

Universidad de las Ciencias Informáticas
Facultad 8.



**Título: “Sistema para la toma de decisiones del
Webmaster en el ámbito del Posicionamiento Web”.**

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

Autores: Natalia M. Alarcón Soto.
Yasniel Reyes Piloto.

Tutor: Ing. Eduardo M. Macías Sotolongo.

Ciudad de La Habana, Junio de 2009
“Año del 50 aniversario del triunfo de la Revolución”



“El futuro de Cuba tiene que ser necesariamente un futuro de hombres de ciencia, de hombres de pensamiento”



Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yasniel Reyes Piloto

Natalia Alarcón Soto

Ing. Eduardo M. Macías Sotolongo.



Agradecimientos

A nuestros padres y familiares que siempre nos apoyaron.

A nuestros amigos que siempre nos han acompañado en los malos y buenos momentos.

A Nelson, primer líder de proyecto y primer tutor del que aprendimos mucho.

A los muchachos del proyecto quienes estuvieron desde los inicios apoyándonos y dándonos nuevas ideas.

A nuestro tutor Eduardo por ayudarnos y por su disposición en todo momento por ayudarnos.

A todos los compañeros de aula que de una forma u otra siempre se preocuparon y nos apoyaron.

Natalia y Yasniel.



Dedicatoria

A mi mamá que en estos momentos se que estaría muy orgullosa de mí por mis resultados obtenidos y por el amor que siempre me dio.

A mi papá por el apoyo que y amor que me ha brindado.

A mis tías y a mi hermano por apoyarme siempre en los momentos más difíciles de mi vida y en general a toda mi familia.

A mi novio por todo el amor que me ha brindado y por su apoyo.

A mis amigos, el piquete que han estado siempre compartiendo los buenos y malos momentos y en especial a Reimis, que siempre a estado apoyándome.

A mis compañeros de aula por compartir estos maravillosos años de mi vida.

A todas las personas que de una forma u otra han contribuido a mi formación para la vida y como profesional.

Natalia



Dedicatoria

A mis padres por apoyarme, por el amor que me han brindado y por estar siempre pendientes de mis estudios.

A mi novia por el amor que siempre me ha dado, por el apoyo incondicional, por su paciencia y sobre todas las cosas porque me ha sabido comprender.

A mis hermanas y sobrino que siempre me han apoyado, por ser las personas más especiales de mi vida después de mis padres.

A mis amigos, el piquete que han estado siempre compartiendo los buenos y malos momentos y en especial a Reimis, que siempre me ha estado apoyándome.

A mis compañeros de aula por compartir estos maravillosos años de mi vida.

A todas las personas que de una forma u otra incidieron en mi vida profesional.

Yasniel



Resumen

La investigación recoge todo el proceso de desarrollo del sistema para la ayuda a la toma de decisiones del Webmaster en el ámbito de Posicionamiento Web.

Para cumplir con los propósitos de la investigación se realizó un estudio de los principales sistemas existentes tanto a nivel nacional como internacional, relacionados con el Posicionamiento Web. Se utilizó Java como lenguaje de programación, como metodología de desarrollo el Proceso Racional Unificado (RUP) junto al Lenguaje Unificado de Modelado (UML); como herramienta de modelado Visual Paradigm y NetBeans como plataforma de desarrollo. Se documentaron y generaron los diferentes artefactos necesarios para la construcción de la aplicación como establece la metodología utilizada.

La aplicación logrará elevar la calidad de las páginas Web mediante la mejora de su código y su contenido para obtener una posición favorable en los buscadores y de esta forma hacer llegar la información exacta que brindan los medios cubanos a una mayor cantidad de usuarios.

La herramienta le da la posibilidad al Webmaster de analizar las páginas web que desee. Le permite validar la estructura interna del HTML para detectar y corregir los errores posibles con el fin de eliminar las malas prácticas que atentan contra el posicionamiento que pueden ser considerados código basura. Obtener un listado de los criterios de búsqueda que mejor recupera la información, así como la frecuencia, peso, densidad y relevancia de cada palabra reflejada en el documento. Analizar las etiquetas importantes para los buscadores que inciden directamente en el Posicionamiento Web. Construir un árbol que mide la profundidad de la página. Generar reportes con las características de la página.

Palabras Claves: Buscadores, Internet, Posicionamiento Web, SEO, Sistema.



Índice

Agradecimientos	IV
Dedicatoria.....	V
Resumen.....	VII
Introducción	14
Capítulo 1: Fundamentación Teórica.....	19
1.1 Introducción.....	19
1.2 Elementos Fundamentales para entender Posicionamiento Web.	19
1.2.1 Internet.....	19
1.2.2 Buscadores.	20
1.2.3 Posicionamiento Web.....	22
1.3 Herramientas relacionadas con Posicionamiento Web.....	29
1.4 Lenguaje de Modelado.....	34
1.4.1 UML.....	34
1.5 Metodologías de Desarrollo.....	34
1.5.1 RUP (Proceso Unificado de Desarrollo).	34
1.6 Lenguaje de Programación.....	35
1.6.1 Java.....	35
1.7 Herramienta CASE (Computer-Aided Software Engineering).	36
1.7.1 Visual Paradigm v 5.3.	36
1.8 Plataforma de Desarrollo.	37
1.8.1 NetBeans v6.5.	37



1.9 Java y los reportes.....	37
1.9.1 Ireport v1.3.3.....	37
1.10 Conclusiones.....	38
Capítulo 2: Análisis y Diseño del Sistema.....	39
2.1 Introducción.....	39
2.2 Propuesta de sistema.....	39
2.3 Requerimientos del sistema.....	41
2.3.1 Requerimientos funcionales.....	41
2.3.2 Requerimientos no funcionales.....	42
2.4 Diagrama de Caso de Uso del Sistema (CUS).....	43
2.5 Análisis y Diseño.....	44
2.5.1 Modelo de Análisis.....	45
2.5.1.1 Diagramas de clases del análisis.....	45
2.5.2 Modelo de Diseño.....	49
2.5.2.1 Diagramas de interacción.....	49
2.5.2.2 Diagramas de clases del diseño.....	55
2.6 Estilo Arquitectónico.....	59
2.6.1 Patrones Arquitectónicos.....	59
2.6.2 Patrones de diseño.....	60
2.7 Prototipo de Interfaz de Usuario.....	61
2.8 Descripción de las clases más importantes.....	68
2.9 Conclusiones.....	73
Capítulo 3: Implementación y Prueba del Sistema.....	75



3.1 Introducción.....	75
3.2 Diagrama de componente.....	75
3.2.1 Estándares de codificación.....	81
3.2.2 Librerías utilizadas.....	82
3.3 Prueba.....	84
3.3.1 Pruebas aplicadas.....	85
3.3.2 Casos de prueba.....	86
3.3.2.1 Casos de prueba de Caja Blanca.....	87
3.3.2.2 Casos de prueba de Caja Negra.....	91
3.3.3 Análisis de los defectos encontrados.....	94
3. 4 Conclusiones.....	96
Recomendaciones.....	98
Referencias bibliográficas.....	99
Bibliografía.....	100
Anexos.....	103



Índice de tablas.

Tabla 1 Diagrama de Caso de Uso del Sistema (CUS).....	44
Tabla 2 Diagrama de Clase del Análisis del CUS: Mostrar comparaciones.....	45
Tabla 3 Diagrama de Clase del Análisis del CUS: Conocer características de una página.	46
Tabla 4 Diagrama de Clase del Análisis del CUS: Mostrar consulta.....	46
Tabla 5 Diagrama de Clase del Análisis del CUS: Mostrar árbol.....	47
Tabla 6 Diagrama de Clase del Análisis del CUS: Validar HTML.....	47
Tabla 7 Diagrama de Clase del Análisis del CUS: Mostrar reporte.....	47
Tabla 8 Diagrama de Clase del Análisis del los CUS: Autenticar.	48
Tabla 9 Diagrama de Clase del Análisis del CUS: Gestionar meta.....	48
Tabla 10 Diagrama de Clase del Análisis del los CUS: Descargar y analizar HTML.	49
Tabla 11 Diagrama de Secuencia perteneciente al CUS: Autenticar.....	50
Tabla 12 Diagrama de Secuencia perteneciente al CUS Conocer características de una página: Información Adicional.....	51
Tabla 13 Diagrama de Secuencia perteneciente al CUS Gestionar Meta: Eliminar etiqueta meta.	51
Tabla 14 Diagrama de Secuencia perteneciente al CUS: Mostrar árbol.....	52
Tabla 15 Diagrama de Secuencia perteneciente al CUS Gestionar Meta: Insertar meta.....	52
Tabla 16 Diagrama de Secuencia perteneciente al CUS: Mostrar consulta.....	53
Tabla 17 Diagrama de Secuencia perteneciente al CUS: Mostrar reporte.....	53
Tabla 18 Diagrama de Secuencia perteneciente al CUS: Validar HTML.....	54
Tabla 19 Diagrama de Secuencia perteneciente a los CUS: Descargar HTML y Analizar HTML.	54
Tabla 20 Diagrama de Secuencia perteneciente al CUS Características de una página: Mostrar etiqueta.....	55
Tabla 21 Diagrama de Clase perteneciente al CUS: Autenticar.	56
Tabla 22 Diagrama de Clase perteneciente al CUS: Reporte.....	56



Tabla 23 Diagrama de Clase perteneciente a los CUS: Descargar y analizar HTML.	57
Tabla 24 Diagrama de Clase de los CUS: (Restantes).....	58
Tabla 25 Interfaz de usuario: Principal.	62
Tabla 26 Interfaz de usuario: Validar HTML.	63
Tabla 27 Interfaz de usuario: Mostrar árbol.....	64
Tabla 28 Interfaz de usuario: Información Autenticar.	65
Tabla 29 Interfaz de usuario: Información adicional.	65
Tabla 30 Interfaz de usuario: Mostrar cantidad de etiquetas.	66
Tabla 31 Interfaz de usuario: Ayuda.....	66
Tabla 32 Interfaz de usuario: Reporte.	67
Tabla 33 Interfaz de usuario: Mostrar comparaciones.....	67
Tabla 34 Descripción de la clase Página.....	69
Tabla 35 Descripción de la clase Anlaizador_Sintactico.....	70
Tabla 36 Descripción de la clase Etiqueta.....	71
Tabla 37 Descripción de la clase Leer_Codigo_Html.	71
Tabla 38 Descripción de la clase Autenticar.....	72
Tabla 39 Descripción de la clase ErroresDataSource.	72
Tabla 40 Descripción de la clase Controladora.	73
Tabla 41 Diagrama de paquete.....	76
Tabla 42 Diagrama de componente del paquete Interfaz de Usuario.	76
Tabla 43 Diagrama de componente del paquete HTML.	77
Tabla 44 Diagrama de componente del paquete Analizador HTML.....	77
Tabla 45 Diagrama de componente del paquete Descargar Página.....	78



Tabla 46 Diagrama de componente del paquete Librerías.	78
Tabla 47 Diagrama de componente del paquete Reportes.	79
Tabla 48 Diagrama de componente de los paquete Controladora.....	79
Tabla 49 Diagrama de componente General.....	80
Tabla 50 Mensajes de error.	83
Tabla 51 Clases de Equivalencia del CU Autenticar.	92
Tabla 52 Clases de Equivalencia del CU Gestionar_Meta.	93
Tabla 53 Clases de Equivalencia del CU Conocer_características_página.....	94
Tabla 54 Análisis de los defectos encontrados.	95
Tabla 55 Análisis de los defectos encontrados.	95



Introducción

El desarrollo de las tecnologías de la información ha evolucionado vertiginosamente en la gran Red de Redes por el incremento de la competencia y el aumento creciente de los usuarios en Internet. Cuba, no se encuentra aislada de este fenómeno e intenta insertarse en el complicado mundo del software, aunque ha pasado por una serie de manipulaciones y limitaciones que le impiden pertenecer a éste.

Por muchas décadas, Cuba ha sido agredida por el Gobierno de los Estados Unidos con el objetivo de destruirla y desmoralizarla ante el mundo. La hostil política norteamericana como parte de sus intentos por derrocar a la Revolución cubana ha mantenido no solo el Bloqueo Económico, Comercial y Financiero, sino que ha utilizado las facilidades que brinda Internet para publicar y tergiversar toda clase de información con el propósito de enmascarar la realidad de la Revolución a través de los medios de comunicación existentes en la red. Esto ha dado lugar a que Cuba busque nuevas estrategias con el fin de contrarrestar la manipulación de la información y las mentiras difundidas por el gobierno norteamericano.

Años atrás era difícil acceder a la información en Internet por la gran cantidad de documentos y páginas existentes. Por esta razón han surgido sistemas automáticos de recuperación de información llamados buscadores que permiten encontrar en tiempo óptimo documentos deseados por los usuarios, mostrando los resultados en varias páginas.

Las tendencias actuales están dirigidas a que los internautas visiten solo los primeros vínculos, muchas veces porque no encuentran la información deseada aún existiendo, ya que no se localizan en las primeras posiciones de los resultados arrojados por los buscadores. Para lograr que una página web sea visitada con frecuencia por una mayor cantidad de personas debe estar ubicada en los primeros enlaces de un motor de búsqueda. Existen diferentes estrategias y técnicas de optimización que se pueden aplicar a una página para alcanzar el objetivo deseado y esto precisamente, es la base del Posicionamiento Web, lo cual se ha convertido en una de las tantas herramientas más cotizadas por los Webmasters.



Los webmasters cubanos no acostumbran a realizar técnicas para promocionar los productos en Internet por la dificultad que requiere esta tarea sin un software que les facilite el trabajo, ya que el análisis y estudio de los sitios requiere tiempo y es engorroso. Además, cada buscador utiliza un algoritmo de indexación diferente que es actualizado y mejorado cada cierto tiempo lo que tardaría y haría aún más costoso el trabajo de los webmasters; sin olvidar que las mejores herramientas de Posicionamiento Web existentes en el mundo no son baratas y las versiones que son gratuitas ofrecen servicios restringidos.

El Grupo de Investigación y Desarrollo sobre Internet (GIDI) de la Universidad de las Ciencias Informáticas, es un proyecto formado por tres sub-proyectos: el Grupo Operativo, Grupo de Desarrollo y el Grupo de Vigilancia Tecnológica y Política. El Grupo Operativo tiene diferentes líneas dentro de las cuales se encuentra la línea de Weblog, que se dedica a publicar información sobre Cuba a través de los Blog, ya que estos se expanden con gran rapidez en el mundo de la comunicación e información. El número de personas que visitan estas páginas es cada vez mayor y algunos investigadores dicen que puede llegar a competir con los medios oficiales de información, aunque no llegarán a sustituirlos. Para sacarle el mejor provecho a los weblog son necesarias una serie de destrezas tecnológicas y habilidades comunicativas, además de que se hace imprescindible que los mismos estén bien posicionados en los diferentes buscadores de internet con le objetivo de lograr que esa información sea vista y consultada por los internautas.

Se puede plantear entonces como **problema científico**: ¿Cómo mejorar el posicionamiento de páginas cubanas en los Buscadores de Internet?

Para que sea posible dar solución a este problema se plantea el siguiente **objetivo general**: Diseñar e implementar una herramienta que ayude a la toma de decisiones del Webmaster, para lograr un buen posicionamiento de las páginas web en los buscadores de internet.

El **objeto de estudio** son los procesos y estrategias de posicionamiento web en los buscadores de internet, lo cual define como **campo de acción** los procesos automatizados que ayuden a la toma de decisiones de los Webmaster en el ámbito de posicionamiento web.



Para cumplir con el objetivo general de la investigación se trazaron los siguientes **objetivos específicos** y **tareas**:

Objetivos específicos:

- Analizar las estrategias de Posicionamiento Web de los Buscadores de Internet.
- Analizar las herramientas existentes sobre Posicionamiento Web.
- Determinar las herramientas y metodologías a utilizar.
- Diseñar el sistema para la ayuda a la toma de decisiones del Webmaster en el ámbito de Posicionamiento Web.
- Implementar el sistema para la ayuda a la toma de decisiones del Webmaster en el ámbito de Posicionamiento Web.
- Analizar los resultados obtenidos durante el desarrollo de la herramienta.

Tareas:

- Analizar el estado actual de la documentación que se utiliza en el mundo y en Cuba sobre Posicionamiento Web.
- Profundizar en el funcionamiento de los buscadores.
- Especificar las técnicas de SEO (del inglés: *Search Engine Optimization*).
- Consultar el estado en que se encuentran nacional e internacionalmente las herramientas como la que se pretende desarrollar.
- Consultar documentos sobre Lenguajes de Modelado, Herramientas CASE y Metodología de desarrollo.
- Precisar el lenguaje de programación a utilizar.
- Especificar la herramienta de modelado.
- Seleccionar patrones necesarios para el desarrollo de la aplicación.
- Confeccionar los diagramas de clases de diseño.
- Diseñar la Interfaz de Usuario.
- Aplicar estándares de codificación.
- Implementar funcionalidades de la herramienta.



- Definir y aplicar los métodos de prueba.

Conociendo lo anteriormente reflejado se puede plantear la siguiente **Idea a Defender**: Con el diseño e implementación de una herramienta que ayude a la toma de decisiones del Webmaster, se logrará un mejor posicionamiento de las páginas web en internet.

Métodos Científicos

Los métodos científicos son los procedimientos que se utilizan para estudiar la realidad, la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Permiten estudiar las características del objeto de investigación que no son observables directamente. Estos se pueden clasificar en: empíricos, teóricos y estadísticos, de los cuales para el presente trabajo se pusieron en práctica los métodos teórico y empírico.

Métodos Teóricos

Análisis y Síntesis: Permitió realizar un estudio de los rasgos que caracterizan a los buscadores, posicionamiento web y las técnicas SEO a partir del análisis de la documentación existente vinculada al desarrollo de estas estrategias, así como en la extracción de los elementos más importantes relacionados con el campo de acción.

Análisis Histórico Lógico: Por medio de este método se estudió de forma analítica la trayectoria histórica de las distintas estrategias de posicionamiento que han ido evolucionando con el paso de los años.

Modelación: Mediante este método se logró la creación de los modelos que permitieron visualizar, descubrir y estudiar nuevas relaciones y cualidades de las estrategias y procesos de implementación para la realización de la herramienta.

Métodos Empíricos

Entrevista: Este método permitió entrevistar a personas calificadas en el tema de posicionamiento web.



El presente trabajo consta de 3 capítulos:

Capítulo 1 titulado “Fundamentación teórica”: Ofrece los conceptos básicos asociados para dominio del problema. Recoge un estudio de algunos los sistemas existentes relacionados con el tema. Se dan a conocer la Metodología de Desarrollo, el Lenguaje de Modelado, la Herramientas CASE, Lenguaje de Programación y la Plataforma donde se desarrolló el sistema.

Capítulo2 titulado “Análisis y Diseño del sistema”: Describe las funcionalidades que debe cumplir el sistema recogidas en la solución propuesta, visualizándose y reflejándose a través de los requerimientos y del diagrama de casos de usos del sistema. En el mismo se modela el análisis y diseño de la aplicación por medio de los diagramas de clases correspondientes y los diagramas de interacción.

Capítulo 3 titulado “Implementación y prueba del sistema”: Describe como fue implementada la aplicación a través del diagrama de componentes. Muestra el estándar de codificación por el cual se rigió la implementación y las librerías utilizadas. Recoge las pruebas realizadas a la aplicación con el objetivo de detectar errores y defectos.



Capítulo 1: Fundamentación Teórica.

1.1 Introducción.

El constante desarrollo de las tecnologías ha cambiado la forma de pensar de las personas. El estudio de los principales conceptos ayudará en este capítulo a entender el mundo del Posicionamiento Web y comprender las tendencias de las diferentes herramientas existentes que determinarán el desarrollo progresivo del software que será implementado. Se fundamentará la selección realizada de la Metodología de Desarrollo, el Lenguaje de Modelado, el Lenguaje de Programación y Herramienta CASE que se emplearon en la realización de la aplicación.

1.2 Elementos Fundamentales para entender Posicionamiento Web.

1.2.1 Internet.

Sus orígenes se remontan a 1967, cuando se estableció la primera conexión de computadoras, Advanced Research Projects Agency Network conocida como (ARPANET), gracias a un contrato con la Agencia de Investigación de Proyectos Avanzados (ARPA) del ministerio de defensa de los EEUU con fines militares.

Hoy en día, Internet es un medio de comunicación pública, cooperativa y autosuficiente en términos económicos, accesible a cientos de millones de personas en el mundo entero. Físicamente usa parte del total de recursos actualmente existente en la redes de telecomunicación.

El desarrollo de Internet ha superado ampliamente cualquier previsión y constituyó una verdadera revolución en la sociedad moderna. El sistema se transformó en un pilar de las comunicaciones, el entretenimiento y el comercio en todos los rincones del planeta.

Es importante saber **World Wide Web**, o simplemente Web, no son sinónimo de Internet sino un subconjunto de la misma. La Web es el universo de información accesible a través de Internet, consiste en



páginas a las que se puede acceder usando un navegador. Utilizando la misma se tiene acceso a millones de páginas de información lo que la hace fuente inagotable del conocimiento humano.

Los **Sitios Web** son un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular típicamente comunes a un dominio de Internet o subdominio en la World Wide Web. Todos los Sitios Web públicamente accesibles constituyen una gigantesca "World Wide Web" de información donde una **Página Web** es un documento electrónico que contiene información específica de un tema en particular creado en formato HTML (Hypertext Markup Language). El lenguaje HTML permite que cualquier persona del mundo pueda crear su propia página web sin tener conocimientos de informática.

1.2.2 Buscadores.

Por la gran cantidad de información existente y por no tener la misma calidad o valor ; buscar información en Internet ha sido siempre un proceso tedioso y difícil, por lo que trajo consigo la aparición de sistemas automatizados que localizan información y recursos deseados que permiten realizarse de manera más fácil y organizada. Estos sistemas son llamados buscadores o motores de búsqueda.

Los buscadores son: "sitio web que gestiona una base de datos confeccionada por robots. Esta base de datos almacena directamente el código de las páginas visitadas por los bots o parte de él". (1)

Para entender el funcionamiento de los buscadores es necesario conocer los elementos principales que lo conforman:

- **Interfaz gráfica:** Es una página que se le muestra al usuario dando la posibilidad de especificar la palabra o las palabras claves del tema que desea buscar; devolviendo como resultado una lista con enlaces a páginas web en cuya descripción o contenidos aparecen las palabras claves.
- **Robot:** Usualmente llamados "Web Wanderers", "Web Crawlers", o "Spiders" (arañas de búsqueda). Es un programa que permite navegar a partir de una página web a través de sus hipervínculos a otras páginas extrayendo sus enlaces y guardándolo en Base de Datos.



Capítulo 1: Fundamentación Teórica

- Base de Datos: Conjunto de información relacionada que se encuentra agrupada ó estructurada. Permite recuperar, comunicar y manejar grandes volúmenes de información y hacen más ordenada la misma.

Aunque en el fondo todos buscadores tienen el mismo objetivo, hacer rápida y sencilla la recuperación de información en Internet, podemos clasificarlos en:

- Buscadores jerárquicos (Arañas o Spiders): La mayoría de grandes buscadores internacionales de uso habitual y conocidos son de este tipo. Requieren muchos recursos para su funcionamiento. No están al alcance de cualquiera. Recorren las páginas recopilando información sobre los contenidos de las páginas. Cuando se busca una información en los motores, ellos consultan su base de datos y presentan resultados clasificados por su relevancia.
- Mixtos Buscador – Directorio: Además de tener características de buscadores, presentan los sitios web registrados en catálogos sobre contenidos, por ejemplo informática, cultura, sociedad y a su vez se dividen en subsecciones.
- FFA - Enlaces gratuitos para todos: FFA (acrónimo del inglés "Free For All"), página de enlaces gratuitos para todos. Cualquiera puede inscribir su página durante un tiempo limitado en estos pequeños directorios. Los enlaces no son permanentes.
- Buscadores de Portal: Bajo este título se engloban los buscadores específicos de sitio, aquellos que buscan información solo en su portal o sitio web, y podrían ser considerados como un directorio.
- Buscadores verticales: Buscadores especializados en un sector concreto, lo que les permite analizar la información con mayor profundidad, disponer de resultados más actualizados y ofrecer al usuario herramientas de búsqueda avanzadas.
- Multibuscadores: Son programas que permiten la consulta simultánea en diversos motores de búsqueda e índices temáticos de la red a partir de una consulta única, mostrando los resultados de cada motor o índice de forma separada.
- Metabuscadores: Genera una lista de páginas web que cumplen las condiciones de búsqueda, incluyendo en dicho listado todos los resultados obtenidos y ordenándolos según un algoritmo propio, el cual suele tener en cuenta la posición que ocupa la página en los resultados de los distintos motores de búsqueda.



Existen los llamados directorios que son: sitios web que gestiona una base de datos confeccionada por humanos. Esta base de datos almacena y clasifica en categorías las URL junto con sus comentarios.

Por el uso cada vez mayor de los buscadores que son los que controlan un enorme porcentaje del tráfico de internet y el gran volumen de información existente en la gran Red de Redes es casi un milagro aparecer en los primeros resultados de un motor de búsqueda. Posicionar los productos es de gran importancia por lo que despertó la curiosidad de muchos webmasters y propietarios de páginas web para hallar una respuesta a la pregunta: ¿qué tengo que hacer para estar en los primeros puestos de los buscadores?

1.2.3 Posicionamiento Web.

El posicionamiento en buscadores consiste en adoptar ciertas estrategias y aplicar diversas técnicas que estudian las características que proporcionan a un sitio o una página, para colocarla en una posición óptima entre los resultados proporcionados por un motor de búsqueda. Se debe tener en cuenta que cada buscador utiliza algoritmos de indexación diferentes que son actualizados. No se dispone del código de los algoritmos de posicionamiento en sí, si no más bien se trata de inferir cómo funciona a partir de su comportamiento observado.

El proceso de posicionar una página web, no implica desviar todo su estudio y esfuerzos orientados en este sentido; hay que saber realmente qué valora cada buscador sobre la base de esto para crear herramientas que optimicen la programación de un sitio web para su indexación en un buscador.

Existen dos tipos de posicionamiento:

- Posicionamiento Orgánico o Natural (SEO del inglés: Search Engine Optimization): Cualquier técnica de desarrollo web que tenga como objetivo mejorar la posición de un determinado sitio web en la lista de resultados de los motores de búsqueda. Para valorar estas técnicas hay que tener en cuenta:



Capítulo 1: Fundamentación Teórica

- Posicionamiento natural: Consigue posicionar una página o un sitio de modo espontáneo, es decir, sin que sea consecuencia de una campaña consciente o planificada.
 - Posicionamiento planificado: Consigue posicionar una página o un sitio web debido a una campaña consciente y planificada. El posicionamiento planificado puede ser ético o fraudulento.
 - Posicionamiento planificado fraudulento: Consigue posicionar una página o un sitio como consecuencia de actividades deliberadas de engaño por parte del responsable de la página web; actividades dirigidas a forzar los resultados de los motores de búsqueda.
 - Posicionamiento Planificado ético: El posicionamiento planificado "ético"(o sea, no fraudulento) debe consistir en conseguir los mismos resultados que se obtendrían con un posicionamiento natural en una situación ideal y en la cual los responsables de los sitios webs los diseñarán con visibilidad óptima para los motores de búsqueda.
- Posicionamiento Patrocinado SEM (Search Engine Marketing): *Significa en español "Marketing (mercado) en motores de búsqueda". Ésta técnica trata sencillamente de comprar las posiciones en los buscadores totalmente contrario con SEO que es el posicionamiento en buscadores. (2)*

Algunos elementos estructurales usados profesionalmente que permiten una aproximación considerable a una posición óptima dentro de la lista de resultados arrojados por un buscador, independientemente de que los algoritmos sean diferentes en cada motor de búsqueda:

- Palabra clave: También conocidas como "conceptos clave" o "keywords". Es una palabra o una combinación de palabras con las que los buscadores clasifican las páginas web dentro de su base de datos. Estas palabras claves, cuando están adecuadamente seleccionadas, serán las mismas que teclean los usuarios cuando usan los buscadores para encontrar información de su interés. Todo proceso de elección de palabras claves lleva tres pasos: la búsqueda de palabras provisionales, el análisis de la totalidad de esas palabras clave, y la elección del grupo más adecuado de palabras clave.



