

Universidad de las Ciencias Informáticas



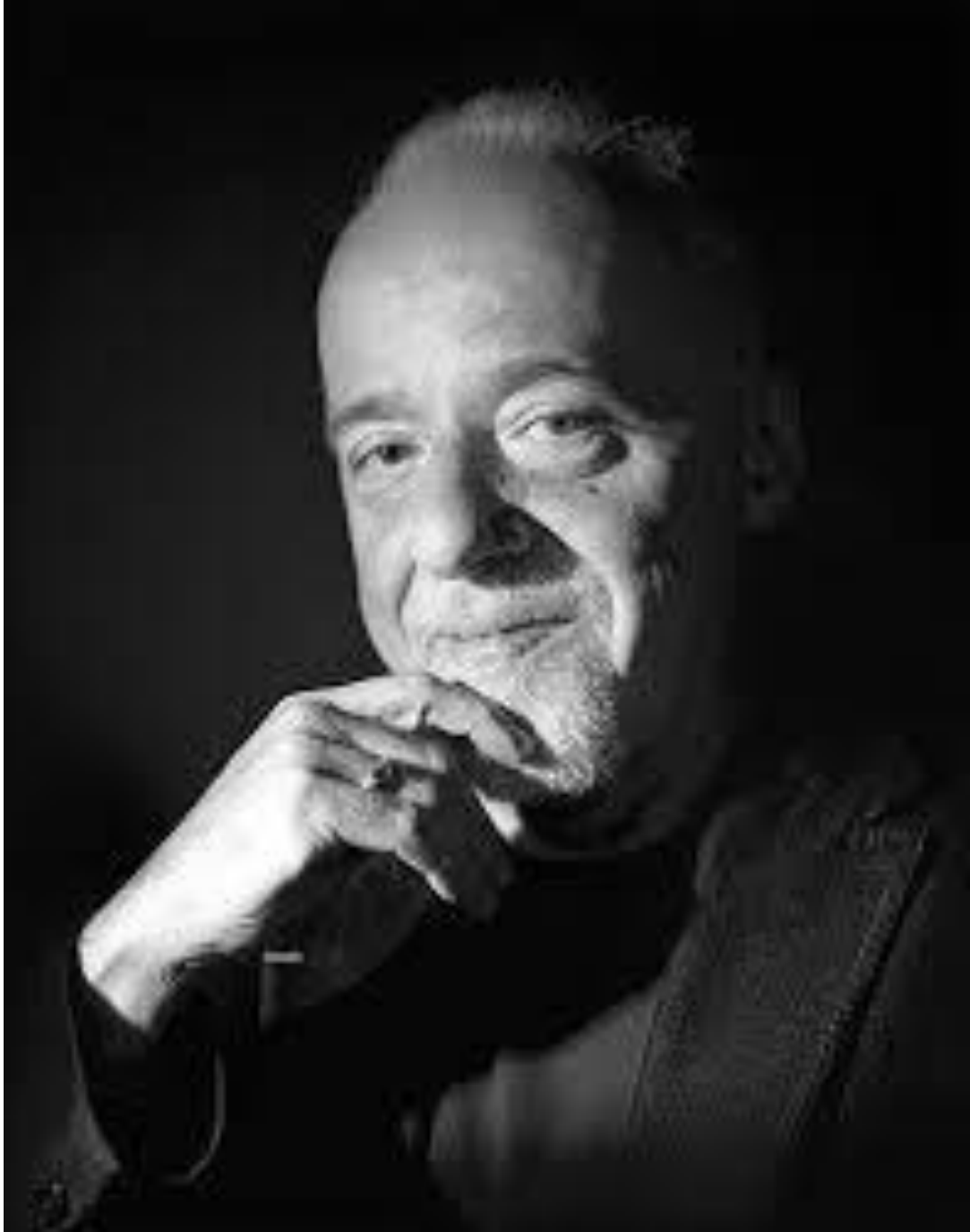
**Herramienta para el análisis webmétrico en la Universidad de las
Ciencias Informáticas**

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

Autores: Gloria Delia Guiamet Pereira
Daniel Ramírez Grave de Peralta

Tutores: Ing. Madisleidy Casanova Hernández
Ing. Yurisleidy Hernández Moya
Ing. Julio Amado López Palma

La Habana, 10 de junio del 2014



La posibilidad de realizar un sueño, es lo que hace que la vida sea interesante.

Paulo Coelho

DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo. Para que así conste se firma la presente a los ____ días del mes de _____ del año _____.

Firma del Autor

Gloria Delia Guiamet Pereira

Firma del Autor

Daniel Ramírez Grave de Peralta

Firma del Tutor

Madisleidy Casanova Hernández

Firma del Tutor

Yurisleidy Hernández Moya

Firma del Tutor

Julio Amado López Palma

Gloria:

Primeramente comenzaré dándole muchísimas gracias a mis padres, que con su sacrificio y empeño, han sabido llevar de la mano todos estos años de mi vida, los cuales no han sido nada fácil, pero ahí han estado. Quiero darle las gracias a mi esposo, por apoyarme en las decisiones que he tomado y mantenerme impulsada para lograr cualquier objetivo que me proponga.

