



Universidad de las Ciencias Informáticas

Facultad 6

Título: Analizador XML y HTML para la optimización de páginas web.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.

Autores: Yusley Orea Rodríguez.

Manuel de Jesús Pérez Núñez.

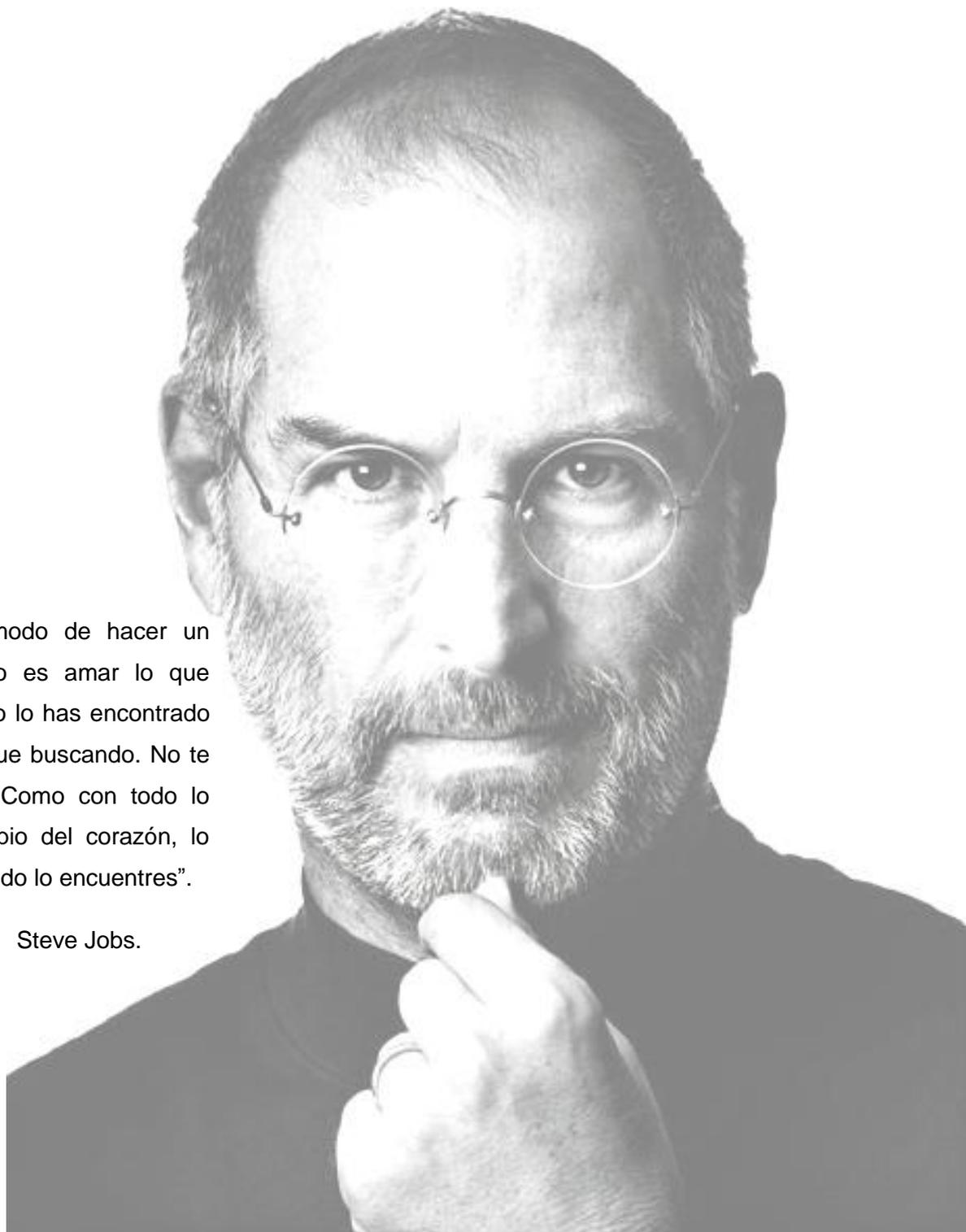
Tutor: Ing. Eduardo M. Macías Sotolongo.

Cotutor: Ing. Yulainne Alonso Hernández.

La Habana, Junio 23 del 2011
“Año 53 de la Revolución”

“El único modo de hacer un gran trabajo es amar lo que haces. Si no lo has encontrado todavía, sigue buscando. No te acomodes. Como con todo lo que es propio del corazón, lo sabrás cuando lo encuentres”.

Steve Jobs.



DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las 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 2011.

Yusley Orea Rodríguez.

Autor

Manuel de Jesús Pírez Núñez.

Autor

Ing. Eduardo M. Macías Sotolongo.

Tutor

Ing. Yulainne Alonso Hernández.

Co-Tutor

Tutor: Ing. Eduardo M. Macías Sotolongo.

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Categoría docente: Instructor

Años de experiencia: 4

Correo electrónico: emmacias@uci.cu

Universidad de la Ciencias Informáticas, La Habana, Cuba.

Co-Tutor: Ing. Yulainne Alonso Hernández.

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Categoría docente: No.

Años de experiencia: No.

Correo electrónico: yalonsoh@uci.cu

Universidad de la Ciencias Informáticas, La Habana, Cuba.

Agradezco a:

A mis padres Alina, Manuel y mi segundo padre Rene por ser personas especiales en mi vida, por todo el apoyo que me han dado, por ser esas luces que me han alumbrado el camino en todo momento, Gracias.

De forma general agradezco a toda mi familia que de una forma u otra se han preocupado por mí y me han estirado su mano.

A mi novia Annis, que siempre ha estado a mi lado desde que la conocí y me ha apoyado en todo, te quiero gata.

A dos hermanos que encontré Laidy (no me hace caso), Orea (pensamientos sincronizados), a inicios de mi carrera universitaria y nunca los olvidare. A todos los amigos, compañeros de la universidad.

A mi tutor Eduardo y a la cotutora Yuly y Ponce quienes no dejaron de batallar por esta tesis.

A Nelson Rodríguez Proenza un ejemplo a seguir en esta disciplina.

Manuel.

Quiero agradecer primeramente a mi madre y a mis hermanos que representan lo más valioso en vida y a toda mi familia por su comprensión y ayuda en todo momento. Al tutor Eduardo y a la cotutora Yuly que nos han apoyado y ayudado en los resultados de la tesis a pesar de las trabas que nos han puesto. A Necho el primer profesor de programación del cual aprendí muchísimo, sobre todo las cosas que no imparten en clases. A todos los que han tenido que ver de una forma u otra con mi formación profesional. A todas las amistades con las que he compartido los buenos y malos momentos a lo largo de estos cinco años. En especial a los que han estado más cerca y han llegado a conocerme un poco mejor, los llevo a todos en mis pensamientos y nunca los olvidaré.

Yusley.

A toda mi familia y en especial a mi madre, mi padre y mi padrasto.

Manuel.

A mi madre por todo el amor, apoyo, comprensión y preocupación.

A mis hermanos Yusniel y Yurian por todo el amor que me han dado, por enseñarme y guiarme por el camino correcto. A ellos que son mi ejemplo a seguir.

A mi padrasto Manuel, por todo su amor y comprensión, por estar siempre presente y sobre todo por todo el apoyo que le ha dado a mi madre.

A mi novia Thais por todo el amor, apoyo y paciencia que ha tenido conmigo.

A mi familia en general por su apoyo y comprensión. En especial a la memoria de mi padre.

A los hermanos que conocí en la universidad Raidel y Pérez, que espero no nos distanciamos y poder compartir unos cuantos años más en la vida.

Yusley.

La investigación recoge todo el proceso de desarrollo del sistema analizador XML y HTML para la optimización de páginas web. Para cumplir con los propósitos de la investigación, se realizó un estudio de los principales sistemas existentes tanto a nivel nacional como internacional, relacionados con el Posicionamiento Web y la optimización para los motores de búsqueda (SEO). Se utilizó Java como lenguaje de programación, como metodología de desarrollo el Proceso Racional Unificado (RUP) junto al Lenguaje Unificado de Modelado (UML); como herramienta de modelado Visual Paradigm y NetBeans como entorno de desarrollo.

La herramienta da la posibilidad al Webmaster de analizar páginas web a partir de una definición de tipo de documento o DTD, donde valida la estructura interna del código fuente para detectar y sugerir la corrección de los posibles errores que contenga. Permite conocer la densidad y relevancia de cada palabra reflejada en el documento. Brinda la opción de comparar múltiples páginas gráficamente. Generar reportes con las características y errores encontrados en el análisis de la página.

Palabras Claves: Posicionamiento Web, SEO, HTML, XML, XHTML, DTD, Buscadores.

The research includes all the system development process XML and HTML parser for optimization of web pages. To fulfill the purposes of research, a study of the principal existing systems both nationally and internationally, related to Web Positioning and Search Engines Optimization (SEO). We used Java as a programming language, such as development methodology Rational Unified Process (RUP) with the Unified Modeling Language (UML), Visual Paradigm and NetBeans as a tool for modeling and integrated development environment respectively.

The tool offers the possibility to analyze web pages from a document type definition or DTD, which validates the internal structure of the source code to detect and suggest the correction of any errors it contains. Lets you know the density and relevance of each word reflected in the document. Provides the option to graphically compare multiple pages. Generate reports with the features and bugs found in the analysis of the page.

Keywords: Web positioning, SEO, HTML, XML, XHTML, DTD, search engines.

Introducción	1
Capítulo 1: Fundamentación Teórica.	6
1.1. Introducción	6
1.2. Internet.....	6
1.2.1 World Wide Web	6
1.2.2 HTML	7
1.2.3 XML.....	8
1.2.4 XHTML.....	8
1.2.5 DTD.....	9
1.3 Motores de Búsqueda.....	9
1.3.1 Optimización para los Motores de Búsqueda.....	11
1.4 Analizadores de XML, HTML, XHTML y herramientas SEO	13
1.5 Metodologías de Desarrollo	19
1.5.1 RUP.....	19
1.6 Lenguaje de Modelado	19
1.7 Herramienta CASE (Computer-Aided Software Engineering)	20
1.7.1 Visual Paradigm	20
1.8 Lenguaje Programación.....	20
1.8.1 Java.....	20
1.9 Reportes.....	21
1.9.1 Ireport v1.3.3.....	21
1.10 IDE	22
1.10.1 NetBeans.....	22
1.11 Conclusiones Parciales.....	23
Capítulo 2: Características del Sistema	24

2.1	Introducción	24
2.2	Modelo de dominio	24
2.2.1	Representación del modelo de dominio	24
2.3	Especificación de los requisitos del sistema	26
2.3.1	Requisitos funcionales	26
2.3.2	Requisitos no funcionales.....	27
2.4	Definición de los casos de uso del sistema	28
2.4.1	Actores del sistema	29
2.4.2	Diagrama de casos de uso del sistema	29
2.4.3	Descripción de los casos de uso del sistema.....	30
2.5	Prototipo de interfaz de usuario.....	37
2.6	Conclusiones Parciales.....	40
Capítulo 3: Diseño del Sistema		41
3.1	Introducción	41
3.2	Patrón Arquitectónico Utilizado.....	41
3.3	Patrones de Diseño Utilizados.....	42
3.3.1	Patrón Creador	42
3.3.2	Patrón Bajo Acoplamiento	43
3.3.3	Patrón Alta Cohesión	43
3.3.4	Patrón Experto	43
3.3.5	Patrón Observador	43
3.3.6	Patrón Singleton.....	43
3.4	Modelo de Diseño.....	44
3.4.1	Diagramas de Clases del Diseño	44
3.4.2	Diagramas de Secuencia.....	44

3.4.3	Conclusiones Parciales.....	45
Capítulo 4:	Implementación y Pruebas.....	46
4.1	Introducción	46
4.2	Diagrama de componentes	46
4.3	Marcos de trabajos utilizados	47
4.3.1	Matra.....	47
4.3.2	JasperReport.....	48
4.5	Pruebas del sistema.....	49
4.5.1	Configuración del entorno de pruebas	49
4.5.2	Diseño de las pruebas de Caja Negra.....	51
4.6	Conclusiones Parciales.....	58
Conclusiones	Generales.....	59
Recomendaciones	60
Bibliografía	Consultada.....	61
Bibliografía	Citada.....	65
Glosario de	Términos.....	66

<i>Tabla 1: Analizadores de XML, HTML, XHTML y herramientas SEO.....</i>	<i>17</i>
<i>Tabla 2: Actores del sistema.....</i>	<i>29</i>
<i>Tabla 3: Descripción del caso de uso Configurar Proxy.....</i>	<i>30</i>
<i>Tabla 4: Descripción del caso de uso Analizar Página Web.</i>	<i>34</i>
<i>Tabla 5: Descripción del caso de uso Comparar Páginas.</i>	<i>35</i>
<i>Tabla 6: Descripción del caso de uso Administrar Metas.</i>	<i>36</i>
<i>Tabla 7: Descripción del caso de uso Generar Reporte a PDF.....</i>	<i>37</i>
<i>Tabla 8: Planificación de las Pruebas.</i>	<i>50</i>
<i>Tabla 9: Primera iteración de las Pruebas.....</i>	<i>51</i>
<i>Tabla 10: Diseño de las pruebas del caso de uso Analizar Página Web.</i>	<i>53</i>
<i>Tabla 11: Diseño de las pruebas del caso de uso Comparar Páginas.</i>	<i>54</i>
<i>Tabla 12: Diseño de las pruebas del caso de uso Administrar Metas.</i>	<i>55</i>
<i>Tabla 13: Diseño de las pruebas del caso de uso Generar Reporte a PDF.....</i>	<i>55</i>
<i>Tabla 14: Diseño de las pruebas del caso de uso Configurar Proxy.....</i>	<i>56</i>
<i>Tabla 15: No Conformidades.....</i>	<i>58</i>

<i>Figura 1: Esquema de evolución de HTML y XHTML.....</i>	<i>9</i>
<i>Figura 2: Diagrama de modelo de Dominio.....</i>	<i>25</i>
<i>Figura 3: Diagrama de casos de uso del sistema.</i>	<i>29</i>
<i>Figura 4: Prototipo de interfaz de usuario: Configurar Proxy.....</i>	<i>38</i>
<i>Figura 5: Prototipo de interfaz de usuario: Analizar Página.....</i>	<i>38</i>
<i>Figura 6: Prototipo de interfaz de usuario: Comparar Páginas.</i>	<i>39</i>
<i>Figura 7: Prototipo de interfaz de usuario: Administrar Metas.</i>	<i>39</i>
<i>Figura 8: Prototipo de interfaz de usuario: Generar Reporte en PDF.....</i>	<i>40</i>
<i>Figura 9: Diagrama de componentes.</i>	<i>46</i>

Introducción

A inicios de la década 1990, surge la WWW (*World Wide Web*), con un conjunto de protocolos que permite la consulta remota de documentos de hipertexto enlazados y accesibles, empleando Internet como medio de transmisión. Cuando Teem Beerners Lee¹ haciendo uso del SGML² (*Standard Generalized Markup Language*) especifica la sintaxis para la inclusión de marcas en los textos, creó el HTML (*Hypertext Markup Language*) o Lenguaje de Etiquetas de Hipertexto, el protocolo HTTP³ (*Hypertext Transfer Protocol*) y el sistema de localización de objetos en la web URL⁴ (*Uniform Resource Locator*), facilitando la forma de comunicación y actualización de la información entre investigadores.

El HTML es hoy en día el tipo de documento más empleado en el mundo, por su flexibilidad y escalabilidad junto con el uso de los navegadores y el URL. Además su sencillez es tal, que cualquier persona puede escribir documentos en este formato sin tener conocimientos de informática y esa fue una de las razones de su gran éxito.