Capítulo 1: Fundamentación Teórica

- Recuento: Total de palabras en un área determinada dentro un documento HTML, es decir, todo el texto que está dentro de la etiqueta body.
- Frecuencia de la palabra clave: Es el número de veces que el término de búsqueda aparece en el área.
- Peso de la palabra clave: El peso es el porcentaje que el término de búsqueda representa sobre el total de palabras en la página o el área analizadas. Dado que el término de búsqueda puede estar formado por varias palabras, la fórmula general del peso se define como:

$$P = n * F / R$$

n = número de palabras del término de búsqueda.

R = Recuento del área.

F = Frecuencia del término de búsqueda.

En cualquier caso hay una amplia discusión sobre el papel del peso y su cálculo exacto. En general, suele rondar el 3% para las páginas mejor posicionadas aunque obviamente esto se debe a que la frecuencia y el recuento de estas suele ser similar. En la práctica vigilar el peso sirve como chequeo de la proporción entre frecuencia y recuento, no tanto como una variable independiente que actúe por si misma.

- Densidad de la palabra clave: La densidad de palabras claves o keyword density, es el porciento de apariciones de una palabra o frase clave contra el total de palabras que componen un texto de <body> a </body>. Por ejemplo, para un texto de 100 palabras, si una palabra aparece 6 veces, su densidad será del 6%. Es recomendable que la densidad de la palabra clave debe encontrarse de 5%-20%.
- Relevancia: El cálculo de relevancia de cada motor de búsqueda combina diversas medidas, pero las más importantes consisten en combinar de alguna forma dos grupos de indicadores:
 - ✓ La frecuencia y la densidad de las palabras claves
 - ✓ El número de enlaces que recibe (visibilidad).



Capítulo 1: Fundamentación Teórica

La relevancia mide la distancia entre el comienzo del área y el lugar donde aparece la palabra clave. Si el término de búsqueda aparece como primera palabra del área la prominencia es 100% y si es la última el 0%. Si aparece varias veces, se promedia. Es decir, a no ser que hagamos recuento muy baja, dejando la palabra clave sólo como primera palabra del body, nunca tendrá un 100%, aunque lo más recomendable es colocarla con cierta profusión en el primer tercio de cada área.

- Popularidad: Número de enlaces que recibe una página o un sitio web. Es en realidad la suma del número de páginas que nos apuntan ponderada por el índice de popularidad de cada una de ellas (medido en porcentaje sobre la máxima puntuación alcanzable).
- PageRank: Es una familia de algoritmos utilizados para asignar de forma numérica (entre 1 y 10) la relevancia de los documentos (o páginas web) indexados por un motor de búsqueda. Sus propiedades son muy discutidas por expertos en optimización de motores de búsqueda. Un enlace a una página cuenta como un voto de apoyo. Una página que está enlazada por muchas páginas con un PageRank alto consigue también un PageRank alto. Si no hay enlaces a una página web, no hay apoyo a esa página específica.

Otros aspectos importantes que se debe tener en cuenta a la hora de posicionar un sitio web es conocer los criterios que utiliza el Robot del motor de búsqueda en relación con las diferentes partes del documento HTML. Sin embargo, la lectura del código HTML por algunos spider no es lineal, es decir, cuando el código se hace muy engorroso o por ejemplo, cuando tiene muchas tablas anidadas, se posterga el análisis del mismo y en ocasiones cuando no es entendible el código se ignora el contenido. También se debe tener en cuenta que algunos robots solo buscan en un determinado orden, de no ser así, no archivará la información que es de vital importancia para un posterior posicionamiento.

Independientemente que las arañas tienen diferentes formas de leer el código HTML existen algunos criterios específicos que no debemos olvidar y que influyen en gran medida en el posicionamiento de los sitios web:



- Área del título: Es un conjunto de palabras claves (10-70 caracteres, sin poner caracteres especiales) que conforman una frase que permite saber de lo que trata una página y por las que queremos que el sitio sea encontrado. Es uno de los factores más importantes para que una página sea indexada por un buscador, bajo una determinada consulta.

- Área de meta información: Permite hacer una breve descripción del contenido del documento. De las etiquetas HTML es la <META>, la cual tiene como atributos (menor de 200 caracteres):

- Importante para el Posicionamiento:

Description: Es una descripción general del contenido de su documento HTML, debe haber coherencia entre el contenido de esta meta-etiqueta y las palabras incluidas en el texto del documento, y en el título de la página.

Keywords: Es una lista de palabras claves, que indican las palabras más importantes en el documento facilitándole a los buscadores información relevante. Se separan por el carácter "," (coma).

- Otras meta-etiquetas recomendadas:

Abstract: Presenta una abstracción breve del documento, un resumen general.

Author: Permite colocar información sobre el autor de un sitio web, se recomienda colocar el email en este atributo, de manera que, si alguien se interesó en contactar al autor de un documento HTML específico tenga de esta forma la información de contacto necesaria.

Expires: Se puede usar para documentos que pierden validez después de pasada alguna fecha. Indica a los motores de búsqueda que el documento se puede eliminar de la base de datos en el momento en que se indique en esta meta-etiqueta.



Capítulo 1: Fundamentación Teórica

Language: Permite indicarle a los motores de búsqueda el idioma en que está hecho el documento, esto puede ser muy importante para directorios que sólo indexen sitios realizados en algún idioma específico.

Revisit: Tiene el objetivo de indicarle a los motores de búsqueda, qué tan frecuentemente deberían visitar el sitio para re-indexar sus contenidos. Su uso es bastante recomendado pero solo en aquellos sitios que realmente están cambiando sus contenidos de una forma muy frecuente.

Robots: Indica cuáles páginas deben ser indexadas por un buscador y cuáles no. Su uso podría ser perjudicial si no se usa con cuidado, en el sentido de que un error acá, o en el archivo robots.txt podría provocar que no se indexen documentos que sí deberían estarse indexando.

- Jerarquía de contenido: Es importante utilizar las etiquetas <H1>, <H2>, <H3>, <H4>, <H5> y <H6> con el objetivo de especificar la importancia que tiene un determinado contenido.
- Texto alternativo: Es un texto que se le asocia al atributo ALT en la etiqueta con el objetivo de hacer una descripción de la imagen.
- Mapa de Navegación: Facilita la indexación de las URL que se encuentran asociadas a los sitios web de forma rápida y económica.
- Tamaños y tipo de fuente de la palabra clave: La palabra clave puede ser destacarla de alguna forma en strong, bold o italic.
- Listas numeradas / no numeradas: La palabra clave debe estar lo más cerca del elemento listado.



Capítulo 1: Fundamentación Teórica

- Palabra clave en url: La primera palabra cuenta más, para el término " **granma** " se valorará más un sitio web con el dominio: *http://www.granma.cubasi.cu/* que otro que sea *http://intranet.uci.cu/granma.php*.
- Stemming: Variante a partir de la raíz de la palabra clave. Ejemplo: posicionamiento posicionar, posicionado, posiciones.
- Edad del sitio que nos enlaza: Mientras más antiguos sean los sitios que nos apuntan mejora la estabilidad del sitio.

Aspectos negativos dentro del sitio web:

- Usar gráfico como texto (es ilegible para el robot y menos usable).
- Exceso o defecto de densidad de la palabra clave.
- Tamaño del texto (o color de fuente igual al del fondo).
- Web dinámica.
- Elevado uso de javascript.
- Presentaciones en Flash.
- Enlazar (o ser enlazado) por granjas de enlaces o servidores de contenido impropio.
- Enlaces rotos.
- No exceder de 100 KB por página.
- Tablas: Los buscadores valoran más el texto del principio del código, no lo primero que se ve en el navegador. Las tablas a veces presentan el texto al lector en el navegador de forma distinta a lo dispuesto en el código. Se pueden colocar tablas en capas o usar tablas definidas con estilos con posiciones.
- Uso de frames.
- Exclusión del robot mediante la etiqueta <no index>.
- Texto muy pequeño.
- Contenido duplicado.
- Que la página tenga errores de programación y cargue lento.



- **Meta-etiqueta Refresh:** Es una forma de hacer que una página re-direccione hacia otra después de cierta cantidad de segundos, sin embargo, no es la forma más elegante de alcanzar este objetivo. Su uso es no recomendado.

1.3 Herramientas relacionadas con Posicionamiento Web.

En la actualidad existen varias herramientas dedicadas a brindar servicios de Posicionamiento Web. Estas aplicaciones permiten a cualquier usuario que decida desarrollar un sitio web y que desee publicarlo en Internet tenga la calidad necesaria para estar en las principales posiciones dentro de la lista de resultados arrojados por los buscadores. Es un paso de avance en estos tiempos donde la información existente en la gran Red de Redes es muy amplia y porque reduce considerablemente el trabajo de los webmasters. Algunos de las herramientas para apoyar la toma de decisiones de los webmasters en el ámbito de posicionamiento web son:

- **Promotor Web:** Este software ha sido desarrollado para ayudar a los webmaster a promocionar sus páginas web, enviándolas a los buscadores de forma automática para lograr un número mayor de visitas. Está diseñado para optimizar los sitios web ayudándole a mejorar y analizar sus páginas, lo que posibilita obtener buenas posiciones en los buscadores. Calcula la popularidad de enlaces de la página y cuenta con generador de palabras claves que permite crear una lista de sugerencias basadas en las principales palabras que posicionarían al sitio web.
- **Active WebTraffic:** La primera versión se publicó originalmente en 1997. Es un software que tiene un conjunto de herramientas SEO para optimizar y posicionar los sitios web. Incluye un generador de palabras claves y de etiquetas metas el cual extrae de los mismos toda la información de las páginas web deseadas y crea automáticamente numerosas etiquetas metas con el objetivo de posicionar las páginas web en los primeros puestos de los buscadores. Dispone de una herramienta de ranking que informa al usuario de las posiciones alcanzadas por el sitio web



especificado según las palabras determinadas en los buscadores más importantes y comprueba como afecta al ranking los enlaces que apuntan hacia la página web. Encuentra palabras claves poco competitivas automáticamente. Determina las palabras más buscadas en Google relacionadas con la web, así como los términos más populares buscados individualmente por buscador. Incluye todo lo que necesita con facilidad y profesionalidad para promocionar un sitio web. Active WebTraffic presenta un constante y continuo flujo de ventas. Para garantizar un funcionamiento correcto debe cumplir con los requerimientos siguientes:

- PC Pentium 100 o superior.
- Windows 95, 98, ME, NT, 2000, 2003, XP, Vista.
- 128 MB de memoria RAM.
- Conexión a Internet.
- 50MB de espacio disco duro.
- 800x600 o mayor resolución de pantalla (1024x768 recomendado).

- Google Monitor Software: Google Monitor es una interesante herramienta gratuita que ayuda en la tarea de buscar y rastrear la posición de una determinada página dentro de la tabla de resultados de Google. Al introducirle la URL de un sitio y un conjunto de palabras claves, devuelve un listado ordenado por ranking con todos los sitios que responden a esas palabras claves y da la posición exacta del sitio en ese listado. Se podrá ver como asciende o desciende el posicionamiento del sitio dentro de los resultados de Google. Es una herramienta práctica ya que no hay que entrar necesariamente al buscador ni recorrer cientos de páginas para encontrar la página a posicionar.

Requisitos del sistema para un funcionamiento correcto:

- PC Pentium 200 MHz.
 - 32 MB de RAM.
 - Windows 98/2000/XP/Vista.
 - Acceso a Internet.
- Active Link Exchange: Su primera versión se publicó originalmente en 2006. Es una popular herramienta de software que da la posibilidad de construir una red de intercambio de enlaces de



forma sencilla, rápida y con una alta calidad; de esta forma, gestiona de una manera más eficiente la visibilidad de la página a través de enlaces en otras webs. Permite crear un número ilimitado de proyectos de intercambio de enlaces para cualquier número de sitios web. El software da la posibilidad de investigar cuales son las páginas web de mayor ranking en un sector, encontrar los sitios clave que enlazan con los competidores. No se tiene la seguridad de que caiga en deficiencias que pueden ser penalizadas por un buscador, un ejemplo podría ser que el sitio se convierta en una “granja de enlaces” es decir que tengas demasiados en enlaces. Para garantizar un funcionamiento correcto debe cumplir con los requerimientos siguientes:

- PC Pentium 100 o superior.
 - Windows 95, 98, ME, NT, 2000, 2003, XP o Vista.
 - 128 MB de RAM de memoria.
 - 10 MB de espacio en disco duro.
 - Conexión a Internet.
 - 800x600 o mayor resolución de pantalla (1024x768 recomendado).
- **SEOAdministrator**: Es una poderosa herramienta de posicionamiento de sitios web que analiza todos los enlaces de una página web en búsquedas de posibles errores tales como hipervínculos rotos, repetidos o híbridos; además posibilita de forma sencilla intercambiar enlaces. Revela los distintos motores de búsqueda por los cuales fueron indexadas las páginas de un determinado sitio web. Identifica sitios donde puede publicar enlaces para aumentar su popularidad. Incluye un generador de archivo de robot.txt, comprueba los posibles errores del código HTML de una página determinada y crea el mapa del sitio donde se recogen todos los enlaces. Obtiene el número de visitantes y las consultas de búsquedas que se utilizaron para recuperar un sitio web. Devuelve de forma automática el valor del PageRank. La herramienta sugiere las palabras claves relevantes para cada una de las páginas que se desee posicionar. Proporciona un informe completo sobre el peso y la densidad de palabras claves. Esta diseñado para:
- Windows 98, 2000, ME, XP, 2003, Vista.
- **SEO Elite**: Es una herramienta de promoción, eficaz que permite optimizar los sitios web con el propósito de que estén bien posicionados en los diferentes buscadores existen en Internet. Evalúa el código HTML para el correcto posicionamiento, crea los contenidos optimizados, da de alta en



Capítulo 1: Fundamentación Teórica

los principales buscadores como Google, Yahoo, AllTheWeb, MSN, Altavista. Permite conocer el número de enlaces que debería tener la página y en que medida afecta al PageRank de la misma. Devuelve el porcentaje de palabras clave a utilizar en los enlaces y añade vínculos recíprocos a su página.

- SEO Studio: Herramienta desarrollada con el objetivo de ayudar al webmasters a conseguir la mayor calificación en los buscadores. Es un software de búsqueda, análisis y administración de vínculos recíprocos, permite construir popularidad de vínculos, promover, optimizar, enviar y jerarquizar los sitios web. Tiene un analizador de palabras claves que le ayuda a obtener la densidad de la misma. Se recomienda que la computadora sea:
 - Pentium 4.
 - Con 256 de RAM o lo requerido por el sistema operativo.
 - Espacio libre en disco duro de 28 MB.
 - Compatible con Microsoft Windows Vista/ XP/ 2000/ 2003.

- Cuwhois: Es una herramienta web que brinda diversa información como: Sistema automática de alta en buscadores, Chequea el estado de los enlaces de una web, busca las palabras más repetidas de una web, calcula el número de visitas mensuales de una web, muestra información de las de palabras claves, posee un validador de código HTML el cual da la posibilidad de solucionar errores comunes y captura metas visibles para buscadores.

- Herramientas sencillas:
 - Site Position: Permite comprobar la popularidad de tu Web con rapidez y eficacia de forma sencilla. Comprueba la posición de tu Web en determinados buscadores de acuerdo con las palabras clave que le especifiques, mostrando los resultados obtenidos de diversas maneras. Puede generar informes con resultados en diversos formatos (CSV, HTML, XML) y funciona



Capítulo 1: Fundamentación Teórica

con los buscadores más populares: Google, Lycos, Altavista, MSN, Yahoo, Webcrawler y otros. Totalmente gratis. Está diseñado para Windows95/98/98SE/2000/ME/XP.

- TopDog Pro: Incrementa el tráfico y analiza la posición que tiene una página web dentro de internet. Determina el índice de efectividad para ser mostrado en los motores de búsqueda facilitando que los sitios obtengan mejor visibilidad y una posición más elevada. Busca simultáneamente en los principales motores y ofrece el resultado en un informe de fácil lectura, con detalles para cada sitio. Actualmente la herramienta no es gratis.
- RankMeter: Es una herramienta web que permite saber la posición de una página determinada en los principales buscadores. RankMeter incluye más de 300 motores de búsqueda entre los principales buscadores del mundo. Este programa es compatible con Windows 95, 98, NT, 2000, XP.
- Link Popularity Check: Programa que muestra la popularidad de un sitio web, basado en los enlaces que apuntan a la página y no en los meta.

Existen diversas herramientas que brindan servicios de posicionamiento web, las cuales en su mayoría son propietarias o brindan servicios restringidos. Muchas de los sistemas anteriormente analizados solo brindan la posibilidad de dar de alta en los buscadores o simplemente hacer intercambio de vínculos para mejorar las visitas y consecuentemente el posicionamiento. Son pocas las herramientas que tienes todos los servicios que den la posibilidad de analizar a profundidad los errores de la web y permita corregir los mismos desde la aplicación. En el caso de aquellos softwares que brindan consejos de optimización, a partir de datos de entrada como son: palabras claves, título, meta y otros presentan incoherencias e imprecisiones.

Como los algoritmos de los buscadores son diferentes, y los cuales son actualizados y mejorados cada día, es importante que las herramientas estén enfocadas no solo a dar algunos aspectos importantes para lograr un buen posicionamiento, sino que den la posibilidad de comparar el sitio que se desea posicionar con otros sitios de algún buscador determinado.



Con el desarrollo de un software que ayude a la toma de decisiones del webmaster en el ámbito de posicionamiento web que logre solucionar los problemas antes mencionados se podría garantizar de manera gratuita para Cuba una aplicación de este tipo, además posibilitaría un compendio de las más importantes funcionalidades que brindan las herramientas similares, dando una mayor importancia al análisis lexicográfico del contenido web.

1.4 Lenguaje de Modelado.

1.4.1 UML

Se utilizó a UML (Lenguaje Unificado de Modelado) ya que en la actualidad está consolidado como un lenguaje estándar para la realización de software. Es un lenguaje expresivo, claro y uniforme, que no garantiza el éxito de los proyectos pero sí mejora sustancialmente el desarrollo de los mismos.

UML es un lenguaje gráfico que se utiliza para visualizar, construir especificar y documentar un sistema de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. UML permite la modelación de sistemas con tecnología orientada a objetos. No es una guía ni un proceso.

Es importante destacar que un modelo UML describe supuestamente lo que hará un sistema, pero no dice cómo implementar dicho sistema. Sirve de apoyo para todos los desarrolladores de un mismo software de forma que todos se comuniquen a través de sus diagramas para un mejor entendimiento y análisis del problema.

1.5 Metodologías de Desarrollo.

1.5.1 RUP (Proceso Unificado de Desarrollo).

Se utilizó a RUP porque constituye una guía rectora que decide quién hace qué, cuándo y cómo lo hace. Constituida por un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Esta metodología no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. RUP es más que



un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Esta metodología de desarrollo de software utiliza el Lenguaje Unificado de Modelado (UML) para expresar gráficamente todos los esquemas de un sistema software. Dentro sus principales características está que es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura. En su modelación define como sus principales elementos: trabajadores, actividades, artefactos y flujos de trabajo. Utilizando RUP la vida de un sistema transcurre a través de ciclos de desarrollo, desde su nacimiento hasta su muerte, en cada ciclo se repite el proceso unificado de desarrollo y consta de cuatro fases: Inicio, Elaboración, Construcción y Transición.

1.6 Lenguaje de Programación.

1.6.1 Java.

Se utilizó a Java por ser un lenguaje bastante potente para desarrollar aplicaciones en cualquier ámbito. Es un lenguaje de programación Orientada a objetos. Las características principales que nos ofrece Java respecto a cualquier otro lenguaje de programación:

Portátil: Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos de manera que funciona en cualquier otro sistema operativo que tenga instalada la máquina virtual java.

Distribuido: Tiene capacidades para interactuar con los protocolos ya establecidos del conjunto de protocolos TCP/IP, como http y ftp. La razón por la que Java es distribuido se debe a sus características de portabilidad.

Robusto: Realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. No utiliza punteros. Fomenta el uso de interfaces en lugar de clases.



Seguro: Fuertes restricciones al acceso a memoria, como son la eliminación de punteros aritméticos y de operadores ilegales de transmisión. Un programa realizado en Java no puede realizar llamadas a funciones globales ni acceder a recursos arbitrarios del sistema, por lo que el control sobre los programas ejecutables no es comparable con otros lenguajes. No ocurre ninguna conversión ilegal de datos. El acceso a los campos de un objeto se sabe que es legal: public, private, protected.

Alto Rendimiento: Pueden convertirse en tiempo de ejecución en código máquina específico de la CPU, ejecutándose casi tan rápidamente (milisegundos) como código compilado de forma nativa C y C++.

Multitarea: Java y las bibliotecas estándares incluyen muchos recursos para ayudarle en este proceso de comunicación, asegurando la seguridad de los subprocesos. Los programas multitarea comparten memoria y deben tener comunicación entre los subprocesos.

Multihilos: Permite atender eficientemente varias tareas a la vez.

1.7 Herramienta CASE (Computer-Aided Software Engineering).

1.7.1 Visual Paradigm v 5.3.

Como herramienta CASE se utilizó a Visual Paradigm por ser una herramienta poderosa y fácil de usar que utiliza UML como Lenguaje de Modelado. Contribuye a que todos los desarrolladores se comuniquen de una mejor forma ya que utiliza un lenguaje estándar común a todo el equipo de desarrollo. Ayuda a una rápida construcción de aplicaciones con una mayor calidad y con un menor coste ya que tiene el diseño centrado en casos de usos y enfocado al negocio.

Permite realizar todos los diagramas especificados en UML como son los diagramas de clases secuencia, colaboración, etc. Da la posibilidad de realizar tanto la ingeniería directa como la inversa, así como generar código fuente partiendo de los distintos diagramas, en diferentes lenguajes de programación. Es una herramienta colaborativa, es decir soporta múltiples usuarios trabajando sobre el mismo proyecto.



El ambiente de Visual Paradigm provee una manera intuitiva para llevar a cabo diseño y análisis en sistemas con programación Orientada a objetos. Entre sus características se incluyen un generador de informes a través de un editor de detalles de casos de uso para la especificación y descripciones de los casos de uso, incluyendo la especificación del modelo general. Presenta la posibilidad de la interoperabilidad con otras aplicaciones como es el Rational Rose. También facilita licencias especiales para fines académicos y es multiplataforma.

1.8 Plataforma de Desarrollo.

1.8.1 NetBeans v6.5.

Se utilizó el NetBeans como herramienta para el desarrollo del sistema, la cual es escrita puramente sobre la base de la tecnología Java, de modo que puede ejecutarse en cualquier ambiente que ejecute Java. Es un producto de código abierto, con todos los beneficios del software disponible en forma gratuita, el cual ha sido examinado por una comunidad de desarrolladores de gran éxito. Ofrece a los desarrolladores numerosas ventajas, en la creación de nuevas aplicaciones multiplataforma.

Las características de NetBeans de flexibilidad entre plataformas, el cumplimiento de UML y la capacidad de administrar la complejidad ayudan a garantizar que las aplicaciones cumplan con los requerimientos específicos del negocio. Da la posibilidad de guardar el trabajo desarrollado con JBuilder y continuar los proyectos en NetBeans.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos (un archivo Java que contiene clases de java) los cuales dan la posibilidad de ser extendidas agregándole nuevos módulos.

1.9 Java y los reportes.

1.9.1 Ireport v1.3.3.

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



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

Características:

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

1.10 Conclusiones.

Los motores de búsqueda siguen siendo la forma más barata y eficaz de encontrar la información deseada en la gran Red de Redes, los cuales se han convertido una de las herramientas más populares en Internet. Un buen posicionamiento solo se logra como resultado de la aplicación de técnicas adecuadas, conociendo la estrategia concebida por los buscadores. La conceptualización de algunos temas generales facilitó un mejor entendimiento sobre el Posicionamiento Web y permitió el estudio general de algunas herramientas utilizadas en el mundo por los Webmasters que se caracterizan en su mayoría por ser propietarias. Cuba no tiene los recursos suficientes para comprar estas herramientas y las aplicaciones gratis que existen no cumplen con las funcionalidades requeridas por el Grupo de Investigación y Desarrollo sobre Internet. Por lo que se hace necesario tomar nuevas estrategias con el fin de que los sitios cubanos estén visibles y los visiten la mayor cantidad de personas.



Capítulo 2: Análisis y Diseño del Sistema.

2.1 Introducción.

El proceso de análisis y diseño de un sistema, lleva consigo un conjunto de elementos arquitectónicos para establecer una arquitectura sólida y firme; en la cual se describan los factores más importantes para la construcción del software.

En este capítulo se generan los artefactos: diagramas de clases del análisis y diseño así como diagramas de interacción los cuales son necesarios como base para comprender, desarrollar y producir económicamente el sistema teniendo en cuenta un dominio amplio y extenso de las exigencias y necesidades del cliente.

2.2 Propuesta de sistema.

Conseguir que los buscadores sepan que las web cubanas existen y que además consideren que realmente habla de aquello que los internautas están buscando y lo coloque en las primeras posiciones de los resultados arrojados por los motores de búsqueda, no es tarea fácil para los webmaster.

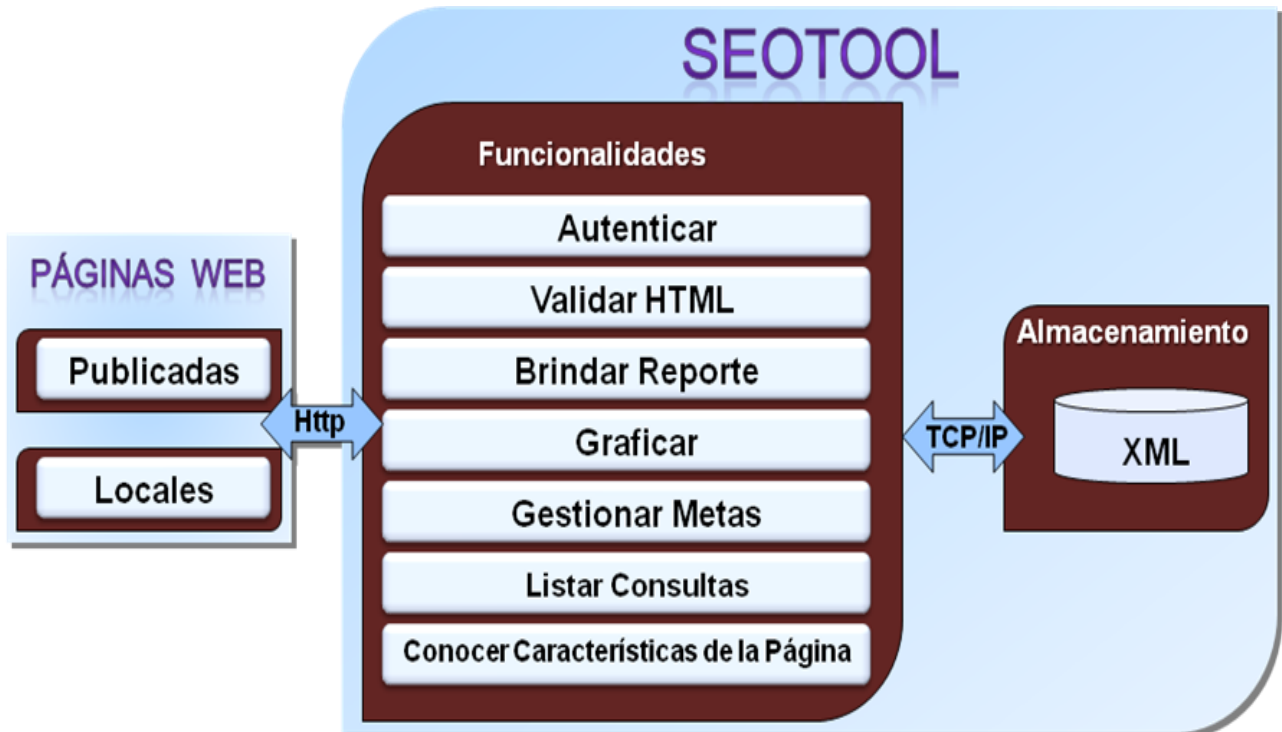
La necesidad de una herramienta que automatice el proceso de análisis de las páginas se hace evidente cada día más; porque es de gran importancia elevar la calidad de las mismas con el fin de obtener una posición favorable en los buscadores para que una mayor cantidad de usuarios visiten las páginas web y de esta forma hacer llegar la información exacta que brindan los medios cubanos a los internautas, pues es conocida la manipulación de la información que por parte de los EE.UU. se ha sufrido a lo largo de más de 40 años.

