



Universidad de las Ciencias Informáticas

Facultad 1

Título: Herramienta informática para análisis de posicionamiento web.

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

Autor: Alexander Leonard Cárdenas.

Tutores: Ing. Yulio Aleman Jimenez.

Ing. Adyanna Pileta Gammalame.

La Habana, junio 2017

“Año 59 de la Revolución”

Declaración de autoría

Yo, Alexander Leonard Cárdenas, con carnet de identidad 92100929603 declaro ser el autor del presente Trabajo de Diploma titulado “Herramienta informática para análisis de posicionamiento web” y autorizo exclusivamente a la Facultad 1 de la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

Para que así conste firmo la presente declaración jurada de autoría en La Habana a los _____ días del mes de _____ del año _____.

Alexander Leonard Cárdenas

Ing. Adyanna Pileta Gammalame

Ing. Yulio Aleman Jimenez



"Y si alguna de las cosas que decimos las explota el enemigo y nos producen profunda vergüenza, ¡¡bienvenida sea la vergüenza!!... ¡¡bienvenida sea la pena!!, si sabemos convertir la vergüenza en fuerza, si sabemos convertir la vergüenza en espíritu de trabajo, si sabemos convertir la vergüenza en dignidad, si sabemos convertir la vergüenza en moral."

A handwritten signature in black ink, which appears to be 'Fidel Castro'. The signature is stylized and written in a cursive script. It is positioned below the quote and is underlined with a single horizontal stroke.

Comandante Fidel Castro Ruz

Dedicatoria

A mi madre:

por todos estos años de sacrificio y dedicación para que yo haya podido tener y ser todo lo que hasta hoy he logrado.

A mi padre:

por todo el apoyo en todos los aspectos de la vida, por ser ejemplo a seguir a lo largo de todos estos años.

A mi hermano:

por estar presente en todo momento.

Al rafa:

Por ayudarme en los momentos críticos de la tesis.

Agradecimientos

A mis padres, por su confianza y apoyo incondicional que siempre me han brindado para lograr mis metas y cumplir mis sueños.

A mi pareja por todo su amor incondicional.

A mis tutores Julio y Adyana, que siempre me han ayudado mucho y toda la paciencia que han tenido conmigo, dos excelentes personas las cuales considero las mejores tutoras que a alguien le puede tocar.

A las personas que durante todo este tiempo han estado presente durante todos estos años. A mis queridísimas Arlene, Anabel, Stephany, Rachel, Aniestsy, Leidys, Zuleimis por todo su cariño durante todos estos años. A mis hermanos de la UCI y a partir de ahora de la vida; Adrian, Feito, Erick, Elío, Asney, Yordan, Gustavo y Ernesto que, aunque ya no está presente todavía se le recuerda.

A mis amigos de la infancia: Jessica y Pototo por ayudarme en todo lo que han podido.

A mis profesores, sobre todo a Yari por todo su cariño durante todo este tiempo.

Resumen

El constante aumento del cúmulo de información en formato digital situada de forma pública en las redes informáticas alrededor del mundo, ha propiciado la dificultad de los usuarios de encontrar lo que realmente necesitan en un momento dado. Para dar respuesta a la necesidad de localizar oportunamente la información requerida, se diseñaron los Sistemas de Recuperación de Información capaces de indexar grandes cantidades de datos y ofrecerlos al público de una manera rápida y concisa. En la Universidad de las Ciencias Informáticas (UCI) y específicamente en el Centro de Ideoinformática se desarrollan sitios web de múltiples instituciones, pero dada la naturaleza de internet, los resultados de ese trabajo solo son posibles de apreciar después de cierto período de tiempo, una vez desplegados físicamente de cara a Internet, por lo que la presente investigación permitió desarrollar una herramienta informática de análisis de posicionamiento web que ayude al centro al diagnóstico temprano de problemas de posicionamiento de los portales web. Para la implementación de la propuesta de solución, guiado por la metodología AUP en su versión para la UCI se seleccionaron como principales tecnologías: el marco de trabajo Springs 2.2 para la programación en JAVA 8.0 y Visual Paradigm 8.0 como herramienta para el modelado. La herramienta informática de análisis de posicionamiento web implementada posee un conjunto de características y funcionalidades que contribuyen, mediante la disponibilidad de interfaces web, al diagnóstico SEO de las principales métricas internacionales que anteriormente era difícil de obtener.

Palabras clave: posicionamiento web, SEO, métricas, herramienta informática.

Índice

Introducción	1
Capítulo 1. Fundamentación teórica.	5
Introducción.	5
1.2 Aspectos fundamentales para entender el surgimiento del posicionamiento.	5
1.3 Herramientas SEO	10
1.4 Tecnologías y herramientas a utilizar	13
1.4 Lenguaje de modelado	25
1.5 Metodología de desarrollo	25
1.6 Marco de trabajo	26
1.7 Conclusiones parciales	26
Capítulo 2. Análisis y diseño de la propuesta de solución.	28
Introducción	28
2.2 Modelado del Dominio	28
2.3 Especificación de los Requisitos del Software	30
2.4 Propuesta del sistema	32
2.5 Modelo de Casos de Uso del Sistema	32
2.6 Estilo arquitectónico	36
2.7 Patrones de diseño	36
2.8 Diagrama de Interacción	38
Conclusiones parciales	39
Capítulo 3. Implementación y validación del sistema.	40
Introducción	40
3.1 Modelo de componentes que integran la solución informática	40
3.2 Estándares de codificación	41
3.3 Principales pantallas de la herramienta informática de análisis de posicionamiento web	42
3.4 Validación de la herramienta de análisis de posicionamiento web implementada	44
3.5 Conclusiones parciales	50
Conclusiones generales	51
Recomendaciones	52
Referencias bibliográficas	53
Bibliografía	54

Anexos	56
Anexo #1	56

Índice de Tablas

<i>Tabla 1: Operacionalización de las variables de la hipótesis (Elaboración propia)</i>	3
<i>Tabla 2: Descripción de las clases del modelo del dominio (Elaboración propia)</i>	28
<i>Tabla 3: Descripción del CU "Registrar nueva url"</i>	35
<i>Tabla 4: Fragmento del caso de prueba basado en el CU "Registrar nueva url"</i>	45
<i>Tabla 5: Variables empleadas en el diseño del caso de prueba basado en el CU "Registrar nueva url"</i>	46
<i>Tabla 6: Cantidad de no conformidades por cada iteración de las pruebas funcionales</i>	46
<i>Tabla 7: Resultados de prueba de usabilidad utilizando lista de chequeo</i>	47
<i>Tabla 8: Resultados del pre experimento para el sitio humanos.uci.cu</i>	48
<i>Tabla 9: Resultados del pre experimento para el sitio android.uci.cu</i>	49
<i>Tabla 10: Resultados del pre experimento para el sitio dragones.uci.cu</i>	49

Índice de Imágenes

<i>Ilustración 1:Funcionamiento de un buscador (Seker, 2012)</i>	6
<i>Ilustración 2:Herramienta woorank (woorank.com)</i>	19
<i>Ilustración 3:Herramienta Siteliner (Siteliner.com)</i>	20
<i>Ilustración 4:Herramienta Metricspot (metricspot.com)</i>	21
<i>Ilustración 5:Diagrama de clases del modelo del dominio (Elaboración propia)</i>	29
<i>Ilustración 6:Jerarquía de usuarios del sistema (Elaboración propia)</i>	32
<i>Ilustración 7:Casos de Uso inicializados por el usuario (Elaboración propia)</i>	33
<i>Ilustración 8:Ejemplo de uso del patrón CRUD Parcial (Elaboración propia)</i>	34
<i>Ilustración 9:Patron de diseño experto de la clase seoResultados (Elaboración propia)</i>	37
<i>Ilustración 10: Patrón de diseño controlador del método principal de la clase Analizar (Elaboración propia)</i>	38
<i>Ilustración 11:Diagrama de secuencia del CU "Registrar nueva url" (Elaboración propia)</i>	39
<i>Ilustración 12:Diagrama de componentes.(Elaboración propia)</i>	41
<i>Ilustración 13:Interfaz de inicio de la herramienta. (Elaboración propia)</i>	43
<i>Ilustración 14:Interfaz de resultados de la herramienta(Elaboración propia)</i>	43
<i>Ilustración 15:Cantidad de no conformidades detectadas por tipo. (Elaboración propia)</i>	47

Introducción

El desarrollo de Internet como medio para publicar y compartir información, ha propiciado el constante aumento de información en la gran red. Esto ha provocado que en algunas ocasiones los usuarios no saben dónde buscar la información que necesitan debido a la gran cantidad de documentos existentes en la web.

Para darle solución a este problema, se diseñaron los Sistemas de Recuperación de Información, capaces de indexar grandes cantidades de datos y ofrecerlos al público de una manera rápida y concisa, en forma de una lista ordenada de resultados de búsqueda. Ejemplos de estos sistemas lo constituyen los buscadores más famosos en la actualidad: Google, Bing, Yahoo!, entre otros. La mayoría de estos sistemas cuentan con mecanismos inteligentes, que clasifican los contenidos de acuerdo a determinados parámetros, según la relevancia respecto a los términos de búsqueda utilizados por los usuarios. En muchos escenarios estos sistemas suelen tener indexados contenidos de gran calidad, y sin embargo no son mostrados en las primeras páginas de búsqueda. (Tolosa y Bordingnon, 2007)

A las técnicas utilizadas para posicionar contenidos de páginas web, en los primeros resultados de un motor de búsqueda, se le denomina Optimización para Motores de Búsqueda (SEO, por sus siglas en inglés) o Posicionamiento Web. (Zolezzi, 2007).

Uno de los aspectos más importantes que atraen visitas de los usuarios a los portales web es, ¿qué tan bien posicionado está?, debido a que la gran mayoría de los usuarios al realizar una búsqueda, revisan a lo sumo de tres a cuatro páginas de resultados. Por lo expresado anteriormente, el Posicionamiento Web se ha convertido en una de las prácticas principales en los últimos tiempos para los desarrolladores. Existen dos tipos de posicionamiento web: el on page, aquel que analiza los elementos internos de la página y el off page, que analiza los elementos externos de la página, estos usan un conjunto de métricas internacionales para medir que tan bueno es el sitio en cuanto a posicionamiento orgánico.

En la actualidad existen diversas herramientas para efectuar análisis SEO; la mayoría de estas son desarrolladas bajo licencias propietarias, por lo que adquirir un software de este tipo sería muy costoso y las versiones que son gratuitas ofrecen servicios restringidos en cuanto a tiempo y funcionalidades, por lo que al análisis de las principales métricas internacionales haciendo uso de estas herramientas se hace bastante difícil de realizar.

En la Universidad de las Ciencias Informáticas (UCI) y específicamente en el Centro de Ideoinformática (CIDI), se desarrollan sitios web de múltiples instituciones de gran importancia social, política y económica para el país. El equipo de desarrollo de portales web se encarga de incluir mecanismos que ayuden al posicionamiento de las páginas web en los principales motores de búsqueda. Sin embargo, dada la naturaleza de Internet, los resultados de ese trabajo solo son posibles de apreciar después de cierto período de tiempo una vez desplegados físicamente de cara a Internet, lo cual impide la detección temprana de errores y la evaluación de las principales métricas internacionales

El equipo de desarrollo de portales web no cuenta hoy con un método de diagnosticar los problemas de SEO que poseen los portales web desarrollados, antes o después de ser desplegados físicamente; debido a los problemas previamente explicados, se hace necesario contar con una herramienta informática que ayude a los desarrolladores a detectar dichos problemas.

De acuerdo a lo expuesto previamente es válido plantear el siguiente **problema de investigación**: ¿Cómo diagnosticar los problemas de SEO existentes en los portales web, desde las primeras etapas del proceso de desarrollo de software?

Tomando como **objeto de estudio** el proceso de optimización para motores de búsqueda en portales web; definiendo como **campo de acción** el análisis de posicionamiento web en portales web.

El **objetivo general** que persigue este trabajo es desarrollar una herramienta informática que permita el análisis de posicionamiento web en los portales web desde las primeras etapas del proceso de desarrollo de software.

Teniendo como **hipótesis de investigación**: El desarrollo de una herramienta informática para diagnosticar los problemas de SEO presentes en los portales web; contribuirá a la detección temprana de malas prácticas de SEO, desde las primeras etapas del proceso de desarrollo de software. Teniendo en cuenta la hipótesis planteada se define como **variable independiente**: la herramienta informática. Este consiste en una aplicación informática la cual, mediante una interfaz le permitirá al usuario analizar la página o sitio web. Como **variable dependiente** se especifica: la detección temprana de malas prácticas de SEO. Dicha variable hace alusión al análisis de los elementos de las páginas de los sitios web. La misma a su vez, se dimensiona en tiempo de resultado de cada consulta realizadas al sitio, constituyendo este último el indicador a medir.

Tabla 1: Operacionalización de las variables de la hipótesis (Elaboración propia)

Variables	Descripción	Dimensiones	Indicadores	Unidad de Medida
Variable independiente	Herramienta informática			minutos
Variable dependiente	La detección temprana de malas prácticas de SEO.	Tiempo de resultado de cada consulta realizadas al sitio	Tiempo	

Como objetivos específicos:

- Caracterizar el marco teórico conceptual y el estudio del estado del arte respecto a las tecnologías actuales para el desarrollo de sistemas de análisis SEO.
- Diseñar un sistema que permita diagnosticar los problemas de SEO presentes en los portales web.
- Implementar un sistema que permita diagnosticar y sugerir soluciones a los problemas de SEO presentes en los portales web.
- Validar el correcto funcionamiento del sistema para análisis SEO.

