ugarte, David.** *El libro del posicionamiento en buscadores*. [pdf] Madrid : s.n., 2002.
6. Atedis. Apoyo Tecnológico para la discapacidad. *Glosario de Términos*. [En línea] http://www.atedis.gov.ar/glosario.php?busca_letra=2&hoja=.
7. **Gosende, Javier.** *Guía Práctica de Posicionamiento en Buscadores*. [pdf] s.l. : Human Level Communications.
8. W3C. World Wide Web. *Guía Breve de Servicios Web*. [En línea] 9 de Enero de 2008. <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
9. **Fernández Calvo, Rafael.** Asociacion de Técnicos de Informática. *Glosario básico inglés-español para usuarios de Internet*. [En línea] julio de 2001. http://www.ati.es/novatica/glosario/glosario_internet.html#eXtensible%20Markup%20Language.
10. Software Zone. *Diccionario Informático*. [En línea] <http://software.adslzone.net/glosario/s-t-y-u/>.
11. **Thomas, Johnson P, Thomas, Mathews y Ghinea, George.** *Modeling of Web Services Flow*. s.l. : IEEE International Conference on E-Commerce, 2003.
12. IBM. *Conozca SOA*. [En línea] http://www-306.ibm.com/e-business/la/ar/soa/soa_2.shtml.
13. **Larman, Craig.** *UML y Patrones*. La Habana : Félix Varela, 2004.
14. **Teleformación.** *Conf#3 Flujo de trabajo de requerimientos*. [pdf] Curso 2007-2008.
15. **Teleformación.** *Conf#6 Fase de Inicio. Flujo de Análisis y Diseño. Modelo de Análisis*. [pdf] Curso 2007-2008.
16. **Teleformación.** *Conf #4 Flujo de trabajo de Prueba*. [doc] Curso 2005-2006.

17. **Teleformación.** *Conf #4 Flujo de trabajo de Prueba.* [doc] Curso 2005-2006.
18. **Peralta, Mario.** *Estimación del esfuerzo basada en casos de uso.* [pdf] Buenos Aires, Argentina : Centro de Ingeniería del Software e Ingeniería del Conocimiento (CAPIS).

Bibliografía

1. Inmersión en Python. [En línea] http://www.gulic.org/almacen/diveintopython-5.4-es/soap_web_services/google.html.
2. **Oakley, Bryan.** Web Services and the Google API. [En línea] 2006. <http://www.tclscripting.com/articles/nov06/article1.html>.
3. Visual Paradigm. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
4. **Zukowski, John.** *Programacion Java2 J2SE 1.4.* [pdf] 2003.
5. OnJava.com. *An Introduction to the Eclipse IDE.* [En línea] <http://www.onjava.com/pub/a/onjava/2002/12/11/eclipse.html>.
6. Slide Share. *Ide Eclipse, Breve Guía.* [En línea] <http://www.slideshare.net/Benedeti/ide-eclipse-breve-gua-201399>.
7. The Apache Software Foundation. *Axis2.* [En línea] <http://ws.apache.org/axis2/>.
8. The Apache Software Foundation. *Apache Tomcat.* [En línea] <http://tomcat.apache.org/>.
9. My SQL. *Las principales características de MySQL.* [En línea] <http://dev.mysql.com/doc/refman/5.0/es/features.html>.
10. Yahoo! Developer Network. *Yahoo! Search Web Services.* [En línea] <http://developer.yahoo.com/search/>.
11. Agent Web Ranking. [En línea] <http://www.agentwebranking.com/es/pricing.htm>.
12. **Teleformación.** *Conf#1 Introducción a la ISW.* [pdf] Curso 2007-2008.
13. **Teleformación.** *Conf#7 Fase de Inicio. Flujo de Análisis y Diseño. Diagramas de Interacción.* [pdf] Curso 2007-2008.
14. **Blasi, Emanuel.** *Resumen de Patrones de Diseño.* [pdf]