El trabajo se centra en una propuesta de sistema que posibilita concentrar de manera accesible y centralizada, las principales funcionalidades SEO que puedan servir para facilitar a los webmaster cubanos la optimización de los sitios web.



Capítulo 2: Análisis Y Diseño Del Sistema

La siguiente figura muestra el sistema que se propone:



El objetivo de la herramienta es elevar la calidad de un sitio web, mediante la mejora de su código y su contenido, partiendo de las principales funcionalidades que el mismo brinda. El sistema permite analizar páginas web locales (que se encuentren en alguna partición del disco duro de la computadora) o publicadas (en un servidor web), así como realizar comparaciones entre varias páginas web para medir la competencia y calidad de las mismas.

El webmaster tiene la posibilidad de elegir la URL de la página que desea analizar. Una vez que esté previamente autenticado podrá descargarla para obtener su código y realizar las diferentes acciones como: validar la estructura interna del HTML para detectar y corregir los errores posibles con el fin de eliminar las malas prácticas que atentan contra el posicionamiento que pueden ser considerado código basura. Obtener un listado de los criterios de búsqueda que mejor recupera la información, así como la frecuencia, peso, densidad y relevancia de cada palabra reflejada en el documento. Analizar las etiquetas importantes para los buscadores que inciden directamente en el posicionamiento web, como son: las etiquetas <META> que deben estar en función del contenido para ampliar más la visibilidad



en Internet; <TITLE>, jerárquicas (<H1>...<H6>) y . Construir un árbol que mide la profundidad de la página. Generar reportes con las características de una o varias página en caso de comparaciones y permitir listar las mejores páginas de un buscador determinado.

2.3 Requerimientos del sistema.

¿Cómo saber lo que se debe automatizar y cómo se expresa esto de manera que sea entendible a clientes y desarrolladores?

Básicamente según la IEEE Standard Glossary of Software Engineering Terminology define requerimiento como la capacidad o condición que necesita un usuario para resolver un problema o lograr un objetivo. Otros autores e instituciones la definen como: condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

Todas las ideas y criterios que se tengan acerca de lo que debe hacer el sistema ya sean de clientes, usuarios o miembros del equipo de proyecto, deben ser analizadas como candidatas a requisitos.

A continuación se presentan los requerimientos que debe cumplir el sistema que se propone, los cuales se han clasificado en: funcionales que son las capacidades o condiciones que debe cumplir un sistema, y no funcionales: son las características que hacen al producto atractivo, usable, rápido y confiable. Estos ponen las pautas a seguir y son imprescindibles para que el sistema tenga aceptación por los usuarios.

2.3.1 Requerimientos funcionales.

- R1. Analizar un sitio determinado.
 - R1.2. Brindar información si posee mapa de navegación.
 - R1.3. Informar la cantidad de etiquetas de cada tipo que posee.
 - R1.4. Mostrar porcentaje de texto que posee.
 - R1.5. Mostrar idioma que posee.
 - R1.6. Mostrar correo del webmaster.



- R1.7. Mostrar enlaces rotos en el sitio.
 - R1.8. Mostrar si está presente en párrafo o en título.
 - R1.9. Mostrar la frecuencia de cada palabra de la página.
 - R1.10. Mostrar el peso de cada palabra de la página.
 - R1.11. Mostrar la relevancia de cada palabra de la página.
 - R1.12. Mostrar el recuento de cada palabra de la página.
 - R1.13. Mostrar la densidad de cada palabra de la página.
 - R1.14. Calcular el tiempo en que demora en cargar la página.
 - R1.15. Mostrar peso de la página.
 - R1.16. Mostrar reporte de errores.
 - R1.17. Mostrar reporte general.
- R2. Mostrar comparaciones entre páginas.
 - R3. Determinar la consulta formada con las palabras claves que mejor recupera el sitio a posicionar.
 - R4. Crear proyecto.
 - R5. Eliminar proyecto.
 - R6. Validar código HTML.
 - R6.1. Crear las etiquetas meta e insertarla correctamente.
 - R6.2. Mostrar errores.
 - R6.3. Adicionar etiquetas.
 - R6.4. Eliminar etiquetas.
 - R6.5. Modificar etiquetas.

2.3.2 Requerimientos no funcionales.

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

Requerimientos de Software: El software debe disponer de Sistema Operativo Linux y Windows 98, 2000, XP, 2003, Vista y la requiere de la maquina virtual de java JDK1.6.



Restricciones en el diseño y la implementación: El diseño de la aplicación será orientado a objetos. Se garantizará la posibilidad de inclusión de nuevas funcionalidades y/o módulos de diseño para futuras iteraciones. Como lenguaje de programación a utilizar es el Java. La herramienta de desarrollo a utilizar es NetBeans v6.5.

Requerimientos de apariencia o interfaz externa: Interfaz intuitiva, amigable, organizada, con una navegabilidad flexible y de fácil comprensión, de forma que el objetivo del sistema para con los clientes, se pueda conseguirse rápidamente.

Requerimientos de eficiencia: El sistema deberá tener un tiempo de respuesta ante peticiones de cómo máximo 3.0 segundos.

Requerimientos de soporte: Se dispondrá de documentación técnica que describa todas las funcionalidades del sistema, de modo que existirá un manual para el uso de la aplicación orientada a clientes y usuarios finales. Deberá designarse un personal con conocimientos medios de informática y con dominio de las tecnologías utilizadas.

Requerimientos de usabilidad: El software podrá ser accedido por cualquier webmaster.

2.4 Diagrama de Caso de Uso del Sistema (CUS).

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

El presente diagrama de Caso de Uso es referenciado de la tesis (Sistema para el apoyo a la toma de decisiones de los webmasters, en el ámbito del Posicionamiento Web. Rol Analista de Sistema. Autora: Aliuska García Machado.), la cual se da continuidad para tener un mejor entendimiento de lo que el sistema debe realizar.

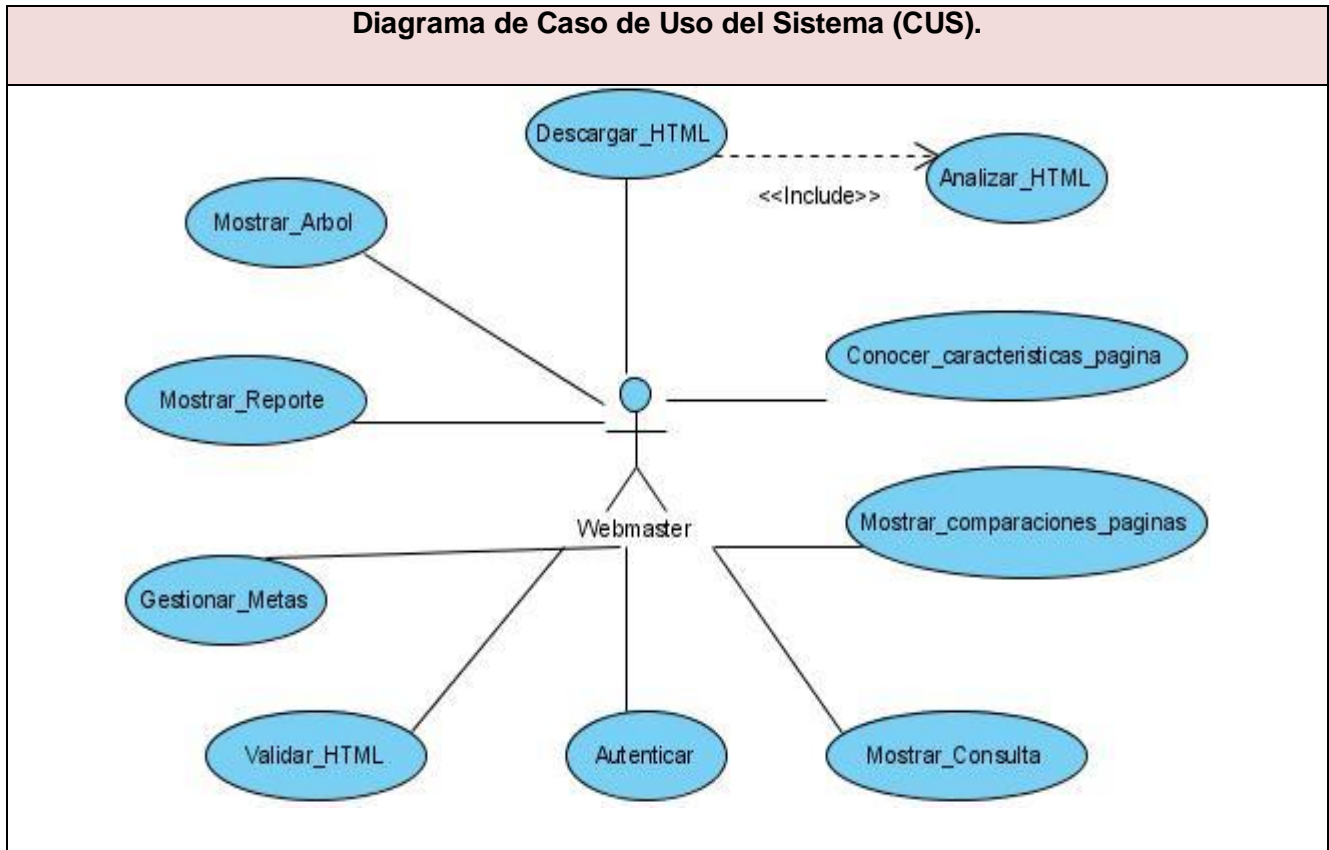


Tabla 1 Diagrama de Caso de Uso del Sistema (CUS).

2.5 Análisis y Diseño.

RUP define 6 disciplinas básicas para el desarrollo de un software, dentro de las cuales se encuentra el flujo de trabajo de Análisis y Diseño, el cual especifica cómo el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar. Tiene como objetivo traducir los requisitos de un sistema a una especificación que describe cómo implementar el mismo.

El Análisis consiste en obtener una visión del sistema, se preocupa de ver QUÉ hace el sistema. Permite razonar sobre aspectos internos del sistema y proporciona una estructura centrada en la flexibilidad ante los cambios y la reutilización. Facilita la entrada al diseño, por lo que es un paso inicial y una primera aproximación conceptual para una vez comprendido los requisitos, aumentar el nivel de



especificidad para garantizar el cubrimiento de los requisitos funcionales y no funcionales. El Análisis sólo se interesa por los requisitos funcionales mientras que el Diseño, realiza un refinamiento del análisis el cual tiene en cuenta los requisitos no funcionales, se preocupa por CÓMO cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades; se preocupa en el QUÉ debe hacer el sistema, pero centra los esfuerzos en él CÓMO el sistema debe funcionar para dar cumplimiento a cada uno de las funcionalidades definidas. Cuando la precisión del diseño es muy grande, la implementación puede ser hecha por un generador automático de código.

2.5.1 Modelo de Análisis.

El Modelo de Análisis proporciona una vista interna del sistema. Contiene clases del análisis y sus objetos organizados en paquetes que colaboran. “Durante el análisis, analizamos los requisitos que se describen en la captura de requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que nos ayude a estructurar el sistema entero, incluyendo su arquitectura”. (5)

2.5.1.1 Diagramas de clases del análisis.

Los diagramas de clases del análisis se centran en los requisitos funcionales. Los mismos exponen las clases que están vinculadas con el dominio del problema, representan conceptos y relaciones del dominio. A continuación se presentan los diagramas de clases del análisis generado durante el desarrollo del sistema.

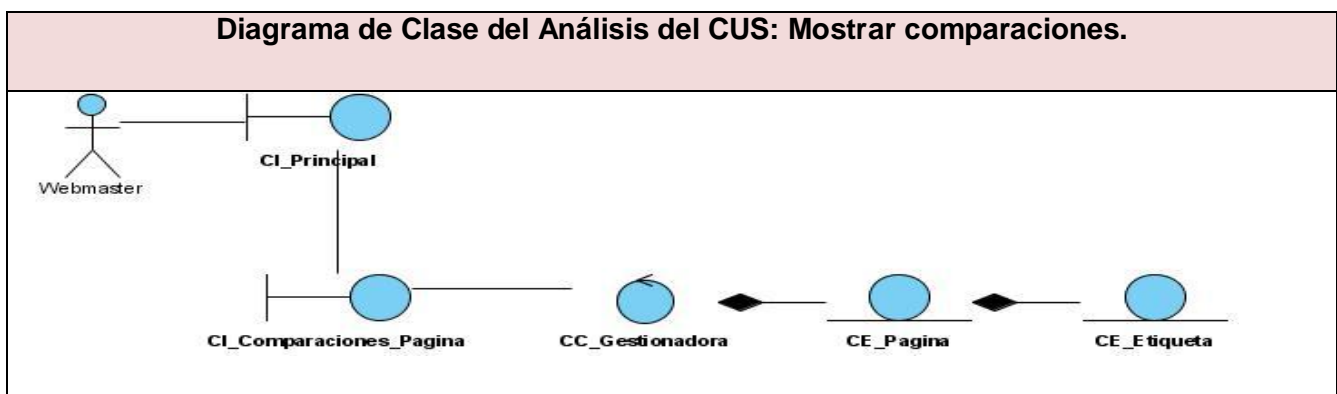


Tabla 2 Diagrama de Clase del Análisis del CUS: Mostrar comparaciones



Capítulo 2: Análisis Y Diseño Del Sistema

Diagrama de Clase del Análisis del CUS: Conocer características de una página.

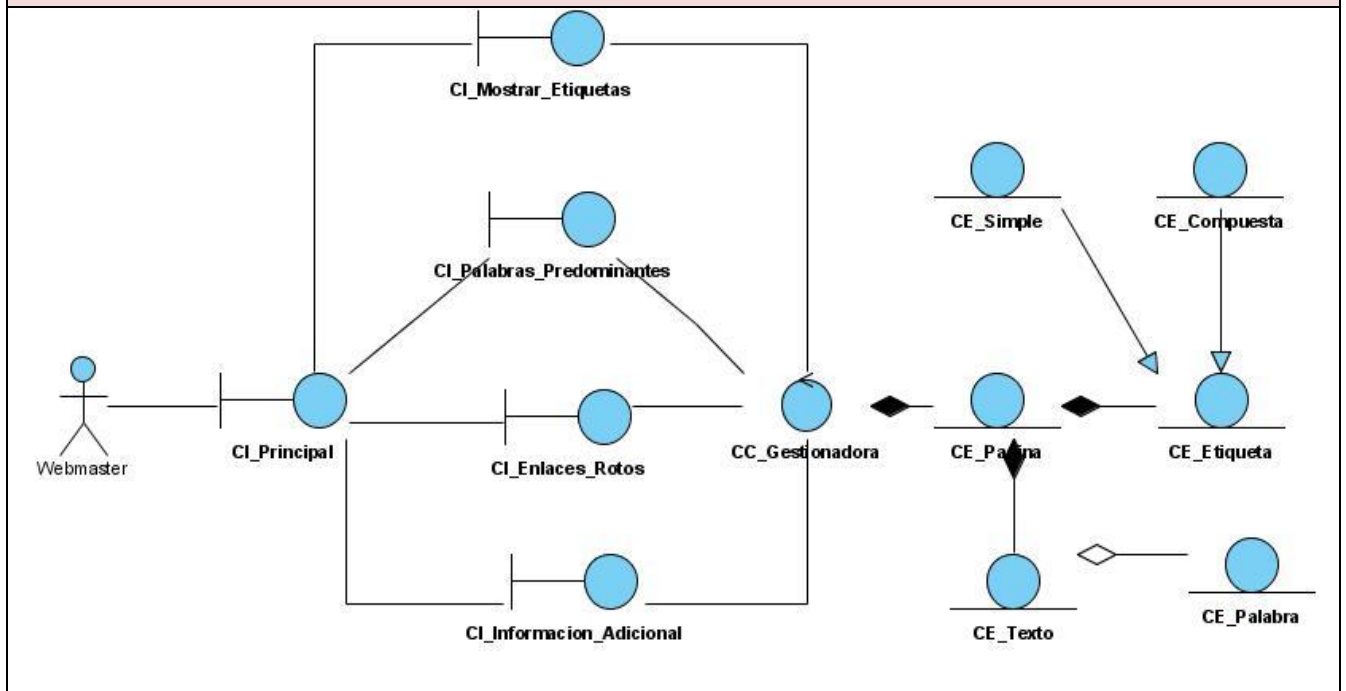


Tabla 3 Diagrama de Clase del Análisis del CUS: Conocer características de una página.

Diagrama de Clase del Análisis del CUS: Mostrar consulta.

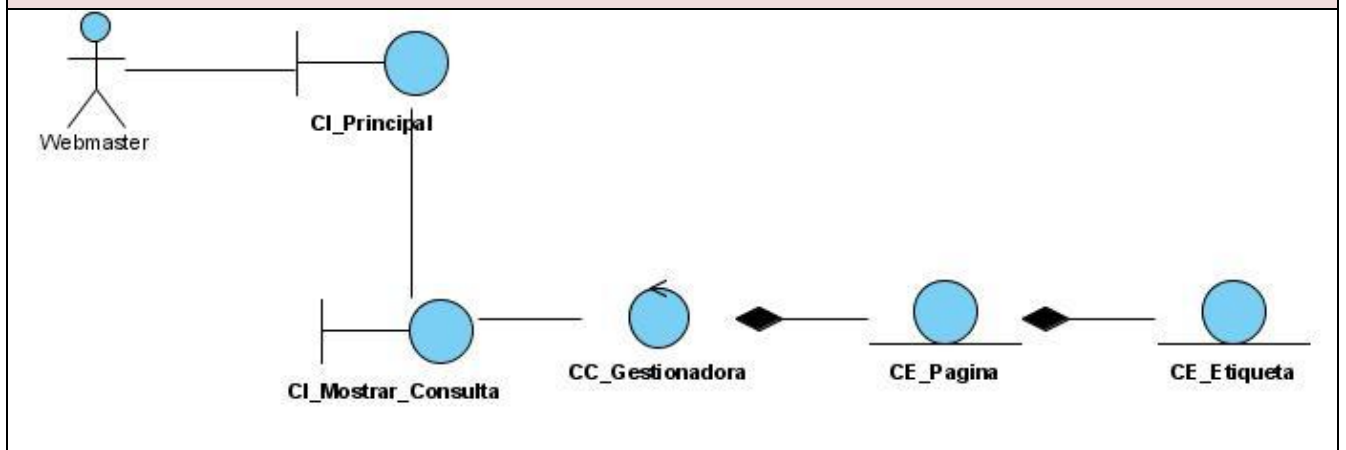


Tabla 4 Diagrama de Clase del Análisis del CUS: Mostrar consulta.



Diagrama de Clase del Análisis del CUS: Mostrar árbol.

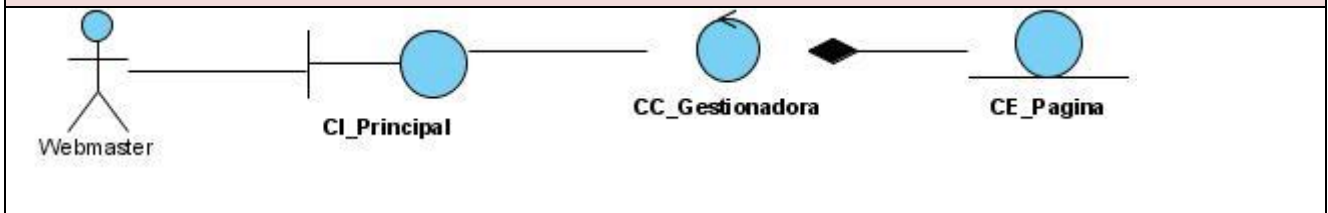


Tabla 5 Diagrama de Clase del Análisis del CUS: Mostrar árbol.

Diagrama de Clase del Análisis del CUS: Validar HTML.

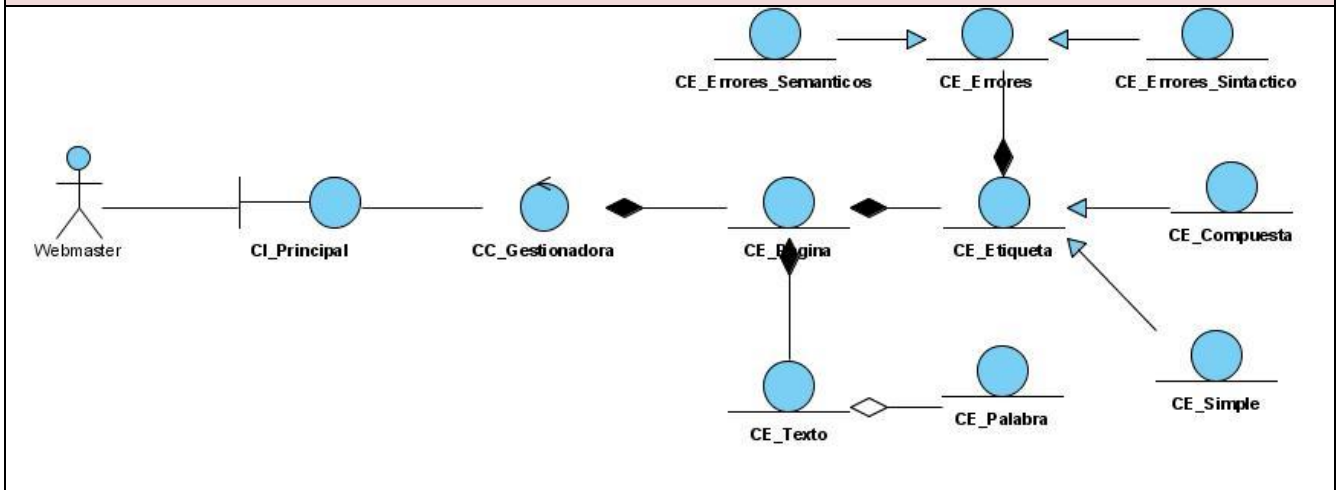


Tabla 6 Diagrama de Clase del Análisis del CUS: Validar HTML.

Diagrama de Clase del Análisis del CUS: Mostrar reporte.

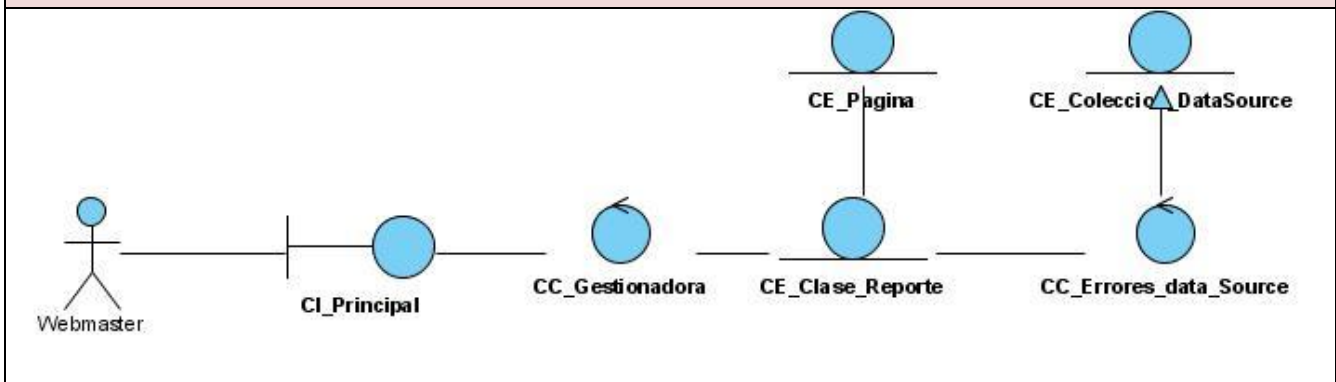


Tabla 7 Diagrama de Clase del Análisis del CUS: Mostrar reporte.



Diagrama de Clase del Análisis del los CUS: Autenticar.

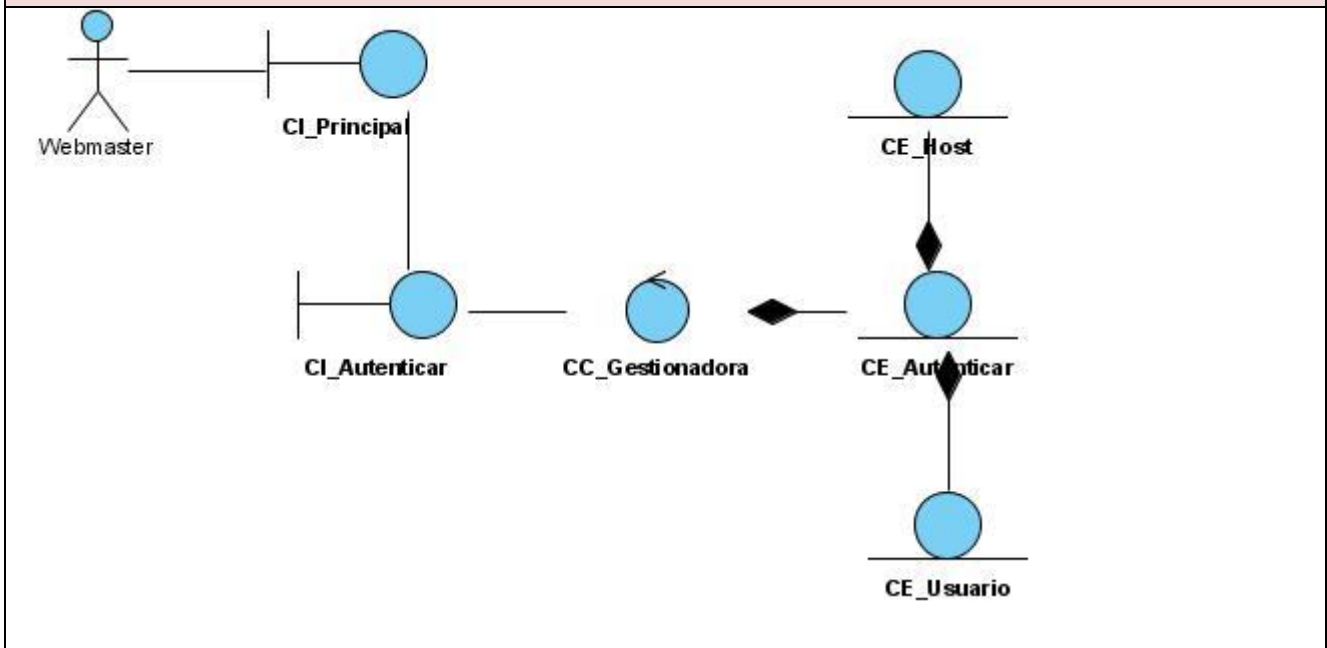


Tabla 8 Diagrama de Clase del Análisis del los CUS: Autenticar.

Diagrama de Clase del Análisis del CUS: Gestionar meta.

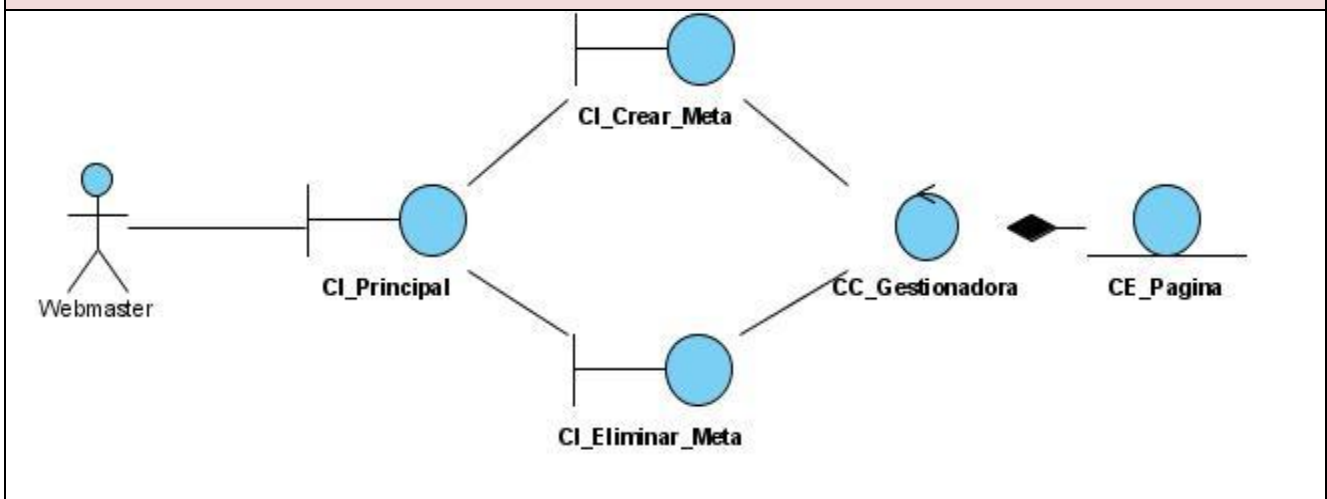


Tabla 9 Diagrama de Clase del Análisis del CUS: Gestionar meta.