Para darle cumplimiento a estos objetivos se han propuesto las siguientes **tareas de investigación:**

- Levantamiento bibliográfico de los principales elementos teóricos y conceptos asociados al análisis de SEO.
- Estudio de sistemas homólogos existentes en la actualidad para análisis SEO.
- Identificar los posibles parámetros de SEO que se podrán medir con el sistema para el análisis SEO a implementar.
- Investigar las técnicas y métodos existentes para la extracción de información asociada al SEO en páginas HTML.
- Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.

- Diseño de la arquitectura, selección de patrones de diseño y arquitectura.
- Implementación de la solución del Trabajo de Diploma.
- Selección de las técnicas de validación útiles del sistema para el análisis SEO

Para darle cumplimiento a las tareas de investigación, se utilizaron los siguientes **métodos de investigación:**

- Histórico – Lógico: Con el objetivo de constatar teóricamente cómo ha evolucionado la Internet, así como la forma en la cual se analiza y procesa la información dentro de la misma, en especial los buscadores web; así como las herramientas y tecnologías usadas en el desarrollo de aplicaciones web.
- Analítico – Sintético: Empleado para el análisis de los elementos esenciales referentes a las teorías, documentos y literatura en general relacionada con SEO.
- Modelación: Utilizado en la representación, mediante el uso de diagramas, de las características del sistema, relaciones entre objetos; y las actividades que intervienen en el proceso de desarrollo de la herramienta informática para el análisis SEO.

Estructura del documento.

Capítulo 1. Fundamentación teórica sobre la Optimización para Motores de

Búsqueda: Se realiza el levantamiento bibliográfico, el estudio de fundamentos teóricos, las tendencias actuales y principales actores relacionados con el proceso de optimización para motores de búsqueda como objeto de estudio del presente trabajo. También se exponen las características más generales respecto a las tecnologías a utilizar en el presente trabajo.

Capítulo 2. Diseño de la propuesta de solución: Se exponen las características del sistema, incluyendo los requisitos funcionales y no funcionales, patrones de diseño y arquitectura utilizados; además de los correspondientes artefactos que requiere la metodología de desarrollo utilizada.

Capítulo 3. Implementación y validación del sistema para análisis de

posicionamiento web: Se exponen algunos aspectos asociados con la implementación de la herramienta informática, así como los componentes que la integran. Además, se presentan los casos de uso de prueba a utilizar en la validación del sistema y se analizan los resultados de las pruebas que permiten evaluar la calidad de la propuesta de solución.

Capítulo 1. Fundamentación teórica.

Introducción.

En este capítulo se definen los principales conceptos y definiciones relacionados con el Posicionamiento Web en los buscadores, para dar un conocimiento básico y una idea a los clientes de lo que se desea desarrollar. Se definen las herramientas y tecnologías utilizadas para llevar a cabo el desarrollo de la aplicación, teniendo en cuenta las necesidades a cumplir.

1.2 Aspectos fundamentales para entender el surgimiento del posicionamiento.

1.2.1 Internet

Desde el año 1969 *Defense Advanced Research Projects Agency* (DARPA, por sus siglas en inglés) comenzó a realizar sus primeros estudios y aplicaciones para conectar computadoras en caso de una guerra atómica que dejara sin comunicación a los seres humanos, con fines principalmente de defensa. Luego en el año 1972 aparece públicamente la primera red denominada ARPANET, financiada por DARPA. Más tarde ARPANET cambia su protocolo NPC por TCP/IP, el cual se usa hoy en día; y varios años más tardes ya existían redes pequeñas en todos los continentes. (Negroponte, 1995)

El aumento de usuarios en la Red ha llegado a formar una gran comunidad, donde todos sus miembros de una forma u otra están conectados entre sí. Tanto es así, que cuando una persona tiene necesidad de conocimiento popular o de conocimiento no escrito en libros recurre a la fuente más fiable, más acorde a su necesidad y más accesible que le sea posible, siendo esta la Internet. Este hecho lejos de ser perjudicial para la especie humana, implica la existencia de un medio capaz de albergar soluciones para problemas que antes eran más difíciles de resolver. (Negroponte, 1995)

Las fuentes en Internet según Monereo y Fuentes (2005) poseen cinco características principales, que son:

- Extenso.
- Mayormente semiestructurada o sin estructura alguna.
- De diversa calidad
- Dinámica.
- Distribuida y autónoma.

Es por ello y por su crecimiento exponencial, que surgió la necesidad de poner un orden, control o algún tipo de clasificación a las páginas Web, documentos y sitios de la red. Los encargados de realizar este análisis se denominan Robots de la Web, pero no todos estos robots son iguales, ni sus técnicas de búsqueda. Cada robot tiene una forma diferente de analizar la información de la Red, así surgieron los buscadores web.

1.2.2 Buscadores web

Los buscadores son unos de los principales componentes o elementos a tener en cuenta para posicionar las páginas o los sitios Web. Esta palabra tiene muchas definiciones en los glosarios de Internet. A continuación, se muestra la definición que más se relaciona a este trabajo.

Los buscadores son herramientas de búsqueda de la red que permiten mediante palabras o combinaciones de palabras, encontrar documentos alojados en páginas Web. (Muñiz, 2001)

En la siguiente figura (Seker, 2012) se muestran los subsistemas que componen un motor de búsqueda de manera general, representando las relaciones internas de comunicación y funcionamiento que se establecen entre ellos.

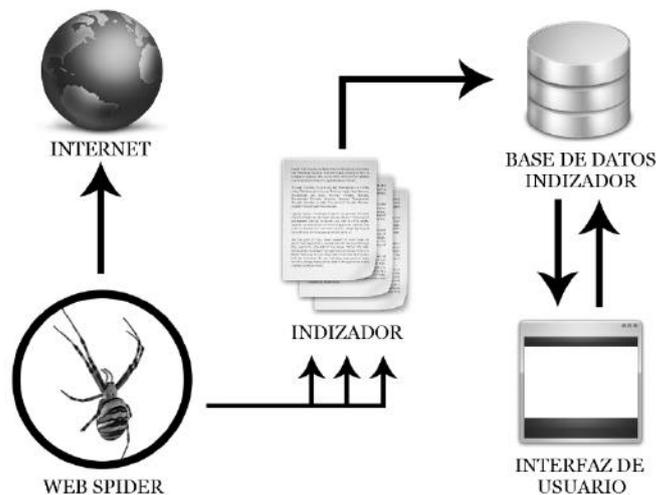


Ilustración 1. Funcionamiento de un buscador (Seker, 2012)

Los buscadores según la revista informática IB Times (2014) están compuestos por:

- Indexador: Realiza diversos métodos para incluir en el índice de Internet el contenido de un sitio web

- Araña o Spider: Navega constantemente la Web para indexar los sitios Web, los documentos y todo el contenido visitado en la red para optimizar la búsqueda.
- Base de datos indexada: Emplea uno o más índices para identificar registros.
- Interfaz de usuario: Es el medio de comunicación entre el usuario y el sistema.

Las búsquedas se hacen por palabras claves o con árboles jerárquicos por temas; el resultado de la búsqueda es un listado de direcciones web en los que se mencionan temas relacionados con las palabras claves buscadas. Los Motores de Búsquedas se clasifican en tres tipos:

- 1 Índices temáticos: Son sistemas de búsquedas por temas o categorías jerarquizadas (aunque también suelen incluir sistemas de búsqueda por palabras claves). Se trata de una base de datos de direcciones web elaboradas manualmente, es decir, hay personas que se encargan de asignar cada página web a una categoría o tema determinado.
- 2 Buscadores: Recorren las páginas recopilando información sobre los contenidos de las páginas. Cuando se busca una información concreta en los buscadores, ellos consultan su base de datos y presentan resultados clasificados por su relevancia para esa búsqueda concreta.
- 3 Metabuscaores: Permite lanzar varias búsquedas en motores seleccionados respetando el formato original de los buscadores. Lo que hacen es realizar búsquedas en auténticos buscadores, analizando los resultados de la página y luego presentando sus propios resultados, según un orden definido por el sistema estructural del metabuscador.

Los motores de búsqueda como operan de forma automática, contienen generalmente más información que los directorios. Sin embargo, estos últimos han de construirse a partir de búsquedas (no automatizadas) o bien a partir de avisos dados por los creadores de páginas. Los buenos directorios combinan ambos sistemas. (IB Times, 2014)

Estos motores de búsqueda le asignan de forma numérica una relevancia a los documentos y páginas web para ser indexados en sus bases de datos, colocando sus direcciones en una base de datos confeccionada por el robot de búsqueda. Esta función es conocida como **popularidad** (*MIT Technology Review*, 2015). El sistema *pagerank* es utilizado por uno de los motores de búsqueda más populares de internet, Google, para ayudarle a determinar la importancia o relevancia de una determinada página. La relevancia es calculada mediante un algoritmo que emplea varios parámetros como el número de enlaces externos e internos, del texto de los enlaces, de la temática de las

páginas enlazadas y del peso de las mismas, de las metaetiquetas, de la frecuencia relativa del término en la búsqueda en el texto, así hasta más de 100 variables, dando como resultado un valor numérico que oscila entre 0 y 10 llamado popularidad, también existen valores negativos que son utilizados por el buscador para penalizar las páginas web.

Con este valor numérico es que el motor de búsqueda de Google ubica en una posición los documentos y páginas. Pero los directorios como Yahoo!, no usan un robot de búsqueda para indexar y almacenar las direcciones de las páginas web, estos directorios gestionan bases de datos confeccionadas por personas. Estas bases de datos almacenan y clasifican en categorías las URLs junto con sus comentarios. El surgimiento de los buscadores y directorios trajo, como consecuencia la preocupación entre los administradores para posicionar las páginas y sitios web, surgiendo así un nuevo término de gran importancia: Posicionamiento Web.

1.2.3 Posicionamiento Web

El posicionamiento web tiene varias definiciones (MC Marcos, 2005) (L Codina, 2005), haciendo un resumen de estas se puede decir que es un conjunto de procedimientos y técnicas que estudian las características que proporcionan a un sitio o una página web la máxima visibilidad en Internet en un lugar óptimo entre los resultados proporcionados por un buscador. (L Codina, 2005)

Existen dos tipos de posicionamiento; el natural u orgánico y el planificado. El posicionamiento natural se conoce como SEO (*Search Engine Optimization*) (Urgarte, 2002), término muy utilizado en la actualidad y significa Optimización de Motores de Búsqueda. Es natural por la característica que tiene de optimizar las páginas web y posicionarlas sin usar marketing. Cuando el motor de búsqueda visita estas páginas web optimizadas y le asigna un ranking, la incorpora automáticamente en su base de datos, es decir, incorpora la URL de la página en la base de datos del buscador, siempre teniendo en cuenta que la misma no fue posicionada pagando dinero ni utilizando ningún tipo de marketing.

El otro tipo de posicionamiento que existe es el denominado SEM (*Search Engine Marketing*, por sus siglas en inglés), este no es más que el Marketing de Buscadores, este tipo de búsqueda está basada en patrocinar las URL de las páginas web en los motores de búsqueda. Como es lógico para este tipo de posicionamiento hay que invertir capital monetario para alcanzar éxito, de lo contrario no serviría seguir esta línea. Por lo que se recomienda el uso del posicionamiento SEO. (Urgarte, 2002)

El uso de SEO o posicionamiento natural tiene como objetivo principal aprovechar los criterios de valoración de los diferentes buscadores. Estas técnicas conllevan a cambiar el contenido de la página, títulos, enlaces, metadatos; de manera que el motor de búsqueda a la hora de procesar la página web para su indexación, le otorgue un valor alto a la misma. (Urgarte, 2002)

Debido al florecimiento de nuevas tecnologías para el desarrollo de software, el crecimiento de la necesidad de procesar información sobre la red y la simplicidad del uso de un navegador web surge un nuevo tipo de aplicaciones.

1.2.4 Técnicas de Optimización para Motores de Búsqueda (SEO)

Con el surgimiento de nuevas áreas de la investigación se hace más complejo organizar y suministrar la información que los usuarios necesitan. Es por ello que, el uso de las técnicas de optimización para motores de búsqueda que apoyen a la toma de decisiones en variados aspectos, contribuyen a evaluar las distintas fuentes de información, proporcionando una visión de la calidad actual, presentada durante el proceso de búsqueda de dicha información. Estas técnicas se usan para evaluar los indicadores métricos. (Mondelo, 2010).

Los indicadores son elementos básicos que brindan información cualitativa y/o cuantitativa del control de cómo funciona una actividad, pues hacen referencia a parámetros estables que sirven de magnitud de comprobación del funcionamiento de esta. Los indicadores deben ser: accesibles (fáciles de identificar y recopilar), pertinentes (para lo que deseamos medir), fieles (que informen con fidelidad de las condiciones de los datos que se recogen), objetivos (no ambiguos en su interpretación), precisos (para la acción que se quiere estimar), unívocos (parámetros exclusivos de lo que se mide) y sensibles (que permitan recoger y estimar variaciones de aquello a lo que referencian). (Mondelo, 2010).