15. **Teleformación.** *Conf #6 Fase de Construcción. Flujo de trabajo de Implementación.* [doc] Curso 2005-2006.
16. **Teleformación.** *Conf#3 Fase de Elaboración. Flujo de Trabajo Implementación.* [doc] Curso 2005-2006.
17. **2008.** AmericaInternet.cl. *Algunas Reglas de Relevancia para el Posicionamiento Web.* [En línea] 13 de Diciembre de 2008. [Citado el: 20 de Mayo de 2009.] <http://www.americainternet.cl/noticias/162-algunas-reglas-de-relevancia-para-el-posicionamiento-web>.
18. **2008.** Informática Hoy. *10 reglas de oro para el posicionamiento web.* [En línea] 2008. [Citado el: 21 de Mayo de 2009.] <http://www.informatica-hoy.com.ar/internet-publicidad-posicionamiento/10-reglas-de-oro-para-el-posicionamiento-web.php>.
19. Reinventia. *Estrategias de posicionamiento web y técnicas SEO para palabras clave.* [En línea] [Citado el: 22 de Mayo de 2009.] <http://madrid.reinventia.net/tecnicas-estrategias-posicionamiento-web-SEO.php>.
20. **Drakic, Branko Fiorilo. 2008.** Sun Microsystems. *Desarrollo y Deploy de Web Services con Sun Java System Application Server.* [En línea] Octubre de 2008. [Citado el: 23 de Mayo de 2009.] http://www.sun.com/bigadmin/content/submitted/es/web_serv_span.jsp.
21. **2008.** La red para profesionales IT. *WebServices en Axis (III) , fácil y sencillo. Seguridad en WS(WS-Security).* [En línea] 2008.
22. **Drakic, Branko Fiorilo. 2008.** Sun Microsystems. *Desarrollo y Deploy de Web Services con Sun Java System Application Server 9.* [En línea] Octubre de 2008. http://www.sun.com/bigadmin/content/submitted/es/web_serv_span.jsp.

Anexos

Diagrama de Secuencia del CUS Obtener Estadígrafos Página Web.

Escenario Cantidad de Imágenes

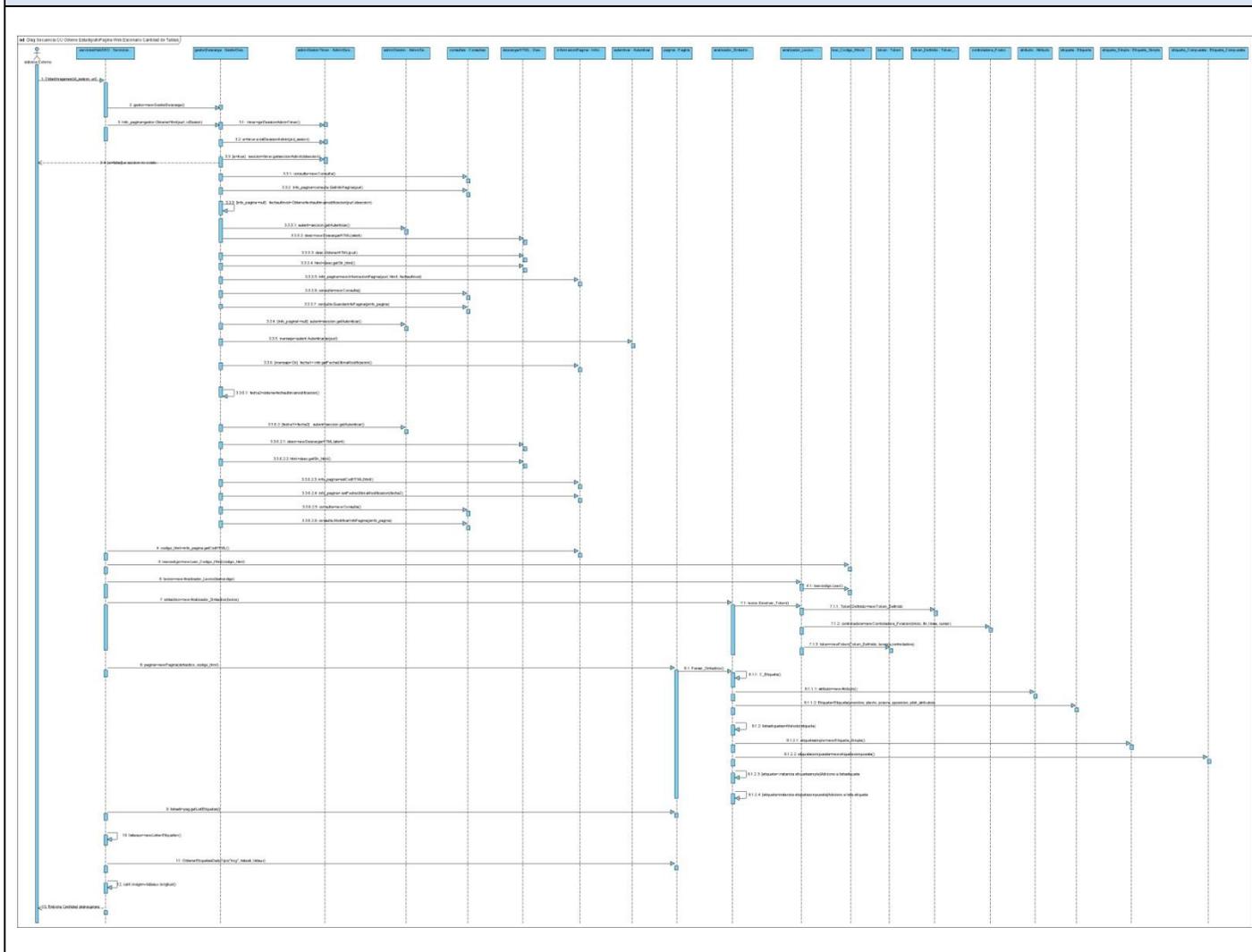


Fig. 28 Diagrama de Secuencia del CUS Obtener Estadígrafos Página Web. Escenario Cantidad de Imágenes.