Con el crecimiento de la web se produjeron innumerables documentos, pero muchos de ellos mal estructurados, principalmente por la no aplicación de los estándares por parte de los diseñadores web. Al mismo tiempo surgía el DTD (*Document Type Definition*) que describe la estructura y sintaxis de un documento.

La respuesta de los problemas surgidos en torno al HTML, fue la creación del XML (*Extensible Markup Language*) por parte de la W3C (*World Wide Web Consortium*) para adaptarlo al campo de internet. Este meta-lenguaje es bien preciso en cuanto a lo que está permitido y lo que no, por lo que todo documento debe cumplir dos condiciones: ser válido y estar bien formado.

1 Teem Beerne Lee, Ingeniero Físico graduado en 1976 en el Queen's College de la Universidad de Oxford. Es considerado como el padre de la web.

2 SGML Consiste en un lenguaje para la organización y etiquetado de documentos.

3 Http es el protocolo usado en cada transacción de la World Wide Web.

4 URL es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.

Actualmente es difícil adquirir contenido específico en Internet, por el gran volumen de páginas y documentos existentes. Por tanto los usuarios que acceden a Internet en busca de datos, lo hacen a través de los sistemas automáticos de recuperación de información llamados buscadores, que permiten localizar en poco tiempo los documentos.

Los resultados de las búsquedas, no siempre son los deseados, aún existiendo la información anhelada, dado que la mayoría de las personas sólo visitan los primeros vínculos de los resultados mostrados por los motores de búsqueda. Para que una página tenga éxito, debe encontrarse en las primeras posiciones de los resultados, de ahí que se aplican una serie de estrategias y técnicas de optimización a las páginas web, lo cual constituye la columna vertebral del posicionamiento web.

Cuba no se encuentra ajena a este fenómeno, intenta insertarse en Internet y que sus sitios web ocupen los primeros lugares en los buscadores para determinados criterios de búsquedas. Los webmasters cubanos, tienen muy poco conocimiento sobre la importancia de alcanzar y mantener un buen posicionamiento web. Además no aplican las estrategias SEO -del inglés *Search Engine Optimization*- y no poseen alguna herramienta automatizada que les ayude en gran parte a realizar esta labor, sin obviar que el análisis y estudio de los sitios web requiere tiempo y es complejo. En la actualidad existen varias herramientas de posicionamiento web, pero la mayoría no son baratas y las gratuitas no están a la altura deseada.

Uno de los principales factores que tienen en cuenta los motores de búsqueda, es la estructura y composición de las páginas web que visitan e indexan en sus grandes bases de datos. Los documentos con errores en su código fuente son más propensos a no aparecer en los primeros resultados de los buscadores.

Por lo anteriormente descrito se plantea como **problema a resolver**: ¿Cómo analizar los documentos XML, XHTML y HTML detectando los errores de estructura, de acuerdo con los estándares definidos por la W3C y que atiendan al buen posicionamiento de las páginas en los motores de búsqueda?

El **objeto de estudio** son los estándares de XML, XHTML y HTML especificados por la W3C. Definiendo como **campo de acción** los procesos de optimización para motores de búsqueda basado en la estructura y composición de las páginas web.

Para dar solución a este problema se propone el siguiente **objetivo general**: Desarrollar un analizador de XML, XHTML y HTML a partir de los estándares definidos por la W3C, que permita detectar errores de estructura y que atiendan al buen posicionamiento de las páginas en los motores de búsqueda.

Para cumplir con el objetivo general de la investigación se trazan los siguientes objetivos específicos:

- Analizar el marco teórico conceptual y el estado del arte respecto a los lenguajes de marcado y el posicionamiento web.
- Diseñar el sistema para analizar y optimizar documentos XML, XHTML y HTML.
- Implementar el sistema para analizar y optimizar documentos XML, XHTML y HTML.
- Aplicar las pruebas de caja negra al sistema.

Con el fin de consumir los objetivos específicos se crean las siguientes tareas de investigación:

- Análisis del surgimiento y evolución de los lenguajes de marcado utilizados para representar el contenido de las páginas web.
- Análisis de la estructura de los lenguajes de marcado utilizados para representar el contenido de las páginas web.
- Análisis del estado del arte a nivel nacional e internacional respecto a los sistemas existentes que procesen lenguajes de marcado.
- Análisis del funcionamiento de los sistemas de posicionamiento web de los motores de búsqueda.
- Identificación de las técnicas de optimización para los motores de búsqueda a utilizar.
- Definición del lenguaje de programación a utilizar.
- Definición de un analizador de DTD a utilizar.
- Identificación de la herramienta de modelado a utilizar.
- Confección de los diagramas de clases de diseño.
- Selección de patrones necesarios para el desarrollo de la aplicación.
- Implementación del Analizador de XML.
- Implementación del Analizador de HTML.

- Implementación del módulo Optimizador para Motores de Búsqueda.
- Aplicación de las pruebas de caja negra.

Con lo anteriormente reflejado se puede plantear la siguiente **idea a defender**: Con el desarrollo y utilización de un sistema capaz de analizar las páginas web, detectar y sugerir la corrección de los errores de acuerdo con los estándares definidos por la W3C. Se logrará mejorar el posicionamiento de las páginas cubanas en los motores de búsqueda.

Como resultados del trabajo de diploma se espera obtener un sistema capaz de analizar los documentos XHTML y detectar los errores presentes en los mismos que atenten contra un buen posicionamiento en los motores de búsqueda. Dicho sistema será lo suficientemente flexible como para adaptarse a los cambios que puedan sufrir en el futuro los lenguajes de marcado, principalmente XHTML y HTML, partiendo del análisis de los DTD y siguiendo los estándares definidos por la W3C. El sistema permitirá corregir los errores que se detecten, sugiriendo en muchos casos las posibles mejoras. El sistema está encaminado a mejorar el posicionamiento de las páginas cubanas en los motores de búsqueda de Internet, lo cual permitiría darle mejor visibilidad a los contenidos de interés nacional. El diseño del sistema será totalmente modular y escalable, lo cual permitirá incorporarlo a otras herramientas que se desarrollen en el futuro y que necesiten un analizador de XHTML.

El presente trabajo ha sido estructurado de la siguiente manera:

Capítulo 1: “Fundamentación Teórica”

Se describen y explican los conceptos esenciales, relacionados con el problema; se muestra una panorámica del estado actual de las herramientas existentes. Además se detalla la Metodología de Desarrollo, el Lenguaje de Modelado, la Herramienta CASE, el Lenguaje de Programación y la Plataforma en la que se desarrollará el sistema.

Capítulo 2: “Características del Sistema”

En este capítulo se describen las características del sistema y se efectúa una descripción de la solución propuesta, definiendo los requisitos funcionales y no funcionales. Se muestra el modelo de dominio y el diagrama de casos de uso del sistema y la descripción de los mismos.

Capítulo 3: “Diseño del Sistema”

Se describe la arquitectura del sistema, haciendo énfasis en el patrón arquitectónico utilizado, así como los patrones de diseño que fueron empleados. Se realiza el diseño del sistema por medio de los distintos artefactos propuestos por la metodología de desarrollo de software.

Capítulo 4: “Implementación y prueba del Sistema”

En este capítulo se describe la implementación del sistema; así como las librerías utilizadas para la solución. De igual forma se obtienen los diseños de casos de prueba, que permiten demostrar el correcto funcionamiento del sistema, así como detectar y corregir los posibles errores.

Capítulo 1: Fundamentación Teórica.

1.1. Introducción

En el desarrollo de todo sistema informático es importante esclarecer los conceptos asociados con el dominio del problema para un mejor modelado de la solución a proponer. Estudiar los sistemas que existen en el mundo semejantes al que se necesita, resulta de vital importancia para no duplicar esfuerzos y aprovechar el conocimiento acumulado.

Por esta razón a continuación se explican las principales definiciones que ayudarán a entender los principales temas asociados con el posicionamiento web. Se fundamentará la selección realizada de la metodología de desarrollo, el lenguaje de modelado, el lenguaje de programación y la herramienta CASE que se emplearán en la realización de la aplicación.

1.2. Internet

Aunque la historia de los medios de comunicación populares a menudo es confusa, el concepto de Internet en realidad es muy simple. Es una colección de redes -una red compuesta por redes- de computadoras que comparten información digital por medio de un conjunto común de protocolos de software y de redes. [1]

Sus orígenes se remontan a un proyecto desarrollado por los Estados Unidos en 1967, para apoyar a sus fuerzas militares. El proyecto estableció la primera conexión de computadoras, *Advanced Research Projects Agency Network* conocida como (ARPANET).

Sin dudas, Internet actualmente es el medio de comunicación mayormente utilizado y tiene un impacto profundo en el trabajo, el ocio y el conocimiento a nivel mundial. Ofrece muchos servicios y protocolos, uno de los servicios más exitosos y conocidos es la WWW, hasta tal punto que es habitual la confusión entre ambos términos. Existen muchos otros como el correo electrónico, transmisión de archivos, telefonía, televisión, mensajería instantánea, entre otros.

1.2.1 World Wide Web

La WWW es un sistema de información distribuido basado en hipertexto o hipermedios, enlazados y accesibles, que utiliza un conjunto de protocolos que permiten la consulta remota de archivos de hipertexto.

Su origen se le atribuye al inglés Tim Berners-Lee en 1989. Ante la necesidad de distribuir e intercambiar información de manera más efectiva, decidió unir Internet con el hipertexto y desarrolló lo que se conoce como HTML o Lenguaje de Etiquetas de Hipertexto, el protocolo HTTP y el sistema de localización de objetos en la web URL. Dándole surgimiento así a la *World Wide Web*.

La Web es el medio de mayor difusión de intercambio personal aparecido en la historia de la humanidad, ha permitido un flujo de comunicación global a una escala sin precedentes. Individuos separados en el tiempo y en el espacio pueden utilizar la web para intercambiar pensamientos, actitudes, cultura, negocios, fotografías, etc., haciéndolo llegar casi de forma inmediata a cualquier punto del planeta. El hipertexto ha hecho posible recopilar en un mismo tiempo y lugar todo el saber acumulado de la humanidad desde sus inicios hasta el momento.

1.2.2 HTML

HTML es un lenguaje de composición de documentos y especificación de ligas de hipertexto, que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica como desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. HTML también le indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos ya sea en su computadora o en otras, así como otros recursos de Internet, como FTP y Gopher. [1]

Como se planteaba anteriormente, fue concebido para el intercambio de documentos científicos y técnicos, con soporte de hipertexto y sirviéndose de un conjunto de etiquetas estructurales y delimitadas para realizar documentos relativamente simples.

La descripción se basa en especificar la estructura lógica del contenido así como los diferentes efectos que se quieran dar. Se puede decir que de algún modo el HTML va en contra de los principios por el cual fue creado, ya que detalla mayormente la forma en la que se va a mostrar el contenido que la información o el mensaje incluido. Las páginas web luego pueden ser vistas por un tipo de aplicación llamada navegador o *browser* en inglés. El navegador muestra el texto y multimedia de acuerdo a la forma especificada o mejor interpretada, puesto a la diversidad de navegadores existentes, los cuales no son capaces de interpretar un mismo código de una manera unificada.

La evolución tan desordenada del HTML ha puesto toda una serie de inconvenientes y deficiencias, que para superarlos se han introducido otras tecnologías accesorias para organizar, optimizar, engalanar, y automatizar el funcionamiento de la web como CSS y JavaScript por citar algunas.

También es importante conocer las limitaciones de este lenguaje. HTML no es una herramienta para procesar texto, una solución de edición electrónica ni tampoco un lenguaje de programación. Su propósito fundamental es definir la estructura y apariencia básica de documentos y conjunto de documentos de tal manera que puedan ser manejados de forma rápida y fácil por un usuario en la red para verlo en diferentes dispositivos.

Sus normas las define un organismo sin ánimos de lucro denominado *World Wide Web Consortium*. Más allá del HTML el W3C tiene la completa responsabilidad de estandarizar cualquier tecnología relacionada con la WWW; administran el estándar de HTTP, lo mismo que lo relacionado con el direccionamiento de documentos en la Web.

1.2.3 XML

Ya que los documentos ricamente estructurados podrían ser utilizados a través de Internet y las únicas alternativas viables, HTML y SGML no eran prácticas para este fin. En 1998 el W3C estandariza por primera vez el XML (Lenguaje de Marcado Extensible del inglés *Extensible Markup Language*) partiendo de las especificaciones de SGML. [2]

XML no especifica ni la semántica ni un conjunto de etiquetas, de hecho es un meta-lenguaje para describir lenguajes de marcas. Proporciona un mecanismo para describir las etiquetas y las relaciones estructurales entre ellas.

Es válido aclarar que se pueden diferenciar dos tipos de documentos XML:

Bien formado: no tiene un DTD asociado pero sigue al pie de la letra todas las especificaciones del lenguaje.

Válido: además de estar bien formado sigue las reglas de un DTD específico.

1.2.4 XHTML

Desde la publicación de HTML 4.01 el 24 de diciembre de 1999, la actividad de estandarización de HTML se detuvo y el W3C se centró en el desarrollo del estándar XHTML. El nuevo lenguaje es una familia de tipos de documentos muy similar al lenguaje HTML, de hecho no es más que una adaptación de HTML al lenguaje XML [3]. En marzo de 2007, W3C retoma la actividad del estándar HTML.

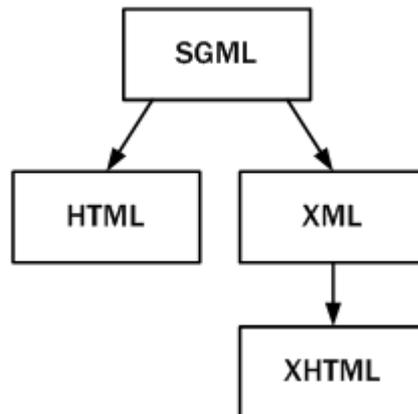


Figura 1: Esquema de evolución de HTML y XHTML.

1.2.5 DTD

La Definición de Tipo de Documento del inglés *Document Type Definition*, es una definición que especifica restricciones en la estructura y sintaxis de documentos XML o HTML. Indica que etiquetas son permitidas y el contenido que poseen, cuales contienen a otras y el orden que ocupan en el documento.

Todas las versiones de HTML estandarizadas tienen asociadas una DTD desde 1992 que se produce la primera Definición de Tipo de Documento llamada HTML1.0. El uso más frecuente que tiene es comprobar que un documento XML sea válido, su aplicación se extiende mucho más, por ejemplo un grupo independiente de personas pueden utilizar una DTD estándar para intercambiar datos y a su vez es una vía para verificar sus propios datos. La principal desventaja que tiene es que es poco flexible a la definición de elementos con contenido mixto y además que no es posible indicar que tipo de dato contiene el valor de un atributo o el texto de un elemento. [4]

1.3 Motores de Búsqueda

En la actualidad, cada vez es más creciente la necesidad de buscar información en Internet, la cual aumenta por minutos y se hace muy difícil el acceso; ya que existe una gran cantidad de documentos e información que son pocos conocidos por los usuarios, debido a que se encuentra dispersa por toda la red. Las páginas aparecen, cambian y desaparecen a un ritmo vertiginoso, de esta necesidad surgieron en Internet unas herramientas que localizan la información y recursos deseados. Estas

herramientas son los motores de búsquedas o también llamados buscadores, cuyo objetivo es organizar, localizar y facilitar el acceso a la información y recursos de la red.