La aplicación de estos indicadores dentro de los buscadores se denominan **buenas prácticas para SEO**, las cuales son empleadas por muchas herramientas a la hora de analizar y mejorar los sitios web.

A continuación, se muestran algunos de los principales elementos a la hora de realizar un análisis SEO:

- 1- Incluir palabras y frases claves en las metaetiquetas “*keyword*” y “*description*” de las páginas web.

- 2- Incluir palabras y frases claves en las regiones más importantes de las páginas web, por ejemplo: títulos y subtítulos de artículos, URL de acceso, dominio del sitio web, etiquetas “alt” de las imágenes, etc.
- 3- Emplear etiquetas de encabezado (h1, h2, ..., h6), para resaltar títulos, subtítulos o secciones importantes en el cuerpo de la página web.
- 4- Uso de mapas de sitios especializados “sitemap.xml”, para ayudar a los spiders a encontrar los contenidos del sitio de forma más rápida y directa.
- 5- Utilizar URLs limpias para el acceso a las páginas web, evitando el empleo de palabras y caracteres sin significado.
- 6- Emplear estilos de diseño adaptativo “*responsive design*”, para adaptar las páginas web a las diferentes resoluciones y dispositivos móviles.
- 7- Empleo correcto de la meta etiqueta “*viewport*”.
- 8- Mantener un promedio de descarga de las páginas web del sitio por debajo de 3 o 5 segundos.
- 9- Emplear URL Canónicas que se usa para evitar contenidos duplicados

1.2.5 Aplicaciones Web

El término aplicación web, según el autor Lujan (2002) plantea la siguiente definición: “Una aplicación web es un tipo especial de aplicación cliente-servidor, donde tanto el cliente como el servidor web y el protocolo mediante el que se comunican (HTTP) están estandarizados”; la cual contrasta con las valoraciones realizadas posteriormente por Hassan, Martín, y Lazza (2004).

Las aplicaciones web suelen componerse por tres capas o niveles que se comunican entre sí y que hacen posible el correcto funcionamiento del software como un todo: la capa superior o interfaz de usuario, la cual se muestra en los navegadores web; la capa intermedia, encargada del procesamiento de información y la capa inferior que proporciona los datos. Esta estructura por capas ofrece mayor comodidad a la hora de implementar las funcionalidades, garantizando en la mayoría de los casos el desacoplamiento de los componentes o módulos que integran la aplicación; brindando excelentes facilidades a la hora de desplegar el software de forma distribuida.

1.3 Herramientas SEO

Según funcionalidades de las herramientas SEO, estas se pueden clasificar en tres tipos (Sullivan, 2004):

- Análisis: Permiten comprobar distintos aspectos del funcionamiento y codificación de las páginas web.

- Mejora: Analizan la página web y ofrecen consejos sobre la estructura y contenido del sitio web, con el objetivo de mejorar su posicionamiento.
- Monitorización: Permite realizar un control de la página web, para comprobar su estado, evolución y cambios a lo largo del tiempo.

Según su uso las herramientas SEO se pueden clasificar en dos tipos (Sullivan, 2004):

- *SEO on page*: Es un término en inglés, se utiliza para referirse a las optimizaciones, que se puedan hacer en la página, para mejorar su posicionamiento orgánico.
- *SEO off page*: Comprende todos los factores externos, es decir los que no pertenecen a la página.

Características de las herramientas *SEO on page*:

- Diseño web *responsive*.
- Que la web pueda ser fácilmente rastreada por las arañas de los buscadores es el primer paso. Los ordenadores de los buscadores tienen que tener acceso a la página web para poder procesarla y mostrarla en los buscadores. Por esa razón, el *crawl budget* o *crawl rate* influye directamente en el posicionamiento: cuanto mayor es la frecuencia de rastreo de una web y más páginas rastrea, mejor es su posicionamiento. Este punto englobaría puntos en los que se especifica más abajo, como pueden ser dejar las páginas más accesibles, eliminar contenido duplicado, reparar errores 4xx, 5xx, 7xx, hacer las webs lo más ligeras posible para que el rastreador consuma menos recursos.
- Crear contenidos de calidad. Es común el dicho de: "El contenido es rey". Desde 2015, Google asigna cada vez más importancia a la llamada "experiencia web del usuario", siendo capaz de medirla en términos estadísticos siempre y cuando dicha web en concreto haya sido indexada por este buscador. La experiencia del usuario tiene relación, sobre todo, con la adaptabilidad a dispositivos móviles, el contenido, la usabilidad y la velocidad del tiempo de carga, entre otros factores. Igualmente, la estructura de enlaces internos es clave para la usabilidad y la experiencia de usuario (*UX, User eXperience*).
- Realizar la estructuración y el diseño de una página web pensando en el posicionamiento, significa prestar atención a que sea funcional, fácil de acceder y que capte la atención del usuario.
- Crear títulos únicos y descripciones pertinentes del contenido de cada página. Cada página es una tarjeta de presentación para el buscador. Los títulos y descripciones son puntos de partida para la identificación de los términos relevantes a lo largo de

la web por los buscadores. Las mejores prácticas recomiendan escribir títulos de entre 60 y 70 caracteres.

Características de las herramientas SEO off page:

- Conseguir que otras webs de temática relacionada enlacen con tu web. Para ello es importante realizar una búsqueda para aquellos términos que consideras deberían llevar tráfico a tu web y estudiar cuáles de ellos tienen un contenido complementario. Si por ejemplo quieres posicionarte por el término "peluquería Madrid" puede ser interesante intentar conseguir *backlinks* de peluquerías de otras ciudades.
- Ahora mismo hay cientos de redes sociales, por ejemplo Hi5, Facebook y Orkut, en las cuales poder participar y obtener visitas de nuestros nuevos «amigos». Para Google, la red social que mayor impacto tiene en el SEO es Google Plus, que ha tomado el lugar en importancia de Twitter y Facebook.
- Darse de alta en directorios importantes como Dmoz y Yahoo!. Los directorios han perdido mucho interés en los buscadores, pero siguen siendo un buen punto de partida para conseguir enlaces o un primer rastreo de tu web por los buscadores. Ambas requieren de un filtro humano para su inclusión lo que asegura la calidad de la web añadida, pero también ralentiza y dificulta su inclusión.
- Registrarse y participar en foros, de preferencia en foros temáticos relacionados a la actividad de su página web. La frecuente participación tiene que ir acompañado de aporte real y valioso como para ser tomado en cuenta como un usuario calificado, el detalle del éxito para conseguir visitas y aumentar el posicionamiento es el enlace hacia su página web presentado en su firma.
- Escribir artículos en otros sitios web. Los artículos son un método muy poderoso para mejorar el posicionamiento y conseguir atraer visitas. Si puedes escribir unos artículos de un curso, de los trucos del día, la utilidad del producto de tu web.

Active Web Traffic

Esta herramienta da de alta a un sitio web en varios buscadores. Tiene incluido un generador de palabras claves. Además, analiza la densidad de palabras claves, comparándolas con los sitios web de la competencia y comprueba la popularidad de los enlaces que apuntan a un sitio web, analizando el comportamiento en cuanto a ranking.

Agent Web Ranking

Limita manualmente el proceso de búsqueda para cada buscador. También crea informes de posiciones seguros y exactos. Además, brinda la posibilidad de emitir estos informes en varios idiomas y es capaz de hacer informes personalizados.

Majestic

Esta herramienta presenta análisis de perfil para saber que está haciendo tu competencia y poder adaptar tu estrategia, además, posee métricas propias (*Citation Flow* y *Trust Flow*) que son muy útiles a la hora de juzgar un perfil de enlaces.

Todas estas herramientas están centradas principalmente en dar de alta los sitios web en los buscadores del mundo, lo cual brinda un gran avance para lograr estar indexado en las bases de datos de los buscadores, lo cual no constituye el objetivo principal de estas herramientas.

El objetivo principal debe ser brindar funcionalidades que brinden un mayor aporte para posicionar los sitios, como por ejemplo: Listar los 10 mejores sitios posicionados, así como algunos parámetros a tener en cuenta de estos: si posee o no mapa de navegación para cada sitio, mostrar las palabras más predominantes, enlaces rotos, determinar la consulta formada con las palabras claves que se obtenga del sitio a posicionar, etc. Esto ayudaría a realizar un análisis de estos parámetros y enfocar nuestro sitio en ese sentido.

1.4 Tecnologías y herramientas a utilizar

1.4.1 Técnicas y mecanismos de rastreo en la web

Apache Nutch

Nutch es un *spider* o *web crawler* libre y de código abierto, desarrollado totalmente en Java por *The Apache Software Foundation*. Ha sido implementado sobre la base de *Apache Lucene*, una librería de alto rendimiento para la búsqueda basada en texto y que utiliza una modificación del algoritmo “*Vector Space Model*” (Modelo de Espacio Vectorial en español), con un enfoque booleano que restringe las estimaciones de los resultados obtenidos (Mahecha, 2009; Apache Nutch, 2013).

Los argumentos expuestos por Mahecha (2009) demuestran que Nutch no altera los resultados obtenidos en las consultas realizadas por los usuarios, sino que trata de ofrecer los mejores resultados posibles de acuerdo al índice construido a partir de sus rastreos. Al analizar el comportamiento de Nutch, en cuanto a la manera que obtiene los datos de las páginas, se comprueba que este no lo realiza de la forma convencional de

otros mecanismos de rastreo; Nutch descarga los datos de una manera especial que solo él puede interpretar. Por lo que analizar diversos parámetros SEO se hace bastante difícil, debido a que se tendría que emplear algún mecanismo para interpretar la información.

Comando de Linux WGET

Según el autor Cesar Iglesias (2014), Wget es una pequeña pero muy útil aplicación de terminal que sirve para descargar archivos. Hay muchas aplicaciones gráficas y (para muchos) más cómodas para ello, pero como siempre digo y recomiendo, hay que tener en cuenta la terminal cuando todo lo demás falle. Wget es muy potente, soporta ftp, http y https.

Este comando tiene varias maneras de descargar la información de un sitio web:

- 1- Colocando directamente la url: Se usa el comando wget y auto seguido la dirección del sitio.
- 2- Descargar a partir de una lista: Se insertan todas las direcciones url en un archivo txt y utilizamos el comando wget -i y el nombre de archivo.
- 3- Descarga recursiva por niveles: Se usa para descargar recursivamente hasta 5 niveles del sitio usando el comando -r y el número de iteración que desea descargar

Debido a la complejidad en el uso de Apache Nutch y al gran volumen de información que brinda no se recomienda para este trabajo debido que ofrece datos adicionales que no se necesitan analizar; Además, Apache Nutch no es capaz de recopilar cierta información de interés, como ejemplo: si el sitio es *responsive* o no, si se emplean mecanismos multi dominio, si contiene flash, entre otros; y además, la forma en que descarga las páginas web, no es la tradicional, por lo que se dificulta la lectura del contenido y la forma de obtener otros datos a partir de técnicas de parsing. El comando WGET no brinda suficiente información para un análisis completo de SEO, por lo que se necesitaría emplear otro mecanismo para adquirir la información restante. Debido a lo explicado anteriormente el autor del presente trabajo decidió implementar un mecanismo de rastreo propio basado en una librería de JAVA llamada jsoup, que descarga las páginas del sitio en texto plano para su posterior análisis, brinda información sobre tiempo de descarga, nombre de dominios, texto, imágenes, archivos multimedia, número de páginas, entre otros muchos elementos.

1.4.2 Técnicas de parsing y extracción de información de páginas HTML

XQuery

Extensible Markup Language (más conocido como XML) es un formato para almacenar datos extremadamente versátiles, utilizado para representar una gran cantidad de información como, por ejemplo, páginas web (XHTML). (Atwood, 2009)

Un conjunto de datos expresados en XML es una cadena de texto donde cada uno de los datos está delimitado por etiquetas de la forma `<T>... </T>` o se incluye como atributo de una etiqueta de la forma `<T A=" ..." >... </T>`.

Actualmente, XML se ha convertido en una herramienta de uso cotidiano en los entornos de tratamiento de información y en los entornos de programación.

El escenario de uso más común se da cuando tenemos que realizar alguna búsqueda en un documento o conjunto de documentos con gran cantidad de datos en XML, o bien hemos de trabajar sólo sobre un subconjunto de todos los datos XML y queremos obtener dicho subconjunto de una forma sencilla, para evitar trabajar con toda la colección de datos. (Ibañez, 2011)

Dar solución a este escenario de uso mediante código escrito con la ayuda de un parser SAX es rápido y sencillo de desarrollar si la consulta y la estructura de la información no tienen una gran complejidad. A medida que la consulta va ganando en complejidad, es necesario definir comportamientos más complejos a ejecutar cuando se encuentre una etiqueta y se necesita un mayor número de almacenamientos temporales de datos. Otro problema es que el código está fuertemente ligado a la estructura de los datos en XML, por lo que cualquier cambio en la consulta o en la estructura de los datos obliga a volver a escribir el código. Además, el código resultante es muy poco parametrizable y reutilizable por lo que para desarrollar una consulta similar habría que volverla a escribir desde el principio, sin poder aprovechar nada de lo escrito. (Ibañez, 2011)

De manera rápida podemos definir XQuery con un símil en el que XQuery es a XML lo mismo que SQL es a las bases de datos relacionales. XQuery es un lenguaje de consulta diseñado para escribir consultas sobre colecciones de datos expresadas en XML. Abarca desde archivos XML hasta bases de datos relacionales con funciones de conversión de registros a XML. Su principal función es extraer información de un conjunto de datos organizados como un árbol n-ario de etiquetas XML. En este sentido XQuery es independiente del origen de los datos. (Ibañez, 2011)

XQuery es un lenguaje funcional, lo que significa que, en vez de ejecutar una lista de comandos como un lenguaje procedimental clásico cada consulta es una expresión que es evaluada y devuelve un resultado, al igual que en SQL. Diversas expresiones pueden combinarse de una manera muy flexible con otras expresiones para crear nuevas expresiones más complejas y de mayor potencia semántica.