Glosario de Términos

API: Siglas de Application Programming Interface, especificación de una librería o utilidad que documenta su interfaz y permite su uso sin conocimiento de su interior.

ARPANET: Siglas de Advanced Research Projects Agency Network, red de computadores creada en EE.UU. con fines militares, a partir de la cual evolucionó Internet.

Buscadores: Sitio web que gestiona una base de datos confeccionada por robots. Esta base de datos almacena directamente el código de las páginas visitadas por los robots o parte de él

CASE: Siglas de Computer Aided Software Engineering, aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software ayudando en todos los aspectos del ciclo de vida de desarrollo del software.

CORBA: Siglas de Common Object Request Broker Architecture, un estándar basado en ORB que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

CU: Caso de Uso.

CUS: Caso de Uso del Sistema.

DCOM: Siglas de Distributed Component Object Model, facilita la distribución transparente de objetos a través de redes y de Internet.

GIDI: Siglas de Grupo de Investigación y Desarrollo sobre Internet.

GPL: Siglas de General Public License, es una licencia libre, sin derechos para software y otro tipo de trabajos.

GRASP: Siglas de General Responsibility Assignment Software Patterns, patrones que describen los principios fundamentales de la asignación de responsabilidades a objetos.

HTML: Siglas de HyperText Markup Language, es el lenguaje de marcado predominante para la construcción de páginas web.

HTTP: Siglas de HyperText Transfer Protocol, protocolo de transferencia de hipertexto usado en cada transacción de la Web.

HTTPS: Siglas de HyperText Transfer Protocol Secure, es la versión segura del protocolo HTTP y viene dada por el protocolo de seguridad SSL.

IEEE: Siglas de The Institute of Electrical and Electronics Engineers, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.

Internet: Conjunto descentralizado de redes de comunicación interconectadas.

J2EE: Siglas de Java 2, Enterprise Edition, plataforma de desarrollo estándar en la industria en soluciones empresariales.

ORB: Siglas de Object Request Broker, es el nombre que recibe una capa de software que permite a los objetos realizar llamadas a métodos situados en máquinas remotas, a través de una red.

Página Web: Documento situado en el Web con información y diversos enlaces con otros documentos también situados en el Web.

Plug-in: Pequeño programa que añade alguna función a otro programa, habitualmente de mayor tamaño.

Posicionamiento web: Consiste en la aplicación de técnicas tendentes a lograr que los buscadores de Internet encuadren nuestra página web en una posición y categoría deseada dentro de su página de resultados para determinados conceptos clave de búsqueda.

RF: Requisito funcional.

Robot: Programa que navega la web yendo desde cada página a todas las que ésta da enlace y enviando e incorporando el código o partes de él a una base de datos de referencia

RUP: Siglas de Rational Unified Process, constituye una metodología estándar de las más usadas para el análisis, implementación y documentación de sistemas orientados a objetos

SEM: Siglas de Search Engine Marketing, se refiere al Posicionamiento Web basado en los resultados patrocinados cuya clasificación depende del dinero que se invierta en los anuncios.

SEO: Siglas de Search Engine Optimization, se refiere al posicionamiento web cimentado en los resultados naturales u orgánicos, que están basados en el algoritmo imparcial de los buscadores.

Servicio Web: Conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web.

Sitio Web: Es un conjunto de páginas colocadas en Internet.

SOA: Siglas de Service Oriented Architecture, es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

SOAP: Siglas de Simple Object Access Protocol, protocolo de acceso que permite la comunicación entre aplicaciones remotas a través de Internet.

SQL: Siglas de Structured Query Language, lenguaje de programación declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

SSL: Siglas de Secure Socket Layer, protocolo que proporciona servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente.

UCI: Universidad de las Ciencias Informáticas.

UDDI: Siglas de Universal Description, Discovery, and Integration, catálogo de negocios de Internet, es un marco independiente de la plataforma para describir servicios, negocios e integrar servicios de negocios

UML: Siglas de Unified Modeling Language, lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

URL: Siglas de Uniform Resource Locator, es la dirección global de cualquier documento o recurso en la Web, visto como localizador.

Webmasters: Persona encargada de la gestión y mantenimiento de un servidor web, fundamentalmente desde el punto de vista técnico.

WSDL: Siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios web.

WWW: Siglas de World Wide Web, sistema de Internet para vincular mediante hipertexto en todo el mundo, documentos multimedia, permitiendo un fácil acceso, totalmente independiente de la ubicación física, a la información común entre documentos.

XML: Siglas de Extensible Markup Language, lenguaje desarrollado por el W3 Consortium para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar y compartir información en la red.