Los buscadores son herramientas que permiten clasificar la información que existe en la red y hacerla localizable en poco tiempo según las preferencias del usuario. [5]

Los motores de búsqueda están formados por una **interfaz gráfica**, que es una página que se le muestra al usuario donde este especifica el término de búsqueda y se muestran los resultados donde aparece esta o estas palabras claves; un **robot** o también denominados “*Bots*”, “*Spiders*”, “*Web Wanderers*” o “*Web Crawlers*” que no es más que un programa que navega a partir de una o varias páginas web a través de sus hipervínculos extrayendo su contenido (texto generalmente) y almacenándolo en grandes bases de datos. Y por último las **bases de datos** que es donde se guarda lo que coleccionan los *Bots*.

Los buscadores pueden clasificarse en:

- **Buscadores jerárquicos:** También conocidos como Arañas o Spiders, requieren muchos recursos para su funcionamiento. La mayoría de los grandes buscadores de uso habitual y conocidos son de este tipo. Recorren las páginas recopilando información sobre sus contenidos. Cuando se busca una información en los motores, ellos consultan su base de datos y presentan resultados clasificados por su relevancia.[6]
- **Mixtos Buscador – Directorio:** Además de tener características de buscadores, presentan los sitios web registrados en catálogos sobre contenidos, por ejemplo informática, cultura, sociedad y a su vez se dividen en subsecciones.[6]
- **Multibuscadores:** Permite lanzar varias búsquedas en motores seleccionados respetando el formato original de los buscadores.[6]
- **Buscadores de Portal:** Bajo este título se engloban los buscadores específicos de sitio, aquellos que buscan información solo en su portal o sitio web, y podrían ser considerados como un directorio.[6]
- **Buscadores Verticales:** Buscadores especializados en un sector concreto, lo que les permite analizar la información con mayor profundidad, disponer de resultados más actualizados y ofrecer al usuario herramientas de búsqueda avanzadas.[6]

- **Metabuscadores:** Los metabuscadores son software que agregan los resultados de varios motores de búsqueda o directorios para localizar las páginas más relevantes sobre un tema concreto. No constan de base de datos propia y su dificultad radica en las diferencias existentes entre unos y otros: los métodos de combinación de buscadores, el orden de los resultados, etc.[6]

Los buscadores son herramientas muy utilizadas por los internautas. El mercado está ahora dominado por Google, Yahoo, Microsoft y otros pocos en menor medida. El resto de los grandes buscadores tienden a ser portales que muestran los resultados de otros buscadores y ofrecen además, otro tipo de contenidos que tienen mayor o menor importancia en la página. En el caso particular de Google, este responde a casi el 60% de las búsquedas en Internet [7]. La búsqueda de información en Internet, es una de las labores más comunes que realizan los usuarios. Por lo que el posicionamiento en los buscadores se ha convertido en los últimos años en el factor más relevante e importante en la estrategia de *marketing online*. Estar bien posicionados en los motores de búsqueda dominantes es tarea imprescindible para toda web que quiera triunfar en Internet.

1.3.1 Optimización para los Motores de Búsqueda

La acción de ajustar la información de las páginas para hacer que aparezcan en las primeras posiciones de los resultados de los buscadores se le denomina optimización para los motores de búsqueda o SEO. Tener un sitio web optimizado para los motores de búsqueda es una tarea ardua y compleja. Solamente aprender a hacerlo bien puede tomar mucho tiempo y dinero. El trabajo para la optimización es amplio ya que incluye al código de programación, el diseño y los contenidos.

Es importante conocer que no todos los buscadores aplican las mismas fórmulas o leyes para otorgarle mayor relevancia a una página web. De hecho cada buscador utiliza un algoritmo de indexación diferente que es permanentemente actualizado y mejorado. Estos algoritmos son considerados secretos industriales por las empresas que los crean. Esto hace que el posicionamiento en buscadores sea un auténtico ejercicio de ingeniería inversa. Ésta es la única forma de saber realmente qué valora cada buscador y tener las herramientas necesarias para optimizar la programación y el diseño de una página o un sitio web completo para su indexación.

El 50% del trabajo de posicionamiento se realiza sobre el código de la página, optimizándolo para que el motor de búsqueda lo destaque entre todas las páginas almacenadas en la base de datos. El otro

50% es más bien “marketing de red” cuyo objetivo es que el bot nos visite con cierta frecuencia y reporte en la base de datos lo importante que es nuestra página para otras muchas. [8]

Existen dos tipos de posicionamiento:

- **Posicionamiento Natural u Orgánico:** hace referencia a la construcción de un sitio. A la forma en que el mismo fue diseñado, desarrollado y a su capacidad de posicionarse naturalmente en los buscadores. Es decir, todo aquel posicionamiento que logra resultados sin invertir dinero en los buscadores.
- **Posicionamiento Patrocinado o SEM:** SEM son las siglas de *Search Engine Marketing* que en español sería marketing en buscadores. Esta técnica se ocupa de promocionar el sitio por medio de otros aspectos, como comprar palabras clave, enlaces patrocinados, anuncio en sitios web, entre otros. Es una forma de comprar posiciones en los buscadores.

Elementos que caracterizan una página web desde el punto de vista de un robot:

- **Información de cabecera:** en la etiqueta <HEAD> o la cabecera del documento, se encuentra toda la información que va a describir la información del documento. Dentro de ella alguna de las áreas más importantes son: la de título y meta-información.
- **Texto alternativo:** es una de las áreas más sensibles y menos cuidadas por los diseñadores.
- **Texto hiperenlazado:** es el conjunto de texto que aparece entre etiqueta <A>, en especial el atributo HREF que es donde se escribe las URLs a las que apunta la página.
- **Palabras claves o Keywords:** facilitan al buscador, información del documento, indicándole las palabras más importante por las que quiere ser encontrado.
- **Frecuencia:** es el número de veces que el término de búsqueda aparece en el área. Varía de acuerdo si el robot es sensible a mayúsculas y minúsculas -*case sensitive* en inglés-, o si la búsqueda es exacta o no.
- **Recuento:** cuenta el total de palabras en un área excluyendo la etiqueta <HTML>.
- **Peso:** es el porcentaje que el término de búsqueda representa sobre el total de palabras en el documento o el área analizada. Dado que el término de búsqueda puede estar formado por varias palabras, la fórmula general podría definirse como: $P = n * F / R$ donde n es el número

de palabras del término de búsqueda, R el recuento del área y F la frecuencia del término de búsqueda.

- **Relevancia:** la importancia que alguna información tiene para una persona o grupo de personas con el fin de cubrir una necesidad específica.
- **Estándares:** lo principal es que cumpla con los estándares definidos por la W3C en cuanto al código de la página en general.

1.4 Analizadores de XML, HTML, XHTML y herramientas SEO

Actualmente existen diversas herramientas dedicadas a ofrecer servicios de Posicionamiento Web, análisis de código HTML, XML y XHTML. A continuación se mencionan algunas herramientas más importantes que se usan en el mundo y se expone con una breve descripción sus principales características.

Xerces: es una potente biblioteca para el análisis, validación y manipulación de documentos XML. Actualmente está desarrollada en los lenguajes Java, C++, Perl. El Xerces está bajo la licencia *Apache Software Foundation* y es de código abierto. La versión 2.x fue implementada desde cero para lograr un código más limpio y organizado, además se presentó con un conjunto nuevo de interfaces internas, llamada XNI (*Xerces Native Interface*), para la creación de componentes y configuraciones de programa de análisis. La biblioteca Xerces a partir de la versión 2.x soporta los siguientes estándares y APIs (*Application Programming Interface*):

- XML 1.0 (4ta Edición)
- XML 1.1 (2da Edición)
- Espacios de nombres en XML 1.0 y 1.1 (2da Edición)
- Esquema XML 1.0 y 1.1 de la W3C
- Esquema XML de W3C lenguaje de definición (XSD)
- XInclude 1.0 (2da Edición)
- OASIS 1.1 Catálogos XML
- SAX 2.0.2

- JAXP 1.4
- StAX 1.0 API de eventos
- DOM Nivel 2 y 3
- Elemento Transversal

SAX (*Simple API for XML*): es una serie de clases e interfaces basada en eventos, para leer datos desde un documento XML. SAX fue la primera API XML en Java. Permite diseñar aplicaciones ligeras, proporciona un control flexible sobre como manipular los errores y fallas. La forma en que lee los documentos XML es unidireccional, se necesita comenzar a escanear el documento para releer un elemento determinado.

NekoHTML: es un analizador HTML, actualmente se encuentra en la versión v1.9.14, escrito en el lenguaje de programación java, utiliza para su funcionamiento el XNI, que le permite usar cualquiera de las herramientas XNI existentes sin necesidad de variar su código. El NekoHTML equilibra etiquetas que permiten acceder a la información utilizando interfaces estándares XML, al mismo tiempo puede corregir algunos errores en los documentos HTML.

HTMLparse: es una librería desarrollada en el lenguaje java, está realizada con el objetivo de analizar el HTML, es muy usado para la transformación y extracción de código HTML. En cuanto a la transformación, permite descargar las páginas web y guárdalas en el disco local, modificar los enlaces que se encuentran en las páginas, además realiza conversiones de las páginas HTML a XML. La extracción es utilizada para conservar la información más relevante de la página, el texto del cuerpo de la página, imágenes, sonidos y los enlaces validos contenidos en la página. Cuenta con una serie de filtros para la información y se pueden agregar etiquetas personalizadas.

Jericho HTML Parser: es una biblioteca java desarrollada a mediados del año 2002, de código abierto, distribuida con la EPL (Licencia Pública Eclipse) y la LGPL (GNU *Lesser General Public License*). Es una biblioteca que proporciona una serie de funcionalidades, entre los más destacados, análisis y manipulación de un documento HTML; reconoce etiquetas interpretadas por los servidores web, PHP, PSP, JSP, ASP y Manson; la estructura de la página web la forma en un árbol y permite cambios sin necesidad de generar un árbol nuevo. Elimina los espacios en blanco innecesarios en el código fuente HTML y además extrae todo el texto contenido en el documento HTML.

JTidy: es una interfaz Java para HTML Tidy, es un analizador de código HTML, elimina errores en los documentos, tales como: la ausencia de las etiquetas de cierre en los elementos, el desorden de las etiquetas de clausura, y agrega el carácter "/" a las etiquetas de cierre. También posibilita adicionar nuevas etiquetas.

TagSoup: es un analizador basado en SAX, implementado en el lenguaje de programación Java, es software libre y de código abierto. Desde la versión 1.2, está sobre lo efecto de la licencia *Apache License Versión 2.0*, que permite a propietarios la reutilización, así como su uso con licencia GPL 3.0 (*General Public License*). Realiza análisis de los documentos HTML muy rápido, sin embargo no está orientado a corregir del todo los errores que estos contienen. El TagSoup no funciona en su totalidad con Java 5 y Java 6.

HotSax: es analizador basado en SAX 2.0, desarrollado en el lenguaje de programación Java, software libre y de código abierto, capaz de analizar HTML, XML y XHTML. Es una herramienta muy sencilla que está desarrollada para crear otras aplicaciones como arañas y convertidores de HTML a otros formatos.

HtmlCleaner: es un analizador de código HTML escrito en Java, es de código abierto, la última versión fue liberada el 2 de septiembre del 2008. En ocasiones es difícil, tanto para las aplicaciones como para una persona con conocimiento básico sobre el lenguaje más propagado en la web, leer documentos HTML mal formados y con errores, es donde el HtmlCleaner juega su papel fundamental permitiendo reparar las sintaxis HTML, ordenando etiquetas, atributos y texto, produciendo XML bien formados. Además se puede hacer uso de reglas de filtrado para las etiquetas. Su funcionamiento no depende de librerías excepto las que tiene por defecto la máquina virtual de java.

Parser: una clase que se encuentra en el SDK (*Software Development Kit*) de Java (`javax.swing.text.html.parser.Parser`), para su funcionamiento requiere de un DTD en formato binario (BDTD), es un formato de archivo utilizado solo por la *Sun Microsystems* en esta implementación. Cuenta solo con un DTD, el de HTML 3.2, de encontrarse un DTD superior como el de HTML 4.01 tendría que realizarse ajustes en el analizador y transformarlo a BDTD. Es muy difícil realizar un BDTD para otras versiones del HTML.

Active WebTraffic: es una aplicación que contiene un conjunto de herramientas para optimizar páginas web y mejorar el Posicionamiento Web en los principales buscadores. Permite generar palabras claves o frases cortas de acuerdo al tema de la página web. Determina la frecuencia de las

palabras claves en el contenido de la web, analiza y valida el código HTML de una página y posee para mayor facilidad, un editor HTML. Crea y comprueba de forma automática etiquetas meta. Dispone de una herramienta de ranking capaz de verificar la posición por palabras claves en los buscadores más importantes. Comprueba los enlaces rotos o inexistentes. Además da de alta a páginas web en múltiples buscadores. La aplicación está a la venta y solo disponible para Windows.

SEO Studio: es una herramienta capaz de generar informes de ranking con ilimitadas palabras claves en los buscadores más importantes. Analiza palabras claves y determina su densidad en la página web. Permite encontrar sitios con el objetivo de compartir enlaces. Está disponible solo para el sistema operativo Windows.

Active Link Exchange: es una herramienta que ayuda a la obtención, gestión y mantenimiento de enlaces recíprocos. Permite conocer cuáles son los sitios más populares de un tema y extraer los enlaces de mayor Pagerank⁵ para garantizar un mayor tráfico a la página o sitio analizado a través del intercambio de direcciones. Los enlaces recíprocos se crean de forma automática para intercambiar con las páginas colaboradoras. Está disponible solamente para el sistema operativo Windows.

Cuwhois: es un sitio web que facilita una variedad de herramientas SEO en línea, determina el Pagerank de un sitio o una página web, comprueba los enlaces rotos, recíprocos y si las direcciones RSS (*Really Simple Syndication*) están bien colocadas en la página. Calcula la densidad de las palabras claves de una URL. Crea de forma automática etiquetas metas. Optimiza y valida código HTML y CSS (Cascading Style Sheets) con la aplicación en línea de la W3C.

Advanced Web Ranking: es una herramienta propietaria que su licencia tiene un elevado costo, se implementó en el lenguaje Java, permite buscar el ranking que tiene las palabras clave asociadas al tema del sitio o página web y compararlas con la de otros sitios. Brinda la posibilidad de hacer una investigación para seleccionar las palabras claves más convenientes para la página web y analiza las de otros sitios relacionados. Está disponible para diversos sistemas operativos como Linux, Windows y Mac OS.

⁵ PageRank es la importancia que tienen las páginas web en el buscador Google.

Tabla resumen:

Herramientas / Características	1	2	3	4	5	6	7	8	9	10	11
Xerces	X	X	X			X					X
SAX	-	X	X			X					X
NekoHTML	-	-		X		X					X
HTMLparse	-	-		X		X					X
Jericho HTML Parser	X	X		X		X					X
JTidy	-	-		X		X					X
TagSoup	X	X		X		X					X
HotSax	X	X	X	X	X	X					X
HtmlCleaner	X	X		X		X					X
Parser		X		X		X			X		X
Active WebTraffic				X		X	X		X		
SEO Studio						X	X				
Active Link Exchange	-					X	X				
Cuwhois		X		X			X			X	X
Advanced Web Ranking				X		X	X				X

Tabla 1: Analizadores de XML, HTML, XHTML y herramientas SEO.

Leyenda:

1. Código abierto.
2. Gratis o software libre.
3. Análisis XML.
4. Análisis HTML.
5. Análisis XHTML.
6. Aplicación de escritorio.
7. Análisis para SEO.
8. Valida XML con DTD.
9. Valida XHTML con DTD.
10. Valida HTML con DTD.
11. Multiplataforma.