XQuery está llamado a ser el futuro estándar de consultas sobre documentos XML. Actualmente, XQuery es un conjunto de borradores. Sin embargo, a pesar de no tener una redacción definitiva ya existen o están en proceso numerosas implementaciones de motores y herramientas que lo soportan.

Aunque XQuery y SQL puedan considerarse similares en casi la totalidad de sus aspectos, el modelo de datos sobre el que se sustenta XQuery es muy distinto del modelo de datos relacional sobre el que sustenta SQL, ya que XML incluye conceptos como jerarquía y orden de los datos que no están presentes en el modelo relacional. Por ejemplo, a diferencia de SQL, en XQuery el orden de los datos es importante y determinante, ya que no es lo mismo buscar una etiqueta dentro de una etiqueta <A> que todas las etiquetas del documento (que pueden estar anidadas dentro de una etiqueta <A> o fuera). (Ibañez, 2011).

XQuery ha sido construido sobre la base de Xpath. Xpath es un lenguaje declarativo para la localización de nodos y fragmentos de información en árboles XML. XQuery se basa en este lenguaje para realizar la selección de información y la iteración a través del conjunto de datos.

SAXON HE

Saxon-HE (*Home Edition* por sus siglas en inglés) es un procesador de XSLT y XQuery creado por Michael Kay. Es un producto de código abierto disponible bajo la Licencia Pública de Mozilla. Proporciona implementaciones de XSLT 2.0, XQuery 1.0 y XPath 2.0 en el nivel básico de conformidad definidos por el W3C. Está disponible para Java y .NET. (Ibañez, 2011)

Saxon permite convertir documentos XML en otros documentos XML; puede convertir un documento XML que obedezca a un DTD (Definición de Tipo de Documento) a otro que obedezca otro diferente, un documento XML bien formado a otro que siga un DTD.

El autor decidió usar SAXON HE debido a su disponibilidad y su facilidad de uso. Además, permite procesar elementos xml como xslt.

1.4.3 Estudio de sistemas homólogos

Alrededor del mundo son muchas las herramientas que existen a la hora y han sido creados para suplir diversas necesidades en particular. Atendiendo a este trabajo, solo se analizarán las principales herramientas internacionales y nacionales, siendo estas últimas herramientas desarrolladas en la propia universidad.

1.4.3.1 Woorank

Woorank es una startup afincada en Bruselas, formada por un equipo multidisciplinar de 12 personas cuya especialización es el posicionamiento web. Woorank es una herramienta web que valora de forma dinámica hasta 100 puntos de una web con el fin de indicar la efectividad del marketing online en un tiempo determinado. (Quiles, 2013)

Además de ofrecer una **revisión de los distintos puntos SEO de la web**, facilita una serie de consejos y recomendaciones punto por punto.

Hay que entender que el posicionamiento web es una especialidad del marketing mucho más profunda y exhausta de lo que la herramienta desarrolla, pero es un buen comienzo para comprender cómo se comporta una página web con respecto a los buscadores.

Por cada IP, se puede realizar un informe a la semana de manera gratuita, solo hay que acceder a la página de Woorank e indicar la web deseada. La herramienta según Quiles analiza la página web en pocos segundos y ofrece los siguientes aspectos:

- **Las 5 prioridades:** es un apartado que refleja los aspectos más importantes para mejorar desde el punto de vista SEO. Estos aspectos son: uso correcto de etiquetas, optimización de enlaces, optimización de imágenes, diseño y uso correcto de meta-tags.
- **Visitantes:** toma como referencia los datos estimados en Alexa.
- **Redes sociales:** los datos estimados en el apartado de “impacto social” no se deberían tener muy en cuenta porque suelen variar sobre los datos reales, sin embargo, el apartado “cuenta de twitter” ofrece datos interesantes como la antigüedad y la ubicación de la cuenta.
- **Móvil:** ofrece información sobre la optimización web en referencia al tiempo de carga en dispositivos móviles, la visualización de la página en un dispositivo móvil y una serie de recomendaciones fundamentales para una mejor optimización.

- **Lo esencial de SEO:** es un punto que trata aspectos fundamentales del SEO que muchas veces se obvian como los archivos robots.txt, canonicalización de la IP, mapa del sitio XML, etc.
- **Contenido SEO:** es el apartado más importante que ofrece la herramienta, ya que, ofrece un análisis del SEO on page y las respectivas recomendaciones en los distintos puntos.
- **Enlaces SEO:** es posiblemente el punto más débil de la herramienta. El apartado de enlaces en el posicionamiento web es uno de los factores más importantes y que requiere de un profundo conocimiento.
- **Palabras clave de SEO:** la herramienta analiza la inserción de palabras en la página descrita.
- **Autoridad SEO:** visión general de la autoridad de la página web en los distintos buscadores.
- **Backlinks SEO:** ofrece un contador de backlinks además de informar si la página está registrada en Dmoz.
- **Usabilidad:** el segundo aspecto más relevante que ofrece la herramienta. Al igual que en el “contenido SEO” ofrece un análisis de los distintos factores con sus respectivas recomendaciones.
- **Seguridad:** la herramienta ofrece distintos aspectos como indicadores de fiabilidad, dominios libres, etc.
- **Tecnologías:** revela información sobre el desarrollo web como las herramientas empleadas, errores en el lenguaje, etc.



Ilustración 2:Herramienta woorank (woorank.com)

1.4.3.2 Siteliner

Siteliner es un servicio en línea que se ocupa de la labor, solo tienes que escribir la URL de tu sitio, hacer clic en el botón “Go” y esperar a que finalice el análisis de tus páginas. El resultado es un informe con múltiples datos y gráficas que te mostrarán los enlaces rotos y los contenidos duplicados de tu sitio.

Puedes hacer clic sobre los enlaces y gráficas para acceder a un listado de las URLs que presentan alguno de estos problemas, así podrás corregirlos y mejorar tu SEO.

El uso de Siteliner es gratuito, aunque esta modalidad está limitada al análisis de 250 páginas. Para eliminar esta limitación, haciendo posible el análisis de hasta 25.000 páginas, deberás suscribirte a la modalidad Premium de esta práctica herramienta SEO.

Siteliner[▶]

Explore your site.

Find duplicate content, broken links, and more...

Ilustración 3:Herramienta Siteliner (Siteliner.com)

1.4.3.3 Metricspot

MetricSpot es una herramienta en fase beta que nos hace un análisis de unos 60 parámetros de nuestro sitio web o una web de la competencia que queramos analizar.

MetricSpot está en español y no es necesario hacer un registro para utilizarla, aunque si nos registramos conseguimos que se guarde todo nuestro historial de análisis, dando la posibilidad de comparar.

MetricSpot se basa en 7 parámetros a la hora de añadir la url del sitio web:

- Resumen: Puntuaciones generales del sitio web, como la autoridad SEO, SEO básico, contenido o usabilidad.
- Autoridad SEO: Aunque son aspectos muy básicos sí nos ayudará a tener una percepción general de un sitio web de la competencia. (PageRank, Dmoz, Backlinks...).
- Seo Básico: Comprueba redirecciones www, metas y archivo como el robots.txt y sitemap.xml.
- Contenido: Realiza un análisis de las imágenes que aparecen en el sitio (alt), enlaces internos, externos y etiquetas html importantes como <h1>, <h2>... o las repeticiones de palabras.

- Usabilidad: Extensión de tamaño de la Url principal, existencia de página 404, tiempos de carga del sitio web, incluso si existe un formulario dentro de la url que estamos analizando.
- Aspectos Técnicos: Como etiquetas metas og, el % de ratio de texto en comparación con el código, tecnología que utiliza y la localización del servidor.
- Redes Sociales: Nos hace un resumen general de la repercusión social del sitio, analizando redes sociales como Facebook y Twitter.

Por lo general, aunque se trata de una herramienta que nos da aspectos muy generales y básicos a la hora de analizar una web, siempre nos va a servir de ayuda teniendo en cuenta los puntos nombrados anteriormente. Además, nos da la opción de guardar el informe generado o imprimirlo directamente.



Ilustración 4:Herramienta Metricspot (metricspot.com)

1.4.3.4 GeWeb

En la actualidad, la web constituye uno de los principales medios de difusión de la información. Hoy en día las páginas web son contenedoras de programas, archivos comprimidos, documentos de texto, archivos multimediales como pueden ser imágenes, videos, animaciones, sonidos, entre otros. Son construidas utilizando diferentes lenguajes de programación y son sustentadas por diversos tipos de servidores web. Por lo tanto, no estaríamos en un error si afirmáramos que la web es un reflejo de las principales tendencias que son seguidas por la mayoría de los usuarios en cuanto a la utilización de las Tecnologías de la Informática.

Generador de estudios webmétricos (GEWeb v1.0), es un producto informático cuyo objetivo principal consiste en realizar un estudio webmétrico (algo muy similar a una fotografía estadística de la web), con múltiples resultados cuantitativos y cualitativos, para asistir a la toma de decisiones respecto a la web analizada. El desarrollo creciente de las tecnologías en Cuba, implica la necesidad de extender estos estudios a todo lo largo y ancho del país.

1.3.4.5 Comparación entre herramientas

A continuación, se expone en la siguiente tabla, una comparación entre los *spiders* analizados, con el objetivo de ilustrar mejor las similitudes y diferencias entre estos.

Herramientas	Portabilidad	Lenguaje de Programación	Solución a los problemas encontrados
Woorank	No	PHP	Si
Sitelineer	No	PHP	Si
Metricspot	No	PHP	Si
GeWeb	Si	JAVA	no

Atendiendo al estudio y comparación de las herramientas anteriores, se determinó que las mismas no constituyen soluciones potenciales a la problemática planteada en este trabajo, dado que cada uno responde a necesidades particulares. Sin embargo, se identificaron aspectos de interés que pueden ser utilizados en la concepción de la solución informática que se propone en este trabajo. Entre los elementos positivos a considerar se destacan: la facilidad de uso de la interfaz de la herramienta woorank, la descripción detallada de elementos de SEO, lo cual evita explicar a los usuarios acerca de los mismos, también se hace énfasis en el posicionamiento natural como opción para el tratamiento de las métricas internacionales. Además, se identificaron como aspectos negativos: Muchas de estas herramientas presentan versiones *premium*, y sus versiones gratis presentan un uso limitado en cuanto a tiempo y funcionalidades, además, la herramienta nacional (GeWeb) no brindan consejos a los problemas de posicionamiento encontrados, ejemplos de estos: Uso indebido de las etiquetas <h1>, <h2>, ..., <h6>, imágenes sin descripción entre otros.

1.4.4 Lenguajes de programación

Debido a que una de las principales características de la herramienta a desarrollar es la portabilidad, se decidió tomar uno de los siguientes lenguajes para la implementación de la solución al problema planteado.

Python

Se presenta como un lenguaje de programación de propósito general, cuya popularidad es relativamente reciente. Se trata de Python, una apuesta por la simplicidad, versatilidad y rapidez de desarrollo. Python es un lenguaje de *scripting* independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo. (Venners, 2003). En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código, menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.

Características del lenguaje

Según el autor Rossum (2003) Python presenta los siguientes aspectos:
Propósito general

Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.

Interactivo

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

Orientado a Objetos

La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

Funciones y librerías

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de *strings*, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en.zip.

Sintaxis clara

Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave *begin* y *end*. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

Java

Java es **un lenguaje de programación orientado a objetos** que se popularizó a partir del lanzamiento de su primera versión comercial de amplia difusión, la JDK 1.0 en 1996. Actualmente es uno de los lenguajes más usados para la programación en todo el mundo.

Java es un lenguaje útil para casi todo tipo de problemas. Podemos citar como funcionalidades de Java varias:

- Aplicaciones “cliente”: son las que se ejecutan en un solo ordenador (por ejemplo, el portátil de tu casa).
- Aplicaciones “cliente/servidor”: son programas que necesitan conectarse a otra máquina (por ejemplo, un servidor de datos).
- Podemos hablar también de “aplicaciones web”, que son programas Java que se ejecutan en un servidor de páginas web.

Estos son sólo algunos ejemplos de todo el potencial que hay detrás de Java como lenguaje para aprender y obtener muchos beneficios con su uso. Obviamente por determinados términos empleados (cliente, cliente/servidor, base de datos, HTML, entre otros). El lenguaje Java tiene mucha potencialidad, pero también de que su conocimiento a fondo requeriría mucho tiempo.