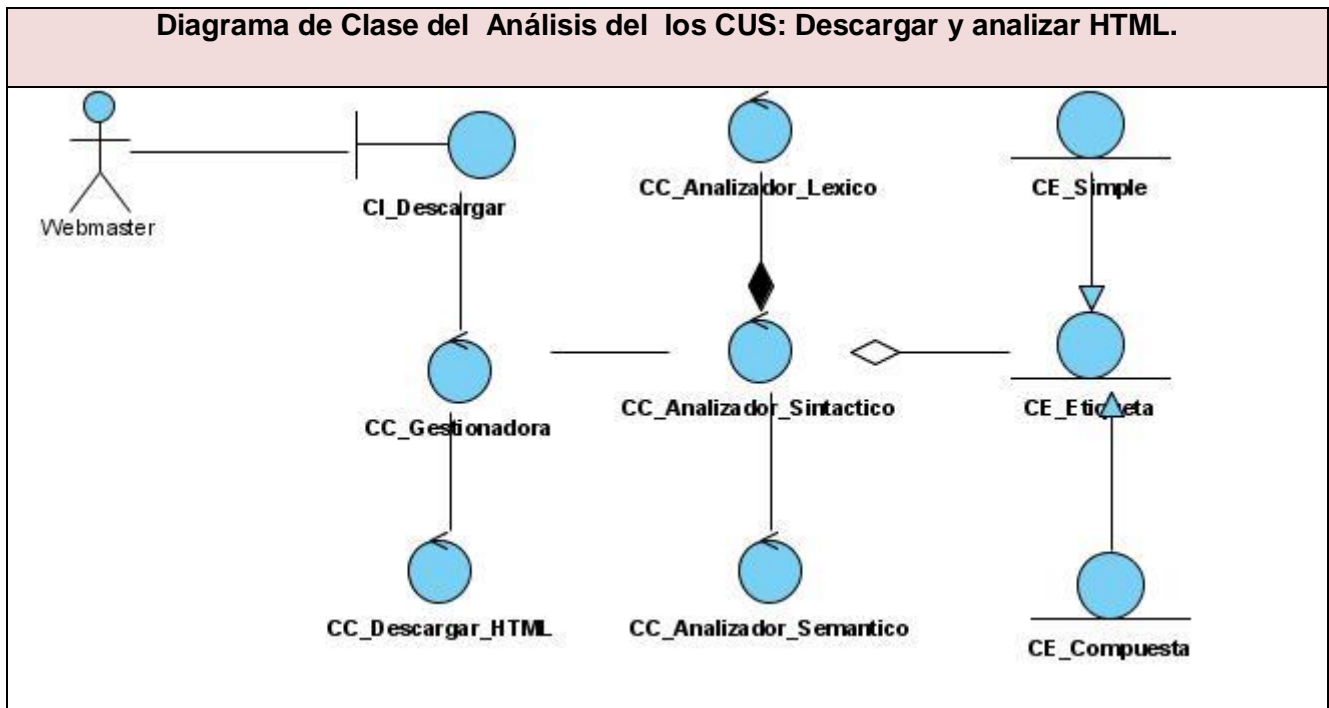


Tabla 10 Diagrama de Clase del Análisis del los CUS: Descargar y analizar HTML.

2.5.2 Modelo de Diseño.

En el flujo de trabajo de análisis y diseño, se modela el sistema de manera que soporte todos los requerimientos, tanto funcionales como no funcionales. Este modelo se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráfica de la aplicación.

“El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. El modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación”. (5)

2.5.2.1 Diagramas de interacción.

Los diagramas de interacción están enfocados a comunicar el aspecto dinámico del sistema, es decir muestran cómo se comunican los objetos, los cuales interactúan para realizar colectivamente, los servicios ofrecidos por las aplicaciones. En esencia, su misión es localizar el comportamiento de los objetos.



Existen dos tipos de diagramas de interacción:

- Diagramas de colaboración: Destaca la organización estructural de los objetos en ejecución, los cuales envían y reciben mensajes. Se prestan más al descubrimiento de abstracciones pues permite representar los objetos en una disposición próxima a la realidad.
- Diagramas de secuencia: Destaca el orden temporal de los mensajes.

A continuación se presentan los diagramas de secuencia generados durante el desarrollo del sistema.

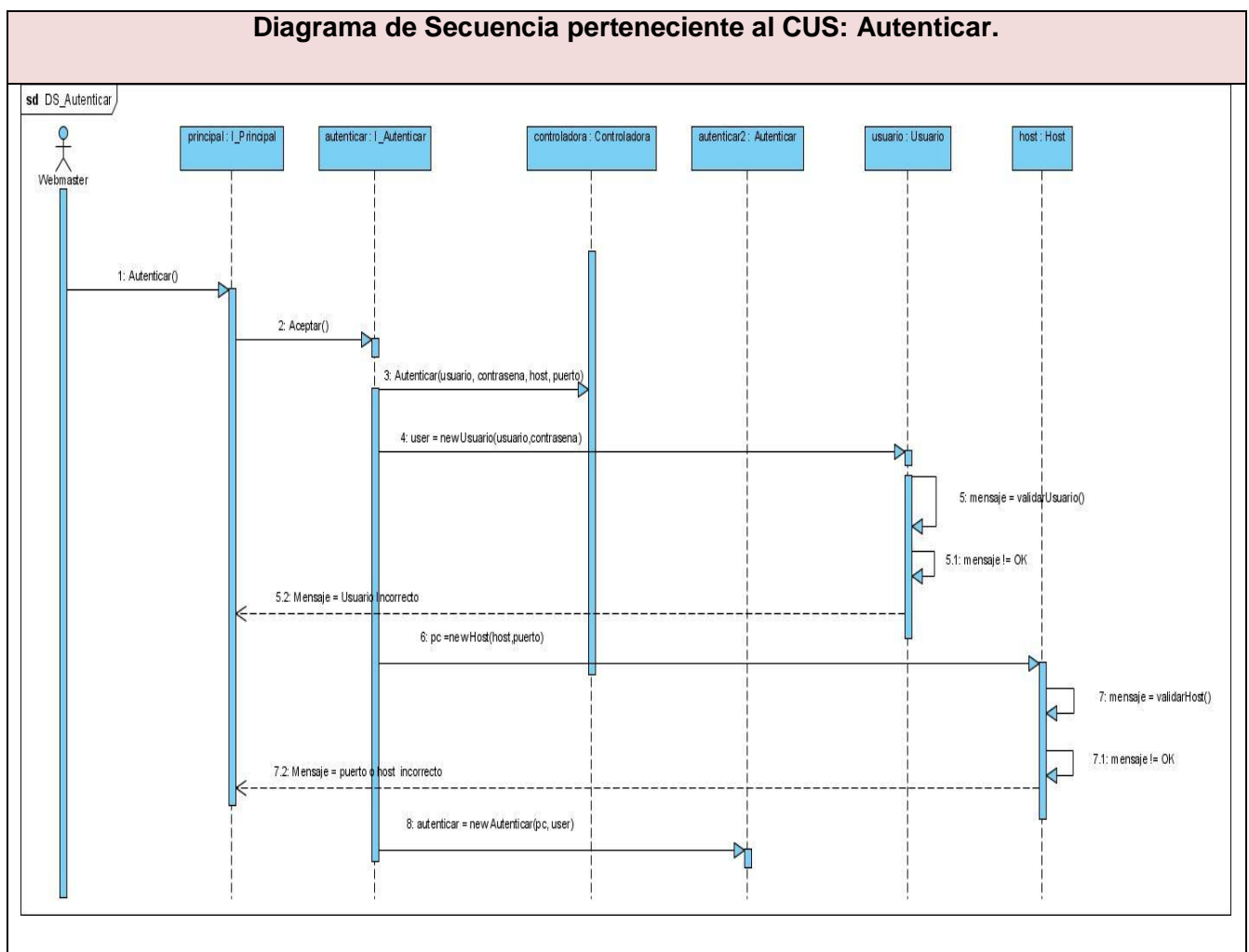


Tabla 11 Diagrama de Secuencia perteneciente al CUS: Autenticar.



Diagrama de Secuencia perteneciente al CUS Conocer características de una página: Información Adicional.

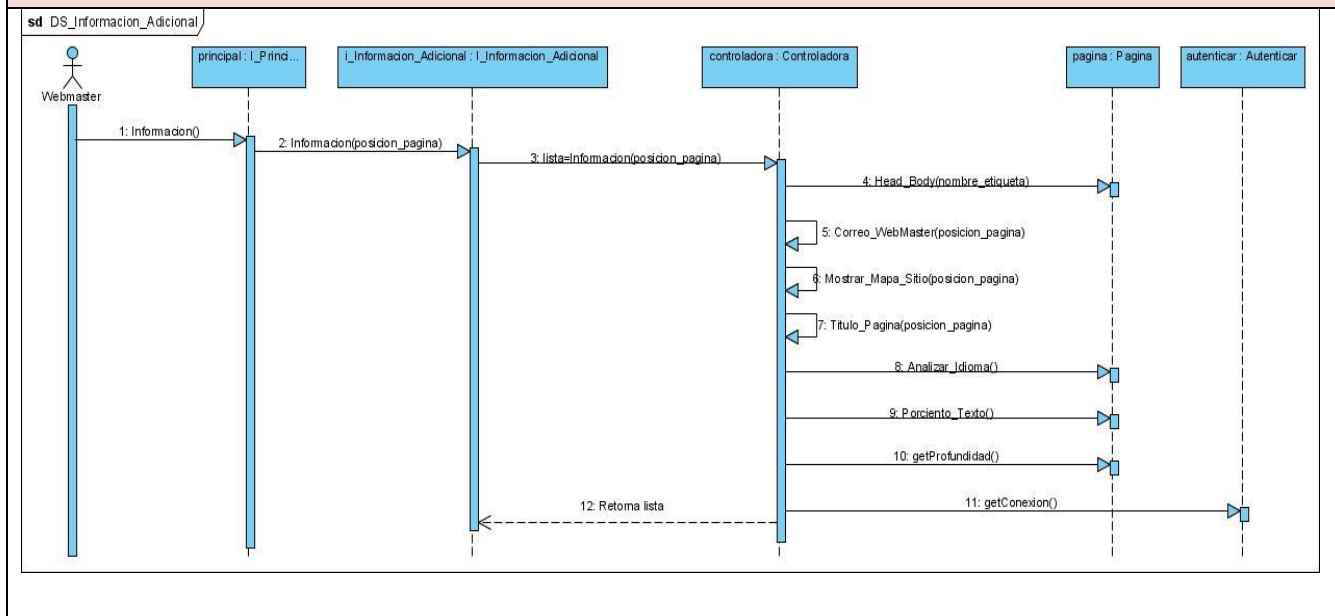


Tabla 12 Diagrama de Secuencia perteneciente al CUS Conocer características de una página: Información Adicional.

Diagrama de Secuencia perteneciente al CUS Gestionar Meta: Eliminar etiqueta meta.

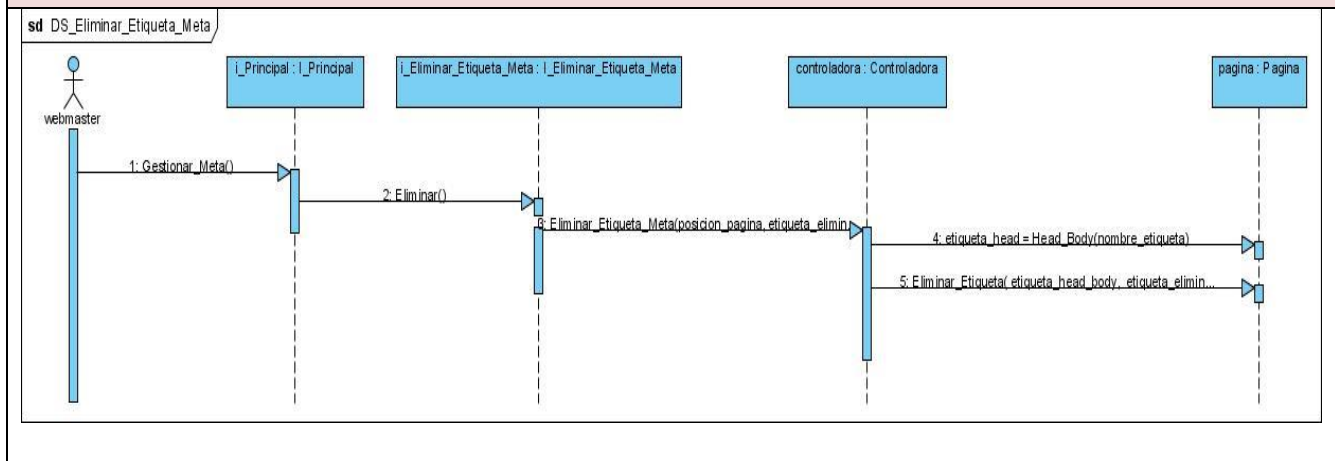


Tabla 13 Diagrama de Secuencia perteneciente al CUS Gestionar Meta: Eliminar etiqueta meta.



Diagrama de Secuencia perteneciente al CUS: Mostrar árbol.

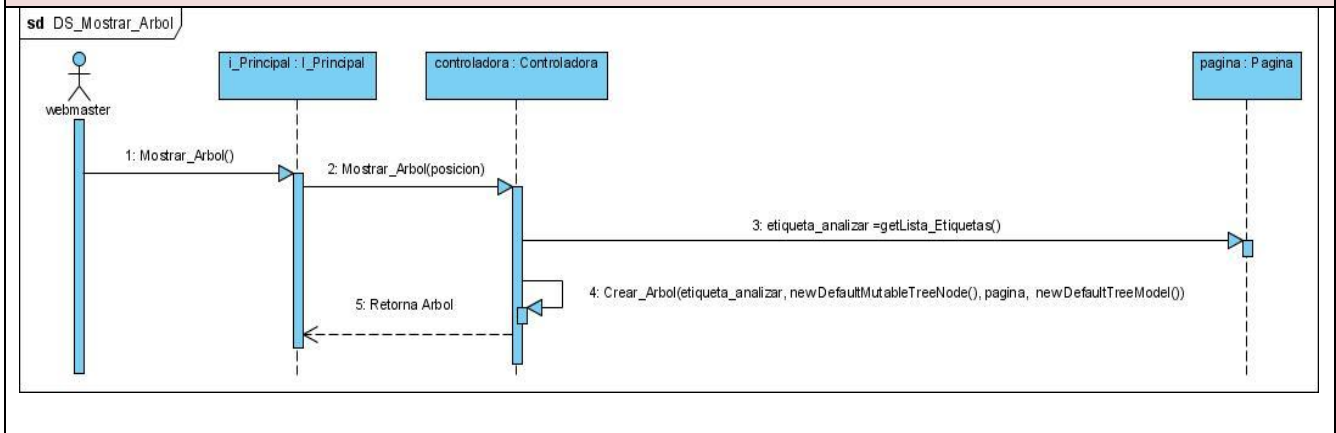


Tabla 14 Diagrama de Secuencia perteneciente al CUS: Mostrar árbol.

Diagrama de Secuencia perteneciente al CUS Gestionar Meta: Insertar meta.

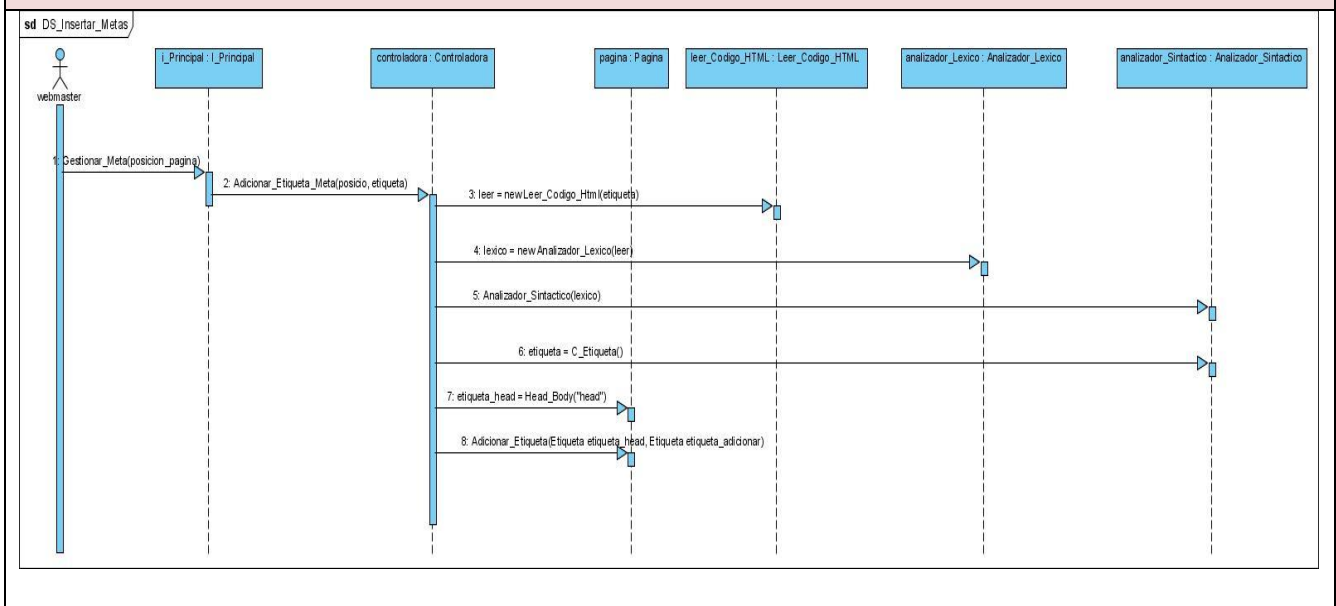


Tabla 15 Diagrama de Secuencia perteneciente al CUS Gestionar Meta: Insertar meta.



Diagrama de Secuencia perteneciente al CUS: Mostrar consulta.

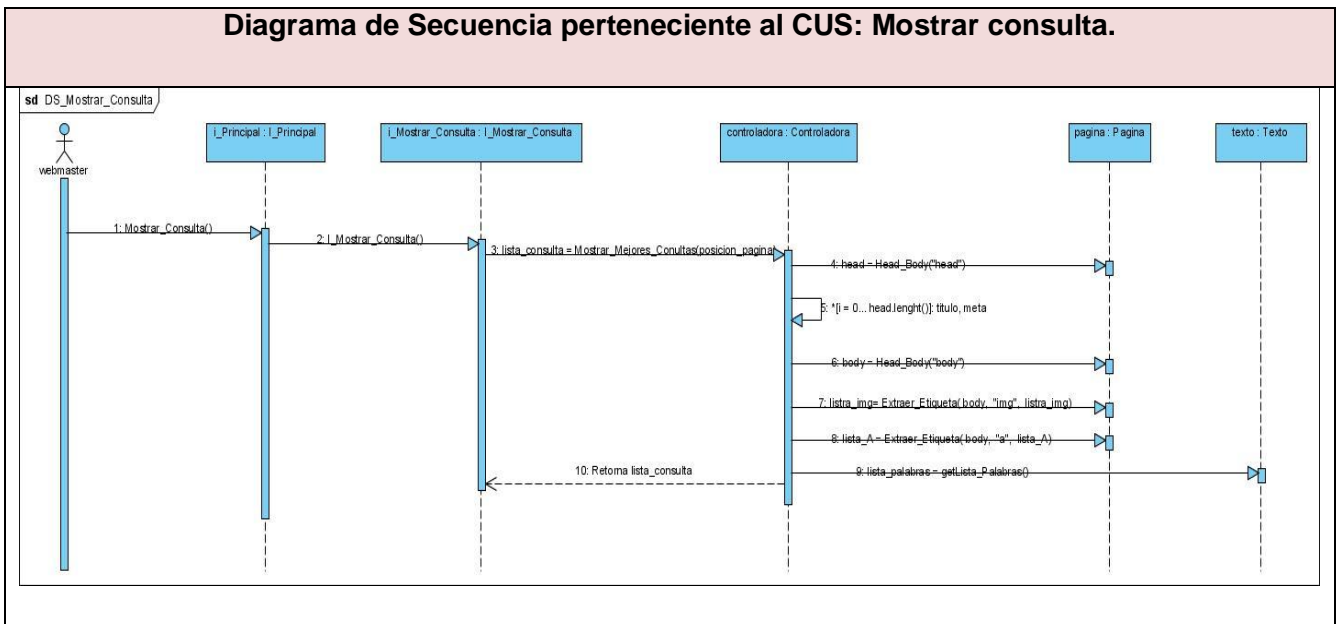


Tabla 16 Diagrama de Secuencia perteneciente al CUS: Mostrar consulta.

Diagrama de Secuencia perteneciente al CUS: Mostrar reporte.

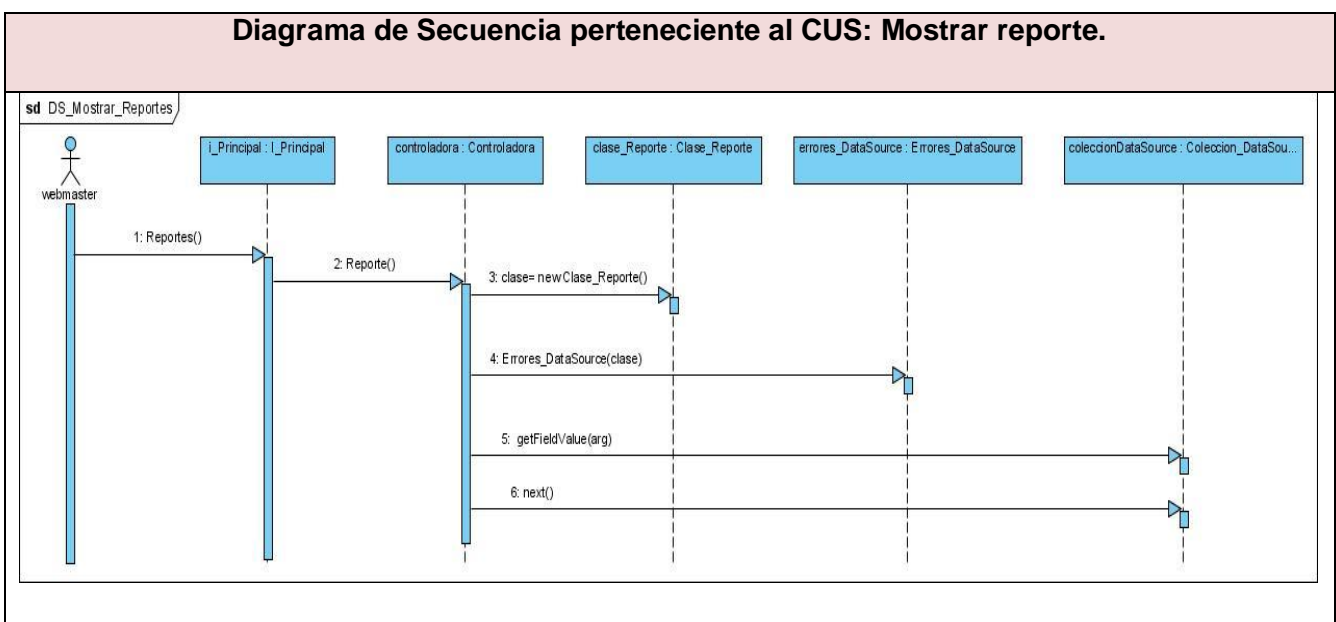


Tabla 17 Diagrama de Secuencia perteneciente al CUS: Mostrar reporte.



Capítulo 2: Análisis Y Diseño Del Sistema

Diagrama de Secuencia perteneciente al CUS: Validar HTML.

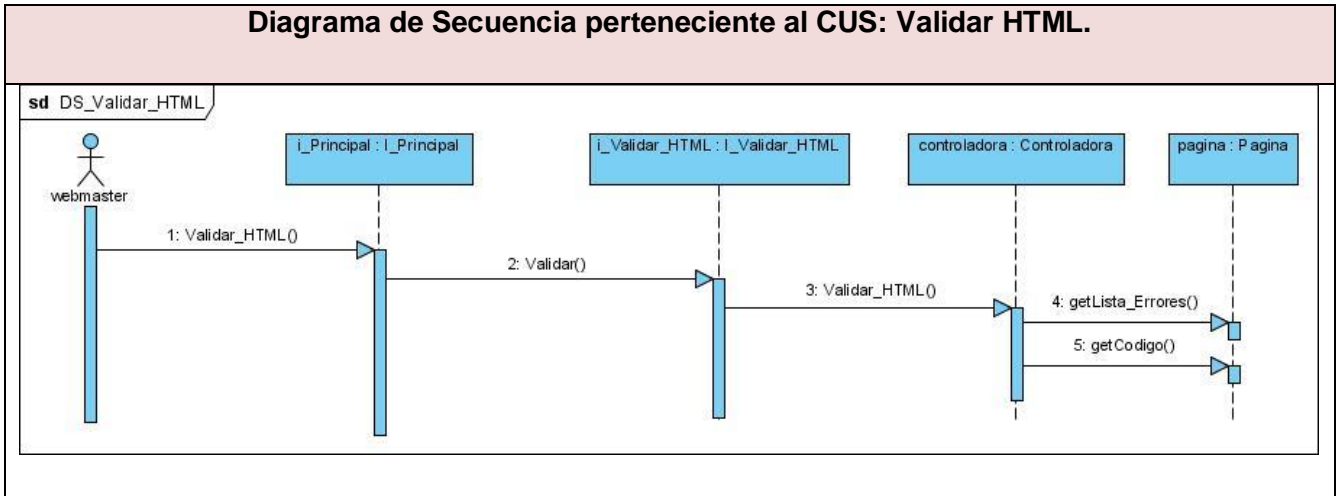


Tabla 18 Diagrama de Secuencia perteneciente al CUS: Validar HTML.

Diagrama de Secuencia perteneciente a los CUS: Descargar HTML y Analizar HTML.