Se han podido observar las diversas herramientas dedicadas a ofrecer servicios de Posicionamiento Web, análisis de código HTML, XML y XHTML. La mayoría de las aplicaciones analizadas, no realizan todas las tareas requeridas, solo dan la posibilidad de analizar XML o sencillamente intercambiar enlaces con otros sitios web para mejorar el posicionamiento. Son escasas las aplicaciones que brindan servicio de análisis, recomendación y reparación de errores HTML y XHTML y posicionamiento web, éstas pocas herramientas son en su mayoría propietarias o de servicio restringido. En el caso de aquellos programas que brindan consejos de optimización, mediante datos de entrada como son: título, palabras claves, meta y otros, muestran incoherencias e imprecisiones.

Con el desarrollo de un sistema que logre solucionar los problemas previamente mencionados se podría garantizar libre de costo un mejor posicionamiento de las páginas cubanas, además posibilitaría reunir las funcionalidades más importantes, brindadas por las herramientas similares en un solo sistema, dando una mayor importancia al análisis lexicográfico del contenido web.

1.5 Metodologías de Desarrollo

Una metodología de desarrollo de software es un conjunto de procedimientos y pasos a seguir para desarrollar un software. ¿Cómo dividir un proyecto?, ¿Qué restricciones deben aplicarse?, ¿Qué tareas se llevan a cabo en cada etapa?, son algunas de las preguntas a las que se le da respuesta en cada metodología y las diferencia de las demás.

1.5.1 RUP

Durante el proceso de desarrollo del software se utilizó la metodología RUP. Ya que es un proceso de desarrollo de software orientado a objetos, el cual proporciona un conjunto de actividades para transformar los requisitos del usuario en un sistema de software. Lo que define quién está haciendo qué, cuándo y cómo para alcanzar el objetivo esperado. Esta metodología pesada como también se le conoce se caracteriza fundamentalmente, por realizar mayor énfasis en el control y planificación del proyecto. Y además describe los requisitos y el modelo o modelado de los flujos de trabajo.

RUP es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, tipos de organizaciones y tamaños de proyectos aunque generalmente se utiliza en proyectos de gran magnitud.

Esta metodología utiliza UML como lenguaje de modelado para expresar gráficamente los esquemas del sistema. Iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura son algunas de sus principales características. Sus principales elementos son los trabajadores, actividades, artefactos y flujos de trabajo y consta de cuatro fases: Inicio, Elaboración, Construcción y Transición.

1.6 Lenguaje de Modelado

Para modelar el análisis y el diseño del software se utilizó UML (*Unified Modeling Language*), ya que permite la modelación de sistemas con tecnología orientada a objetos y está implícito en la metodología utilizada.

UML es un lenguaje de modelado que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software. Está respaldado por el OMG (*Object Management Group*), y su objetivo es lograr modelos que además de describir con cierto grado de formalismo los sistemas, puedan ser entendidos por los clientes o usuarios de aquello que se modela. Es importante resaltar que es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para

llegar a construirlo en términos concretos, en otras palabras indica que va hacer un sistema pero no como lo hará.

1.7 Herramienta CASE (*Computer-Aided Software Engineering*)

Es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo, su objetivo es acelerar el proceso para el que han sido diseñadas, en el caso de CASE para automatizar o apoyar una o más fases del ciclo de vida de vida del desarrollo del desarrollo de sistemas. [9]

1.7.1 Visual Paradigm

Como herramienta CASE se utilizó Visual Paradigm por ser una herramienta que ofrece un entorno de creación de diagramas para UML. Además facilita una mayor comunicación entre el equipo de desarrollo ya que utiliza un lenguaje estándar común. El diseño está centrado en casos de usos y enfocado al negocio lo que proporciona un desarrollo de aplicaciones con una mayor calidad y un menor coste.

Está disponible en múltiples plataformas, permite realizar todos los tipos de diagramas de clases y generación de código para varios lenguajes. Soporta el modelado colaborativo con CVS6 y Subversion. Además permite generar reportes y documentación en HTML/PDF.

1.8 Lenguaje Programación

1.8.1 Java

Para el desarrollo del software se utilizó Java como lenguaje de programación. Por ser un lenguaje orientado a objetos, muy potente y con una gama de facilidades para el desarrollo de aplicaciones de y para Internet. La principal característica es que Java es compilado a un código intermedio o bytecode, el cual es interpretado por una máquina virtual. Es la máquina virtual la que hace posible que aplicaciones desarrolladas en java puedan ser ejecutadas en diferentes sistemas y plataformas. A continuación se muestra un resumen de las características de java:

- **Orientado a objetos:** se diseñó desde un principio para ser orientado a objetos. Los primitivos tipos numéricos, carácter y booleano son las únicas excepciones. Una clase es la unidad básica de compilación y ejecución de Java; todos los programas de Java son clases.

⁶ Concurrent Versioning System : es una aplicación informática que implementa un sistema de control de versiones

- **Arquitectura neutral y portátil:** el compilador de Java genera bytecode para la máquina virtual de Java (*JVM-Java Virtual Machine*) en vez de código nativo de máquina. Como los bytecode de Java no dependen de la plataforma, los programas se pueden ejecutar en cualquier plataforma en la que esté instalada la máquina virtual.
- **Distribuido:** proporciona un soporte de alto nivel para redes. Las clases del paquete `java.net` hacen que la lectura de un archivo o fuente remota sea tan fácil como leer un archivo local. La principal razón por la que Java es distribuido es por sus características de portabilidad.
- **Robusto:** revisa la memoria, liberando y corrigiendo la corrupción de memoria. No utiliza punteros y promueve el uso de interfaces en lugar de clases.
- **Seguro:** ningún lenguaje o entorno es 100% seguro pero a pesar de ello, Java ofrece un nivel práctico de seguridad para la mayoría de las aplicaciones. Dada la naturaleza distribuida de Java, sin un aseguramiento de la seguridad no sería prudente ejecutar un código en el ordenador local proveniente de cualquier sitio de Internet. Una de las características más sobresalientes de su seguridad, es la firma digital incluida desde Java 1.1.
- **Alto rendimiento:** nunca será tan rápido como un lenguaje compilador. Pero puede convertir en tiempo de ejecución al código máquina específico del sistema o plataforma casi tan rápidamente como código compilado de forma nativa. Esta velocidad es más adecuada para ejecutar aplicaciones interactivas basadas en red y GUI (*Graphic User Interface*).
- **Multitarea y Multihilo:** ya que es una aplicación de red basada en GUI, es fácil imaginar que se llevan a cabo diferentes tareas o procesos al mismo tiempo. Java proporciona soporte de lenguaje incorporado para los hilos, lo que hace que la programación para hilos sea muy sencilla.

1.9 Reportes

1.9.1 Ireport v1.3.3

La herramienta iReport es un constructor/diseñador de informes visuales, poderoso, intuitivo y fácil de usar para JasperReport escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes y subinformes.

IReport está además integrado con JFreeChart, una de las bibliotecas gráficas de código abierto más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos incluso múltiples uniones JDBC, TableModels, JavaBeans y XML.

Características:

- Soporta JavaBeans como orígenes de datos (éstos deben implementar la interface JRDataSource).
- Maneja el 98% de las etiquetas de JasperReports.
- Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, subreports.
- Recopilador y exportador integrados.
- Soporta JDBC.
- Tiene asistentes para las plantillas.

1.10 IDE

Un Entorno de Desarrollo Integrado (*IDE-Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación como son: editor de código, compilador, depurador y constructor de interfaz gráfica (GUI). Los IDE proveen un marco de trabajo amigable y pueden soportar múltiples lenguajes. Lo que permite simplificar el desarrollo de una cualquier aplicación informática.

1.10.1 NetBeans

En el proyecto en cuestión se utilizó NetBeans v6.9 como entorno de desarrollo. Es un proyecto de código abierto de gran éxito y con una gran comunidad de usuarios y socios. Está escrito completamente en Java por lo que puede ejecutarse en cualquier ambiente que tenga la JVM instalada. Ofrece a los desarrolladores numerosas ventajas en el desarrollo de aplicaciones multiplataforma. Es un IDE multilenguaje completo y modular, tiene soporte para Java SE, Java EE, Java ME y con vinculación de gran cantidad de módulos (plugins). Permite desarrollar aplicaciones de escritorio, web y mobile (para móviles). Entre los lenguajes que soporta se encuentra: Java, C/C++, PHP, JavaScript, Python, Ruby on Rails, entre otros.

1.11 Conclusiones Parciales

En este capítulo se analizó el marco teórico conceptual y el estado del arte respecto a los lenguajes de marcado y el posicionamiento web. Se abordaron los principales conceptos para una mejor comprensión del problema a resolver. Se realizó un estudio de las herramientas SEO existentes así como analizadores de documentos XML, HTML y XHTML, detectando múltiples inconformidades en cada uno de ellos respecto a lo que se desea alcanzar. Como resultado de la investigación se definió RUP como metodología así como UML como lenguaje de modelado y Visual Paradigm como herramienta CASE a utilizar. Además de Java y NetBeans como lenguaje de programación y entorno de desarrollo respectivamente.

Capítulo 2: Características del Sistema

2.1 Introducción

En el presente capítulo se brinda una breve descripción de la solución propuesta, sus requisitos funcionales y no funcionales, así como los actores que intervienen en ella. Se presenta el diagrama de casos de uso del sistema, además de la descripción de los mismos.

2.2 Modelo de dominio

Debido a la poca estructuración de los procesos del negocio que tienen que ver con el objeto de estudio y para poder entender el contexto en que se emplaza el sistema se describe el funcionamiento de la aplicación mediante una serie de conceptos, entidades y sus relaciones, agrupándolos en un modelo de dominio.

El modelo de dominio es una representación de los conceptos u objetos del mundo real, significativos para un problema. Tiene como objetivo fundamental la descripción de las clases más importantes en el sistema y representa conceptos del mundo real.

2.2.1 Representación del modelo de dominio

A continuación se muestra el diagrama de clases del modelo de dominio del presente trabajo, donde se puede apreciar que el webmaster estudia una o varias páginas web, las cuales son analizadas por un analizador de estándares, definido por un DTD y un analizador SEO, el primero captura errores y advertencias, el segundo advertencias y tanto para los errores como para las advertencias de ambos analizadores sugiere una o varias posibles soluciones.

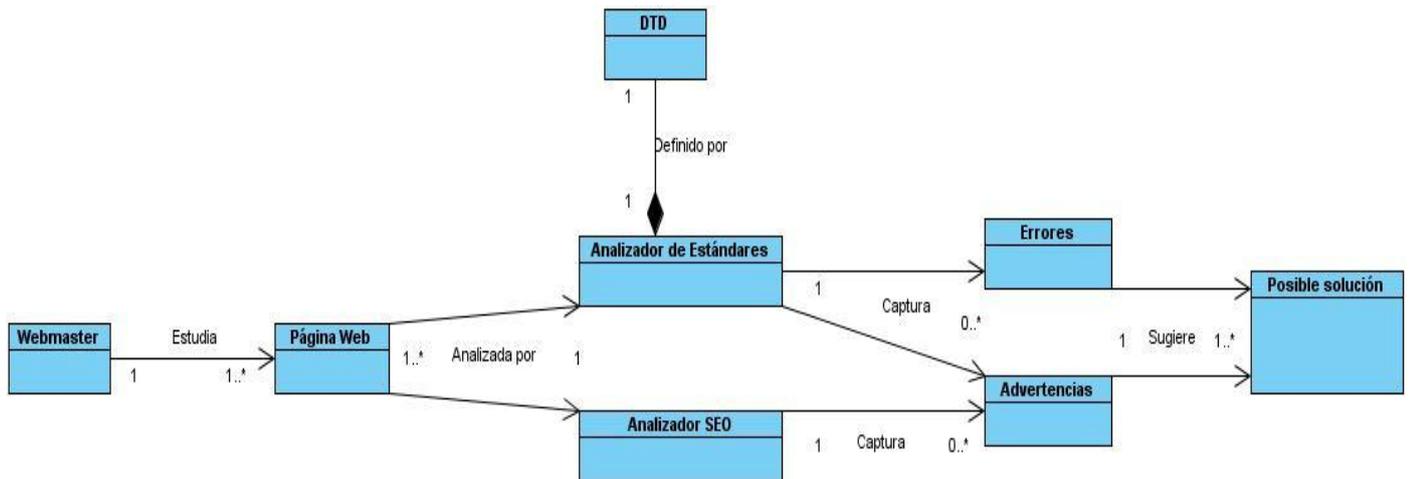


Figura 2: Diagrama de modelo de Dominio.

A continuación se describen cada una de las clases mostradas en el diagrama anterior:

- **Webmaster:** Administrador del sitio o página web.
- **Página Web:** Documento electrónico escrito en HTML o XHTML.
- **Analizador de Estándares:** Conjunto de funciones y clases que analizan la página web en cuanto a su estructura y composición.
- **DTD:** Fichero que especifica la estructura que debe tener la página web, es definido por la W3C.
- **Analizador SEO:** Conjunto de funciones y clases que analizan la página web en cuanto a optimización para los motores de búsqueda.
- **Errores:** Errores estructurales que posee la página de acuerdo a los estándares.
- **Advertencias:** Advertencias estructurales o desde el punto de vista SEO para mejorar la página web.
- **Posible Solución:** Sugerencia o información para corregir y mejorar los errores y advertencias.

2.3 Especificación de los requisitos del sistema

Los requisitos son condiciones o capacidades que necesita el usuario para resolver un problema o conseguir un objetivo determinado. Esta definición se extiende y se aplica a las condiciones que debe cumplir o poseer un sistema -o uno de sus componentes-, para satisfacer un contrato, una norma o una especificación.

2.3.1 Requisitos funcionales

Los requisitos funcionales de un sistema describen su funcionalidad, los servicios que de él se esperan, o los que proveerá, entre ellos: sus entradas, salidas y excepciones. A continuación se presentan los requisitos por casos de uso.

CU1: Configurar Proxy

RF1.1: Configurar el proxy del sistema.

CU2: Analizar página web

RF2.1: Descargar el código fuente de la página.

RF2.2: Analizar código fuente de la página.

RF2.3: Capturar errores y advertencia del documento.

RF2.4: Capturar errores y advertencia de etiquetas.

RF2.5: Optimizar la página para los motores de búsqueda.

RF2.6: Capturar advertencias para la optimización de los motores de búsqueda.

RF2.7: Mostrar errores y advertencia del documento.

RF2.8: Mostrar errores y advertencia de etiquetas.

RF2.9: Mostrar advertencias para la optimización de los motores de búsqueda.

RF2.10: Mostrar tamaño en memoria de la página.

RF2.11: Mostrar cantidad de etiquetas de cada tipo.

RF2.12: Calcular y mostrar la relevancia de cada palabra de la página.

RF2.13: Calcular y mostrar la densidad de cada palabra de la página.

RF2.14: Mostrar enlaces rotos.

RF2.15: Mostrar árbol de etiquetas.

CU3: Comparar páginas

RF3.1: Realizar comparaciones entre páginas.

CU4: Administrar metas

RF4.1: Mostrar Metas.

RF4.2: Modificar Metas.

CU5: Generar Reporte a PDF

RF5.1: Generar reporte en formato PDF.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable, por ejemplo, pudiera desearse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado. Normalmente están vinculados a requisitos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto.

Apariencia o interfaz externa:

La aplicación deberá tener una interfaz externa amigable, que sea sencilla y fácil de entender, para así evitar que el usuario se pierda dentro de la aplicación.