El autor del presente trabajo elige el lenguaje Java, pues presenta un gran dominio acerca del mismo y a la gran cantidad de librerías y documentación acerca del mismo.

1.4.5 Herramientas CASE

Las herramientas CASE son aplicaciones informáticas que permiten agilizar la realización de determinadas actividades del proceso de desarrollo de software. Utilizadas mayormente en la modelación y análisis de procesos y sistemas, a través de diagramas; pueden contribuir eficazmente a la reducción de costos en términos de tiempo y dinero. Lo anterior garantiza el aumento de la productividad en el desarrollo de software de manera significativa. Entre las facilidades más apremiadas que proporcionan estas herramientas se encuentran: el diseño de proyectos, cálculo de costos, implementación automática de parte del código dado un diseño previo, compilación automática, documentación, detección de errores, entre otras.

1.4 Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML) es el lenguaje estándar especificado por el *Object Management Group* (OMG) para visualizar, especificar, construir y documentar los artefactos de un sistema, incluyendo su estructura y diseño. Utiliza un conjunto de símbolos y notaciones para representar gráficamente los diversos componentes que forman parte de la arquitectura de software. Permite el modelado de procesos de negocio y el modelado de requisitos apoyándose en el análisis orientado a objetos (*Object Management Group*, 2013).

1.5 Metodología de desarrollo

Una metodología de desarrollo de software constituye una filosofía de trabajo que proporciona una base de procesos para llevar a cabo con éxito cualquier proyecto informático. Existen dos grupos principales de metodologías: metodologías pesadas o tradicionales y metodologías ágiles. Cada una en particular propone una serie de procedimientos, técnicas, herramientas y soporte documental necesarios en todo proceso de ingeniería de software y que, además, se corresponde con un determinado enfoque que define la forma en que se realizan las actividades. Las metodologías de desarrollo brindan soporte a la toma de decisiones en un equipo de trabajo; es decir, permiten conocer qué persona hace una determinada actividad, cuándo y cómo la debe hacer.

La Universidad de las Ciencias Informáticas (UCI) realizó una versión de la metodología de desarrollo ágil de software AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma, y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, denominada ejecución, y agregándose también una nueva fase denominada Cierre. (Sánchez, 2015).

1.6 Marco de trabajo

Un marco de trabajo o “*framework*” como popularmente se le conoce, puede definirse como un conjunto bien estructurado y organizado de librerías que ponen a disposición determinadas funcionalidades básicas implementadas con anterioridad, y que permiten a los programadores centrarse objetivamente en las nuevas funcionalidades a desarrollar. Usualmente cada *framework* posee una filosofía o metodología de trabajo estándar, así como uno o varios patrones de diseño e implementación que definen cómo utilizarlos de la manera más eficiente posible. Su estructura a menudo suele tratarse como el esqueleto o núcleo para la construcción de nuevas aplicaciones, posibilitando el desarrollo basado en módulos, garantizando así la futura extensibilidad de los sistemas. (Dirt, 2000)

Spring

Spring ayuda a solventar este problema ya que cambia las responsabilidades y en vez de que el propio desarrollador sea el encargado de generar los objetos de cada uno de los *frameworks* es Spring basándose en ficheros xml o anotaciones el encargado de construir todos los objetos que la aplicación va a utilizar. (Álvarez, 2014).

De esta manera al ser Spring el encargado de inicializar todos los objetos de los distintos *frameworks*, es también el responsable de asegurarnos que se integran de la forma correcta.

1.7 Conclusiones parciales

Durante el análisis de los preceptos teóricos abordados en el capítulo, se pudo constatar que existe gran diversidad de elementos y herramientas SEO, atendiendo a sus clasificaciones y uso. Los mismos realizan análisis complejos, con un gran número de indicadores a analizar tanto internos como externos.

Con la selección de la metodología, herramientas y tecnologías con soporte multiplataforma y basadas en software libre, se logró obtener una base tecnológica adecuada que permitirá el desarrollo de la solución.

Capítulo 2. Análisis y diseño de la propuesta de solución.

Introducción

En este capítulo se abordarán los aspectos fundamentales relacionados con el diseño del sistema a desarrollar. Entre los elementos a destacar se encuentran el diagrama del modelo del dominio, mediante el cual se representan los objetos reales que intervienen en el proceso de análisis de posicionamiento web. Como vía para definir las futuras funcionalidades de la aplicación y qué usuarios podrán tener acceso a las mismas, se generaron los artefactos relacionados a la especificación de los requerimientos funcionales y no funcionales que deberá poseer el software; así como la especificación de los casos de uso del sistema. Como parte del diseño de la aplicación se definieron los estilos y patrones de arquitectura y diseño que se emplearán para lograr buenas prácticas de diseño y programación. Se presenta el diseño de la base de datos que se empleará para la persistencia de la información a manipular. A lo largo del capítulo se mostrarán los principales artefactos de ingeniería de software correspondientes a las funcionalidades o casos de uso más críticos.

2.2 Modelado del Dominio

Un modelo del dominio, también conocido como modelo conceptual, es una representación visual en forma de diagrama de las clases conceptuales u objetos del mundo real, que son significativos en un dominio de interés (Larman, 2003).

Considerando que los procesos y roles del negocio que intervienen en el problema real identificado son difíciles de definir, se hace necesario describir el procedimiento de modelado de la herramienta para análisis de posicionamiento web mediante una serie de conceptos, entidades y sus relaciones, agrupándose en un modelo del dominio con el fin de contribuir a la comprensión del contexto actual del problema.

2.2.1 Descripción de Clases del Modelo del Dominio

La modelación del dominio constituye la herramienta fundamental para garantizar la comprensión y descripción de las clases o conceptos y sus relaciones más importantes dentro del contexto del problema. A continuación, en la Tabla No.1 se presenta la descripción de los conceptos identificados.

Tabla 2: Descripción de las clases del modelo del dominio (Elaboración propia)

Concepto	Descripción
----------	-------------

Usuario	Persona que interactúa con la interfaz.
Interfaz	Constituye el entorno principal de la herramienta donde aparecen las diversas operaciones que se realizan al sitio.
Instancias de parsing y rastreo	Constituyen cada una de las instancias de los diversos mecanismos y técnicas con el objetivo del rastreo y parsing determinados contenidos específicos en la Web.
Archivos SAXON	Se refiere a los archivos de configuración escritos en formato XML y XSLT.
Archivos TXT	Se refiere a los archivos de configuración escritos en formato TXT.

2.2.2 Diagrama de clases del modelo del dominio

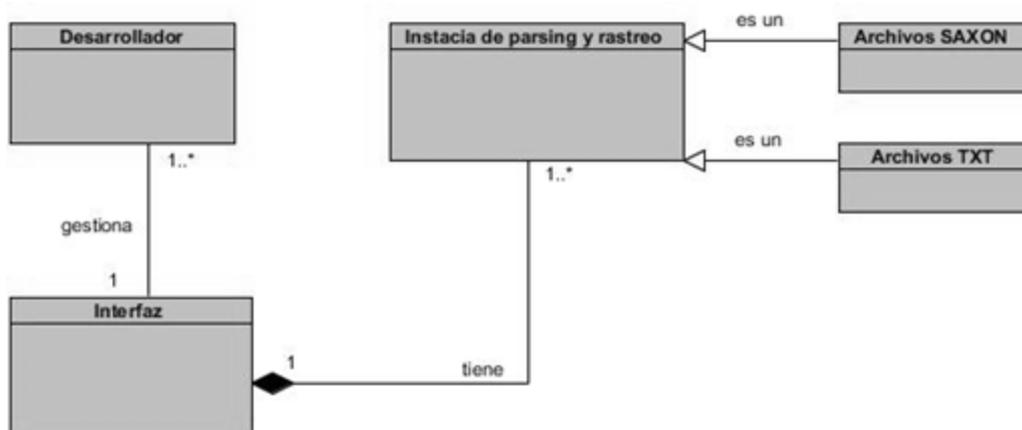


Ilustración 5: Diagrama de clases del modelo del dominio (Elaboración propia)

El modelo del dominio representado en la ilustración 2, muestra la relación existente entre los objetos fundamentales que intervienen en el análisis de posicionamiento web. Inicialmente se observa como los desarrolladores puede gestionar la interfaz de la herramienta. Este último posee en su estructura interna cierta cantidad de instancias de parsing y rastreo, además de archivos de configuración que definen el comportamiento de la herramienta. Además, los archivos de configuración pueden encontrarse en distintos formatos de escritura, según la función para la que se utilicen.

2.3 Especificación de los Requisitos del Software

Los requisitos del software constituyen las descripciones de los servicios y funcionalidades que brinda una aplicación informática, teniendo en cuenta además las restricciones operativas a las cuales están sujetas. De manera general, estos requisitos expresan las necesidades objetivas que presentan los usuarios, ante un sistema que resuelve un problema en particular de un determinado dominio (Sommerville, 2005).

A continuación, se muestran los requerimientos funcionales y no funcionales, de acuerdo con el objetivo planteado.

2.3.1 Requisitos funcionales

El objetivo fundamental de los requisitos funcionales es detallar lo que debe ser capaz de hacer el sistema, y con ello, guiar su desarrollo; pudiendo identificar la reacción a entradas particulares y cómo se debe comportar en situaciones concretas (Sommerville, 2005).

Requisitos funcionales de la herramienta informática para análisis de posicionamiento web:

RF 1. Registrar nueva url.

RF 2. Mostar url del sitio.

RF 3. Mostar palabras claves de la url.

RF 4. Mostar título del sitio.

RF 5. Mostar la longitud del título del sitio.

RF 6. Mostrar la meta descripción del sitio.

RF 7. Mostrar la longitud de la meta descripción del sitio.

RF 8. Mostrar las palabras claves del sitio.

RF 9. Mostrar cantidad de títulos del sitio.

RF 11. Mostrar la cantidad de imágenes del sitio.

RF 12. Mostrar la cantidad de imágenes del sitio con atributo alt.

RF 13. Mostar el número de frames del sitio.

- RF 14. Mostrar el número de iframes del sitio.
- RF 15. Mostrar la cantidad de enlaces del sitio
- RF 16. Mostrar la cantidad de enlaces internos del sitio.
- RF 17. Mostrar la cantidad de enlaces externos del sitio.
- RF 18. Mostrar velocidad promedio de descarga.
- RF 19. Optimización de Robot.txt
- RF 20. Mostrar la cantidad de enlaces rotos del sitio.
- RF 21. Código limpio
- RF 22. Mostrar la cantidad de imágenes con atributo title.
- RF 23. Uso de diseño adaptable.
- RF 24. Creación y envío de sitemap a buscadores.
- RF 25. Mostrar la cantidad de imágenes con atributo description.

2.3.2 Requisitos no funcionales

Los requerimientos no funcionales se presentan, en la mayoría de los casos, como las propiedades o cualidades que el sistema debe poseer. Sin embargo, desde otras aristas, pueden concebirse como las restricciones de las funcionalidades del sistema (Pressman, 2010). Independientemente del punto de vista, estos requisitos deben tratarse como las características que hacen al producto usable, rápido y/o confiable. Los requerimientos no funcionales, aunque no aportan funcionalidades propiamente dichas dentro de la aplicación, son de vital importancia para una puesta en marcha exitosa del software, y para lograr que este responda a las expectativas del usuario.

Requerimientos de software

RNF 1. Máquina virtual de JAVA.

Requerimientos de diseño e implementación

RNF 2. Paradigma de Programación Orientada a Objetos (POO).

Requerimientos de interfaz de usuario

RNF 3. Interfaz visualmente agradable y que posibilite al usuario una navegabilidad intuitiva por las funcionalidades del sistema.

2.4 Propuesta del sistema

La herramienta a desarrollar debe proporcionar un análisis de cualquier sitio, mediante su url. Dicha herramienta brinda información detallada acerca de los principales aspectos de posicionamiento web, además de una breve descripción de cada elemento; una vez analizado el sitio debe brindar una valoración general y posibles soluciones a los problemas encontrados.

2.5 Modelo de Casos de Uso del Sistema

El modelo de casos de uso del sistema es un diagrama que explica de forma simbólica la interacción entre los actores y los casos de uso (CU) que representan las funcionalidades de la aplicación. Además, son representadas las relaciones existentes entre casos de uso. El actor es una entidad externa del sistema que participa en la ejecución del caso de uso. Generalmente el actor estimula el sistema con eventos de entrada o recibe algo de él.

2.5.1 Diagrama de Caso de Uso del Sistema

A continuación, se presenta un diagrama donde se representa la relación existente entre los actores que deberán interactuar con el sistema.

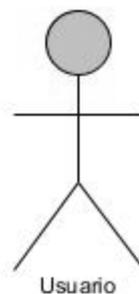


Ilustración 6: Jerarquía de usuarios del sistema (Elaboración propia)

A continuación, se muestran los casos de usos de dicho actor del sistema con el objetivo de favorecer la comprensión del mismo.