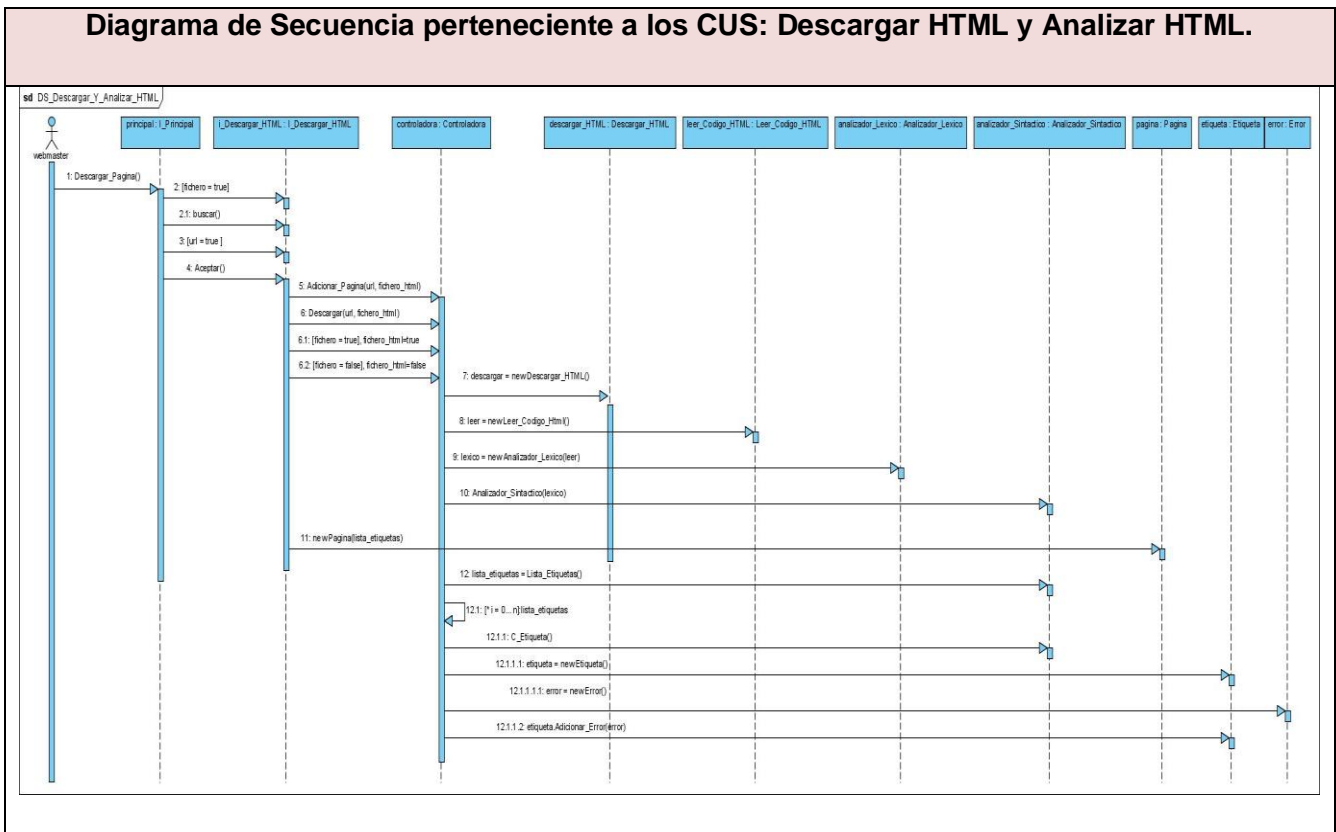


Tabla 19 Diagrama de Secuencia perteneciente a los CUS: Descargar HTML y Analizar HTML.

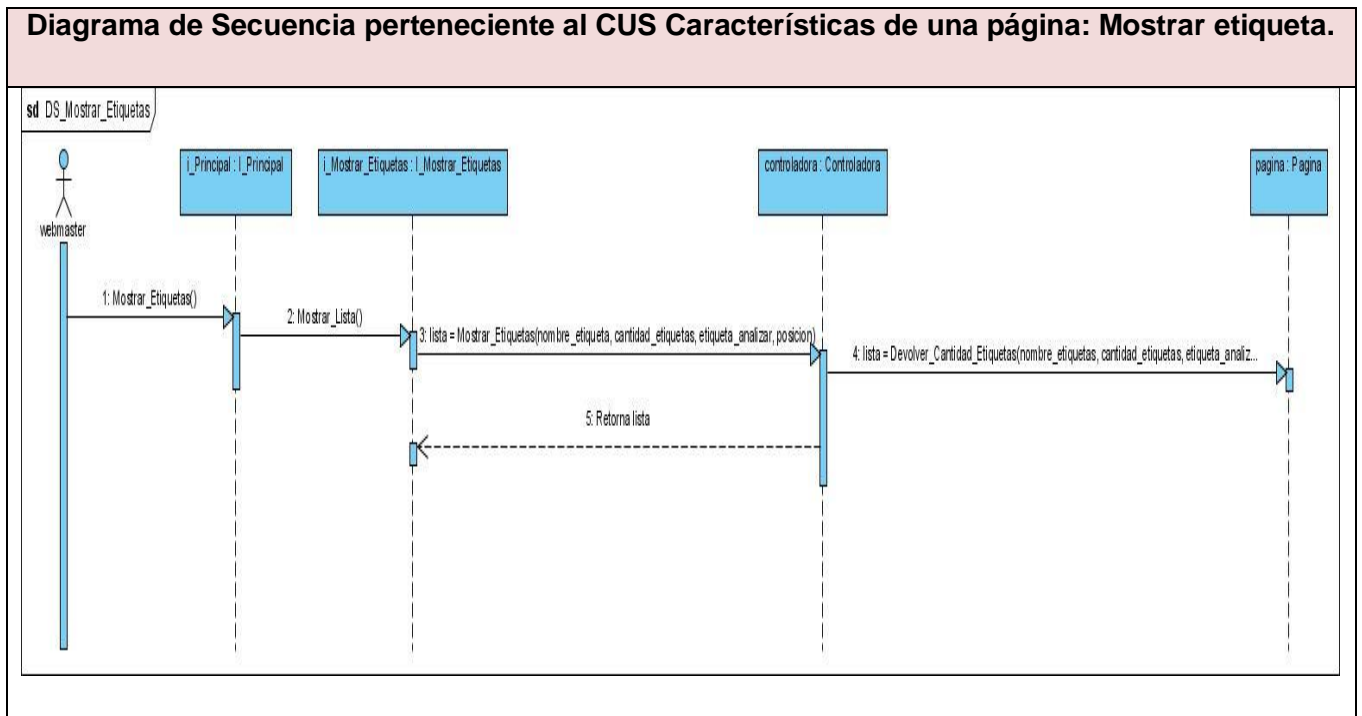


Tabla 20 Diagrama de Secuencia perteneciente al CUS Características de una página: Mostrar etiqueta.

2.5.2.2 Diagramas de clases del diseño.

Los diagramas de clases de diseños son uno de los artefactos generados durante el flujo de trabajo Análisis y Diseño de la fase de elaboración la cual tiene como objetivo lograr una mayor estabilidad en la arquitectura, requerimientos y planes, así como mitigar los riesgos.

Los diagramas de clases de diseños están regidos y estructurados por clases, las cuales están compuestas por atributos y métodos que tienen correspondencia directa con los métodos en la implementación. Estos atributos y métodos se les define la visibilidad: **public**, **protected** o **private** y se especifican utilizando la sintaxis del lenguaje de programación elegido.



Capítulo 2: Análisis Y Diseño Del Sistema

A continuación se muestran los diagramas de clases generados durante el desarrollo del sistema.

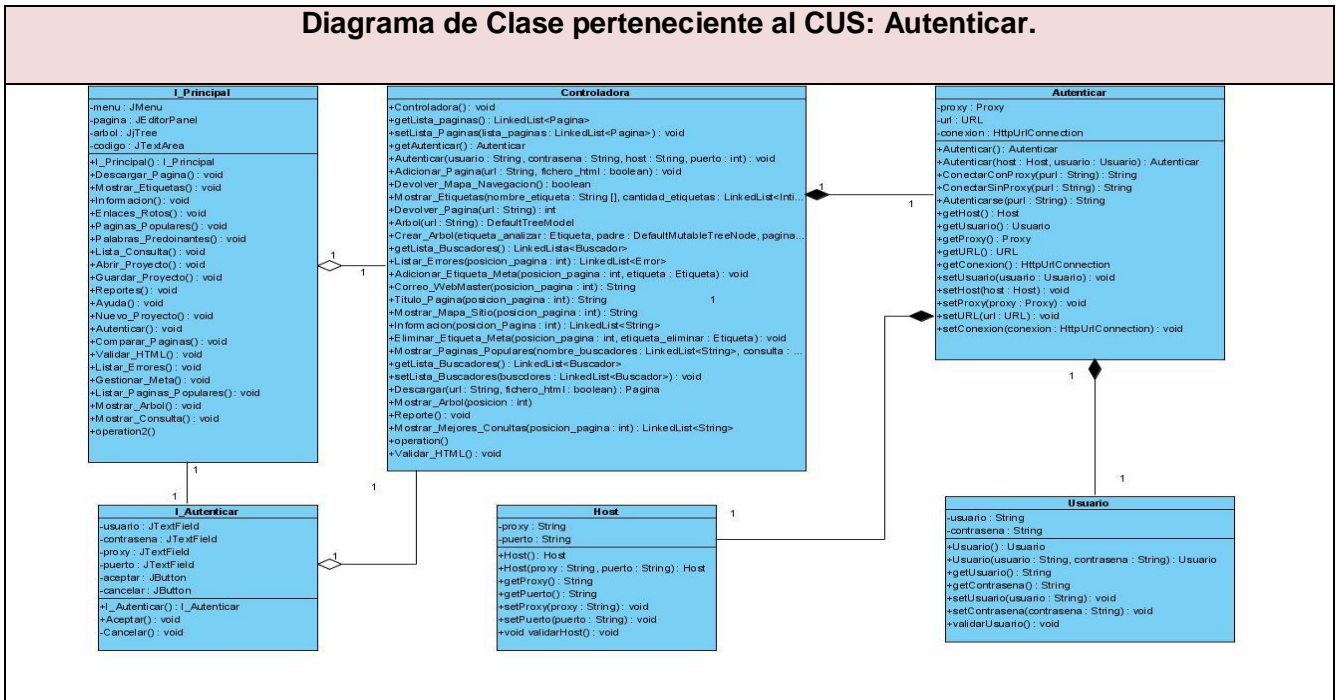


Tabla 21 Diagrama de Clase perteneciente al CUS: Autenticar.

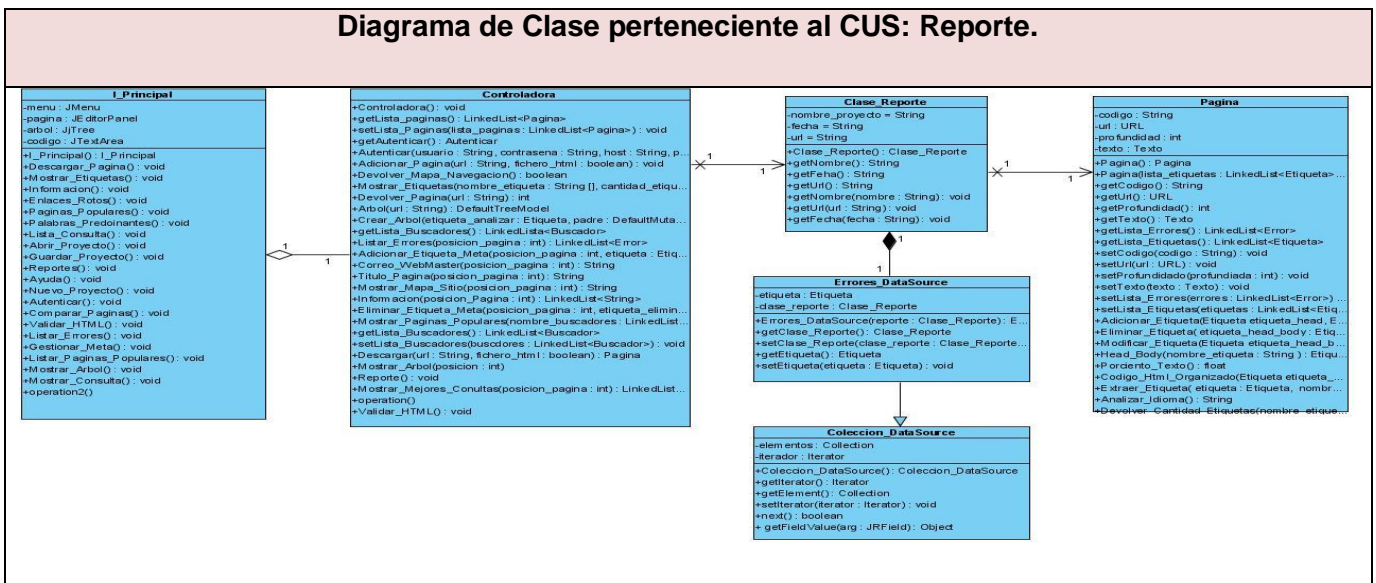


Tabla 22 Diagrama de Clase perteneciente al CUS: Reporte.



Diagrama de Clase perteneciente a los CUS: Descargar y analizar HTML.

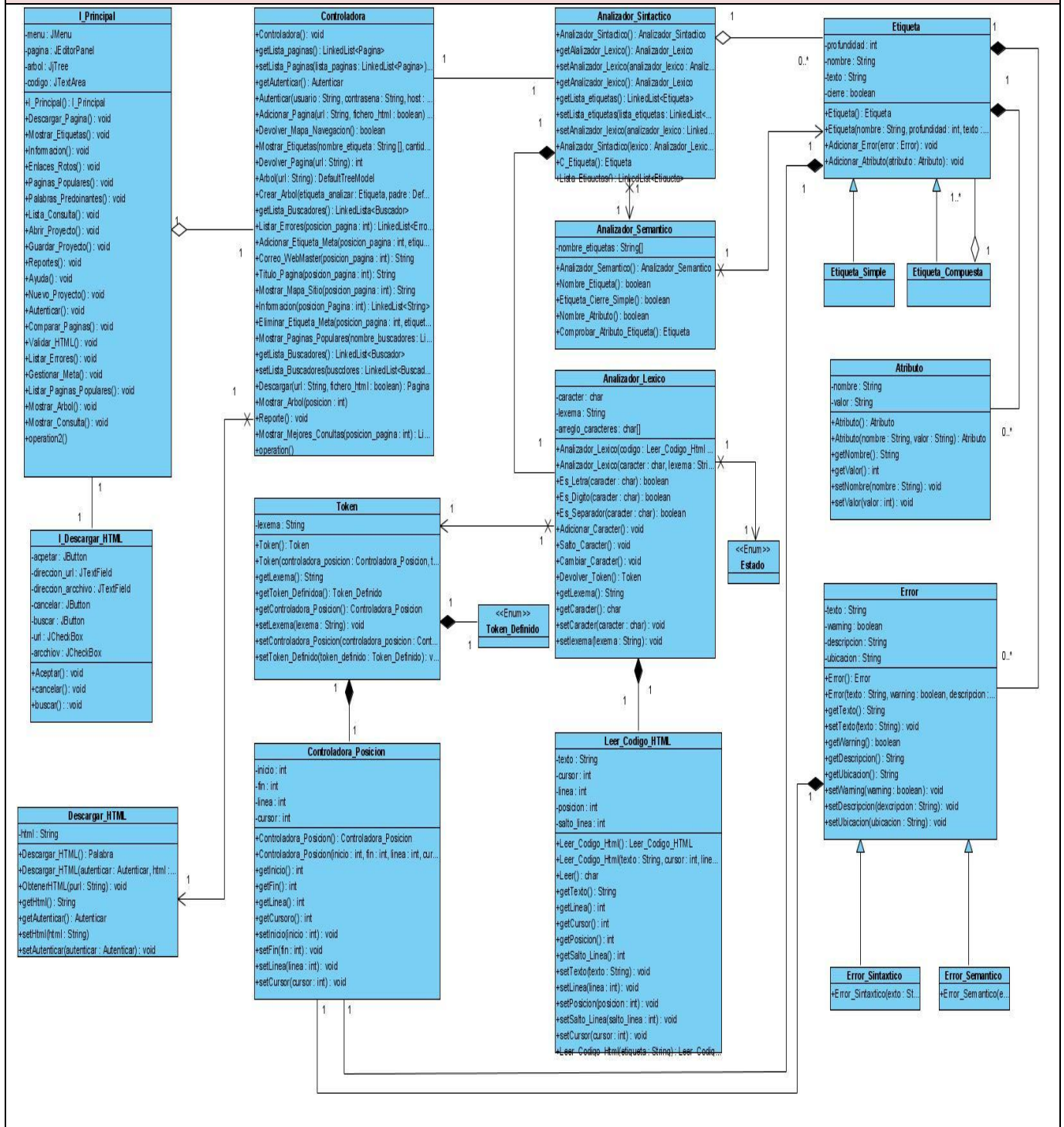


Tabla 23 Diagrama de Clase perteneciente a los CUS: Descargar y analizar HTML.



Diagrama de Clase de los CUS: (Restantes).

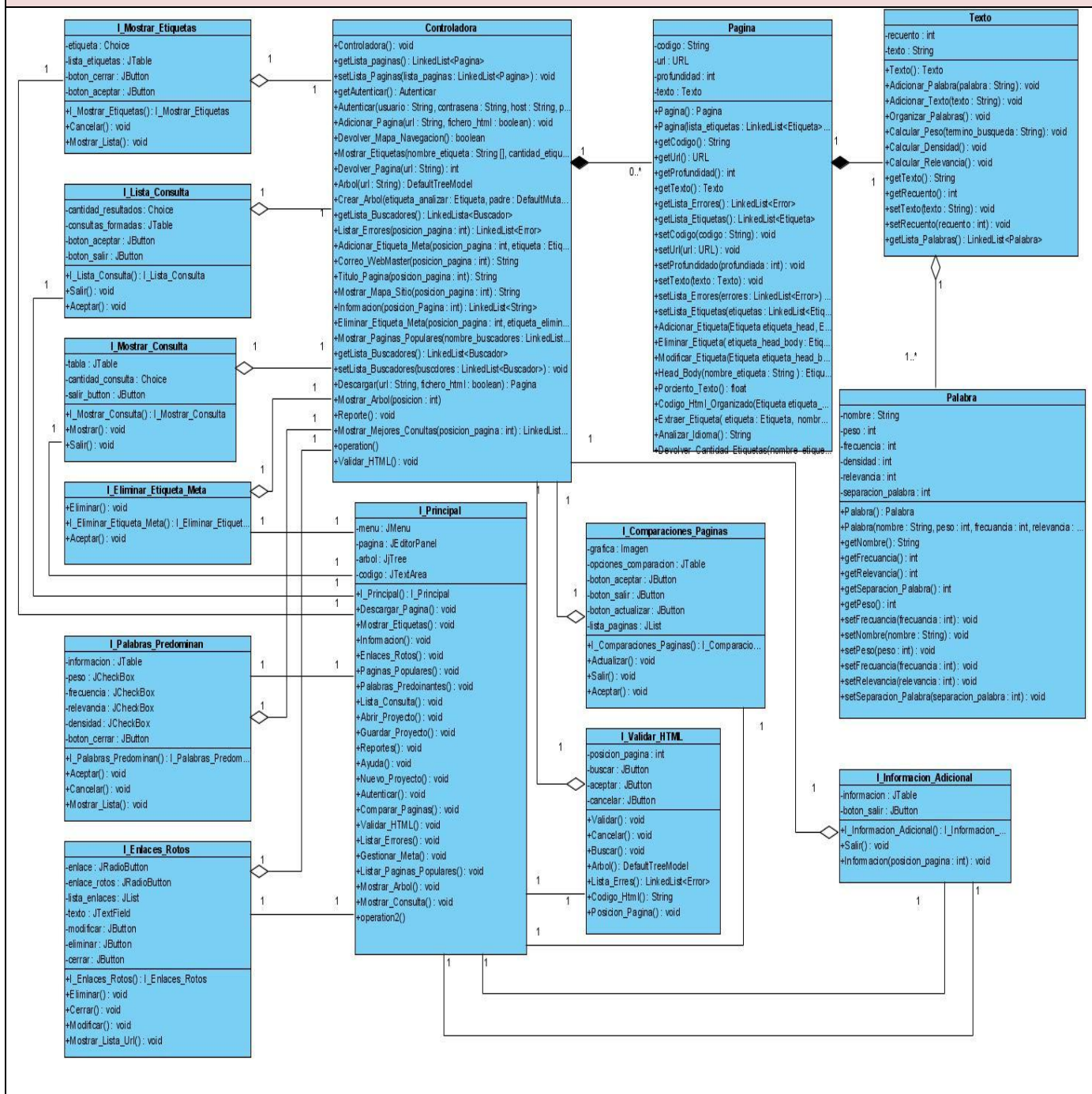


Tabla 24 Diagrama de Clase de los CUS: (Restantes).



2.6 Estilo Arquitectónico.

Para que un sistema sea desarrollado con la calidad necesaria es importante establecer desde sus inicios una arquitectura que cumpla tanto con los requerimientos funcionales como con los no funcionales. La arquitectura define los estilos o grupos de estilos adecuados para cumplir con las peticiones del cliente.

Según la IEEE 1471-2000 la arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

Los estilos arquitectónicos son un conjunto de reglas de diseño que identifica las clases de componentes y conectores que se pueden utilizar para componer en sistema o subsistema, junto con las restricciones locales o globales de la forma en que la composición se lleva a cabo. (3)

Los patrones surgen de la experiencia acumulada de seres humanos al tratar de lograr ciertos objetivos con el fin de promover buenas prácticas. Un patrón es un modelo que describe un problema que ocurre una y otra vez en el entorno, de modo que pueda ser usada en más de una ocasión sin hacerlo dos veces de la misma forma. También es una solución estándar para un problema común de programación que conduce a una arquitectura más pequeña, simple y comprensible.

2.6.1 Patrones Arquitectónicos.

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

Arquitectura en capas.

Se utilizó el patrón arquitectura n capas debido a que permite organizar el modelo de diseño, donde estas suelen ser entidades complejas compuestas de varios paquetes o subsistemas. Las capas pueden estar físicamente distribuidas y los componentes de una capa solo pueden hacer referencia a componentes en capas inmediatamente inferiores.



Garlan y Shaw definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. (4)

El estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. Proporciona una amplia reutilización. Admite muy naturalmente optimizaciones y refinamientos. Simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores.

La aplicación se divide en tres capas lógicas:

Capa de presentación: Recoge todos los aspectos del software que tiene que ver con las interfaces. Presenta el sistema al usuario de modo que interactúa con los diferentes tipos de usuarios comunicando y capturando información. Incluye el manejo y aspecto de las ventanas formularios y botones.

Capa de aplicación: Reciben las peticiones del usuario y se envían las respuestas tras el proceso. Contiene las clases controladoras y entidades del negocio.

Capa de datos: Almacena todos los datos del sistema. Los datos son recogidos en cualquier fuente de información ya sea una base de datos, un servicio de directorio como el servidor LDAP o un conjunto de documentos XML.

2.6.2 Patrones de diseño.

Un patrón de diseño es: una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular.

A continuación se muestran los patrones utilizados en la realización de la aplicación

GRASP: Describen los principios fundamentales de la asignación de responsabilidades a una clase.



- ✓ **Experto**: Su objetivo es asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir determinada responsabilidad. Distribuye el comportamiento entre las clases que cuentan con la información requerida.
- ✓ **Creador**: Plantea que una instancia de un objeto tiene que ser creada por el objeto que tiene la información necesaria.
- ✓ **Bajo Acoplamiento**: Su objetivo es asignar las responsabilidades de tal forma que las clases se comuniquen con el menor número de clases que sea posible.
- ✓ **Alta Cohesión**: Su objetivo es asignar a las clases, responsabilidades de tal forma que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad.

GOF:

- ✓ **Singleton**: Garantiza que una sola clase sólo tenga una instancia, de modo que proporcione un punto de acceso global a ella. Permite encapsular su única instancia permitiendo tener un control estricto sobre como y cuando acceden los clientes a ella.

2.7 Prototipo de Interfaz de Usuario.

Otros de los aspectos a tener en cuenta para el desarrollo de la aplicación son los diseños de las interfaces de usuario (UI) que han sido definidos y elaborados. Se realizaron con el objetivo de posibilitar las pruebas al diseño de la de la UI. De esta forma se asegura que se está construyendo el sistema correcto con las funcionalidades requeridas, antes de perder mucho tiempo y recursos en el desarrollo.

Se hace necesario tener un diseño con calidad, de tal forma que los usuarios se sientan a gusto con la aplicación, de modo que las interfaces deben ser agradables y amigables. Para lograr lo planteado se trabajó sobre la base de algunos principios básicos del diseño como: se evitaron fuertes contrastes que puedan dificultar la concentración del usuario y que puedan resultar llamativos; se usaron los mismos tipos de menú definidos en el sistema y se trató de tener siempre la misma posición para los componentes para que los usuarios logran tener un mejor entendimiento del entorno.



Capítulo 2: Análisis Y Diseño Del Sistema

A continuación se presentarán las interfaces de usuarios que responderán a los requisitos funcionales más importantes.

Interfaz de usuario: Principal.

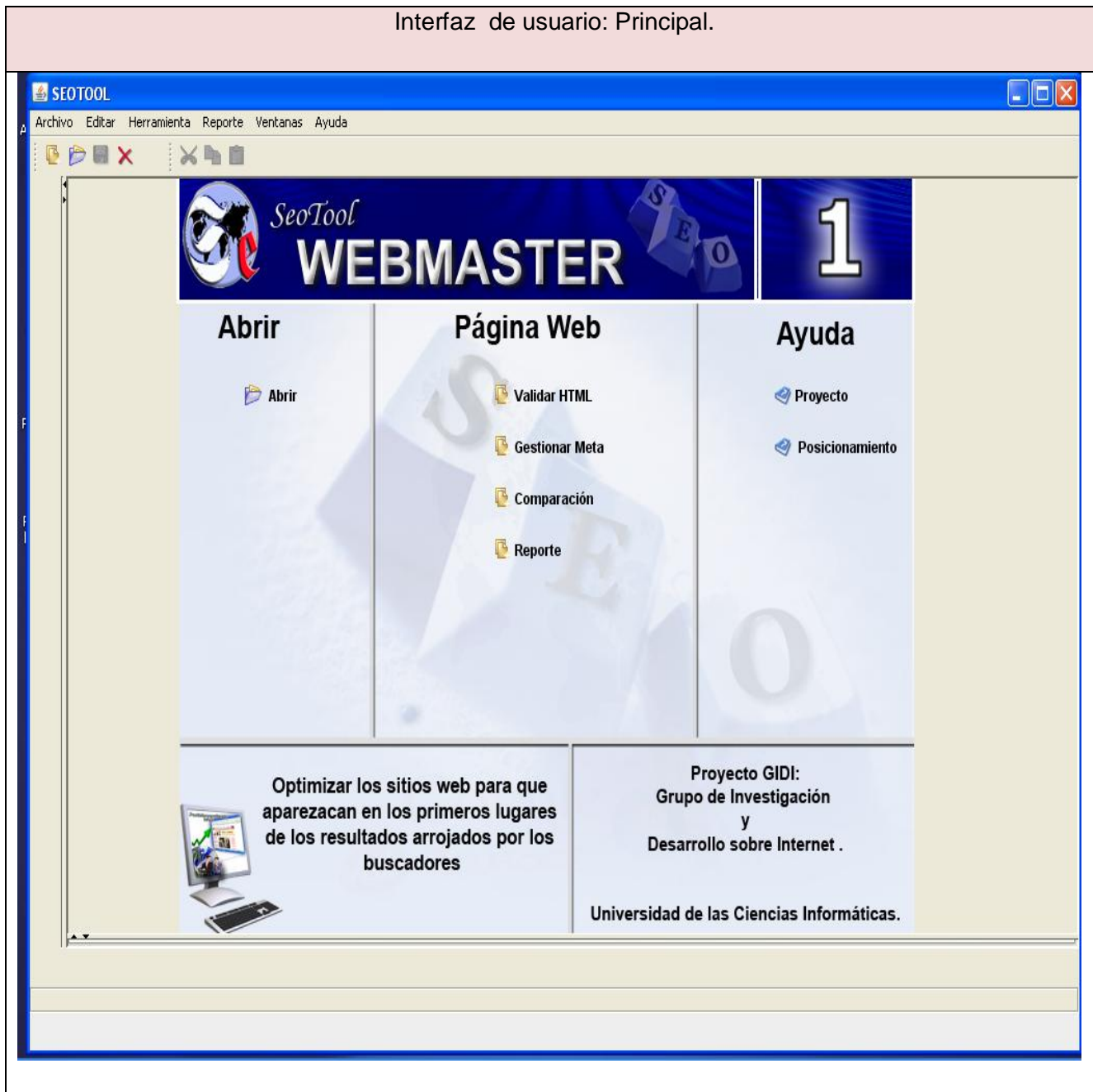


Tabla 25 Interfaz de usuario: Principal.



Interfaz de usuario: Validar HTML.

The screenshot displays a web browser window titled "Interfaz de usuario: Validar HTML." The browser's address bar shows "http://intranet...". The main content area displays HTML code for a menu, including a table with a colspan="3" and a list of links with href and title attributes. The left pane shows the DOM tree structure, and the bottom pane shows a metadata table.

Información	
Título	Bienvenidos al Portal de la Ciudad Digital
Peso de la página	-----
Mapa de Navegación	No posee
Por ciento de texto	0.0 %
Idioma	Español
Fecha de Última Modificación	-----

Tabla 26 Interfaz de usuario: Validar HTML.