Software:

El sistema podrá ejecutarse en cualquier ambiente que posea la máquina virtual de java JDK1.6.

Hardware:

El sistema debe disponer de una tarjeta de red para la conexión a internet con una capacidad de 128 de RAM o superior.

Restricciones en el diseño y la implementación:

El diseño de la aplicación será orientado a objetos. Se garantizará la posibilidad de inclusión de nuevas funcionalidades y/o módulos de diseño para futuras iteraciones. Se utilizará Java como lenguaje de programación, NetBeans v6.9 como entorno de desarrollo y Visual Paradigm como herramienta de modelado.

Seguridad:

Se garantizará la confidencialidad del usuario y contraseña cuando se haga uso del proxy.

Usabilidad:

El sistema podrá ser accedido por cualquier webmaster o personal con conocimientos básicos en la programación web del lado del cliente y/o SEO.

Rendimiento:

El sistema deberá tener un tiempo de respuesta ante peticiones de cómo máximo 5.0 segundos después de haber descargado el código fuente de la página.

2.4 Definición de los casos de uso del sistema

Los casos de uso son descripciones de las funcionalidades del sistema y se utilizan para obtener información de cómo este debe trabajar. Independientemente de la implementación, describen, bajo la forma de acciones y reacciones, el comportamiento de un sistema desde el punto de vista del usuario.

2.4.1 Actores del sistema

Se le llama Actor a toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Esto incluye a los operadores humanos, pero también incluye a todos los sistemas externos, así como a entidades abstractas como el tiempo.

Actores	Descripción
Webmaster	Representa al usuario que va a hacer uso del sistema, y quien tiene la posibilidad de interactuar con todas las funcionalidades del mismo.

Tabla 2: Actores del sistema.

2.4.2 Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema sirve para especificar la comunicación y el comportamiento de un sistema, mediante su interacción con los usuarios y/u otros sistemas. A continuación se muestra el diagrama de casos de uso del presente trabajo.

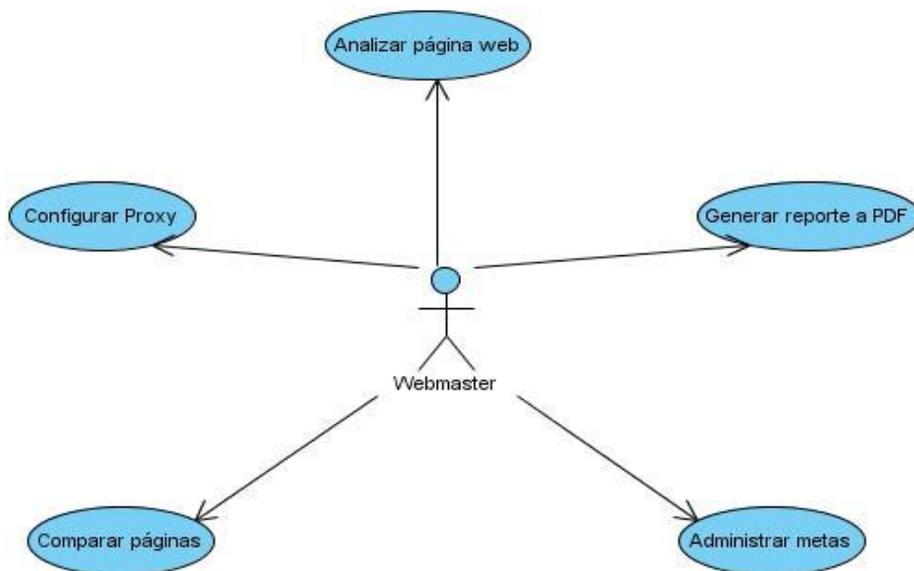


Figura 3: Diagrama de casos de uso del sistema.

2.4.3 Descripción de los casos de uso del sistema

Caso de Uso:	Configurar proxy	
Actores:	Webmaster	
Resumen:	El caso de uso se inicia cuando el webmaster selecciona la opción configurar proxy.	
Precondiciones:	Ninguna	
Referencias	RF1.1	
Prioridad	Alta	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El webmaster selecciona la opción configurar proxy.	1.1 El sistema muestra una ventana de diálogo donde el webmaster debe entrar los datos de configuración.	
2. El webmaster entra el usuario, contraseña, proxy, puerto y oprime el botón aceptar.	2.1 El sistema se configura para usar el proxy, puerto, usuario, clave y cierra el cuadro de diálogo.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
1 El webmaster oprime el botón cancelar	1.1 El sistema cierra el cuadro de diálogo correspondiente.	
Poscondiciones	El sistema queda configurado para poder acceder a sitios de internet.	

Tabla 3: Descripción del caso de uso Configurar Proxy.

Caso de Uso:	Analizar página web	
Actores:	Webmaster	
Resumen:	El caso de uso se inicia cuando el webmaster selecciona la opción analizar página web.	
Precondiciones:	Ninguna	
Referencias	RF2.1, RF2.2, RF2.3, RF2.4, RF2.5, RF2.6, RF2.7, RF2.8, RF2.9, RF2.10, RF2.11, RF2.12, RF2.13, RF2.14, RF2.15.	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección “Descargar el código fuente de la página”		
Acción del Actor	Respuesta del Sistema	
1. El webmaster selecciona la opción nuevo análisis.	1.1 El sistema muestra una ventana de diálogo solicitando la dirección de la página.	
2. El webmaster entra los datos requeridos, selecciona el DTD y oprime el botón analizar.	2.1 El sistema verifica que los datos entrados sean correctos. 2.2 Se conecta y descarga el código fuente de la página como texto.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	2.1 El sistema muestra un mensaje advirtiendo del error.	
	2.2 Si el sistema no puede conectarse, muestra un mensaje indicando el motivo.	
Sección “Analizar el código fuente de la página”		
Acción del Actor	Respuesta del Sistema	

CARACTERÍSTICAS DEL SISTEMA

	<p>1.1 El sistema lee el código fuente de la página y devuelve los tokens definidos.</p> <p>1.2 El sistema hace un análisis a partir de los tokens detectando errores y advertencias en la sintaxis de las etiquetas, y devuelve las etiquetas con sus respectivos atributos.</p> <p>1.3 El sistema analiza las etiquetas detectando errores semánticos en las mismas, construye y devuelve un documento que contiene todo el análisis del código fuente de la página incluyendo errores en el documento.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Sección "Optimizar la página para los motores de búsqueda"	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none">1. El sistema carga un fichero con las características y parámetros para el análisis SEO.2. El sistema analiza el documento y captura advertencias y sugiere consejos para una buena optimización de la página para los motores de búsqueda.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema

Sección "Mostrar resultados del análisis"	
Acción del Actor	Respuesta del Sistema
	<p>1.1 El sistema muestra los errores y las advertencias del documento encontrada en la sección "Análisis del código fuente de la página".</p> <p>1.2 El sistema muestra las advertencias capturadas en la sección "Optimización de los motores de búsqueda".</p> <p>1.3 El sistema muestra el tamaño en memoria de la página web analizada.</p> <p>1.4 El sistema muestra la cantidad de etiquetas de cada tipo.</p> <p>1.5 El sistema calcula la relevancia de cada palabra de la página analizada y la visualiza.</p> <p>1.6 El sistema calcula y muestra la densidad de cada palabra encontrada en la página.</p> <p>1.7 El sistema muestra los enlaces inaccesibles o rotos.</p> <p>1.8 El sistema visualiza un árbol de etiquetas a partir del documento.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema

Poscondiciones	
-----------------------	--

Tabla 4: Descripción del caso de uso Analizar Página Web.

Caso de Uso:	Comparar páginas	
Actores:	Webmaster	
Resumen:	El caso de uso se inicia cuando el webmaster selecciona la opción comparar página.	
Precondiciones:	El sistema debe haber analizado alguna página web	
Referencias	RF3.1	
Prioridad	Normal	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El webmaster selecciona la opción comparar página.	1.1 El sistema muestra: un listado de los proyectos creados recientemente, un Jtextfield para introducir la URL de la página sino se encuentra en la lista de proyectos, una lista de las páginas seleccionadas para comparar, un conjunto de etiquetas e indicadores por los cuales desea comparar y además una gráfica donde muestra el resultado de las comparaciones.
	2. El webmaster entra y selecciona los datos requeridos y presiona el botón Graficar.	2.1 El sistema verifica que no se repitan las páginas a comparar. 2.2 El sistema muestra en la gráfica los resultados de la comparación.
Flujos Alternos		

Acción del Actor	Respuesta del Sistema
	2.1 El sistema muestra un mensaje advirtiendo del error.
Poscondiciones	

Tabla 5: Descripción del caso de uso Comparar Páginas.

Caso de Uso:	Administrar metas	
Actores:	Webmaster	
Resumen:	El caso de uso se inicia cuando el webmaster selecciona la opción Mostrar o Modificar meta.	
Precondiciones:	El sistema debe haber analizado alguna página web	
Referencias	RF4.1, RF4.2	
Prioridad	Normal	
Flujo Normal de Eventos		
1. El webmaster selecciona una de las siguientes opciones: <ul style="list-style-type: none"> • Mostrar Meta • Modificar Meta 	1.1. El sistema ejecuta una de las siguientes secciones: <ul style="list-style-type: none"> • Si selecciona la opción Mostrar Meta ir a la sección Mostrar Meta. • Si selecciona la opción Modificar Meta ir a la sección Modificar Meta. 	
Sección "Mostrar Meta"		
Acción del Actor	Respuesta del Sistema	
1. El webmaster selecciona la opción mostrar metas.	1.1 El sistema muestra las etiquetas meta que posee la página web y muestra las	

CARACTERÍSTICAS DEL SISTEMA

	etiquetas metas que se pueden adicionar.
2. El webmaster elige la etiqueta que desea adicionar	2.1 El sistema solicita la información de la etiqueta seleccionada.
3. El webmaster entra la información solicitada y oprime el botón adicionar.	3.1 El sistema muestra la etiqueta meta con su estructura.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Sección "Modificar Meta"	
Acción del Actor	Respuesta del Sistema
1. El webmaster selecciona la opción modificar metas.	1.1 El sistema muestra las etiquetas meta que posee la página web y muestra las etiquetas metas que se pueden adicionar.
2. El webmaster selecciona una de las etiquetas que posee la página web.	2.1 El sistema solicita la información de la etiqueta seleccionada.
3. El webmaster entra la información solicitada y oprime el botón modificar.	3.1 El sistema muestra la etiqueta meta con su nueva estructura.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Poscondiciones	

Tabla 6: Descripción del caso de uso Administrar Metas.

Caso de Uso:	Generar Reporte a PDF
Actores:	Webmaster

Resumen:	El caso de uso se inicia cuando el webmaster oprime el icono del PDF.	
Precondiciones:	El sistema debe haber analizado alguna página web.	
Referencias	RF5.1	
Prioridad	Normal	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El webmaster hace clic en el icono del PDF.		1.1 El sistema muestra una ventana donde se representa como queda diseñado y estructurado el reporte.
2. El webmaster selecciona la opción de guardar.		2.1 El sistema guarda el reporte en el formato PDF.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
Poscondiciones		

Tabla 7: Descripción del caso de uso Generar Reporte a PDF.

2.5 Prototipo de interfaz de usuario

A continuación se muestran los prototipos de interfaz de usuario para cada caso de uso definido.

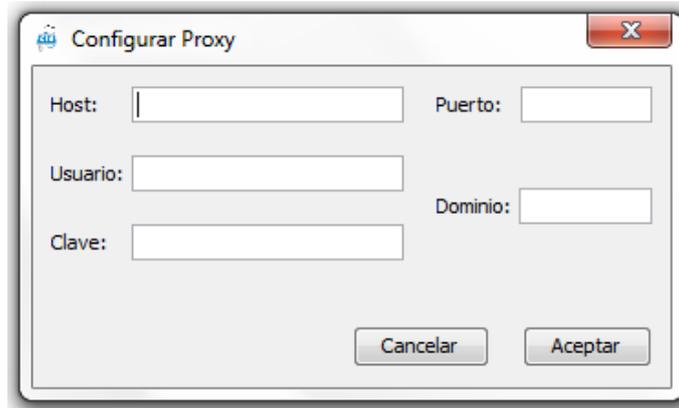


Figura 4: Prototipo de interfaz de usuario: Configurar Proxy.

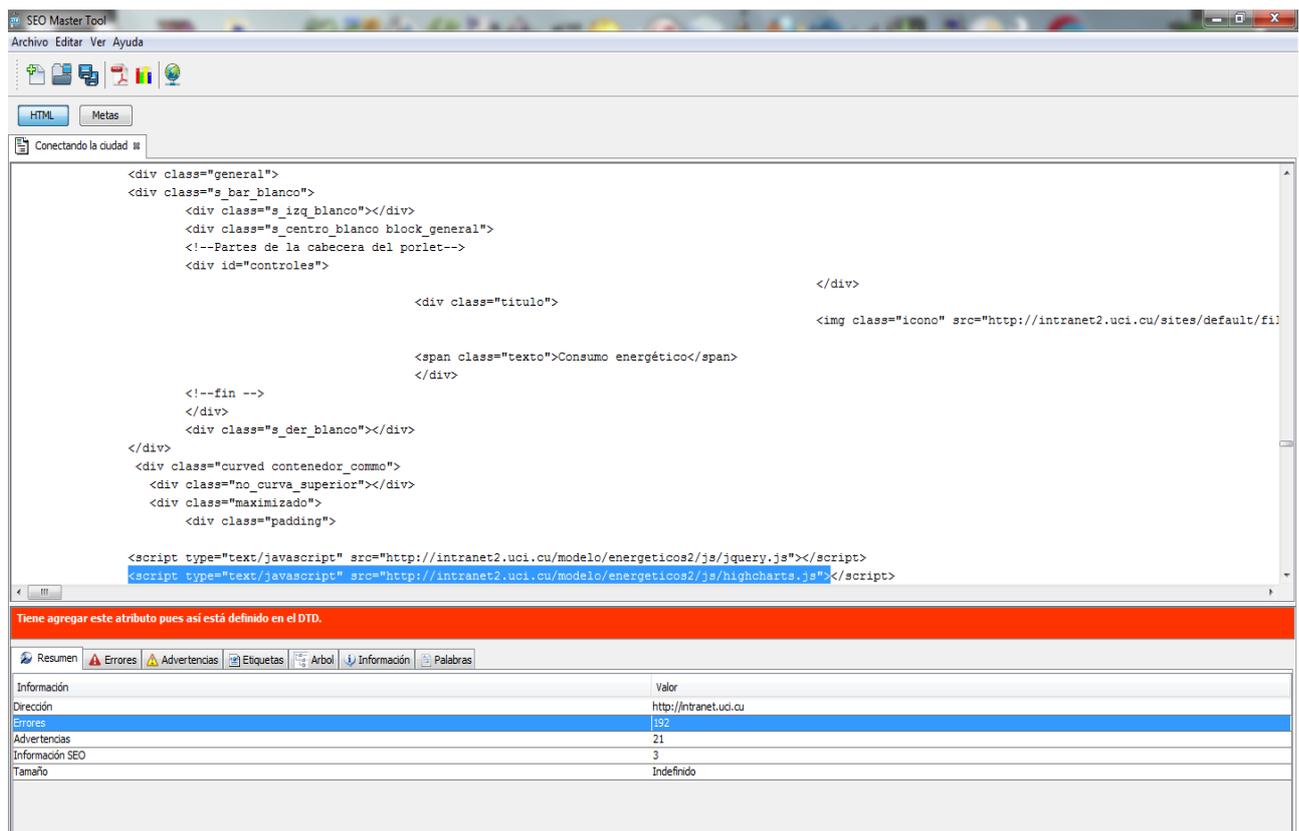


Figura 5: Prototipo de interfaz de usuario: Analizar Página.