Ilustración 7: Casos de Uso inicializados por el usuario (Elaboración propia)

En el diagrama se identificó un total de 24 casos de uso, así como el actor que interviene en el proceso. Se identificaron varios CU que se consideran críticos para el correcto funcionamiento de la herramienta, el resto de los CU aparecen en el anexo #1:

- **CU Mostrar cantidad de títulos del sitio:** Se encarga de mostrar en una tabla la cantidad de etiquetas <h1>, <h2>, ..., <h6> tiene el sitio.
- **CU Mostrar meta descripción del sitio:** Se encarga de mostrar la una breve descripción del sitio, la cual es analizada por los motores de búsqueda a la hora de buscar determinada información.
- **CU Mostrar palabras claves del sitio:** Se encarga mostrar las palabras más importantes del sitio
- **CU Mostrar cantidad de enlaces del sitio:** Se encarga de mostrar la cantidad de etiquetas <href> del sitio.
- **CU Mostrar cantidad de enlaces internos del sitio:** Se encarga de mostrar la cantidad de enlaces internos del sitio.

- **CU Mostrar cantidad de enlaces externos del sitio:** Se encarga de mostrar la cantidad de enlaces externos del sitio.
- **CU Mostrar cantidad de enlaces rotos del sitio:** Se encarga de mostrar la cantidad de enlaces rotos del sitio.
- **CU Mostrar velocidad promedio de descarga del sitio:** Muestra el tiempo de descarga del sitio en milisegundos.
- **CU Optimización de Robot.txt:** Se encarga de mostrar si se el archivo Robot.txt se encuentra optimizado.
- **CU Código limpio:** Muestra si el código es correcto según los estándares internacionales de programación establecidos.
- **CU Creación y envío de sitemap a buscadores:** Comprueba si existe el archivo sitemap.xml.
- **CU Uso de diseño adaptable:** Se encarga de mostrar si el sitio presenta diseño adaptable.

2.5.2 Patrones de Casos de Uso utilizados

Los patrones de casos de uso permiten una mejor comprensión del comportamiento del sistema y generalmente son utilizados como plantillas que describen cómo deberían ser estructurados y organizados los casos de uso. A continuación, se describen aquellos empleados en la representación de los diagramas de casos de uso descritos en la sección anterior.

CRUD Parcial: El empleo de este patrón permite simplificar el tamaño y facilita el análisis del modelo mediante la asociación de varias operaciones dentro de un caso de uso, cuando todas estas contribuyen al mismo valor del negocio, y existe alguna que, por su complejidad, extensión o significación, debe modelarse como un caso de uso independiente.

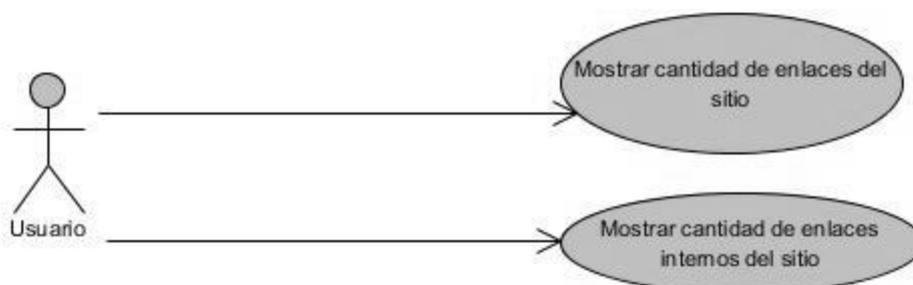


Ilustración 8: Ejemplo de uso del patrón CRUD Parcial (Elaboración propia)

2.5.3 Especificación de Casos de Uso

A continuación, se realiza la descripción del CU del sistema.

Tabla 3: Descripción del CU "Registrar nueva url"

Objetivo		Hacer uso de las diferentes técnicas y mecanismos de rastreo y parsing para generar un análisis.
Actores	Usuario.	
Resumen	El caso de uso se inicia cuando el usuario introduce la dirección o la url del sitio que se desea analizar.	
Complejidad	Alta.	
Prioridad	Alta.	
Precondiciones	El nombre o url del sitio debe ser válida.	
Postcondiciones	Se realizó correctamente el análisis, presione "Mostar resultados"	
Flujo de eventos		
Flujo básico Generar análisis		
Pasos	Actor	Sistema
1.	El actor introduce el nombre o url del sitio que desea analizar	
2.	Pulsa el botón asociado a la acción: <ul style="list-style-type: none"> Analizar 	
3.		Ejecuta el análisis y una vez terminado muestra la vista asociada a los resultados del análisis.
4.		Termina el CU

2.6 Estilo arquitectónico

En la arquitectura de software. Estos, indican los tipos de componentes y conectores involucrados, las restricciones de interconexión entre ellos, y el empleo de buenas prácticas para el diseño de software. Las buenas prácticas para el diseño de arquitecturas de sistemas informáticos, han sido descritas en forma de patrones arquitectónicos, los cuales han condicionado históricamente la toma de decisiones estructurales coherentes. A partir de la identificación de los estilos, se recomienda realizar un análisis crítico para su reutilización en las situaciones semejantes que se presenten en el futuro (Reynoso, 2004).

Para el desarrollo de la aplicación se decide emplear el patrón arquitectónico Modelo - Vista - Controlador (MVC).

El patrón MVC pertenece a la familia de los estilos de llamada y retorno, la cual enfatiza su trabajo en aumentar la capacidad de modificación y la escalabilidad de las aplicaciones. El mismo propone el diseño de arquitecturas de software, donde la lógica de negocio se encuentre separada de la interfaz gráfica de usuario, de manera tal que cualquier cambio en esta última no afecte la lógica de negocio y viceversa (Reynoso, 2004).

2.7 Patrones de diseño

De acuerdo a las valoraciones de Larman (2003), los patrones de diseño representan la descripción de un problema particular y recurrente, que aparece en contextos específicos, y presenta un esquema genérico demostrado con éxito para su solución; este último se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí.

De acuerdo a las valoraciones de Larman (2003), los patrones de diseño representan la descripción de un problema particular y recurrente, que aparece en contextos específicos, y presenta un esquema genérico demostrado con éxito para su solución; este último se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí.

En el diseño del módulo de configuración se tuvieron en cuenta los siguientes Patrones Generales de Software para Asignación de Responsabilidades (GRASP), que describen los principios fundamentales de la asignación de responsabilidades a objetos:

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

En la solución implementada se hace uso de este patrón en la clase seoResultados, debido a que esta clase conoce la información suya propia y la de Analizar, por lo cual puede hacer modificaciones y llamadas a ésta, para reutilizar código.



```
40     }
41     private void getPageData(Analizar analize, long timeResponse){
42         this.urlCode.setText(analize.getUrl());
43         this.textFieldUrl.setText(analize.getUrl());
44         URLLongitud.setText("<html>longitud: </html>"+analize.getUrl().length());
45         jLabelText1.setText("<html>Resultados para </html>"+analize.getUrl());
46         site_cBox.removeAllItems();
47         site_cBox.addItem(analize.getUrl());
48         list_urls = analize.links_text();
49         for(int i = 0; i<list_urls.getItemCount();i++){
50             site_cBox.addItem(list_urls.getItem(i));
51         }
52         DescriptionCode.setText("<html>"+analize.htmlDescription()+"</html>");
53         keywordsCode.setText("<html>"+analize.htmlKeyword()+"</html>");
54         titleCode.setText("<html>"+analize.htmlTitle()+"</html>");
55         TitleLongitud.setText("Longitud: "+analize.htmlTitle().length());
56         descripcionLongitud.setText("Longitud: "+analize.htmlDescription().length());
57         tableTitles.setText(getTable(analize.cantH1(), analize.cantH2(), analize.cantH3(), analize.cantH4(), analize.cantH5(), analize.cantH6()));
58         imgCode.setText("Se han encontrado "+analize.cantImg()+" imágenes");
59         imgAttribute.setText(analize.cantImgAttrAlt()+" con atributo alt");
60         palabras_claves.setText("Cantidad: "+ analize.htmlKeyword().length());
61         jLabelConsejos1.setVisible(false);
62         int[] links = analize.links();
63         enlace_interno1.setText("Internos: "+ links[0]);
64         enlace_externo.setText("Externos: "+ links[1]);
65         frame_1b.setText("Frame: "+ analize.cantFrame());
66         iframe_1b1.setText("Iframe: "+ analize.cantIframe());
67         vel_descarga.setText(Long.toString(timeResponse)+" ms");
```

Ilustración 9: Patrón de diseño experto de la clase seoResultados (Elaboración propia)

Creador: La instanciación de una clase es una de las actividades fundamentales en un sistema orientado a objetos. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, con lo que se logra menos dependencia y mayores oportunidades de reutilización de código.

Este patrón se pone de manifiesto en la creación del método principal de la clase Analizar.

```

33 public class Analizar {
34     private Document doc;
35     private String url;
36     public Analizar(String url) {
37         this.url = url;
38         try {
39             doc = Jsoup.connect(url).userAgent("Mozilla").get();
40         } catch (IOException e) {
41             e.printStackTrace();
42         }
43     }

```

Ilustración 10: Patrón de diseño controlador del método principal de la clase Analizar (Elaboración propia)

Bajo acoplamiento: El acoplamiento es una medida de la fuerza con que un elemento está conectado, tiene conocimiento o confía en otros elementos. El objetivo de este patrón consiste en mantener un bajo nivel de dependencia de otros elementos, por lo que constituye un principio que debe estar presente en todas las decisiones de diseño con lo que se reduce el impacto de los cambios.

Alta cohesión: En el diseño orientado a objetos, la cohesión es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento (clase o subsistema). Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que colaboran entre sí y con otros objetos para simplificar su trabajo. Una clase con alta cohesión es relativamente fácil de mantener, entender y reutilizar.

La aplicación de estos patrones de diseño se manifiesta, por ejemplo, en la implementación de las clases “Analizar” y “seoResultados”, creadas con el objetivo de manejar la información del sitio para posteriormente ser analizada. La clase “Analizar” es la encargada de realizar las operaciones de extracción y recopilación de la información, mientras que la clase “seoResultados” se encarga de realizar las operaciones de escritura del análisis del sitio.

2.8 Diagrama de Interacción

Los diagramas de interacción son utilizados para modelar los comportamientos dinámicos que caracterizan un sistema informático. Estos suelen representar un conjunto de objetos o clases y sus relaciones, así como los mensajes que se pueden enviar entre ellos. Entre los diagramas de interacción más conocidos en la ingeniería de software se encuentran el diagrama de secuencia y el diagrama de colaboración.

2.8.1 Diagrama de secuencia

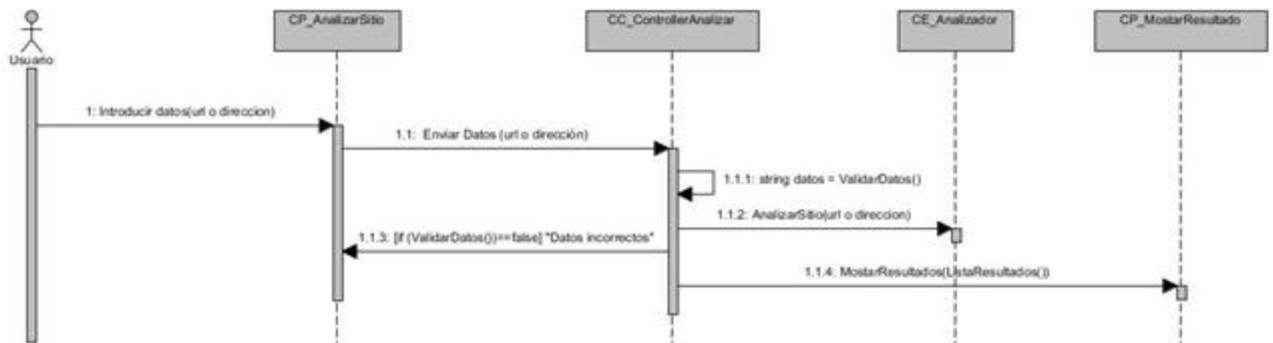


Ilustración 11:Diagrama de secuencia del CU "Registrar nueva url" (Elaboración propia)

Conclusiones parciales

Durante el análisis y diseño de la herramienta informática para análisis de posicionamiento web, la representación y descripción de los artefactos generados garantizaron un mejor entendimiento de los flujos de trabajos presentes en el proceso de análisis SEO. La especificación de los requisitos funcionales y no funcionales del sistema, dieron paso a una mejor comprensión, por parte de los autores, de los resultados que se pretenden obtener de una manera precisa y sirvieron de guía para la codificación del sistema. La definición de la arquitectura y los patrones de diseño a utilizar, permitieron establecer las bases para fomentar la reutilización y las buenas prácticas de programación entre los desarrolladores durante la fase de implementación, así como disminuir el impacto de los cambios futuros en el código fuente.

Capítulo 3. Implementación y validación del sistema.

Introducción

La implementación del sistema es una de las fases imprescindibles dentro del proceso de desarrollo de software. Esta fase comprende la materialización, en forma de código, de todos los artefactos, descripciones y arquitectura propuestos en la etapa de análisis y diseño; con el objetivo de conformar el producto final requerido por el cliente.

Aparejado al proceso de implementación, el software que se construye debe ser sometido a determinadas pruebas que corroboren la correspondencia entre el producto y los requisitos definidos en las etapas anteriores. A esta etapa se le conoce como validación del sistema y en ella, pueden realizarse diferentes tipos de pruebas en función de los objetivos de las mismas.