Interfaz de usuario: Mostrar árbol.

```
Página
├── <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
├── <html xmlns="http://www.w3.org/1999/xhtml">
│   ├── <head>
│   │   ├── <title>
│   │   ├── <SCRIPT LANGUAGE="JavaScript">
│   │   ├── <script type="text/javascript">
│   │   ├── <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
│   │   ├── <style type="text/css" media="all">
│   │   ├── <style type="text/css" media="all">
│   │   └── </head>
│   └── <body >
│       ├── <div class="general_region" id="total">
│       │   ├── <div class="top_region" id="banner">
│       │   │   ├── <table cellpadding="0" cellspacing="0" border="0" id="table_banner">
│       │   │   │   ├── <tr>
│       │   │   │   ├── <tr>
│       │   │   │   └── </table>
│       │   │   └── </div>
│       │   ├── <div class="center_region" id="content">
│       │   │   ├── <table id="table_content" cellpadding="0" cellspacing="0" border="0">
│       │   │   │   ├── <tr>
│       │   │   │   │   ├── <td valign="top" class="main">
│       │   │   │   │   ├── <td valign="top" class="center_block">
│       │   │   │   │   └── <td valign="top" class="side_right">
│       │   │   │   └── </tr>
│       │   │   │   ├── <tr>
│       │   │   │   └── </table>
│       │   │   └── </div>
│       │   └── </div>
│       └── </body>
└── </html>
```

Tabla 27 Interfaz de usuario: Mostrar árbol.



Interfaz de usuario: Autenticar.

Autenticarse

Usuario: rsalazar

Contraseña: ●●●●●●●●●●

Proxy: 10.0.0.1

Puerto: 8080

Aceptar Cancelar

Tabla 28 Interfaz de usuario: Información Autenticar.

Interfaz de usuario: Información adicional.

	Información
Título	Bienvenidos al Portal de la Ciudad Digital
Peso de la página	-----
Mapa de Navegación	No posee
Porcentaje de texto	52.837593 %
Idioma	Español
Fecha de Última Modificación	-----
Tiempo en cargar la Página	-----
Profundidad	15
Correo del Webmaster	NO tiene correo

Salir

Tabla 29 Interfaz de usuario: Información adicional.



Interfaz de usuario: Mostrar cantidad de etiquetas.

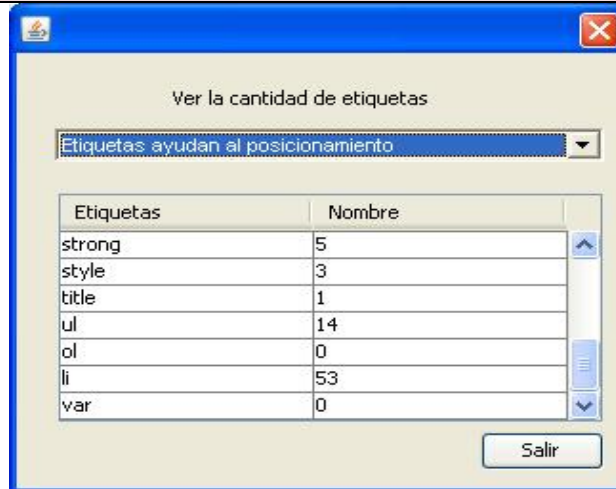


Tabla 30 Interfaz de usuario: Mostrar cantidad de etiquetas.

Interfaz de usuario: Ayuda.

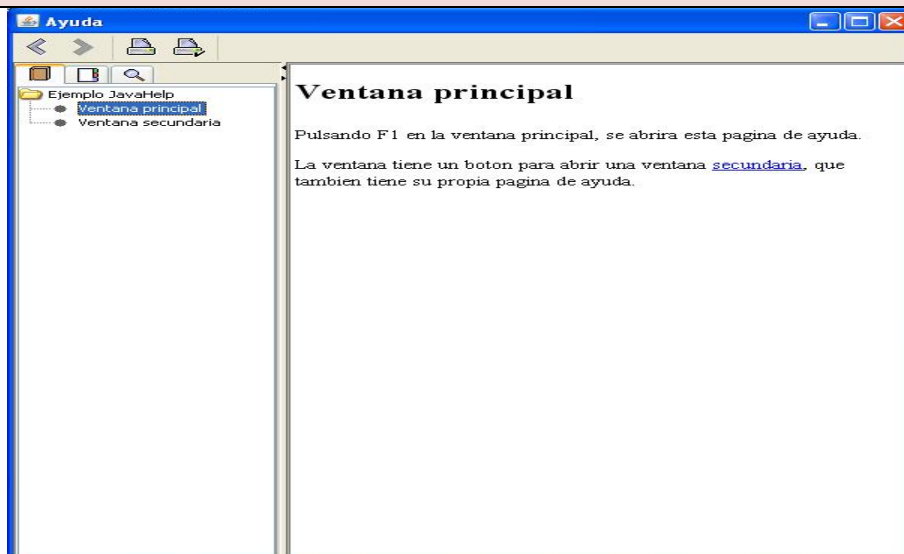


Tabla 31 Interfaz de usuario: Ayuda.



Interfaz de usuario: Reporte.

(en Internet)

Informe de Posicionamiento

http://intranet

Este informe se ha generado para ofrecer información sobre la valoración de distintos aspectos que influyen del Posicionamiento Web.

Datos que has utilizado para realizar este informe

Dirección web: http://intranet

Nombre de la Empresa:

Palabras Claves:



Tabla 32 Interfaz de usuario: Reporte.

Interfaz de usuario: Mostrar comparaciones.

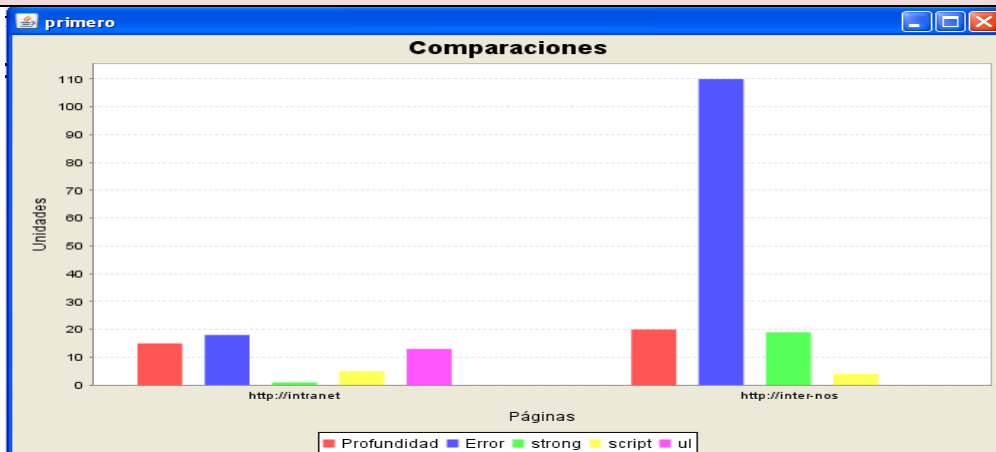


Tabla 33 Interfaz de usuario: Mostrar comparaciones.



2.8 Descripción de las clases más importantes.

A continuación se describen las clases más importantes de la aplicación especificando de manera más clara los atributos y métodos que poseen las mismas con el fin de obtener de una forma más clara cual es su papel que juega dentro del sistema.

Nombre: Pagina	
Tipo de clase: Entidad	
Descripción : Esta clase es la que recoge todos los atributos que debe tener una página web.	
Atributo	Tipo
lista_etiquetas	LinkedList<Etiqueta>
codigo	String
texto	Texto
lista_errores	LinkedList<Error>;
profundidad	int
url	String
peso_pagina	int
tiempo_cargar	int
Para cada responsabilidad:	
Nombre:	public Pagina(LinkedList<Etiqueta> lista, String codigo, String url)
Descripción:	Constructor de la clase encargado de inicializar todos los atributos de la misma.
Nombre:	Adicioanr_Etiqueta(Etiqueta etiqueta_head, Etiqueta etiqueta_adicionar)
Descripción:	Este método es el encargado de adicionar nuevas etiquetas a la lista de etiquetas.
Nombre:	Etiquetas_Linielaes(Etiqueta etiqueta, LinkedList<Etiqueta> lista)
Descripción:	Esta función permite listar todas las etiquetas
Nombre:	Eliminar_Etiqueta(Etiqueta etiqueta_head_body, Etiqueta etiqueta_eliminar)
Descripción:	Permite eliminar una etiqueta pasada por parámetros.
Nombre:	Devolver_Cantidad_Etiquetas(String[] nombre_etiquetas, LinkedList<Integer> cantidad_etiquetas, Etiqueta etiqueta_analizar)
Descripción:	Devuelve la cantidad de etiquetas de un tipo específico.
Nombre:	Enlaces_Rotos()
Descripción:	Devuelva una lista con los enlaces rotos que tiene la página.



Nombre:	Extraer_Etiqueta(Etiqueta etiqueta, String nombre, LinkedList<Etiqueta> retornar_lista_etiqueta)
Descripción:	Devuelve la etiqueta pasada por parámetros de una etiqueta compuesta especificada.
Nombre:	Analizar_Idioma()
Descripción:	Analiza y devuelve el idioma de la página.

Tabla 34 Descripción de la clase Página.

Nombre: Anlaizador_Sintactico	
Tipo de clase: Entidad	
Descripción : Esta clase es la recoge todos los atributos que tiene una etiqueta HTML	
Atributo	Tipo
lista_etiquetas	LinkedList<Etiqueta>
codigo	String
texto	Texto
lista_errores	LinkedList<Error>;
profundidad	int
url	String
peso_pagina	int
tiempo_cargar	int
Para cada responsabilidad:	
Nombre:	public Pagina(LinkedList<Etiqueta> lista, String codigo, String url)
Descripción:	Constructor de la clase encargado de inicializar todos los atributos de la misma.
Nombre:	Adicioanr_Etiqueta(Etiqueta etiqueta_head, Etiqueta etiqueta_adicionar)
Descripción:	Este método es el encargado de adicionar nuevas etiquetas a la lista de etiquetas.
Nombre:	Etiquetas_Linielaes(Etiqueta etiqueta, LinkedList<Etiqueta> lista)
Descripción:	Esta función permite listar todas las etiquetas
Nombre:	Eliminar_Etiqueta(Etiqueta etiqueta_head_body, Etiqueta etiqueta_eliminar)
Descripción:	Permite eliminar una etiqueta pasada por parámetros.
Nombre:	Devolver_Cantidad_Etiquetas(String[] nombre_etiquetas, LinkedList<Integer> cantidad_etiquetas, Etiqueta etiqueta_analizar)



Capítulo 2: Análisis Y Diseño Del Sistema

Descripción:	Devuelve la cantidad de etiquetas de un tipo específico.
Nombre:	Enlaces_Rotos()
Descripción:	Devuelva un alista con los enlaces rotos que tiene la página.
Nombre:	Extraer_Etiqueta(Etiqueta etiqueta, String nombre, LinkedList<Etiqueta> retornar_lista_etiqueta)
Descripción:	Devuelve la etiqueta pasada por parámetros de una etiqueta compuesta especificada.
Nombre:	Analizar_Idioma()
Descripción:	Analiza y devuelve el idioma de la página.

Tabla 35 Descripción de la clase Anlaizador_Sintactico.

Nombre: Etiqueta	
Tipo de clase: Entidad	
Descripción : Esta clase es la que recoge todos los atributos que caracterizan a una etiqueta HTML.	
Atributo	Tipo
lista_atributo	LinkedList<Atributo>
nombre	String
profundidad	int
texto	String
cierre	boolean
posicion	Controladora_Posicion
lista_errore	LinkedList<Error>
Para cada responsabilidad:	
Nombre:	Etiqueta(LinkedList<Atributo> lista_atributo, String nombre, String texto, boolean cierre, Controladora_Posicion posicion, LinkedList<Error> lista_errores, int profundidad)
Descripción:	Constructor de la clase encargado de inicializar todos los atributos de la misma.
Nombre:	Adicionar_Atributo(Atributo atributo)
Descripción:	Adiciona un nuevo atributo a la etiqueta.
Nombre:	Adicionar_Error(Error error)
Descripción:	Adiciona un nuevo error la etiqueta.



Tabla 36 Descripción de la clase Etiqueta.

Nombre: Leer_Codigo_Html	
Tipo de clase: Entidad	
Descripción : Esta clase es la que recoge todos los atributos que permiten ir leyendo el código HTML por token.	
Atributo	Tipo
texto	String
cursor	int
línea	int
posicion	int
salto_linea	int
Para cada responsabilidad:	
Nombre:	Leer_Codigo_Html()
Descripción:	Constructor de la clase encargado de inicializar todos los atributos de la misma.
Nombre:	Leer()
Descripción:	Este método es el que permite leer el código HTML, determinando que tipo de token al que pertenece el mismo.

Tabla 37 Descripción de la clase Leer_Codigo_Html.

Nombre: Autenticar	
Tipo de clase: Entidad	
Descripción : Esta clase es la que recoge todos los atributos que permiten ir leyendo el código HTML por token.	
Atributo	Tipo
usuario	Usuario
host	Host
proxy	Proxy
conexion	URLConnection
url	URL



Para cada responsabilidad:	
Nombre:	Autenticar(Usuario pusuario,Host phost)
Descripción:	Constructor de la clase encargado de inicializar todos los atributos de la misma.
Nombre:	ConectarConProxy(String purl)
Descripción:	Este método es el que establece la conexión sin especificación de un proxy.
Nombre:	Autenticarse(String purl)
Descripción:	Establece la conexión en caso de que sea con proxy o sin el mismo.
Nombre:	ConectarSinProxy(String purl)
Descripción:	Este método es el que establece la conexión a través del puerto y el proxy determinados por el usuario.

Tabla 38 Descripción de la clase Autenticar.

Nombre: ErroresDataSource	
Tipo de clase: Controladora.	
Descripción : Esta clase es la encargada de construir el reporte a partir de los datos que posee la clase Clase_Reporte.	
Atributo	Tipo
etiqueta	Etiqueta
clase_reporte	Clase_Reporte
Para cada responsabilidad:	
Nombre:	ErroresDataSource(Clase_Reporte clase)
Descripción:	Constructor de la clase encargado de inicializar todos los atributos de la misma.
Nombre:	next()
Descripción:	Retorna el próximo elemento de la lista de páginas y lo almacena en etiqueta.
Nombre:	getFieldValue()
Descripción:	Es el encargado de asignarle los valores al reporte.

Tabla 39 Descripción de la clase ErroresDataSource.

Nombre: Controladora.
Tipo de clase: Controladora.



Descripción : Esta clase es la encargada de la gestión de toda la información de la aplicación.	
Atributo	Tipo
lista_paginas	LinkedList<Pagina>
autenticar	Autenticar
Para cada responsabilidad:	
Nombre:	Controladora()
Descripción:	Constructor de la clase encargado de inicializar todos los atributos de la misma.
Nombre:	Autenticar(String usuario, String contraseña, String host, int puerto)
Descripción:	Permite construir el objeto autenticar con los datos necesarios para establecer una conexión.
Nombre:	Arbol(int posicion_pagina)
Descripción:	Construye un árbol con la estructura de la página web determinada.
Nombre:	Mapa_Navegacion(int posicion)
Descripción:	Permite saber si una página tiene o no mapa de navegación.
Nombre:	Texto_Pagina(Etiqueta etiqueta_body, Pagina pagina)
Descripción:	Extrae el texto de la página que se encuentra dentro de la etiqueta body.
Nombre:	Correo_WebMaster(int posicion_pagina)
Descripción:	Permite saber si una página tiene o no correo webmasters.
Nombre:	Adicionar_Etiqueta_Meta(int posicion_pagina, String etiqueta)
Descripción:	Permite adicionar etiquetas meta el código HTML.
Nombre:	Palabras_Claves(int posicion_pagina)
Descripción:	Devuelve una lista de palabras claves para el posicionamiento.

Tabla 40 Descripción de la clase Controladora.

2.9 Conclusiones



Capítulo 2: Análisis Y Diseño Del Sistema

La descripción y análisis de la solución propuesta, y el estudio de los diferentes patrones facilitó la realización de los diferentes artefactos generados: clases de análisis y diseño y diagramas de interacción correspondiente al flujo de trabajo Análisis y Diseño. También permitió crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases. Posibilitó ir construyendo una arquitectura solida, robusta y fuerte que soportará el sistema.



Capítulo 3: Implementación y Prueba del Sistema.

3.1 Introducción.

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

3.2 Diagrama de componente.

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

El Diagrama de Componente define cómo las clases, artefactos y otros elementos de bajo nivel, se unen para formar componentes de alto nivel y las conexiones entre ellos. Los componentes son artefactos de software compilados que trabajan acoplados para brindar el comportamiento requerido dentro de las restricciones definidas en el proceso de captura de requisitos.

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

Los diagramas que se presentan a continuación: diagrama de paquetes y un diagrama de componente por cada paquete, tienen como objetivo, explicar gráficamente la estructura y organización general del sistema.

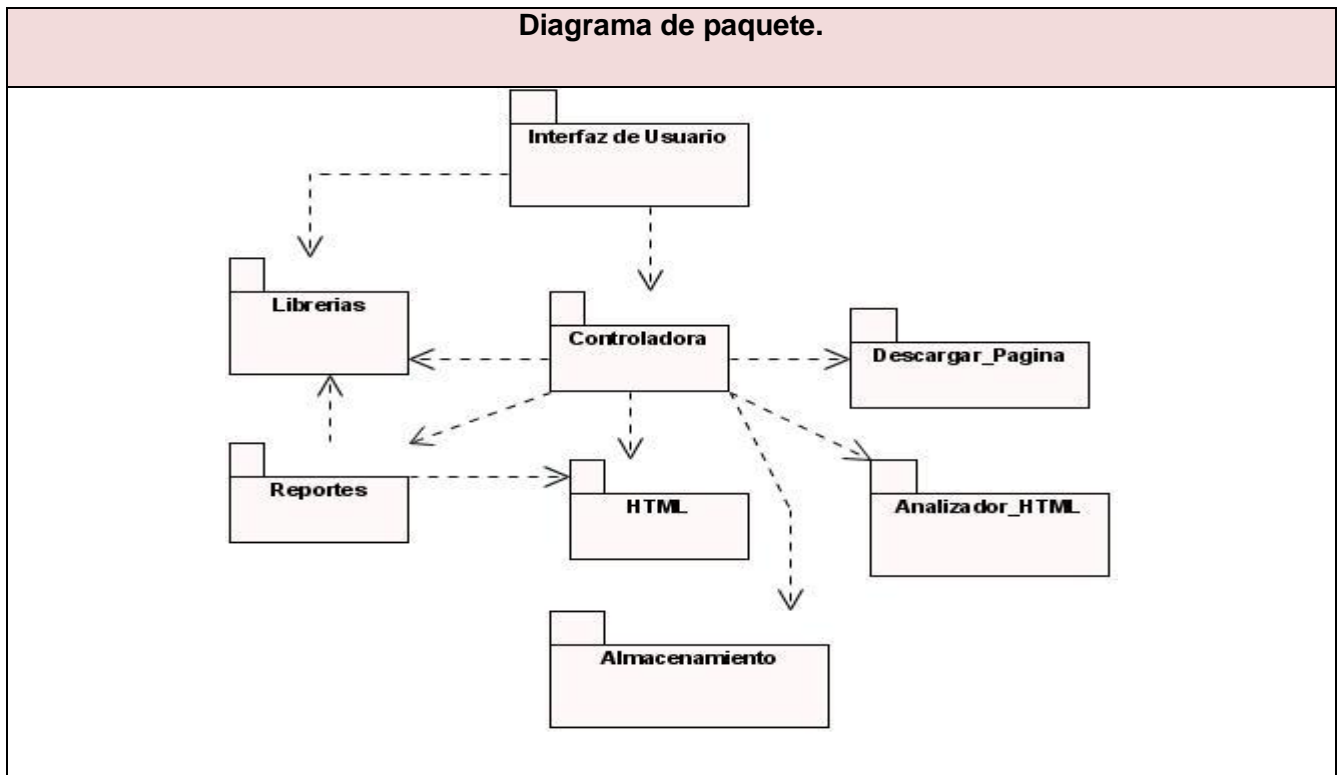


Tabla 41 Diagrama de paquete.

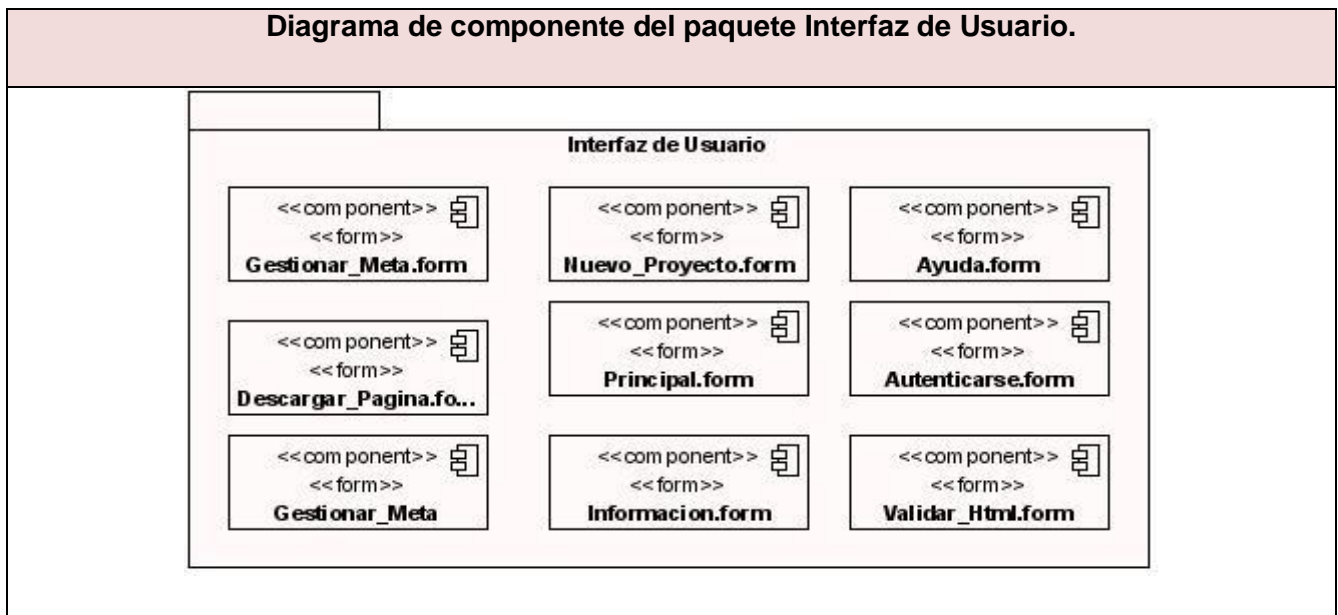


Tabla 42 Diagrama de componente del paquete Interfaz de Usuario.

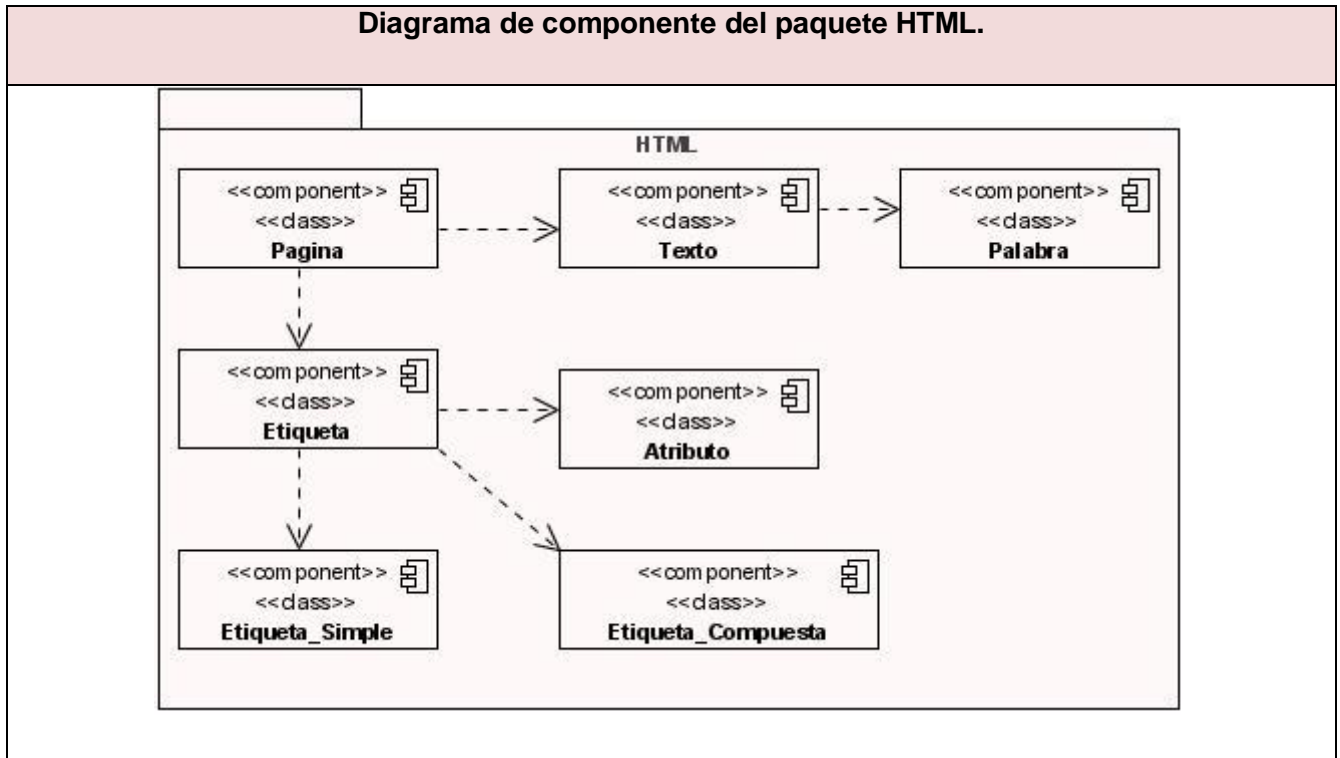


Tabla 43 Diagrama de componente del paquete HTML.

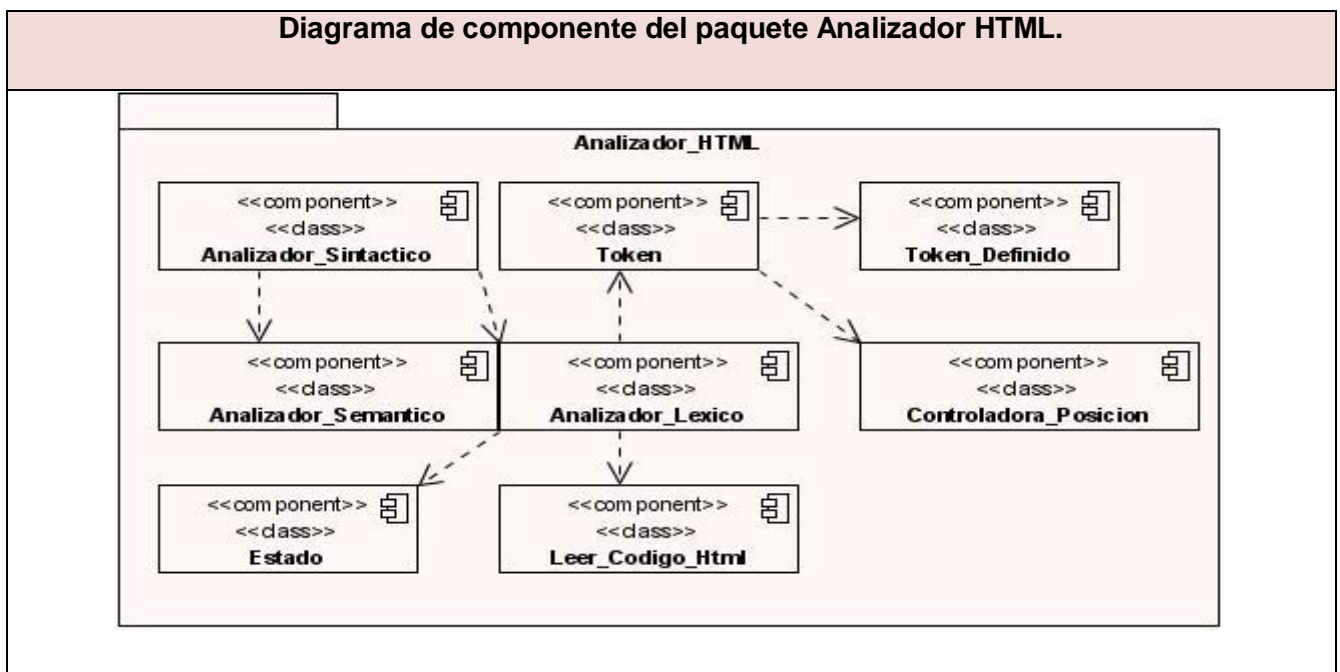


Tabla 44 Diagrama de componente del paquete Analizador HTML.