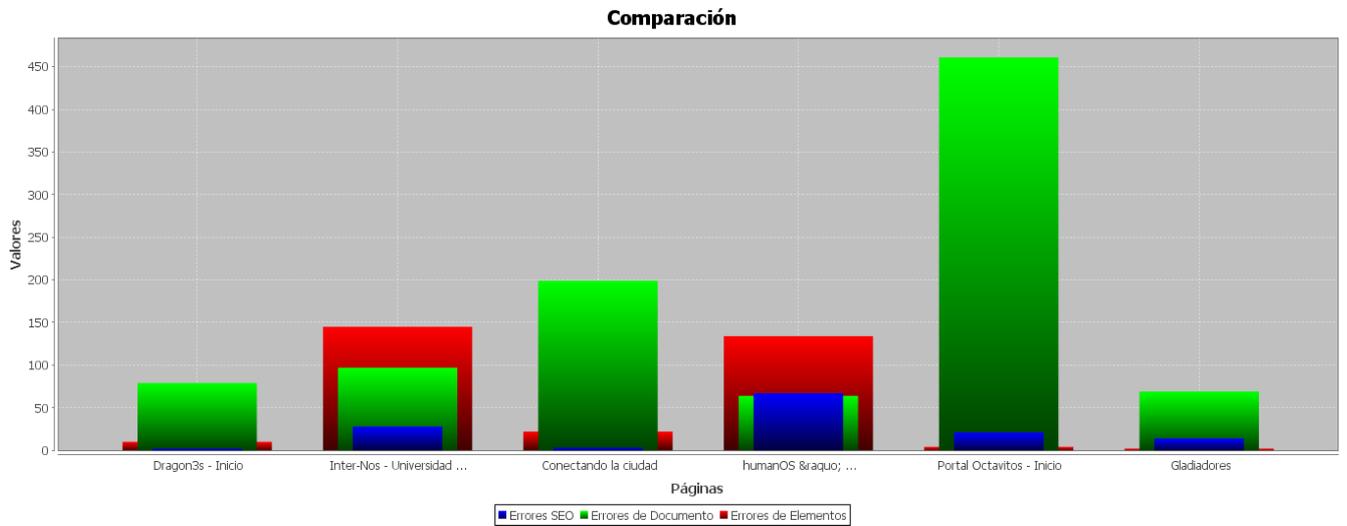


Figura 6: Prototipo de interfaz de usuario: Comparar Páginas.

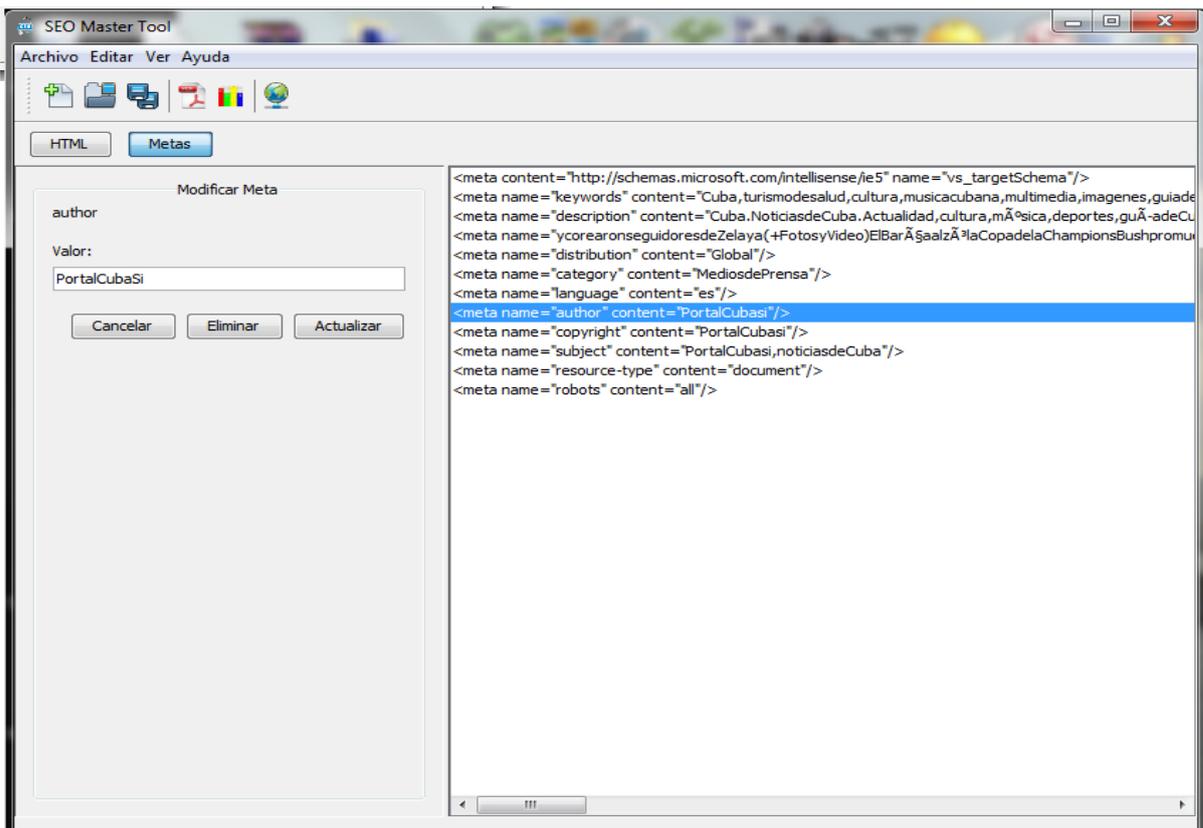


Figura 7: Prototipo de interfaz de usuario: Administrar Metas.

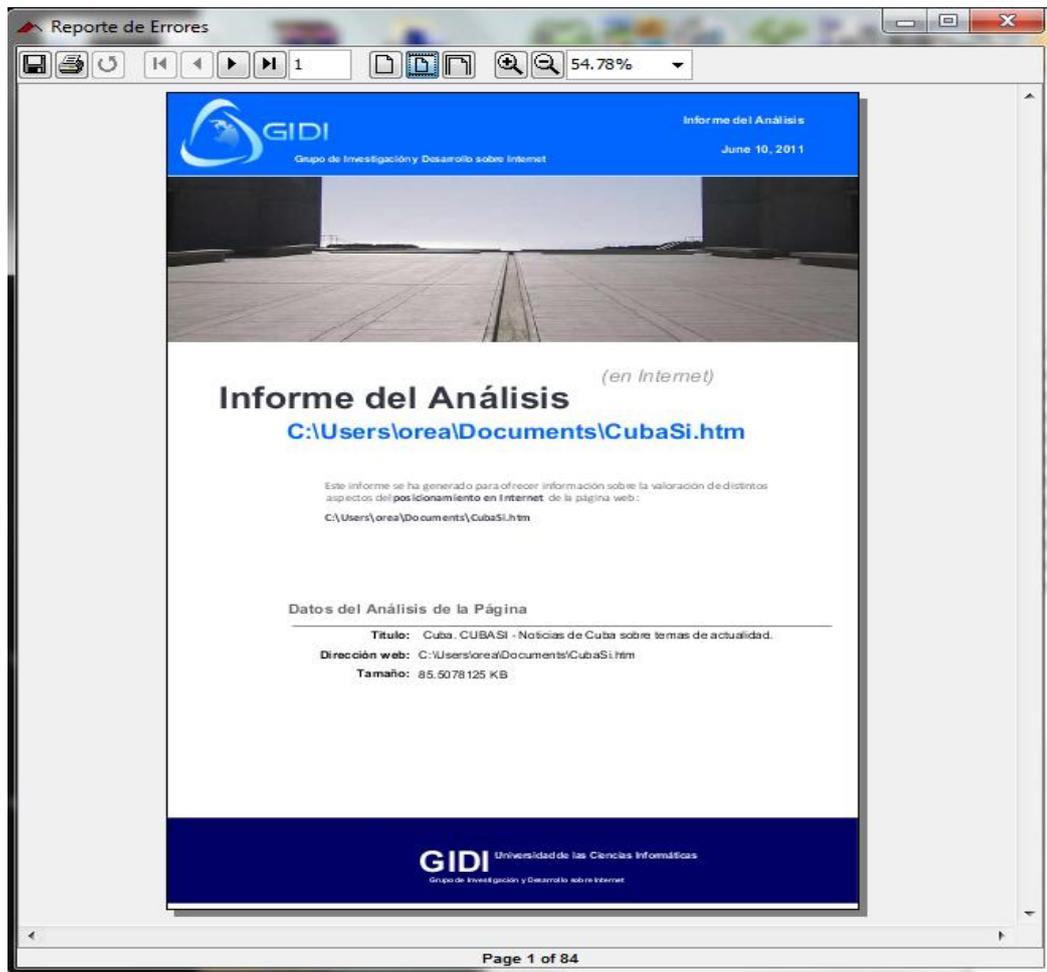


Figura 8: Prototipo de interfaz de usuario: Generar Reporte en PDF.

2.6 Conclusiones Parciales

En este capítulo, se brinda una clara definición de los requisitos que debe cumplir el sistema. Seleccionando al actor webmaster, así como su interacción con los cinco casos de uso del sistema, se ganó claridad en cuánto al sistema a construir. También se sentaron las bases para las restantes fases del proceso de diseño e implementación a través de la descripción de los casos de uso.

Capítulo 3: Diseño del Sistema

3.1 Introducción

En el presente capítulo se realizará un refinamiento de la descripción del sistema. Se dará una descripción del estilo arquitectónico empleado y los patrones de diseño utilizados para el desarrollo de la herramienta. Se realiza el diseño del sistema por medio de los distintos artefactos, propuestos por la metodología de desarrollo de software.

3.2 Patrón Arquitectónico Utilizado

Los patrones arquitectónicos describen los principios fundamentales de la arquitectura de un sistema de software. Estos identifican los subsistemas dentro de un sistema, centrándose en la comunicación y el acoplamiento de los mismos. Expresan un esquema organizativo estructural, de modo que establecen reglas y guías para organizar las relaciones entre ellos.

Garlan y Shaw definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. [10]

Para el desarrollo de la herramienta se utilizó el patrón Modelo Vista Controlador (*Model-View-Controller-MVC*). Clásico patrón de diseño utilizado para diseñar aplicaciones con sofisticadas interfaces, donde la lógica de interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio.

La aplicación se divide en tres capas lógicas:

Modelo: Almacena todos los datos del sistema. Representa específicamente el dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos. El modelo encapsula los datos y las funcionalidades.

Vista: Presenta el sistema al usuario de modo que interactúa con los diferentes tipos de usuarios comunicando y capturando información. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

Controlador: Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

Entre las principales ventajas que presenta este patrón arquitectónico se encuentran:

- Permite agregar múltiples representaciones de los datos o información.
- Facilita agregar nuevos tipos de datos según sea requerido por el sistema.
- Proporciona con mayor facilidad el mantenimiento del sistema en caso de errores.
- Ofrece formas más sencillas de probar el correcto funcionamiento del sistema.

3.3 Patrones de Diseño Utilizados

Un patrón de diseño describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces, sin hacerlo ni siquiera dos veces de la misma forma.[11]

Los patrones de diseño nombran, abstraen e identifican los aspectos clave de un diseño estructurado y común, que lo hace útil para la creación de diseños orientados a objetos reutilizables. Los patrones de diseño identifican las clases participantes y las instancias, sus papeles y colaboraciones, y la distribución de responsabilidades. Cada patrón de diseño se enfoca sobre un particular diseño orientado a objetos.

A continuación se muestran los patrones utilizados en la elaboración de la herramienta.

3.3.1 Patrón Creador

Este patrón es el encargado de que una clase B cree una instancia de una clase A, siempre que:

- La clase B contenga a la clase A.
- B sea una agregación (o composición) de A.
- B tenga los datos de inicialización de A (datos que requiere su constructor).
- B utilice A.

Se puede evidenciar este patrón en las clases Elemento, Documento, Controladora entre otras.

3.3.2 Patrón Bajo Acoplamiento

Este patrón asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades. El controlador no las realiza, las delega en otras clases con las que mantiene un modelo de alta cohesión. Fue aplicado en clases como `CodigoRecurso`, `Documento` y `Texto`.

3.3.3 Patrón Alta Cohesión

La cohesión es una medida de fuerza con las que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento de nuestro diseño debe realizar una labor única y auto identificable dentro del sistema; no desempeñada por otro de ellos. Una clase con baja cohesión hace muchas operaciones no relacionadas, o trabaja en demasía. Se puede apreciar la utilización en las clases `AnalizadorSEO`, `Controladora`, `Semántico`, `Sintáctico` y `Lector`.

3.3.4 Patrón Experto

Su objetivo es asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir determinada responsabilidad. Distribuye el comportamiento entre las clases que cuentan con la información requerida. `Controladora`, `Texto` y `AnalizadorSEO` son clases donde se evidencia la utilización de este patrón.

3.3.5 Patrón Observador

Permite a los objetos captar dinámicamente las dependencias entre objetos, de tal forma que un objeto notificará a los objetos dependientes de él cuando cambia su estado, siendo actualizados automáticamente. Encapsular estos aspectos en objetos diferentes permite variarlos y reutilizarlos. Se utilizó este patrón en las clases `Elemento`, `ValoresObservables`, `Sintáctico` y `Proyecto`.

3.3.6 Patrón Singleton

Garantiza que solamente se crea una instancia de la clase y provee un punto de acceso global a él. Todos los objetos que utilizan una instancia de esa clase usan la misma instancia. Las clases `Controladora`, `Proxy`, `ReporteErrores` y `FormularioPrincipal` demuestran el uso del patrón Singleton.

3.4 Modelo de Diseño

Es una abstracción del Modelo de Implementación y su código fuente, el cual se emplea fundamentalmente para representar y documentar su diseño. Se usa como entrada esencial en las actividades relacionadas con la implementación. Representa a los casos de uso en el dominio de la solución.

3.4.1 Diagramas de Clases del Diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema; también muestra sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, cuando se crea el diseño conceptual de la información que se manejará en el sistema, conjuntamente con los componentes que se encargarán del funcionamiento y la relaciones entre unos y otros.

- Diagrama de clases del diseño del caso de uso Administrar Metas. **(Ver Anexo 1)**
- Diagrama de clases del diseño del caso de uso Analizar Página Web. **(Ver Anexo 2)**
- Diagrama de clases del diseño del caso de uso Comparar Página Web. **(Ver Anexo 3)**
- Diagrama de clases del diseño del caso de uso Configurar Proxy. **(Ver Anexo 4)**
- Diagrama de clases del diseño del caso de uso Generar Reporte a PDF. **(Ver Anexo 5)**

3.4.2 Diagramas de Secuencia

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos.

- Diagrama de secuencia del caso de uso Administrar Metas. **(Ver Anexo 6 y 7)**
- Diagrama de secuencia del caso de uso Analizar Página Web. **(Ver Anexo 8)**
- Diagrama de secuencia del caso de uso Comparar Página Web. **(Ver Anexo 9)**
- Diagrama de secuencia del caso de uso Configurar Proxy. **(Ver Anexo 10)**

- Diagrama de secuencia del caso de uso Generar Reporte a PDF. **(Ver Anexo 11)**

3.4.3 Conclusiones Parciales

La descripción y análisis de la solución propuesta, y el estudio de los diferentes patrones facilitó la realización de los diferentes artefactos generados: clases de análisis, diseño y diagramas de interacción correspondiente al flujo de trabajo Diseño. También permitió crear una entrada apropiada y un punto de partida para actividades de implementación. Posibilitó ir construyendo una arquitectura sólida y robusta para el sistema utilizando el patrón arquitectónico MVC.