3.1 Modelo de componentes que integran la solución informática

El modelo de componentes representa la forma en que es estructurado un sistema informático atendiendo a las diferentes partes que lo componen. Partiendo de este punto, Sommerville (2005) puntualiza que cada componente debe ser tratado como una unidad de composición independiente e indispensable dentro de un sistema, y que puede contraer relaciones de dependencia con otros componentes. Algunos ejemplos de componentes físicos lo constituyen los archivos, módulos, librerías, ejecutables, binarios, entre otros.

3.1.1 Diagrama de componentes

Este tipo de diagrama ilustra la relación que existe entre componentes de software, así como la ubicación de cada uno de ellos dentro del sistema. A continuación, se muestra el diagrama de componentes de la herramienta informática de análisis de posicionamiento web; cuya organización se encuentra acorde con la arquitectura MVC propuesto por Spring y descrita en el capítulo anterior de este trabajo.

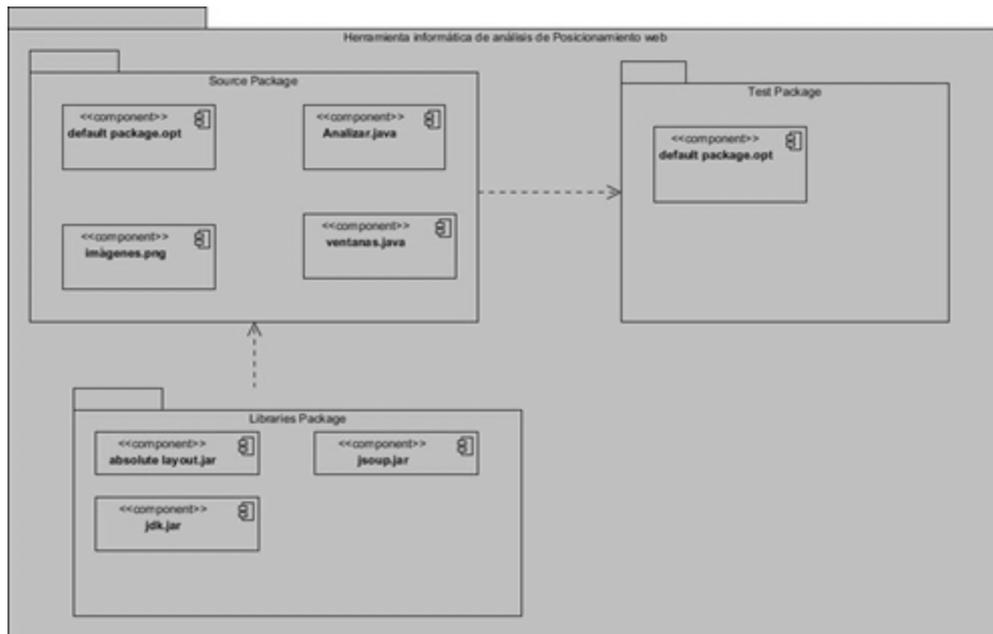


Ilustración 12:Diagrama de componentes.(Elaboración propia)

3.2 Estándares de codificación

Los estándares de codificación son especificaciones o estilos que establecen la forma de generar el código funcional de las aplicaciones informáticas. Puesto que, en muchas ocasiones, los sistemas de cómputo son implementados por varios programadores, la adopción inicial de un único estilo de codificación constituye uno de los factores de mayor peso en la calidad, rendimiento, legibilidad y capacidad de mantenimiento del producto final.

Para facilitar el entendimiento del código y precisar un modelo a seguir, se adoptaron determinados estándares de codificación que a continuación se describen, agrupados por los aspectos en los que fueron utilizados.

Identificadores:

Para la definición del nombre de los métodos se utilizó el estilo *lowerCamelCase*. Este estilo establece que la separación entre palabras internas de los identificadores deberá realizarse escribiendo la letra inicial en mayúscula, a excepción de la primera palabra. Además, no deberá colocarse ningún carácter especial entre palabras de los identificadores.

Ejemplos de uso:

- Funciones:

Private String htmlTitle()

```
Private void htmlParser()
```

Indentación

Esta costumbre enfatiza en comenzar a escribir cada línea de código a diferentes distancias desde el borde izquierdo del área de edición. La distancia deberá regirse por la jerarquía que se forma al introducir sentencias dentro de bloques de estructuras. Gracias al uso de NetBeans como IDE de desarrollo, los espacios de indentación son ajustados automáticamente, permitiendo a los programadores enfocarse en otras funciones de mayor importancia.

Para establecer cierta homogeneidad y mayor legibilidad, se escribirá cada sentencia en una línea de código, y si fuera necesario cortar las líneas, se hará luego de una coma o antes de un operador. La sección de la derecha de la línea que se corte se ubicará en la línea siguiente indentada al nivel de la expresión correspondiente en la línea superior.

Llaves

En cuanto a la ubicación de estos operadores de delimitación, existen diversos criterios en los diferentes lenguajes de programación que lo usan para la definición de bloques de código. En este trabajo las llaves de apertura se colocarán inmediatamente al final de la línea de cabecera del bloque, así como en las estructuras *if*, *for*, *while*, *else*, *switch*, *foreach*. Las llaves de cierre se colocarán solitarias en la línea siguiente a la última línea dentro del bloque e indentadas al nivel de la línea cabecera del bloque.

3.3 Principales pantallas de la herramienta informática de análisis de posicionamiento web

Para ilustrar de una mejor manera las funcionalidades implementadas, se muestran a continuación varias imágenes de la herramienta informática de análisis de posicionamiento web, así como una breve descripción de cada una de ellas.



Ilustración 13: Interfaz de inicio de la herramienta. (Elaboración propia)

En esta interfaz es donde se inserta la dirección del sitio que se desea analizar, para su posterior análisis.



Ilustración 14: Interfaz de resultados de la herramienta (Elaboración propia)

En esta interfaz se muestran los resultados luego de haberse realizado el análisis SEO. En el mismo se muestran las principales métricas analizadas del sitio, el estado en la cual se encuentran, además de consejos que ayudarían a mejorar dicho aspecto.

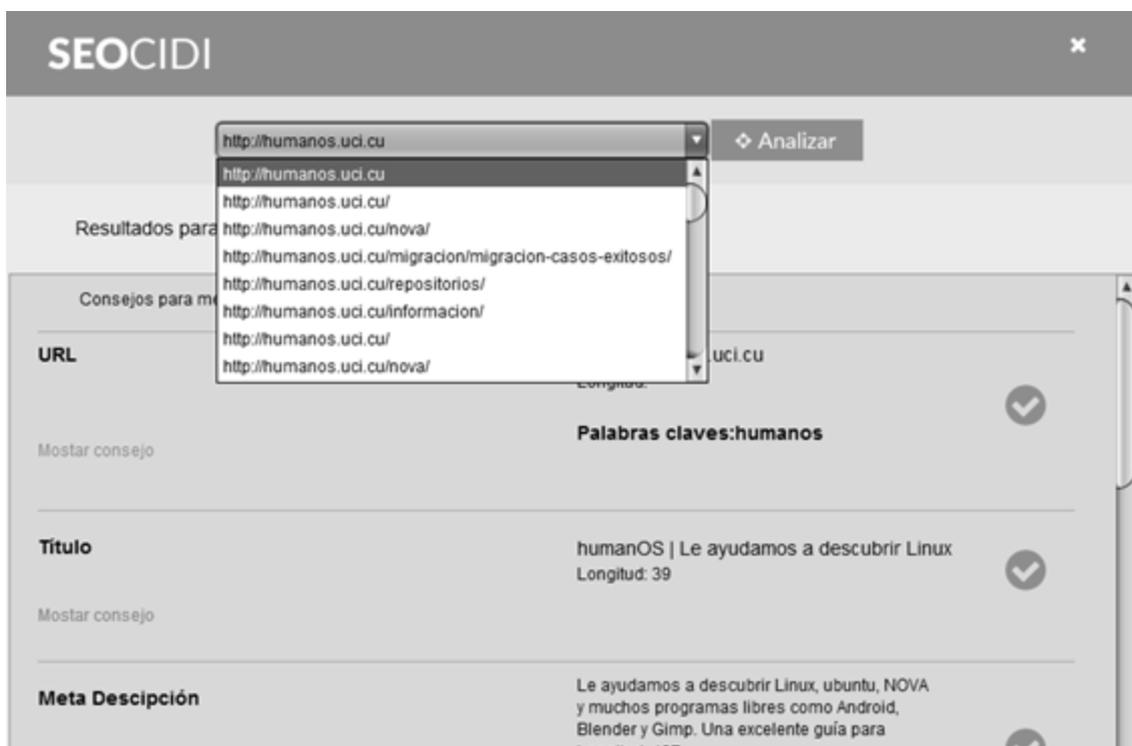


Ilustración 14: Interfaz de resultados de la herramienta (Elaboración propia)

La herramienta además brinda la posibilidad de realizar el mismo análisis a todos los enlaces del sitio, lo cual contribuye a un análisis de posicionamiento web más concreto.

3.4 Validación de la herramienta de análisis de posicionamiento web implementada

A continuación, se detallan los tipos de pruebas de software aplicados a la herramienta de posicionamiento web implementada. Las mismas persiguen como objetivo fundamental, la detección de las no conformidades respecto a las funcionalidades de la aplicación, la medición del grado de usabilidad de las funcionalidades implementadas, así como también la correcta integración entre los diferentes componentes de la arquitectura del sistema.

3.4.1 Pruebas funcionales

Las pruebas funcionales son aquellas que se aplican a un software determinado, con el objetivo de validar que las funcionalidades implementadas funcionen de acuerdo a las especificaciones de los requisitos definidos con anterioridad. Para la ejecución de este

tipo de pruebas, suelen emplearse dos métodos fundamentales: el método de Caja Blanca y el método de Caja Negra. El primero se centra en las pruebas al código de las aplicaciones; mientras que el segundo permite a los probadores enfocar su atención en el funcionamiento de la interfaz, a través del análisis de los datos de entrada y los de salida.

En este epígrafe se exponen los aspectos concernientes a las pruebas funcionales realizadas utilizando el método de Caja Negra, a partir de los casos de prueba diseñados, empleando la técnica Partición Equivalente.

Diseño de los casos de prueba basados en casos de uso

A continuación, se muestra un fragmento del diseño del caso de prueba basado en caso de uso "Registrar nueva url".

Caso de uso: Registrar nueva url

Escenarios: 1.1 y 1.2

Condiciones de ejecución: Ninguna

Tabla 4: Fragmento del caso de prueba basado en el CU "Registrar nueva url"

Escenario	Descripción	Dirección	Respuesta de sistema	Flujo central
EC 1.1 Insertar la url de forma exitosa.	El sistema introduce la url de forma exitosa.	V	El sistema procede a analizar la url y luego muestra los resultados en la interfaz seoResultados.	1- El usuario inserta la url del sitio deseado 2- El usuario pulsa el botón " Analizar ". 3- El sistema muestra la interfaz seoResultados
EC 1.2 Insertar la url de forma incorrecta.	El sistema introduce la url de forma incorrecta	I	El sistema no encuentra la url insertada y muestra el mensaje Error: url no existe o es incorrecta	

Variables empleadas en el diseño del caso de prueba basado en el CU "Registrar nueva url"

Tabla 5: Variables empleadas en el diseño del caso de prueba basado en el CU "Registrar nueva url"

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	url	Campo de texto	No	Debe tener el formato valido para una url

Resultados de las pruebas funcionales

Para llevar a cabo la detección de las no conformidades presentes en la aplicación desarrollada, se realizaron 2 iteraciones de pruebas funcionales. En la siguiente tabla se muestran los resultados obtenidos en cada iteración de prueba a la herramienta de análisis de posicionamiento web, así como la corrección de cada uno de los errores.

Tabla 6: Cantidad de no conformidades por cada iteración de las pruebas funcionales

No conformidades	1ra iteración	2da iteración
Detectadas	13	7
Resultadas	5	7
Pendientes	7	0

Las no conformidades detectadas pueden ser agrupadas atendiendo a diferentes tipologías. En la siguiente figura se muestra una gráfica con la cantidad de no conformidades detectadas por cada tipo.