Diagrama de componente del paquete Descargar Página.

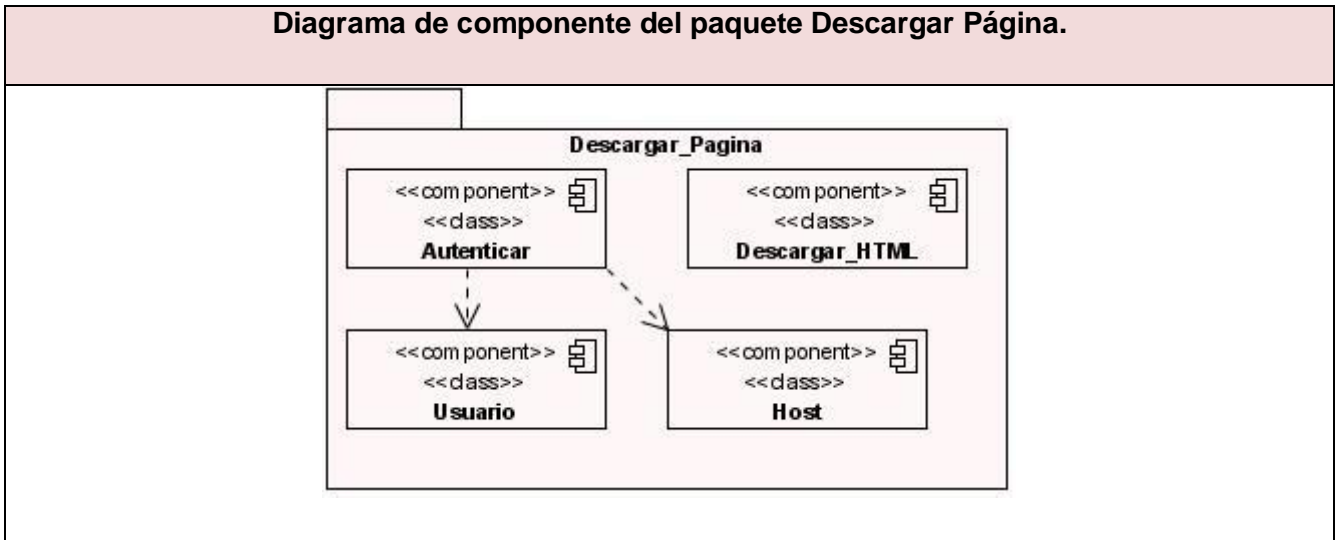


Tabla 45 Diagrama de componente del paquete Descargar Página.

Diagrama de componente del paquete Librerías.

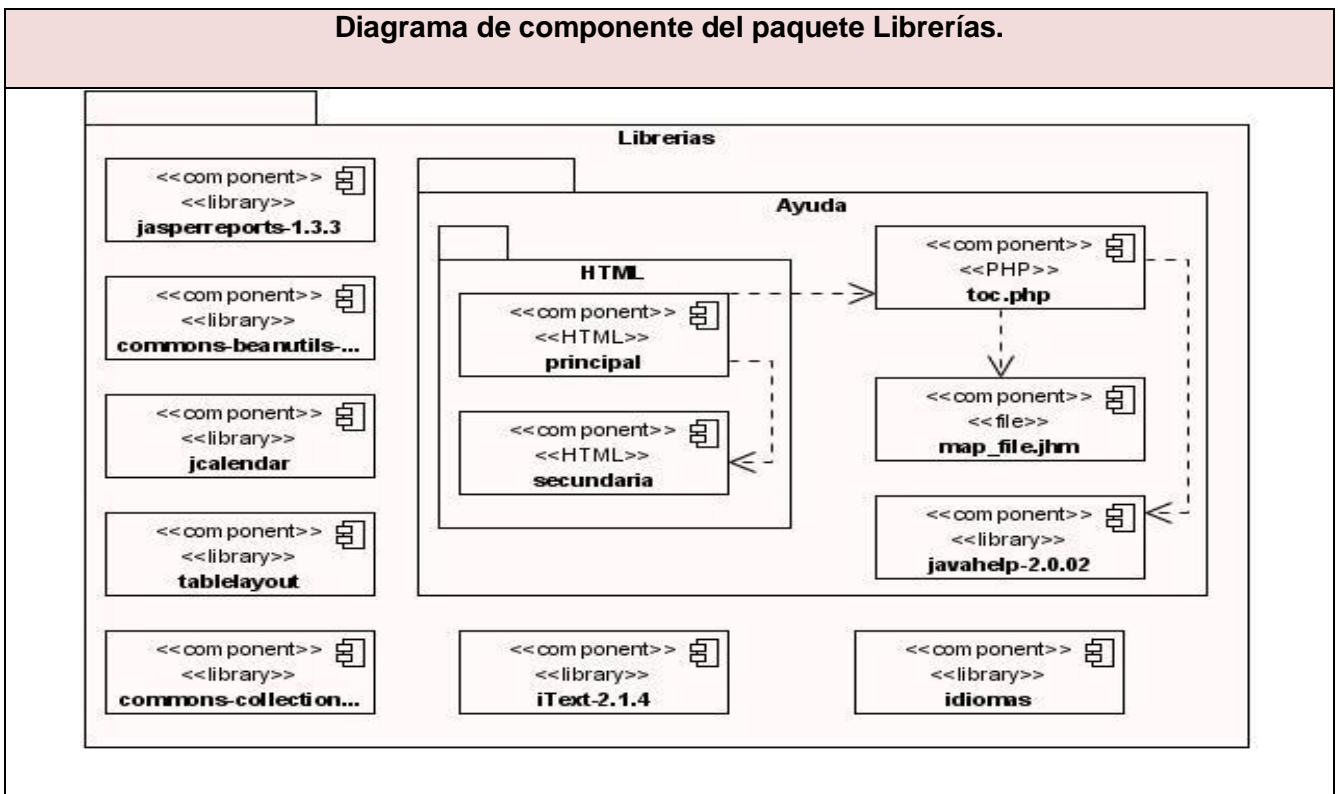


Tabla 46 Diagrama de componente del paquete Librerías.



Capítulo 3: Implementación y Prueba del Sistema

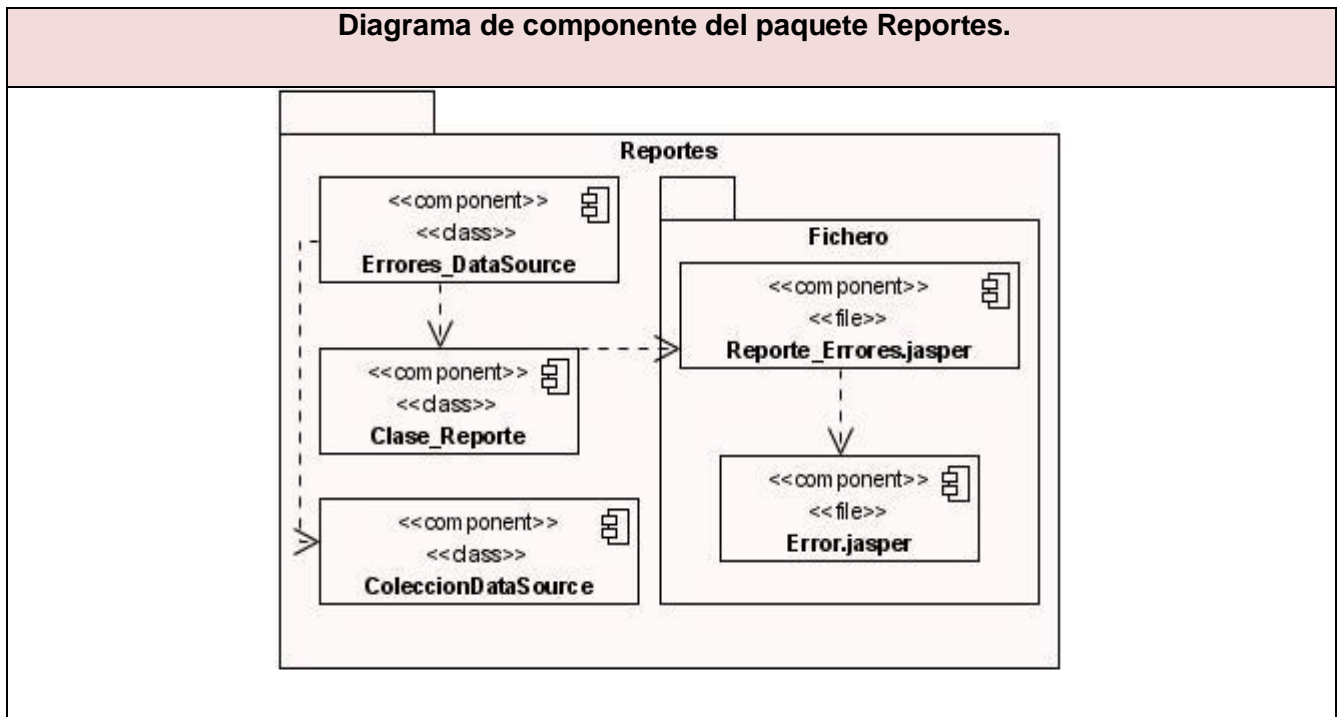


Tabla 47 Diagrama de componente del paquete Reportes.

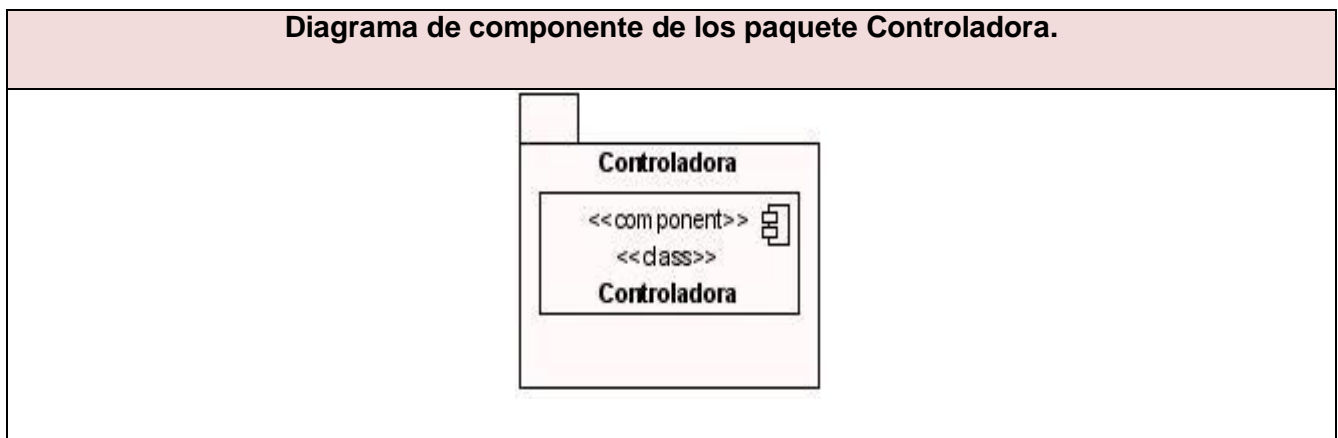


Tabla 48 Diagrama de componente de los paquete Controladora.



Capítulo 3: Implementación y Prueba del Sistema

Diagrama de componente General.

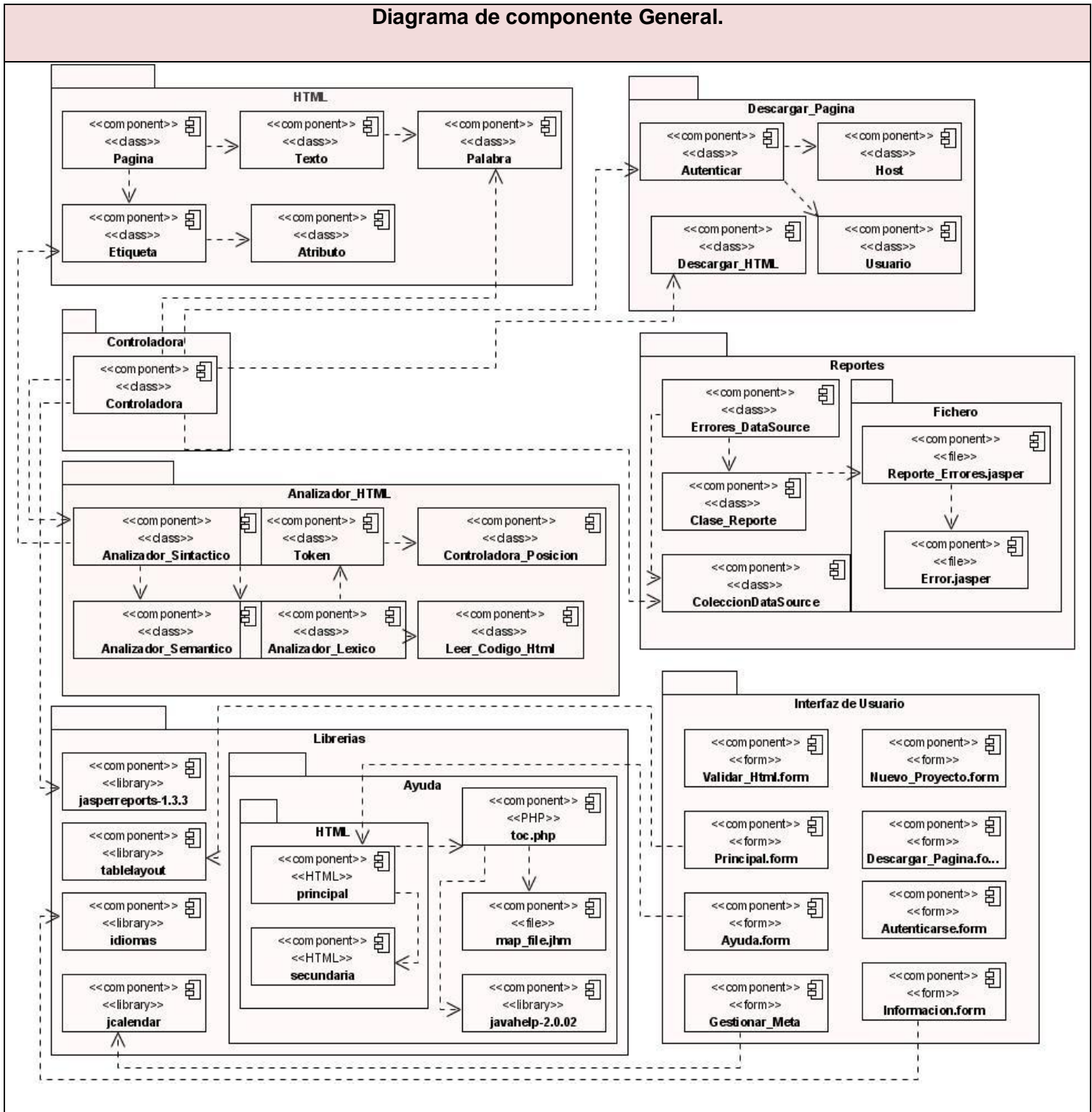


Tabla 49 Diagrama de componente General.



3.2.1 Estándares de codificación.

Cuando se trabaja en equipo es necesario hacer un código legible y entendible, no sólo para quien lo escribe, sino también para quien lo lee, de esta forma se contribuye a la comunicación de los desarrolladores. A continuación se muestra el estándar de codificación por el cual se regirá la implementación del sistema.

Las **clases** deben estar escritas con letra inicial mayúscula y en caso de estar compuesta por más de una palabra las mismas se separarán con un guion bajo (_).

Los **atributos** se especificarán de una forma clara y entendible de modo que el nombre que se le asigne no debe escribirse con abreviaturas, serán en letra minúscula y si está conformado por más de una palabra el mismo se separará por un guion bajo (_) como se muestra en el siguiente ejemplo: `privete boolean etiqueta_simple`. Se debe evitar variables de una sola letra.

Los **métodos** comenzarán con letra mayúscula y si está conformado por más de una palabra se separará un guion bajo (_) y las mismas deben comenzar con mayúscula. En caso de tener parámetros los mismos se denotarán con letra minúscula y si están compuestas por más de una palabra las mismas se separarán con un guión bajo (_). Las llaves de abrir y de cierre deben estar en líneas solas y alineada una con otra. Después de la llave de apertura del mismo debe existir tres espacios y todas las líneas de código que se encuentren dentro deben estar alineadas a un mismo nivel.

Instrucciones if, else, while, for, do while, try, catch: las llaves deben estar debajo y a 3 espacios del inicio de la fila, todas las líneas de código que se encuentren dentro deben estar alineadas a un mismo nivel.

Los **comentarios** deben ser entendibles para todo el personal que lo lea, especificados de esta forma: `(//comentario)` cuando es un sola línea, en caso de ser varias se pondrá de esta mara: `(/* comentario */)`.

Los **mensajes de salida** al usuario deben ser claros y precisos de tal forma que el usuario comprenda lo que se quiere transmitir, no deben tener abreviaturas.



3.2.2 Librerías utilizadas.

Frameworks JasperReport.

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

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

Los reportes generados pueden ser exportados a una multitud de formatos como PDF, XLS, RTF, HTML, XML, TXT, jasper viewer, CVS (valores separados por coma) y texto plano.

Los reportes son capaces de presentar los datos de manera textual o a través de gráficos: no sólo son capaces de mostrar los datos que le son pasados sino que pueden generar o calcular con esos datos otros datos de forma dinámica y mostrarlos.

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

JFreeChart.

JFreeChart es una biblioteca libre de código abierto desarrollada para la plataforma java, centrada en la generación de gráficas (GNU LGPL). Puede generar gráficos de pastel, de Barras (con vistas 3D), de líneas, de puntos, de series, de Gantt, de paretto, de intervalos, planillas métricas, planillas de candelabro, alto-bajo-abierto-cerrado entre otros y combinaciones de todos ellos. Exporta a mapas de imágenes PNG, JPEG, PDF, SVG y HTML.



JavaHelp

JavaHelp es una librería opcional de Java que nos permite poner ventanas de ayuda las aplicaciones de forma sencilla. Dichas ventanas serán como las que estamos acostumbrados en cualquier aplicación. La misma contiene una tabla de contenidos, un índice, permite buscar dentro de la ayuda.

3.2.3 Tratamiento de Errores.

El tratamiento de errores en un sistema influye directamente sobre la estabilidad y la calidad del mismo, por esa razón en que se ha manejado un conjunto de excepciones que persigue el correcto funcionamiento de la aplicación. Estas excepciones muestran diferentes mensajes en dependencia del error garantizando que el usuario conozca la situación actual de la operación.

Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias. Son una forma clara para controlar los errores sin confundir el código con muchas instrucciones de control del error. Cuando se verifica un error se pone en marcha una excepción que, si se recibe enseguida, permite gestionar un error.

Los mensajes de error llegarán a los usuarios en un lenguaje común, con el objetivo de facilitar su comprensión. En la siguiente tabla se describen las posibles acciones, que pueden producir errores.

Error	Causa	Respuesta Sistema
Autenticación	El usuario intenta descargar el código HTML de una página y existe problemas con la conexión al pasar por el proxy.	Notificar al usuario que no fue posible descargar la página especificada porque necesita autenticación o los datos de la conexión es incorrecto.
Conexión_URL	El usuario intenta descargar el código HTML de una página y existe problemas con la conexión.	Notificar al usuario que no fue posible descargar la página especificada por problemas de conexión, es decir: la url está mal formada, la página no existe, hay demora al leer el código o el servidor no da respuesta.

Tabla 50 Mensajes de error.



3.3 Prueba.

El aseguramiento de la calidad de los softwares se ha convertido en una necesidad prioritaria y en una tarea vital en el desarrollo de cualquier sistema informático, por la importancia que tiene garantizar el tiempo de desarrollo, el costo y la calidad del producto. Por esta razón es necesario establecer un conjunto de pruebas que garanticen que el software cumpla con los requisitos especificados y que no presente errores.

Para determinar la calidad de un producto de software se deben efectuar medidas y desarrollar actividades que permitan comprobar el grado de cumplimiento de las especificaciones iniciales del sistema, y es aquí donde las pruebas de software desempeñan un papel fundamental. Estas verifican el desarrollo que va alcanzando el producto durante todas sus etapas, identificando posibles fallos de implementación, calidad, o usabilidad de un programa.

Según Roger Pressman, en su libro *Un Enfoque Práctico*, las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. (6)

Una de las características típicas durante el desarrollo de un software basado en el ciclo de vida de RUP es la realización de controles periódicos. Estos controles pretenden evaluar la calidad de los productos generados para poder detectar posibles defectos cuanto antes. Todo sistema o aplicación, independientemente de estas revisiones, debe ser probado mediante su ejecución, controlado antes de ser entregado al cliente. Estas ejecuciones o ensayos de funcionamiento, posteriores a la terminación del código del software, se denominan habitualmente pruebas.

La IEEE (Standard Glossary of Software Engineering Terminology) plantea que la prueba es la actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan y almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente. Según RUP las pruebas son una disciplina en el proceso de ingeniería de software cuyo objetivo es integrar y poner a prueba el sistema.

Estos son algunos de los principales conceptos de prueba manejados actualmente en el mundo del software. Todos ellos apuntan a la significativa importancia de este proceso durante todo el desarrollo



Capítulo 3: Implementación y Prueba del Sistema

de un producto. Se considera que la prueba de software es una etapa imprescindible durante todo el proceso de desarrollo, pues una vez que se genera código fuente, el software debe ser probado para descubrir y corregir el máximo número de errores posibles antes de su entrega al cliente.

3.3.1 Pruebas aplicadas.

Las pruebas aplicadas al software fueron las Unitarias. La misma tiene fundamentalmente dos enfoques: Pruebas de Caja Blanca (o Pruebas Estructurales) y las Pruebas de Caja Negra (o Pruebas Funcionales).

Las **pruebas de Caja Blanca** suelen ser llamadas estructurales o de cobertura lógica. En ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones, bloques, etc. que han sido ejecutados por los casos de prueba. Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.(6)

Existen varios métodos de prueba de Caja Blanca, entre los que se encuentra: Prueba de camino básico que es la que se empleó para verificar que el software no presentara errores. Es una técnica que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecute por lo menos una vez cada sentencia del programa. (6)

Las **pruebas de Caja Negra** demostrarán que las funciones del software son operativas, que la entrada se acepta de forma adecuada, que se produce una salida correcta, y que se mantiene la integridad de la información externa. Estas pruebas se centran en los requisitos funcionales, se llevan a cabo sobre la interfaz del software y son completamente indiferentes al comportamiento interno y la estructura del programa. La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca.



Capítulo 3: Implementación y Prueba del Sistema

Las Pruebas de la Caja Negra intentan encontrar errores de: funciones incorrectas o inexistentes, errores relativos a las interfaces, errores en estructuras de datos o en accesos a bases de datos externas, errores debidos al rendimiento, errores de inicialización o terminación.

Dentro de los métodos de Caja Negra se encuentra la partición equivalente, el cual fue utilizado para la detectar los de errores de la aplicación. Esta técnica divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia.

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. (6).

A menudo se plantea que las pruebas a los software nunca terminan, simplemente se transfiere del desarrollador al cliente. Cada vez que el cliente usa el programa está llevando a cabo una prueba. Aplicando el diseño de casos de pruebas al software en cuestión se puede conseguir una prueba más completa y descubrir y corregir el mayor número de errores antes de que comiencen las “*pruebas del cliente*”.

3.3.2 Casos de prueba.

Los casos de pruebas tienen como objetivo comprobar que el producto que se encuentra en desarrollo, satisface realmente las peticiones del usuario final, tal y como se describe en la especificación de los requerimientos y los casos de uso.

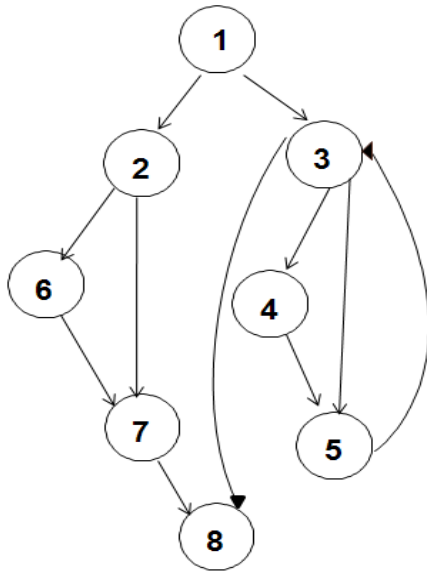
Los casos de pruebas no son más que un conjunto de entradas con datos de prueba, unas condiciones de ejecución, y unos resultados esperados cuyo propósito es identificar y comunicar las condiciones que se llevarán a cabo en la prueba.



3.3.2.1 Casos de prueba de Caja Blanca.

A continuación se presentan algunos de los casos de prueba de Caja Blanca realizados a la aplicación.

Método private Pagina Texto(Etiqueta etiqueta_body, Pagina pagina)



Complejidad ciclomática.

$$V(G) = 5.$$

Caminos Independientes.

1-2-6-7-8.

1-2-7-8.

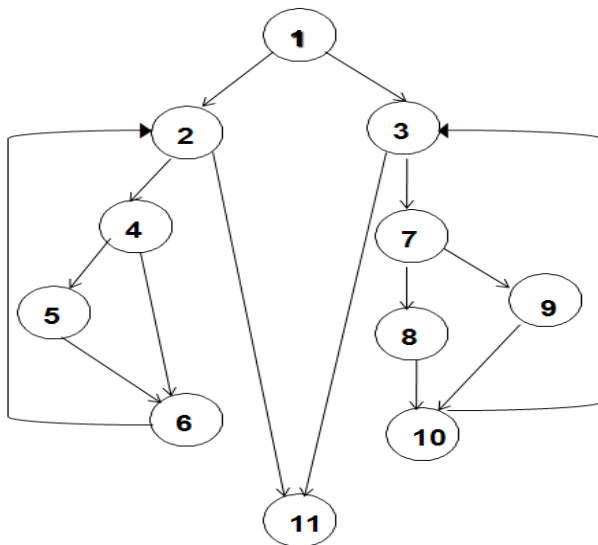
1-3-8.

1-3-4-5-3-8.

1-3-5-3-8.

Figura 1 Grafo de flujo del método Texto.

Método private DefaultTreeModel Crear_Arbol(Etiqueta etiqueta_analizar, DefaultMutableTreeNode padre, Pagina pagina, DefaultTreeModel modelo)



Complejidad ciclomática.

$$V(G) = 6.$$

Caminos Independientes.

1-2-11.

1-3-11.

1-2-4-5-6-2-11.

1-2-4-6-2-11.