Capítulo 4: Implementación y Pruebas

4.1 Introducción

En el capítulo se describe cómo los elementos del modelo de diseño se implementan en términos de componentes, para ello se muestra el diagrama de componentes y los estándares de codificación. Además se especifican las pruebas realizadas al software con el objetivo de verificar que cumpla con los requerimientos y validar el correcto funcionamiento de la aplicación.

4.2 Diagrama de componentes

El objetivo principal del flujo de trabajo de implementación es convertir los elementos del diseño en elementos de implementación, los cuales pueden ser: códigos fuentes, ejecutables, binarios, entre otros. Entre los artefactos generados durante esta etapa se encuentra los diagramas de componentes los cuales muestran las organizaciones y dependencias lógicas entre componentes de software.

Desde el punto de vista del diagrama de componente, se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización y las restricciones impuestas por el lenguaje de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

A continuación se muestra el diagrama de componente de la aplicación desarrollada:

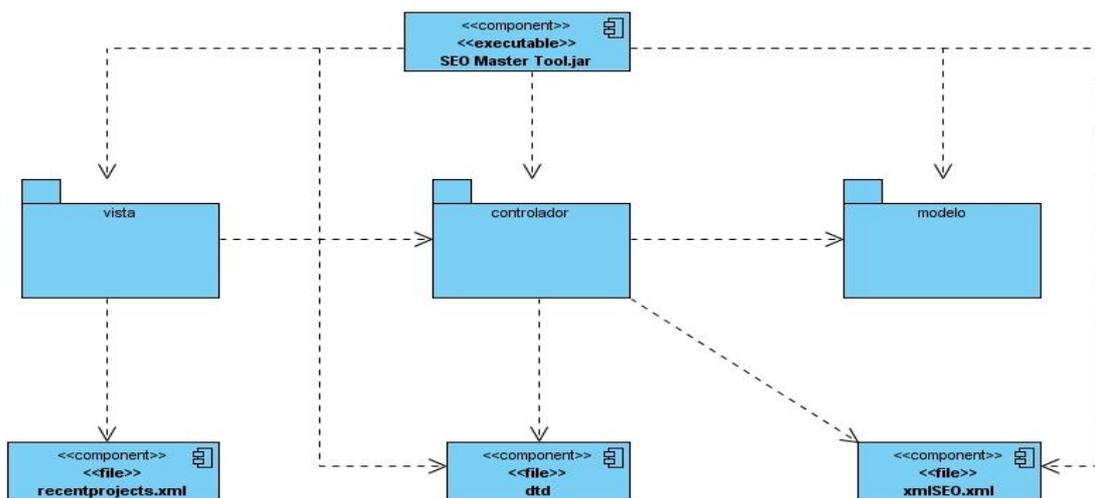


Figura 9: Diagrama de componentes.

Debido a lo extenso del diagrama se agruparon los componentes en paquetes, los cuales se corresponden a la arquitectura utilizada. A continuación se exponen los diagramas de componentes de los paquetes:

- Diagrama de componentes del paquete modelo. **(Ver Anexo 12)**
- Diagrama de componentes del paquete controlador. **(Ver Anexo 13)**
- Diagrama de componentes del paquete vista. **(Ver Anexo 14)**

4.3 Marcos de trabajos utilizados

Un marco de trabajo o framework es una mini arquitectura reusable que provee una estructura y comportamiento genéricos para una familia de abstracciones de software. [12]

Para el desarrollo de la aplicación se hace uso de una librería de clases denominada Matra la cuál provee funcionales muy importantes e imprescindibles.

4.3.1 Matra

El proyecto comenzó como un simple generador de árbol de DTD, pero para generar el árbol se necesitaba un programa que analizara el DTD. De esta forma surgió esta librería de clases que puede ser utilizada para:

- Analizar el DTD: analiza el DTD y comprueba si hay errores.
- Combinar DTD: muchos DTD se dividen en varios módulos, Matra puede crear un DTD que contiene todos los tipos de elementos y declaraciones de atributos en el DTD y todos sus módulos incluidos.
- Generar el Árbol de DTD: crea una representación visual del DTD donde todos los nodos del árbol es una representación visual del elemento.

Una vez analizado un DTD, permite conocer características minuciosas pero fundamentales de cada elemento atributo y entidades del DTD que reflejan como debe comportarse los documentos regidos por ellos. Como por ejemplo qué atributos tiene un elemento, si un elemento tiene hijos, que elementos se encuentran dentro de otros, entre otras características.

4.3.2 JasperReport

JasperReport es una poderosa herramienta de código abierto, se compone de un conjunto de librerías java para facilitar la generación de informes en aplicaciones tanto web como de escritorio. Permite incluir en los reportes: textos, imágenes, gráficos y hasta genera imágenes de fondo para utilizarlo como marcas de agua con el propósito de identificar el reporte o simplemente por motivos de seguridad.

Los reportes se pueden dividir en secciones opcionales que son: título del reporte, el encabezado de página, una sección para los detalles del reporte, el pie de página y una sección de resumen que aparece al final del reporte. Pueden generar sub-reportes que permiten la creación de reportes dentro de otro reporte, lo que facilita bastante el diseño.

Los reportes generados pueden ser exportados a una multitud de formatos como PDF, XLS, RTF, HTML, XML, TXT, jasper viewer, CVS (valores separados por coma) y texto plano. Los reportes son capaces de presentar los datos de manera textual o a través de gráficos: no sólo son capaces de mostrar los datos que le son pasados, sino que pueden generar o calcular con esos datos otros datos de forma dinámica y mostrarlos.

Sostiene la generación de gráficos estadísticos a través de la librería JFreeChart. En caso de querer generar datos directamente desde la aplicación, los gráficos pueden crearse independientemente como imágenes o generados por datos de entrada, incluso utilizando una de las numerosas librerías de código libre disponibles para la creación de gráficos. La imagen producida será mostrada usando un componente de imagen generado a través de la librería JFreeChart.

4.4 Librerías utilizadas

4.4.1 JFreeChart

JFreeChart es una biblioteca de gráficos de Java, que hace que sea fácil para los desarrolladores mostrar gráficos de calidad profesional en sus aplicaciones. Contiene un amplio conjunto de funciones entre las que se destacan:

- Una consistente y bien documentada API, que admite una amplia gama de tipos de gráficos.
- Un diseño flexible que es fácil de extender.

- Soporte para los tipos de salida, incluyendo los componentes Swing, archivos de imagen (incluyendo PNG y JPEG) y gráficos vectoriales en formato de archivo (incluyendo PDF, EPS y SVG).
- Es de código abierto, se distribuye bajo los términos de la GNU *Lesser General Public Licence* (LGPL), que permite su uso en aplicaciones propietarias.

4.4.2 JDOM

JDOM es una biblioteca de código abierto para manipulaciones de datos XML optimizados para Java, es simplemente, una representación Java de un documento XML. JDOM proporciona una manera de representar ese documento para la lectura fácil y eficiente, la manipulación, y la escritura. Dispone de un sencillo API, es ligero y rápido, y está optimizado para el programador Java.

4.5 Pruebas del sistema

Las pruebas de software son una 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. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

Un plan de pruebas está constituido por un conjunto de pruebas. Cada prueba debe dejar claro: qué tipo de propiedades se quieren probar (corrección, robustez, fiabilidad, amigabilidad); cómo se mide el resultado, especificar en qué consiste la prueba y definir cuál es el resultado que se espera. En lo adelante se abordara todo lo relacionado con las pruebas realizadas al sistema.

4.5.1 Configuración del entorno de pruebas

La configuración del entorno donde se vayan a ejecutar las diferentes pruebas que se realizan a un software, es un aspecto muy importante dentro del proceso de pruebas, pues si no se analizan bien los recursos de software y hardware que necesita el producto que se está construyendo, a la hora de probarlo se prescindirá de los elementos necesarios para la ejecución de un proceso de pruebas exitoso.

Por ello se tuvo en cuenta algunos requerimientos de hardware y de software que hicieron posible un mejor desarrollo de las pruebas, en aras de que se logaran minimizar los errores de la aplicación en

desarrollo. Los requerimientos que se consideraron necesarios para las pruebas se referencian a continuación:

Requerimientos de software:

- PC con Windows XP o superior.
- Máquina Virtual de Java 1.6.

Requerimientos de hardware:

- PC con Microprocesador Pentium IV o superior.
- 160 GB de Disco Duro o superior.
- 512 MB de memoria RAM o superior.

Fecha de Inicio	Fecha de Fin	Actividades	Personal Implicado
20/05/2011	20/05/2011	Aceptación y firma del Plan de Prueba.	Manuel Pérez Núñez. Yusley Orea Rodríguez. Eduardo Manuel Macías.
22/05/2011	22/05/2011	Verificación de las condiciones previas para el inicio de las pruebas.	Manuel Pérez Núñez. Yusley Orea Rodríguez. Eduardo Manuel Macías.

Tabla 8: Planificación de las Pruebas.

23/05/2011	23/05/2011	Ejecución de las pruebas de caja negra al caso de uso Configurar Proxy.	Manuel Pérez Núñez. Yusley Orea Rodríguez.
23/05/2011	25/05/2011	Ejecución de las pruebas de caja negra al caso de uso Analizar Página Web.	Manuel Pérez Núñez.

			Yusley Orea Rodríguez. Eduardo Manuel Macías.
26/05/2011	26/05/2011	Ejecución de las pruebas de caja negra al caso de uso Comparar Páginas.	Manuel Pérez Núñez. Yusley Orea Rodríguez.
26/05/2011	26/05/2011	Ejecución de las pruebas de caja negra al caso de uso Administrar Metas.	Manuel Pérez Núñez. Yusley Orea Rodríguez.
27/05/2011	27/05/2011	Ejecución de las pruebas de caja negra al caso de uso Generar Reporte a PDF.	Manuel Pérez Núñez. Yusley Orea Rodríguez.
28/05/2011	28/05/2011	Análisis de las no conformidades y las solicitudes de cambio.	Manuel Pérez Núñez. Yusley Orea Rodríguez. Eduardo Manuel Macías. Yulainne Alonso Hdez.

Tabla 9: Primera iteración de las Pruebas.

4.5.2 Diseño de las pruebas de Caja Negra

Las pruebas de caja negra son aquellas que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene.

Secciones a probar en los Casos de Uso:

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 1: Descargar	EC 1.1: Descargar código fuente de una página web	El webmaster selecciona la opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo.	Selecciona la opción nuevo análisis en la

código fuente.	remota.	Aparece una ventana donde el webmaster introduce la URL de la página, escoge el dtd para el análisis y oprime el botón Analizar. El sistema se conecta y descarga el código fuente de la página si es posible para el análisis.	barra de herramientas, o selecciona la opción en el menú Archivo.
	EC 1.2: Descargar código fuente de una página web local.	El webmaster selecciona la opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo. Aparece una ventana donde el webmaster, escoge la página en el ordenador y el dtd para el análisis y oprime el botón Analizar. El sistema lee el fichero y obtiene el código fuente del mismo para el análisis.	Selecciona la opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo.
SC 2: Analizar código fuente de la página.	EC 2.1: Analizar sintácticamente la página.	El webmaster selecciona la opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo. Aparece una ventana donde el webmaster indica la página que se va analizar y oprime el botón Analizar. El sistema a partir del código fuente de la página analiza la misma en busca de errores sintácticos.	Selecciona la opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo.
	EC 2.2: Analizar semánticamente la página.	El webmaster selecciona la opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo. Aparece una ventana donde el webmaster indica la página que se va analizar y oprime el botón Analizar. El sistema a partir del código fuente de la página analiza la misma en busca de errores semánticos.	Selecciona la opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo.
SC 3:	EC 3.1: Análisis de	El webmaster selecciona la opción nuevo	Selecciona la

Optimizar la página para los motores de búsqueda.	la página desde el punto de vista SEO.	análisis en la barra de herramientas, o selecciona la opción en el menú Archivo. Aparece una ventana donde el webmaster indica la página que se va analizar y oprime el botón Analizar. El sistema luego de haber analizado sintáctica y semánticamente la página. Realiza un análisis a partir de un fichero de configuración en busca de inconformidades.	opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo.
SC 4: Mostrar resultados del análisis.	EC 4.1: Mostrar resultados del análisis.	El webmaster selecciona la opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo. Aparece una ventana donde el webmaster indica la página que se va analizar y oprime el botón Analizar. El sistema luego de haber analizado completamente la página. Muestra los errores, advertencias, sugerencias SEO, árbol de etiquetas, cantidad de etiquetas de cada tipo, tamaño de la página, densidad y relevancia de cada palabra en el documento, así como los enlaces y si son válidos o no.	Selecciona la opción nuevo análisis en la barra de herramientas, o selecciona la opción en el menú Archivo.

Tabla 10: Diseño de las pruebas del caso de uso Analizar Página Web.

El diseño de las pruebas por cada sección, definidos anteriormente, se encuentran en el anexo 15, 16, 17 y 18.

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 1: Comparar páginas.	EC 1.1: Comparar páginas.	El webmaster oprime el botón comparar en la barra de herramientas, o selecciona la opción en el menú Ver. Aparece una ventana donde el webmaster selecciona	Oprime el botón comparar en la barra de herramientas, o

		las páginas a comparar y oprime el botón Aceptar. El sistema muestra en una gráfica, características de las páginas.	selecciona la opción en el menú Ver.
--	--	--	--------------------------------------

Tabla 11: Diseño de las pruebas del caso de uso Comparar Páginas.

El diseño de las pruebas por cada sección, definidos anteriormente, se encuentran en el anexo 19.

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 1: Adicionar Metas.	EC 1.1: Adicionar Metas.	El webmaster oprime el botón Metas. Aparece una interfaz donde se muestra las metas que posee el documento a la derecha, y otras que pudiera adicionar en la izquierda. El webmaster selecciona (en la izquierda) el nombre de la meta que desea adicionar al documento, introduce el valor de la misma y oprime el botón Adicionar. El sistema crea una nueva meta la cual es adicionada al documento y mostrada en el listado de metas (en la derecha).	Oprime el botón Metas.
SC 2: Modificar Metas.	EC 2.1: Modificar Metas	El webmaster oprime el botón Metas. Aparece una interfaz donde se muestra las metas que posee el documento a la derecha, y otras que pudiera adicionar en la izquierda. El webmaster selecciona (en la derecha) la meta que desea modificar. El sistema cambia la interfaz izquierda y brinda la posibilidad de introducir un nuevo valor de la meta seleccionada. El webmaster introduce el nuevo valor y oprime el botón Actualizar. El sistema cambia el valor de la meta y muestra la misma con su nuevo valor en el listado de	Oprime el botón Metas.

		metas (en la derecha).	
SC 3: Eliminar Metas.	EC 3.1: Eliminar Metas.	El webmaster oprime el botón Metas. Aparece una interfaz donde se muestra las metas que posee el documento a la derecha, y otras que pudiera adicionar en la izquierda. El webmaster selecciona (en la derecha) la meta que desea eliminar. El sistema cambia la interfaz izquierda y brinda la posibilidad de eliminar la meta seleccionada. El webmaster oprime el botón Eliminar. El sistema elimina la meta y actualiza el listado de metas (en la derecha).	Oprime el botón Metas.

Tabla 12: Diseño de las pruebas del caso de uso Administrar Metas.

El diseño de las pruebas por cada sección, definidos anteriormente, se encuentran en el anexo 20, 21 y 22.