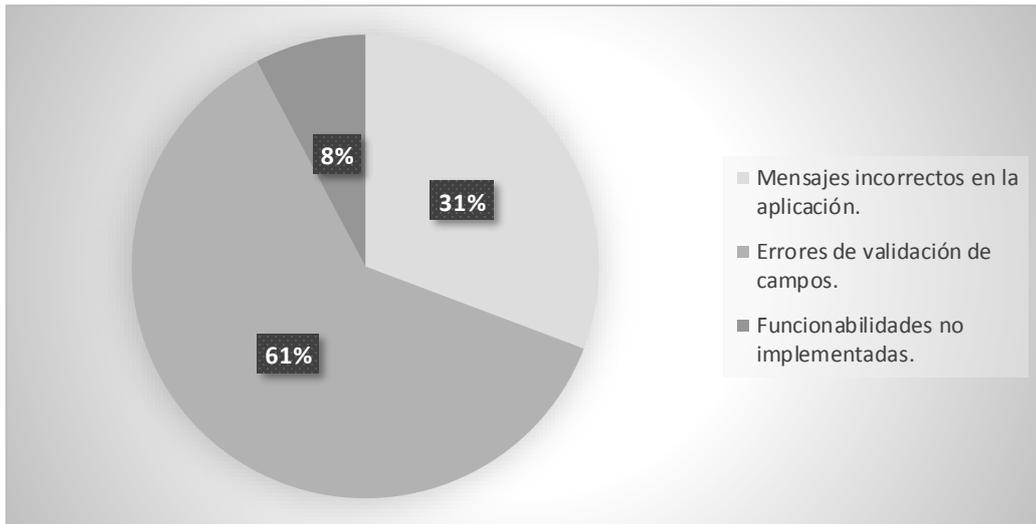


Ilustración 15: Cantidad de no conformidades detectadas por tipo. (Elaboración propia)

3.4.2 Pruebas de usabilidad

En el contexto del desarrollo de software, la usabilidad está considerada como uno de los factores de calidad de mayor importancia para el éxito de un proyecto. De manera general, el término usabilidad es empleado para referirse a la capacidad que posee un producto de ser utilizado por los usuarios de forma fácil, eficiente y con satisfacción, en un determinado contexto de uso. Jakob Nielsen (2010), en su libro *Usability Engineering*, describe la usabilidad como un término multidimensional en el que intervienen cinco atributos fundamentales: capacidad de aprendizaje, eficiencia en el uso, facilidad de memorizar, tolerante a errores y subjetivamente satisfactorio.

La realización de pruebas de usabilidad contribuye, en cierta medida, a la adquisición de aplicaciones de alta calidad y gran facilidad de uso por parte de los usuarios finales. Para ello, se utilizó una lista de chequeo que proporciona un conjunto de preguntas clasificadas en varias categorías. A continuación, se muestran los resultados de la aplicación de la lista de chequeo a la herramienta informática de análisis de posicionamiento web.

Tabla 7: Resultados de prueba de usabilidad utilizando lista de chequeo

Categorías de indicadores	Indicadores	Correctos
Visibilidad del sistema	7	5
Adecuación del sistema	16	14
Consistencia y estándares	17	14

Flexibilidad y eficiencia de uso	7	7
Estética	21	19
Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores	3	3
Accesibilidad	35	21
Comprobaciones técnicas	9	8
Total	115	91

En la tabla anterior se puede apreciar que inicialmente, de un total de 115 indicadores de usabilidad, el módulo de configuración web implementado cumple con 91 indicadores, cifra que representa el 79,13 %, luego fueron corregidos 12 indicadores para una usabilidad resultante de 89,57%.

3.4.3 Evaluación del tiempo de resultado de las consultas de la herramienta informática de análisis de posicionamiento web.

Con el propósito de evaluar el indicador referente al tiempo de resultado de las consultas realizadas al sitio, correspondiente a la variable dependiente definida como parte de la hipótesis de investigación, se realizó un pre experimento donde se comparan los resultados obtenidos teniendo en cuenta el uso de la herramienta desarrollada y en segundo lugar sin hacer uso de la herramienta, haciendo uso del inspector de código. Para esta tarea se realizó un análisis a 3 sitios de la universidad, analizando de estos un total de 5 páginas.

La ejecución de dicho pre experimento arrojó los siguientes resultados en minutos:

Sitio humanos.uci.cu

Tabla 8: Resultados del pre experimento para el sitio humanos.uci.cu

Parámetros analizados	Haciendo uso del sistema	Forma manual
Títulos	0:04:36	0:17:48
Meta descripción	0:02:16	0:07:30

Palabras claves	0:01:21	0:04:25
Cantidad de imágenes	0:02:10	0:12:06
Número de enlaces	0:03:21	0:11:03
Número de frames	0:01:25	0:03:27

Sitio android.uci.cu

Tabla 9:Resultados del pre experimento para el sitio android.uci.cu

Parámetros analizados	Haciendo uso del sistema	Forma manual
Títulos	0:02:35	0:09:48
Meta descripción	0:01:16	0:04:30
Palabras claves	0:02:21	0:06:25
Cantidad de imágenes	0:04:10	0:09:06
Número de enlaces	0:03:03	0:08:54
Número de frames	0:02:25	0:07:27

Sitio dragones.uci.cu

Tabla 10:Resultados del pre experimento para el sitio dragones.uci.cu

Parámetros analizados	Haciendo uso del sistema	Forma manual
Títulos	0:07:19	0:18:46
Meta descripción	0:03:14	0:07:28
Palabras claves	0:01:47	0:04:30
Cantidad de imágenes	0:02:17	0:06:04

Número de enlaces	0:02:21	0:11:43
Número de frames	0:01:20	0:03:27

A partir del análisis de la comparación de medias y desviaciones estándar obtenidas en las pruebas, se evidencia una reducción significativa del tiempo del proceso de configuración mediante el uso de la herramienta implementada.

3.5 Conclusiones parciales

La deficiente realización de pruebas de software a los sistemas de cómputo, constituyen hoy en día una de las causas de mayor impacto que provoca la baja calidad de las aplicaciones informáticas. Por esta razón, la ejecución de pruebas a la herramienta de análisis de posicionamiento web permitió detectar las deficiencias presentes, subsanarlas en el menor tiempo posible y ofrecer una aplicación con mayor calidad, seguridad y usabilidad.

Conclusiones generales

En el presente trabajo se ha llevado a cabo un proceso de desarrollo de software completo, dividido en flujos de trabajos e iteraciones, con el objetivo de lograr un producto de calidad en el tiempo establecido. Una vez completada la presente investigación, se puede concluir que:

- A partir de los fundamentos teóricos analizados en la presente investigación, se determinó que existe gran diversidad en cuanto a las herramientas SEO existentes. Estas, constituyen sistemas informáticos complejos y con un gran número de parámetros SEO analizados, por lo que es necesario una explicación detallada que auxilien a los desarrolladores a tener una mejor comprensión de los mismos.
- Una vez estudiados los elementos que intervienen en la herramienta informática de análisis de posicionamiento web, fue posible la modelación de los artefactos que contribuyeron al diseño de la propuesta de solución. Se identificaron los patrones de diseño y arquitectura, que por sus características, ofrecen mayor soporte a la implementación de los requisitos previamente expresados por el cliente; garantizando la estructura base para la organización lógica del código fuente y la disminución del impacto ante futuras modificaciones en la aplicación.
- Se obtuvo una herramienta de análisis SEO que permitirá la detección temprana de errores de posicionamiento desde las primeras etapas del proceso de desarrollo de software.
- La evaluación de las pruebas de software realizadas permitió erradicar las insuficiencias detectadas a la herramienta desarrollada, que comprometían la facilidad de uso de las funcionalidades presentes.

Recomendaciones

Se recomienda como parte de la investigación:

- Implementar nuevas métricas que ayuden a realizar un análisis más exhaustivo de los sitios web.

Referencias bibliográficas

ALARCÓN N., S., M. y REYES P., Y. *Sistema para la toma de decisiones del Webmaster en el ámbito del Posicionamiento Web*. Tesis de Ingeniería Informática. Universidad de las Ciencias Informáticas. La Habana, Cuba, 2009.

AgentWebRanking. "Search engine ranking software for SEO Experts." Retrieved 26 abril del 2008, 2008, from <http://www.agentwebranking.com/>.

Clements, P. (1996). Coming attractions in Software Architecture. Alemania.

Codina, L. "Posicionamiento Web: Conceptos y Ciclo de Vida." Retrieved 26 de abril del 2008, 2008, from http://www.hipertext.nefflNeb/paq216_print.htm.

David Garlan, M. S. (1993). An Introduction to Software Architecture. New Jersey, World Scientific Publishing Company.

evosdesigner. Retrieved 26 abril del 2008, 2008, from <http://www.evosdesign.com/>.

Exchange, L. (2006). "Link Building is what we do!" Retrieved 26 abril del 2008, 2008, from <http://www.linkexchanaeexperts.com>.

Iniesta, V. (2006, 5 de abril de 2006). "Recuperacion y Organizacion de la Informacion." Retrieved 10 de diciembre, 2007, from <http://www.telefonica.net/web2/herramientas-seo/herramientasmeiora.html>.

Larman, C. "UML y Patrones." 507.

Muñiz, R. Marketing en el Siglo XXI.

Myrosoft. "Active WebTraffic." Retrieved 26 abril del 2008, from <http://www.myrosoft.com/activewebtraffic/activewebtrafficsp.htm>.

Reino, A. (2000). "Introducción a Xml en Castellano."

Reynoso, C. B. (2006, 10 diciembre 2007). "Introducción a la Arquitectura de Software." from http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arg/intro.mspix.

Ugarte, D. d. (2002). El libro del posicionamiento en buscadores. Madrid.

TOLOSA, G. H.; BORDIGNON, F. R. A. *Introducción a la Recuperación de Información*. 2008.

Bibliografía

"Promotor Web - Software para alta y posicionamiento en buscadores en español".

"Vocabulario SEO y Terminos Google." Retrieved 28 de noviembre., 2007, from <http://www.xeoweb.com/buscadores/terminos-seo.php>.

(1991). "ISO 9126: The Standard of Reference."

(2006). "Arquitectura de aplicaciones de .NET: Diseño de aplicaciones y serftios."

(2007). "Introducción a Microsoft .NET Framework 3.0 ".

(2008). "Fase de Elaboración."

abansys. "Posicionamiento Web ¿En qué consiste?" Retrieved Abril 26 del 2008., 2008, from [http://www.abansys.com/posicionamientoWeb en_que_consiste.html](http://www.abansys.com/posicionamientoWeb_en_que_consiste.html).

AgentWebRanking. "Search engine ranking software for SEO Experts." Retrieved 26 abril del 2008, 2008, from <http://www.agentwebranking.com/>.

Clements, P. (1996). Coming attractions in Software Architecture. Alemania.

Clements, P. (1996). A Survey of Architecture Description Languages. Alemania.

Codina., L. "Posicionamiento Web: Conceptos y Ciclo de Vida." Retrieved 26 de abril del 2008, 2008, from http://www.hipertext.neffWeb/pag216_print.htm.

David Garlan, M. S. (1993). An Introduction to Software Architecture. New Jersey, World Scientific Publishing Company.

evosdesign. Retrieved 26 abril del 2008, 2008, from <http://www.evosdesign.com/>.

Exchange, L. (2006). "Link Building is what we do!" Retrieved 26 abril del 2008, 2008, from <http://www.linkexchangeexperts.com>.

Iniesta, V. (2006, 5 de abril de 2006). "Recuperación y Organización de la Información." Retrieved 10 de diciembre, 2007, from <http://www.telefonica.net/web2/herramientas-seo/herramientasmejora.html>.

Kiccillof, C. R.-N. (2004). Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.

Larman, C. "UML y Patrones." 507.

Muñiz, R. Marketing en el Siglo XXI.

Myrosoft. "Active WebTraffic." Retrieved 26 abril del 2008, 2008, from <http://www.myrosoft.com/activewebtraffic/activewebtrafficsp.htm>.

R. Kazman, M. K., P. Clements (1998). Evaluating Software Architectures for Real-Time System.

Real., U. d. C.-L. M. E. d. I. d. C. Arquitectura Multicapa. Prácticas Ingenieras del Software 3.

Reino, A. (2000). "Introducción a Xml en Castellano."

Reynoso, C. B. (2006, 10 diciembre 2007). "Introducción a la Arquitectura de Software." from http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.mspix.

Rodríguez, A. "Diseño de Aplicaciones Three Tier." Retrieved 3 de diciembre, 2007, from <http://www.fpress.com/revista/Num9711/Nov97.htm>.

Seco, J. A. G. (2001). "El lenguaje de programación C#."

SEOguru. (2007). "uso de los Backlinks en el Web." Retrieved 28 de noviembre, 2007, from <http://www.pagoclick.com.mx/blog/2007/10/24/uso-de-los-backlinks-en-el-Web/>.

SUBMISSION, D. "Search Engine Submission Software ", from <http://www.dynamicsubmission.com>.

Anexos

Anexo #1

A continuación, se muestran los casos de uso presentes en diseño de la herramienta:

- **CU Registrar nueva url:** Se encarga de analizar la url insertada en la interfaz de inicio de la aplicación
- **CU Mostrar url del sitio:** Muestra el nombre de la url.
- **CU Mostrar palabras claves de la url:** Muestra las palabras más importantes de la url.
- **CU Mostrar título del sitio:** Muestra el título del sitio.
- **CU Mostrar longitud del título del sitio:** Muestra la longitud del título del sitio.
- **CU Mostrar longitud de la meta descripción del sitio:** Muestra la longitud de la meta descripción del sitio.
- **CU Mostrar la cantidad de imágenes del sitio:** Muestra la cantidad de imágenes que contiene el sitio.
- **CU Mostrar la cantidad de imágenes del sitio con atributo alt:** Muestra la cantidad de imágenes con atributo alt que contiene el sitio.
- **CU Mostrar la cantidad de imágenes del sitio con atributo title:** Muestra la cantidad de imágenes con atributo title que contiene el sitio.
- **CU Mostrar la cantidad de imágenes del sitio con atributo description:** Muestra la cantidad de imágenes con atributo description que contiene el sitio.
- **CU Mostrar número de frame del sitio:** Muestra la cantidad de frame (marcos) que contiene el sitio.
- **CU Mostrar número de iframe del sitio:** Muestra la cantidad de iframe (marcos) que contiene el sitio.