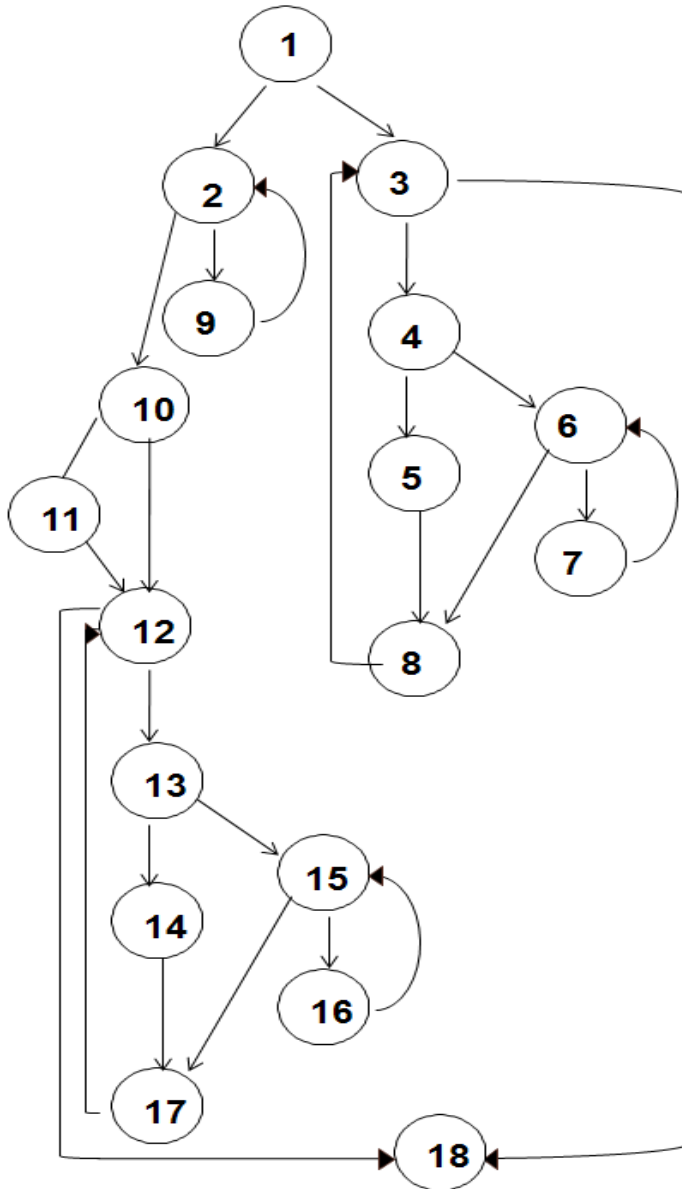
1-3-7-8-10-3-11.

1-3-7-9-10-3-11.

Figura 2 Grafo de flujo del método Crear Árbol.



Método public void Adicionar_Error(Etiqueta etiqueta_analizar)



Complejidad ciclomática.

$$V(G) = 9.$$

Caminos Independientes.

1-2-10-11-12-13-14-17-12-18.

1-2-9-10-11-12-13-14-17-12-18.

1-2-9-10-12-13-13-14-17-12-18.

1-2-9-10-12-13-13-15-16-15-17-12-18.

1-2-9-10-12-18.

1-3-18.

1-3-4-6-8-3-18.

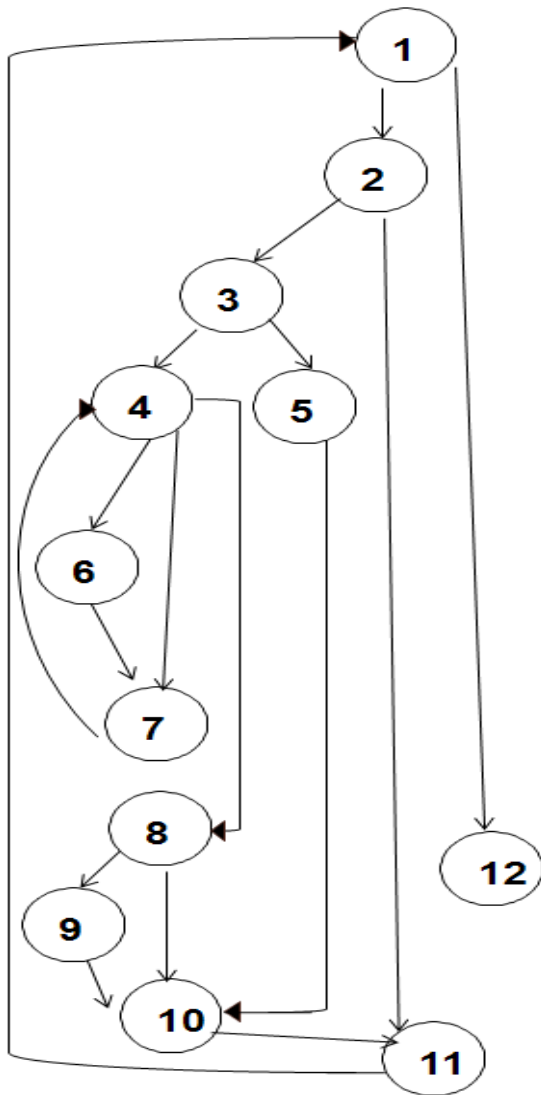
1-3-4-6-7-6-8-3-18.

1-3-4-5-8-3-18.

Figura 3 Grafo de flujo del método Adicionar_Error.



Método public void Modificar_Etiqueta(Etiqueta etiqueta_head_body)



Complejidad ciclomática.

$$V(G) = 7.$$

Caminos Independientes.

1-12.

1-2-11-1-12.

1-2-3-5-10-11-1-12.

1-2-3-4-8-9-10-11-1-12.

1-2-3-4-6-7-4-8-9-10-11-1-12.

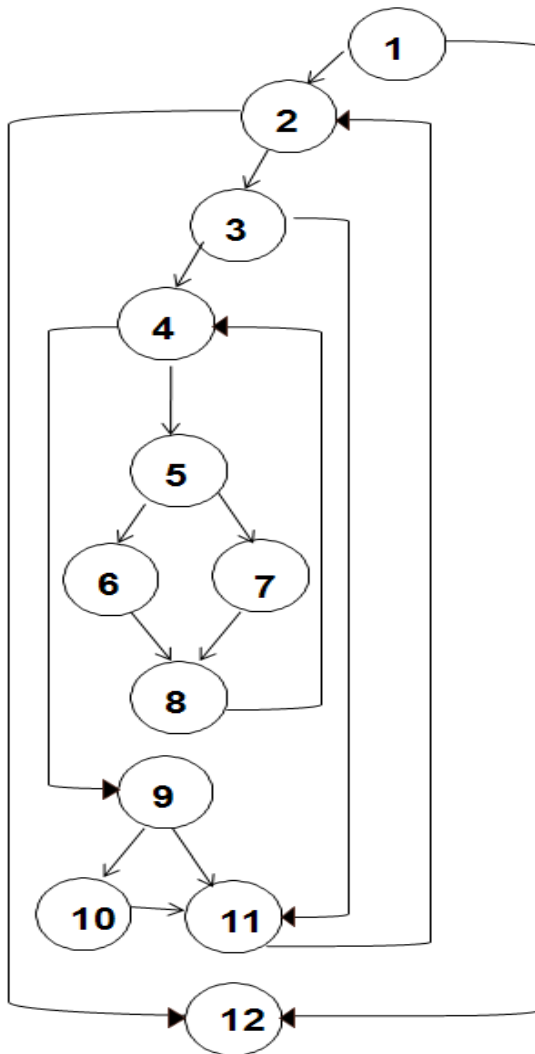
1-2-3-4-7-4-8-9-10-11-1-12.

1-2-3-4-7-4-8-10-11-1-12.

Figura 4 Grafo de flujo del método Modificar_Etiqueta.



Método `public String Correo_WebMaster(int posicion_pagina)`



Complejidad ciclomática.

$$V(G) = 7.$$

Caminos Independientes.

1-12.

1-2-12.

1-2-3-11-2-12.

1-2-3-4-9-10-11-2-12.

1-2-3-4-5-6-8-4-9-10-11-2-12.

1-2-3-4-5-7-8-4-9-10-11-2-12.

1-2-3-4-5-7-8-4-9-11-2-12.

Figura 5 Grafo de flujo del método `Correo_Webmaster`.

Luego de tener elaborados los Grafos de Flujos y los caminos a recorrer, se prepararon los casos de prueba que forzaron la ejecución de cada uno de los caminos. Se escogieron los datos de entradas de forma que todas las condiciones y caminos de cada método se ejecutara con el fin de reflejar los problemas que presentaba la aplicación. En el mismo se compararon los resultados con los esperados.



3.3.2.2 Casos de prueba de Caja Negra.

A continuación se muestran tanto las clases válidas como inválidas identificadas de algunos de los casos de usos escogidos para la realización de las pruebas de Caja Negra.

Clases de Equivalencia del CU Autenticar.

Descripción

Este caso de prueba consiste en que el usuario introduce el usuario, contraseña, proxy y puerto con el fin de autenticarse en el sistema.

Flujo Central.

El usuario solicita autenticarse, donde el sistema muestra una pantalla con los campos necesarios para introducir el usuario, contraseña, puerto y proxy.

El usuario introduce los datos especificados y el sistema en caso de que exista algún error notifica una respuesta en correspondencia con el error introducido de lo contrario se activan las opciones que el sistema brinda

Condición de entrada	Clases válidas	Clase inválidas
Usuario	Cadenas de caracteres que incluyen: <ul style="list-style-type: none">(1) A – Z	Cadenas de caracteres que incluyen: <ul style="list-style-type: none">(2) Espacios(3) extraños tales como: ; “ \$ % & / () = ¿ ? Ç * ^ “ _ : ; a @
Contraseña	Cadenas de caracteres que incluyen: <ul style="list-style-type: none">(4) A – Z(5) Números(6) Espacios(7) Comienzan con Espacios(8) Incluye todo tipo de caracteres extraños tales como: ; “ · \$ % & / () = ¿ ? Ç * ^ “ _ : ; a	



Capítulo 3: Implementación y Prueba del Sistema

Puerto	Cadena de solo 4 caracteres que incluyen: <ul style="list-style-type: none">(9) Números	Cadenas de caracteres que incluyen: <ul style="list-style-type: none">(10) Espacios(11) Letras(12) caracteres extraños tales como: ¡ “ \$ % & / () = ¿ ? ¢ * ^ “ _ : ; ^ @
Proxy	Cadena de solo 4 caracteres que incluyen: <ul style="list-style-type: none">(13) Números(14) Caracteres punto (.)	Cadenas de caracteres que incluyen: <ul style="list-style-type: none">(15) Espacios(16) Letras(17) caracteres extraños tales como: ¡ “ \$ % & / () = ¿ ? ¢ * ^ “ _ : ; ^ @

Tabla 51 Clases de Equivalencia del CU Autenticar.

Clases de Equivalencia del CU Gestionar_Meta.

Descripción

Este caso de prueba consiste en que el usuario a hecho una petición ya sea insertar, eliminar o modificar algunas de las etiquetas metas.

Flujo Central.

Si el usuario selecciona la opción eliminar el sistema muestra una lista con todas las etiquetas metas que contiene la página a la que desea realizarle la modificación. Para ello el usuario selecciona la etiqueta deseada y el sistema elimina y muestra los cambios realizados a la página.

Si el usuario selecciona la opción adicionar, el sistema muestra una ventana con los campos necesario para construir la etiqueta meta de la página a la que desea realizarle la modificación. Para ello el usuario introduce los datos pertinentes y el sistema adiciona y muestra los cambios realizados a la página.

Si el usuario selecciona la opción modificar el sistema muestra una lista con todas las etiquetas metas que contiene la página a la que desea realizarle la modificación. Para ello el usuario selecciona la



Capítulo 3: Implementación y Prueba del Sistema

etiqueta deseada e introduce los datos pertinentes, y el sistema modifica la etiqueta y muestra los cambios realizados a la página.

Condiciones de Ejecución

El usuario debes estar previamente autenticado en el sistema.

Condición de entrada	Clases válidas	Clase inválidas
nombre de la etiqueta	Cadenas de caracteres que incluyen: <ul style="list-style-type: none"> (1) A – Z 	Cadenas de caracteres que incluyen: <ul style="list-style-type: none"> (2) Espacios (3) extraños tales como: ; “ \$ % & / () = ¿ ? Ç * ^ ” _ : ; ^ @
atributo	Cadenas de caracteres que incluyen: <ul style="list-style-type: none"> (4) A – Z (5) carácter guión (-) 	Cadenas de caracteres que incluyen: <ul style="list-style-type: none"> (6) Espacios (7) extraños tales como: ; “ \$ % & / () = ¿ ? Ç * ^ ” _ : ; ^ @
valor del atributo	(8) Cadenas de caracteres.	

Tabla 52 Clases de Equivalencia del CU Gestionar_Meta.

Clases de Equivalencia del CU Conocer_características_página.

Descripción

Este caso de prueba consiste conocer las características que posee una página en particular.

Flujo Central.

El usuario solicita ver las características que posee una página seleccionando dicha opción como respuesta el sistema muestra una lista con las particularidades que presenta la misma, tales como: correo webmaster, cantidad de etiquetas, profundidad e idioma que posee.



Capítulo 3: Implementación y Prueba del Sistema

Condición de entrada	Clases válidas	Clase inválidas
Página	(1) Exista página descargada.	(2) No exista la página

Tabla 53 Clases de Equivalencia del CU Conocer_características_página.

Clases de Equivalencia del CU Descargar_Página.

Descripción

Este caso de prueba consiste descargar una determinada página especificada por el usuario.

Flujo Central.

El usuario solicita descargar una página, como respuesta el sistema muestra una ventana dándole la posibilidad al mismo de especificar la url de una página que se encuentre publicada en algún servidor web o seleccionar una página que se encuentre dentro de algún sector del disco duro de la máquina en al que se está trabajando. El oprime el botón aceptar y el sistema muestra el código de la página.

Condiciones de Ejecución

El usuario debes estar previamente autenticado en el sistema en caso que la página no sea local.

Condición de entrada	Clases válidas	Clase inválidas
url	Cadenas de caracteres que incluyen: <ul style="list-style-type: none">• (1) A – Z• (2) Números	Cadenas de caracteres que incluyen: <ul style="list-style-type: none">• (3) * :”

La ejecución de los Casos de Prueba de Caja Negra reveló principalmente algunos errores en la implementación de la interfaz visual. La mayoría debido a descuidos o a errores en la lógica de programación pero en general los resultados fueron satisfactorios. Finalmente se corrigieron dichos errores y se comprobó nuevamente el resultado.

3.3.3 Análisis de los defectos encontrados.

Las pruebas realizadas al software permitieron verificar que el mismo cumplía con todo los requisitos funcionales especificados e identificar cuáles fueron las principales deficiencias detectadas. Los Casos de Pruebas que se definieron anteriormente fueron ejecutados satisfactoriamente por lo que se puede

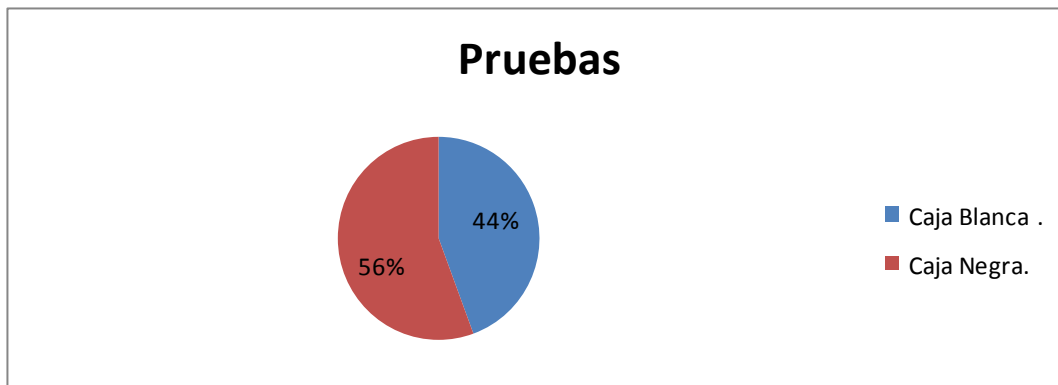


Capítulo 3: Implementación y Prueba del Sistema

plantear de forma general que el sistema sólo contenía algunas fallas de validación, errores ortográficos y de tratamiento de errores. Los distintos errores encontrados que a simple vista no hubieran sido fáciles identificar se produjeron por las siguientes causas: descuido, falta de comprensión del problema a solucionar y falta de revisión durante la codificación. Los errores fueron clasificados según la prioridad (alta, media, baja), los cuales fueron corregidos en la siguiente tabla.

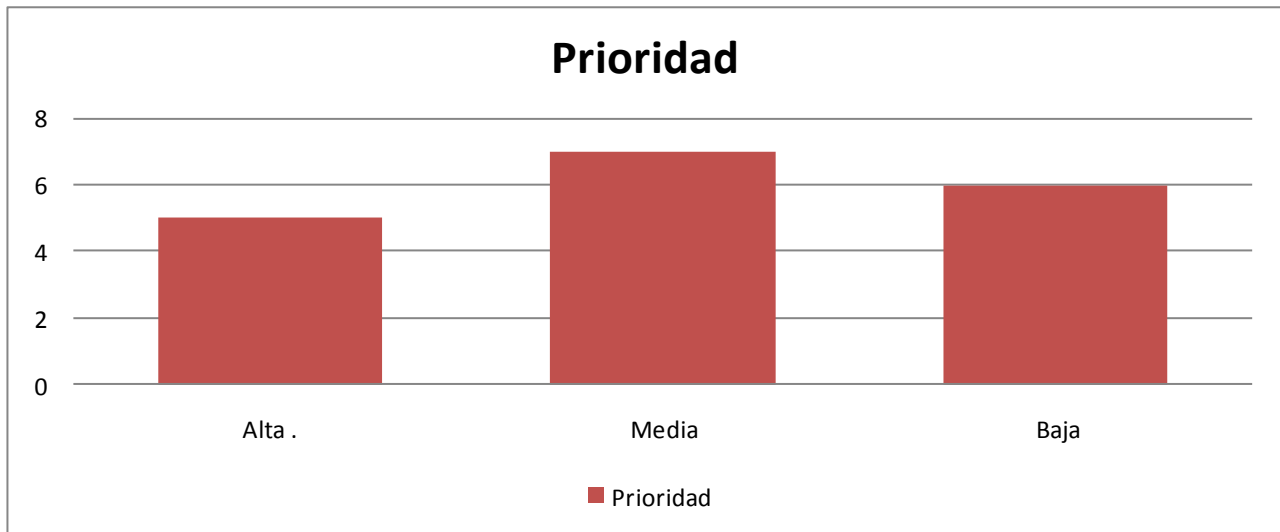
Prueba	Errores Encontrados
Caja Blanca	8
Caja Negra	10

Tabla 54 Análisis de los defectos encontrados.



Prioridad	Cantidad
Alta	5
Media.	7
Baja.	6

Tabla 55 Análisis de los defectos encontrados.



3. 4 Conclusiones.

En el capítulo se mostraron los resultados obtenidos durante el flujo de trabajo de implementación quedando los subsistemas y clases codificadas definidos en el capítulo anterior. Se realizó el diagrama de componentes con el objetivo de brindar una idea de cómo se implementó el software en término de componentes dando una visión de cómo quedó construida y distribuida la aplicación. Mediante los casos de prueba se validó el software, verificando el correcto funcionamiento del sistema, comprobándose que no existían errores en las funciones operativas. Posibilitó la obtención de las principales deficiencias que presentaba el software.



Conclusiones

Al concluir la investigación se planteó que se cumplió con el objetivo general propuesto. El análisis de las tecnologías actuales y el estudio de los principales elementos del posicionamiento web permitieron consolidar una base sólida para el desarrollo del sistema. Basado en la metodología de desarrollo RUP, a lo largo de todo el trabajo se transitó por los flujos de trabajo de análisis y diseño, implementación y pruebas, documentándose en cada flujo los principales artefactos necesarios para el desarrollo de la solución propuesta; logrando así obtener una aplicación funcional que permite elevar la calidad de las páginas Web mediante la mejora de su código y su contenido.

La aplicación en conjunto con otros factores que intervienen en el posicionamiento web influirá en que los sitios web cubanos aparezcan dentro de los primeros resultados arrojados por los buscadores para cualquier criterio de búsqueda y de esta forma hacer llegar la información exacta que brindan los medios de comunicación a los internautas.



Recomendaciones.

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

- Profundizar en el estudio de las estrategias y técnicas de posicionamiento web, con vista a detectar posibles cambios o modificaciones en el algoritmo de posicionamiento de los diferentes buscadores.

- Realizar un estudio a los weblog del Grupo Operativo perteneciente al Grupo de Investigación y Desarrollo sobre Internet, con la utilización de la herramienta, para mejorar el posicionamiento y aumentar su visibilidad.

- Continuar con el proceso de desarrollo del sistema con el fin de incorporarle nuevas funcionalidades para facilitarle el trabajo del Webmaster.
 - ✓ Generar el fichero robots.txt que le restringe o permite a los spider de los motores de búsqueda, las páginas que deben o no ser indexadas.
 - ✓ Generar el mapa de navegación del sitio Web con el objetivo de indicarle a los buscadores donde se encuentran las páginas web y de esta forma facilitarle la indexación.
 - ✓ Verificar si el sitio tiene información duplicada, la cual es penalizada por los buscadores.
 - ✓ Validar los estilos de la página para determinar si el texto tiene el mismo color del fondo de la página para evitar posibles fraudes.



Referencias bibliográficas.

1. El libro del posicionamiento en buscadores. [aut. libro] David Ugarte. Madrid : s.n., 2002.
2. [En línea] [Citado el: 10 de dic de 2009.] <http://www.bloginformatico.com/definiendo-algunos-terminos-seo.php>.
3. Clements., Mary Shaw y Paul. A field guide to Boxology: Preliminary classification of architectural styles for software systems. Carnegie Mellon University. : s.n., Abril de 1996.
4. Shaw, David Garlan y Mary. An introduction to software architecture. s.l. : CMU Software Engineering Institute Technical Report, 1994. CMU/SEI-94-TR-21, ESC-TR-94-21.
5. IVAR JACOBSON, G. B., JAMES RUMBAUGH. . Addison Wesley Publishing Company (November 2000), 1999. El Proceso Unificado de Desarrollo de Software. s.l.
6. Pressman, Roger S. Ingenieria del software. Un enfoque práctico. Parte I. La Habana 2005. : s.n.



Bibliografía.

1. El libro del posicionamiento en buscadores. [aut. libro] David Ugarte. Madrid : s.n., 2002.
2. http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html.
3. Citado el: 11 de dic de 2008.] <http://www.bloginformatico.com/definiendo-algunos-terminos-seo.php>.
4. Parte 1, Página 12,
\\datalab\Documentacion\Visual_Paradigm\documentacion\Visual_Paradigm_4UML_User_Guide\vp-uml_users_guide_part_1. Visual Paradigm for UML.
5. [Citado el: 13 de dic de 2008.] <http://www.visual-paradigm.com/product/vpuml/>.
6. [Citado el: 13 de dic de 2008.]
<http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm>.
7. Becerril, Francisco. "Java a su alcance", 1998, Parte 1, Páginas 11-20 . México : McGRAW-H.
8. Zukowski, John. ;"Java 2, J2SE 1.4", Tomo I Parte 1, Páginas 33-39. Madrid : Edición española, 2003.
9. "UML y Patrones, Tomo I Capítulos 1, Páginas 3-16. México : s.n., 1999.
10. Introducción a la Ingeniería de Software ,2007-2008. [En línea] [Citado el: 14 de dic de 2008.]
<http://teleformacion.uci.cu/Larman, Craig>.
11. Codina., Lluís. Posicionamiento Web: Conceptos y Ciclo de Vida . [En línea] [Citado el: 15 de dic de 2008.] ,<http://www.hipertext.net>, Publicado 29/09/2008, <http://www.articuloz.com/seo-articulos/que-es-una-palabra-clave-328792.html>.
12. Buitrago, Camilo. Que es una palabra clave. 09-02-2008.
13. <http://es.wikipedia.org/>.
14. http://es.wikipedia.org/wiki/Sitio_web.



15. [Citado el: 13 de dic de 2008.]
<http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm>.
16. [Citado el: 13 de dic de 2008.] http://www.pergaminovirtual.com.ar/definicion/Pagina_web.html.
17. [Citado el: 14 de dic de 2008.] <http://www.posicionar-mi-web.com.ar/definiciones.html>.
18. [Citado el: 14 de dic de 2008.] <http://active-web-traffic..>
19. [Citado el: 15 de dic de 2008.]
20. [Citado el: 15 de dic de 2008.] <http://active-link-exchange.uptodown.com/>.
21. [Citado el: 16 de dic de 2008.] http://www.zonagratis.com/a-utilidades/software_google/google_monitor.htm.
22. [Citado el: 16 de dic de 2008.] <http://www.herramientas-plr.com/activelinkexchange.html>.
23. [Citado el: 16 de dic de 2008.] <http://active-link-exchange.uptodown.com/>.
24. [En línea] [Citado el: 10 de 1 de 2009.] <http://gratis.portalprogramas.com/Active-Link-Exchange.html>.
25. Citado el: 10 de ene de 2009.] http://www.geomundos.com/descargas/active-link-exchange-133_p683.html.
26. Citado el: 11 de ene de 2009.] <http://www.seoadministrator.com>.
27. [Citado el: 11 de dic de 2008.] <http://www.netbeans.org/>.
28. [Citado el: 11 de dic de 2008.] http://es.wikipedia.org/wiki/NetBeans_IDE.
29. [Citado el: 12 de ene de 2009.]
http://www.freedownloadscenter.com/es/Autoria_Web/Herramientas_para_Promocion_de_Sitios_Web/SEO_Studio_Web_Promotion_Software.html.
30. [Citado el: 13 de ene de 2009.] <http://www.softbull.com/site-position.html>.
31. [Citado el: 16 de ene de 2009.] <http://www.softbull.com/site-position.html>.
32. [Citado el: 17 de ene de 2009.] http://www.descargaspremium.com/topdog_pro_695.html.



33. [Citado el: 20 de ene de 2009.] <http://www.topdogsoftware.biz/downloads/TDProSetup.exe>.
34. [Citado el: 20 de ene de 2009.] <http://rankmeter.programas-gratis.net/>.
35. [Citado el: 20 de enr de 2009.] <http://www.tucows.com> .
36. [Citado el: 20 de enero de 2009.]
http://www.freedownloadmanager.org/es/downloads/Estudio_de_SEO_9742_p/free.htm.
37. [Citado el: 20 de ene de 2009.] <http://es.brothersoft.com/SEO-Administrator-128899.html>.
38. [Citado el: 22 de ene de 2009.] <http://www.indianwebs.com/diccionario-posicionamiento-seo/diccionario-posicionamiento-definicion.asp?Id=250> .
39. [Citado el: 22 de ene de 2009.] http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/concepto.htm.
40. [Citado el: 5 de Feb de 2009.] <http://maestroweb.ticoblogger.com/2006/10/definicin-de-blog.html>.
41. Clements., Mary Shaw y Paul. A field guide to Boxology: Preliminary classification of architectural styles for software systems. Carnegie Mellon University. : s.n., Abril de 1996.
42. Shaw, David Garlan y Mary. An introduction to software architecture. s.l. : CMU Software Engineering Institute Technical Report, 1994. CMU/SEI-94-TR-21, ESC-TR-94-21.
43. Adictos al trabajo. [En línea] [Citado el: 2009 de feb de 10.] <http://www.adictosaltrabajo.com>.
44. 2009 de feb de 12. <http://jasperreports.sourceforge.net/>.
45. 2009 de feb de 12. <http://www.javahispano.org/>.
46. 2009 de feb de 12. <http://www.mygnet.net>.
47. 2009 de feb de 13. <http://ireport.sourceforge.net/>.
48. 2009 de feb de 13. <http://diesel-java.blogspot.com>.



Anexos.

Anexo 1

Para avalar nuestra investigación se realizó una encuesta con el objetivo saber el conocimiento que poseían las personas de la universidad acerca del posicionamiento web.

Esta es una encuesta muy sencilla para una investigación que se está llevando a cabo en el Grupo de Investigación y Desarrollo sobre Internet (GIDI) para saber los conocimientos que usted posee sobre Posicionamiento Web. Agradeceríamos mucho su colaboración.

Pregunta:

¿Qué sabes de Posicionamiento Web? Poco___ Medio___ Nada___

En caso afirmativo, explique brevemente que considera usted que es. El posicionamiento Web.

Muchas gracias.

Resultados de la encuesta:

Se encuestaron 410 estudiantes, de los cuales 100 son estudiantes de cuarto año, 150 estudiantes de quinto año, 80 de estudiantes de tercer año, 40 estudiantes de segundo año y 20 estudiantes de primer año; de las 10 facultades y de ambos sexos. Se encuestaron además 20 profesores de la Universidad de la Ciencia Informáticas.



Respuestas

De las personas encuestadas respondieron: Poco 150, Nada 200, Medio 60.





Glosario de términos.

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

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

Googlebot: Es el robot que utiliza Google para rastrear los sitios web en Internet

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

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

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

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

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

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

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

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



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

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

Weblog: Conocido como blog, es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.