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 1: Comparar páginas.	EC 1.1: Comparar páginas.	El webmaster oprime el botón exportar a PDF en la barra de herramientas, o selecciona la opción en el menú Ver. El sistema muestra un prototipo del PDF donde permite guardarlo. El webmaster oprime el botón guardar. El sistema muestra una ventana donde permite al webmaster escoger el destino del mismo e indicar un nombre al fichero. El webmaster oprime el botón guardar. El sistema guarda el fichero en el lugar indicado.	Oprime el botón exportar a PDF en la barra de herramientas, o selecciona la opción en el menú Ver.

Tabla 13: Diseño de las pruebas del caso de uso Generar Reporte a PDF.

El diseño de las pruebas por cada sección, definidos anteriormente, se encuentran en el anexo 23.

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
Configurar Proxy.	Configurar el proxy para acceder a internet.	El webmaster oprime el botón configurar proxy en la barra de herramientas, o selecciona en el menú Editar la opción Proxy. Aparece una ventana con las opciones de configuración, donde el usuario introduce la dirección del proxy, puerto, usuario, contraseña, dominio y oprime el botón aceptar. El sistema se configura para en caso de necesitar usar proxy utilizar los datos entrados por el usuario.	Oprime el botón configurar proxy en la barra de herramientas, o selecciona en el menú Editar la opción Proxy.

Tabla 14: Diseño de las pruebas del caso de uso Configurar Proxy.

El diseño de las pruebas por cada sección, definidos anteriormente, se encuentran en el anexo 24.

4.5.3 No Conformidades detectadas

Elemento	No	No Conformidad	Aspecto correspondiente	Etapas de detección	Significativo	No Significativo
Interfaz	1	No mostraba un mensaje de error.	Cuando se analizaba una dirección no válida.	Etapas de implementación y aplicación de pruebas al caso de uso Analizar Página	X	

IMPLEMENTACIÓN Y PRUEBAS

				Web.		
Interfaz	2	No cambiaba la interfaz del panel HTML al panel Metas.	Cuando se daba doble clic al botón Meta.	Etapa de implementación y aplicación de pruebas al caso de uso Analizar Página Web.	X	
Interfaz	3	No actualizaba las listas de la ventana Comparar páginas.	Cuando se daba clic al botón comparar de la barra de herramientas.	Etapa de implementación y aplicación de pruebas al caso de uso Comparar Página Web.	X	
Interfaz	4	No actualiza los campos: host, puerto, dominio, usuario y contraseña con los valores	Cuando se selecciona configurar proxy más de una vez.	Etapa de implementación y aplicación de pruebas al caso de uso		X

		anteriores.		Configurar Proxy.		
--	--	-------------	--	----------------------	--	--

Tabla 15: No Conformidades.

4.6 Conclusiones Parciales

En el capítulo se mostraron los resultados obtenidos durante el flujo de trabajo de implementación. Se realizó el diagrama de componentes con el objetivo de brindar una idea de cómo se implementó el software en término de componentes, dando una visión de cómo quedó construida y distribuida la aplicación. Se le realizaron pruebas de caja negra al sistema y se detectaron cuatro no conformidades, las cuales fueron registradas y solucionadas para un mejor funcionamiento de la aplicación.

Conclusiones Generales

Al concluir la investigación se cumplió con el objetivo general propuesto. El análisis de las tecnologías actuales y el estudio de los principales elementos del posicionamiento web permitieron consolidar una base para el desarrollo del sistema. Basado en la metodología de desarrollo RUP, a lo largo de todo el trabajo se transitó por los flujos de trabajo de análisis y diseño, implementación y pruebas, documentándose en cada flujo los principales artefactos necesarios para el desarrollo de la solución propuesta; logrando así desarrollar un analizador de XML, XHTML y HTML a partir de los estándares definidos por la W3C, que permita detectar errores de estructura y que atenten al buen posicionamiento de las páginas en los motores de búsqueda. Con la utilización de la aplicación en conjunto con otros factores que intervienen en el posicionamiento web se logrará mejorar el posicionamiento de las páginas cubanas en los motores de búsqueda.

Recomendaciones

Con el objetivo de lograr mejoras en el funcionamiento de la aplicación se recomienda:

- Profundizar en el estudio de las estrategias y técnicas de posicionamiento web, con vista a detectar posibles cambios o modificaciones en el algoritmo de posicionamiento de los diferentes buscadores.
- Continuar con el proceso de desarrollo del sistema con el fin de incorporarle nuevas funcionalidades para facilitar el trabajo del Webmaster.
 - ✓ Generar el fichero robots.txt que le restringe o permite a los spider de los motores de búsqueda, las páginas que deben o no ser indexadas.
 - ✓ Generar el mapa de navegación del sitio Web con el objetivo de indicarle a los buscadores donde se encuentran las páginas web y de esta forma facilitarle la indexación.
 - ✓ Validar los estilos de la página para determinar si el texto tiene el mismo color del fondo de la página para evitar posibles fraudes.
- Aplicar las pruebas de caja blanca al sistema.
- Actualizar las librerías utilizadas para la generación de los reportes, debido al mal funcionamiento de estas, en sistemas operativos donde no fueron diseñadas las plantillas.

Bibliografía Consultada

1. **Zuñiga Gómez, Graciela.** monografias.com. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://www.monografias.com/trabajos81/internet-trabajo/internet-trabajo.shtml>.
2. **Shaw, Mary y Clements, Paul.** A field guide to Boxology: Preliminary classification of architectural styles for software systems. Carnegie Mellon University. : s.n., 1999.
3. **Shaw, David Garlan y Mary.** *An introduction to software architecture.* s.l.: CMU Software Engineering Institute Technical Report, 1994. CMU/SEI-94-TR-21.
4. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado, Manual de Referencia.*
5. **Rodriguez Porensa, Nelson.** *Buscadores en la UCI.* Grupo de Investigaciones de Estudios de Internet, UCI. 2008.
6. **Roche, Conrad.** Matra. [En línea] [Citado el: 16 de mayo de 2010.] <http://matra.sourceforge.net>.
7. **Rebolledo, Julissa y Santacoloma, Andrea.** [En línea] [Citado el: 28 de noviembre de 2010.] www.angelfire.com/ultra2/pecanpie/.../Glosario.htm.
8. **Ramirez, Veronica.** monografias.com. [En línea] [Citado el: 21 de Noviembre de 2010.] <http://www.monografias.com/trabajos81/que-es-internet/que-es-internet.shtml>.
9. **Raaz, Ana.** monografias.com. [En línea] <http://www.monografias.com/trabajos81/introduccion-internet/introduccion-internet.shtml>.
10. **Montalvo, Marlene Melián.** *XML el nuevo lenguaje universal.*
11. **Miñarro, David Campoy.** Cuwhois. [En línea] 2010. [Citado el: 1 de Diciembre de 2010.] <http://www.cuwhois.com/>.
12. **Kaisler, Stephen. H.** *Software Paradigms.* 2005. ISBN 0-471-48347-8.
13. **Jericho.** Jericho HTML Parser. [En línea] [Citado el: 29 de noviembre de 2010.] <http://jericho.htmlparser.net/docs/index.html>.

14. **IVAR JACOBSON, G. B., JAMES RUMBAUGH.** El Proceso Unificado de Desarrollo de Software. [aut. libro] Addison Wesley Publishing Company. *El Proceso Unificado de Desarrollo de Software*. 2000.
15. **Howland, Ed.** The HotSAX project. [En línea] Junio de 2001. [Citado el: 28 de noviembre de 2010.] <http://hotsax.sourceforge.net/>.
16. **google.dirson.com.** Posicionamiento web en Google. [En línea] [Citado el: 25 de noviembre de 2010.] <http://google.dirson.com/posicionamiento.net/>.
17. **Fresno Fernández, Víctor.** Lenguajes de marcado.
18. **Eguíluz Pérez, Javier.** *Introducción a XHTML*. 2008.
19. **Pfravioli.** monografias.com. [En línea] <http://www.monografias.com/trabajos7/html/html.shtml>.
20. **Javier Francisco y Martínez, Juan.** *Guía de construcción de software en Java con patrones de diseño*. Oviedo : s.n.
21. *Recuperación de información en Internet: Google, un buscador singular.* **García de León y Alicia Ma.** 3, Diciembre de 2001, Revistas Ciencias de la Información, Vol. 32. ISBN 959-234-032-3.
22. **David de Ugarte.** *El Libro del posicionamiento en buscadores*. Madrid : s.n., 2002.
23. **Chuck Musciano y Kenedy, Bill.** *HTML La guía completa*. Segunda Edición. s.l. : McGRAW-HILL INTERAMERICANA, 1999. ISBN-1-56592-235.
24. **Christopher Alexander, y otros.** *A pattern Language: Towns/Building/Construction*. s.l. : Oxford University Press, 1977.
25. **Andy Clark y Marc Guillemot.** CyberNeko HTML Parser. [En línea] 2009. [Citado el: 28 de noviembre de 2010.] <http://nekohtml.sourceforge.net/>.
26. **W3C.** *XHTML™ 1.0 The Extensible HyperText Markup*. Segunda Edición. 2002.
27. **Caphyon Ltd.** Web Ranking Software. [En línea] 2010. [Citado el: 3 de Diciembre de 2010.] <http://www.advancedwebranking.com/buy-website-ranking.html>.

28. **TrendMetrix Software Inc.** TrendMetrix Software. [En línea] 2010. [Citado el: 2 de Diciembre de 2010.] <http://www.trendmx.com/website-promotion-optimization-software/website-promotion-download.aspx>.
29. TagSoup. [En línea] [Citado el: 28 de noviembre de 2010.] <http://mercury.ccil.org/~cowan/XML/tagsoup/>.
30. Scitation. [En línea] [Citado el: 3 de Diciembre de 2010.] <http://scitation.aip.org/vsearch/servlet/VerityServlet?KEY=FREESR&smode=results&maxdisp=10&possible1=XML&possible1zone=article&OUTLOG=NO&viewabs=CCEJEL&key=DISPLAY&docID=5&page=0&chapter=0>.
31. **DeltaOrion.** Posicionamiento Web. [En línea] [Citado el: 2 de Diciembre de 2010.] <http://www.posicionamientow3.com.ar/posicionamiento-natural.php>.
32. Osmosis Latina. [En línea] 2008. [Citado el: 26 de Noviembre de 2010.] <http://tecncliente.osmosislatina.com/curso/html.htm>.
33. Myrosoft-activelinkexchange. [En línea] 2010. [Citado el: 1 de Diciembre de 2010.] <http://www.myrosoft.com/activelinkexchange/activelinkexchangesp.htm>.
34. Myrosoft. [En línea] 2010. [Citado el: 2 de Diciembre de 2010.] <http://www.myrosoft.com/activewebtraffic/activewebtrafficsp.htm>.
35. Jtidy-Jtidy. [En línea] [Citado el: 27 de noviembre de 2010.] <http://jtidy.sourceforge.net/>.
36. JDOM. [En línea] [Citado el: 12 de Febrero de 2011.] <http://www.jdom.org/docs/faq.html>.
37. Htmlcleaner. [En línea] 2006. [Citado el: 27 de noviembre de 2010.] <http://htmlcleaner.sourceforge.net/>.
38. HTML Parser. [En línea] 2006. [Citado el: 27 de noviembre de 2010.] <http://htmlparser.sourceforge.net/>.
39. **Google Inc.** Google. [En línea] 2010. [Citado el: 5 de Diciembre de 2010.] <http://www.google.com/intl/es/corporate/tech.html>.
40. Estrategias para Máximo Posicionamiento Web. [En línea] [Citado el: 2 de Diciembre de 2010.] <http://www.maximoposicionamientoweb.com/>.

41. Encyclopedia Britannica. [En línea] [Citado el: 3 de Diciembre de 2010.] <http://www.britannica.com/>.
42. Document Type Definiton (DTD). [En línea] [Citado el: 3 de Diciembre de 2010.] <http://www.ulpgc.es/otros/tutoriales/xml/DTD.html>.
43. **The Apache Software Foundation.** Apache Xerces. [En línea] 2010. [Citado el: 26 de noviembre de 2010.] <http://xerces.apache.org/xerces-j/>.
44. **SlideShare Inc.** [En línea] [Citado el: 23 de noviembre de 2010.] <http://www.slideshare.net/Giorlysole/herramientas-case-5004699>.

Bibliografía Citada

1. **Chuck Musciano y Kenedy, Bill.** *HTML La guía completa.* Segunda Edición. s.l. : McGRAW-HILL INTERAMERICANA, 1999. ISBN-1-56592-235.
2. **Montalvo, Marlene Melián.** *XML el nuevo lenguaje universal.*
3. **Eguíluz Pérez, Javier.** *Introducción a XHTML.* 2008.
4. **W3C.** *XHTML™ 1.0 The Extensible HyperText Markup.* Segunda Edición. 2002.
5. **Rebolledo, Julissa y Santacoloma, Andrea.** [En línea] [Citado el: 28 de noviembre de 2010.] www.angelfire.com/ultra2/pecanpie/.../Glosario.htm.
6. **Rodríguez Porensa, Nelson.** *Buscadores en la UCI.* Grupo de Investigaciones de Estudios de Internet, UCI. 2008.
7. **google.dirson.com.** Posicionamiento web en Google. [En línea] [Citado el: 25 de noviembre de 2010.] <http://google.dirson.com/posicionamiento.net/>.
8. **David de Ugarte.** *El Libro del posicionamiento en buscadores.* Madrid : s.n., 2002.
9. **SlideShare Inc.** [En línea] [Citado el: 23 de noviembre de 2010.] <http://www.slideshare.net/Giorlysole/herramientas-case-5004699>.
10. **Shaw, David Garlan y Mary.** *An introduction to software architecture.* s.l. : CMU Software Engineering Institute Technical Report, 1994. CMU/SEI-94-TR-21.
11. **Christopher Alexander, y otros.** *A pattern Language: Towns/Building/Construction.* s.l. : Oxford University Press, 1977.
12. **Kaisler, Stephen. H.** *Software Paradigms.* 2005. ISBN 0-471-48347-8.

Glosario de Términos

Enlace: Conocido como link e hipervínculo. Es una ruta virtual que conecta a un documento (o página web) con otro documento dentro de una misma red o dentro de un mismo computador.

Etiqueta: Son los símbolos básicos del HTML (< >). Se conocen también por el nombre “tag”, e “identificador”.

HTML: *HyperText Markup Language*, lenguaje utilizado para la escritura y creación de páginas web.

Hipertexto: Conjunto de documentos que contienen vínculos a otros documentos. Un ejemplo muy ilustrativo puede ser una página web. En ella hay enlaces a otros documentos que se muestran con tan sólo hacer un clic con el ratón.

Metatags: Es una etiqueta o tag oculta que se coloca en el apartado de una página web y contiene información sobre esa página, pero no se puede visualizar dentro de la ventana del navegador.

Navegador web: Del inglés, web browser, es un programa que permite visualizar la información que contiene una página web.

PageRank: Parámetro que mide la importancia que tiene una página para un buscador. Es una medida de la popularidad de los sitios web.

Parámetro: Valor que se le pasa a una función que variará su comportamiento.

SEO: Término en inglés *Search Engine Optimization* que en español significa Optimización para Motores de Búsqueda.

Servidor web: Es un programa que implementa el protocolo http. Se ejecuta continuamente en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

URL: Localizador Uniforme de Recursos (*Uniform Resource Locator*.) Es la dirección única y global de un documento o página web.

Webmaster: Es una palabra de origen Inglés, que traducido al español es “Maestro Web”. Es la persona responsable de un sitio web.