Agradezco a todos mis familiares y amistades que a pesar de lo difícil que ha sido este último año para mí por alejarme de mis hijos Anthony y Kelly Brooks Guiamet, hicieron que fuera un poco flexible el sacrificio, cuidándolos y ayudando a mi mamá en cualquier situación que se presentara con ellos, estando yo aquí.

Le agradezco de todo corazón a mi compañero de tesis Daniel, por este logro exitoso que hemos tenido, porque a decir verdad, hemos formado un buen equipo de trabajo para cumplir con nuestro principal objetivo aquí en la universidad.

También quiero agradecer a todas las personas que de una forma u otra han aportado su grano de arena en mi formación como profesional, y yo personalmente, les garantizo que su esfuerzo no ha sido en vano, entre ellos, están los profesores: William Santana, Graciela González, Paula Yuleni Hechavarría, Dayron Pérez Roldan, Gendry Alfonso, Yariel Ramos, José Ramón Fernández, Alexander Hernández Chapman, Orlando Cruzata, Yankiel Aparicio, Luis Domínguez, Gilberto Lissabet, Hubert Viltres y las secretarías de la antigua facultad 10 Milagros Elvira Ramírez (Vicky) y Cecilia Borges Zayas.

Gracias a todas las amistades que me han acompañado en el largo camino de la carrera, las que ya no están por algún motivo u otro, las que se han mantenido y las que he hecho nuevas en este curso de retorno a la escuela.

En fin, mi agradecimiento va dirigido a todas las personas que han sido parte de esta etapa estudiantil en mi vida y por ende, se han ganado mi aprecio por su generosidad, solidaridad y compañía.

Daniel:

A Dios y a la Revolución por construir esta hermosa obra que es la UCI y permitirme estudiar aquí.

A mis tutores, que han estado a mi lado, dándome apoyo; a mi oponente Ruben Reynaldo Bonachea, que ha revisado el documento innumerables veces en función de hacerlo lo más completo posible.

A mis padres, abuelos, mis tíos, primos y demás, que me han apoyado, enseñado y demostrado lo que es una familia que confía en uno, que cree en sus éxitos y siempre se puede contar con ellos para todo.

A la luz de vida, mi esposa Ruhaine, que ha sido un gran apoyo en todos los sentidos a lo largo de mis años de estudio.

A todos mis compañeros del 125 por aguantarme y siempre contar conmigo.

A todos los profesores del proyecto y otros que me ayudaron con sus ideas y sus consejos.

A mis compañeros de proyecto, que me apoyaron en todo momento.

A todas las personas que se preocuparon por el desarrollo de esta investigación.

Resumen

El presente trabajo de diploma lleva por título: “Herramienta para el análisis webmétrico en la Universidad de las Ciencias Informáticas (UCI)”. Presenta los resultados de una investigación realizada con el propósito de facilitar la toma de decisiones de los *webmasters* sobre la gestión de los sitios web en la UCI, con el fin de mejorar su estructura, composición y calidad. Para lograr el objetivo propuesto, se realizó un estudio de las características de la Webmetría como disciplina métrica de la información. En el mismo, se analizan los principales indicadores webmétricos que deben tenerse en cuenta para la optimización de los sitios web. A partir de la selección de estos indicadores, se desarrolla una herramienta para análisis webmétrico, la cual brinda la posibilidad de generar reportes estadísticos enfocados en las métricas que proporcionarán información del comportamiento de los sitios web. Para el desarrollo de la herramienta mencionada anteriormente se emplea el gestor de base de datos SQLite3. En el diseño, implementación y documentación se utiliza como metodología de desarrollo OpenUp. Además, fueron utilizadas las herramientas Visual Paradigm 8.0, IntelliJIDEA 12 y Grails 2.2.1, como lenguajes se usaron Groovy 2.0.7 y el Lenguaje de Modelado Unificado (UML). Como parte de las pruebas realizadas a la herramienta desarrollada, se analizaron los sitios web de la UCI, de los cuales se obtuvieron estadísticas relacionadas con la cantidad de sitios analizados, cantidad de objetos multimedia, de páginas estáticas y dinámicas, así como la cantidad de documentos por idiomas y el porcentaje que representan.

Palabras claves: indicadores webmétricos, reportes estadísticos, sitios web, webmetría.

Índice

Introducción	1
1 Capítulo I: Aspectos teóricos generales sobre webmetría.....	5
1.1 Introducción.....	5
1.2 Conceptos Fundamentales	5
1.2.1 Webmetría.....	5
1.2.2 Indicadores webmétricos	6
1.3 Estudio de sistemas homólogos en el ámbito nacional e internacional.	8
1.4 Webcrawler.....	10
1.5 Tecnologías, lenguajes y metodología de desarrollo.....	11
1.5.1 Metodología de desarrollo	11
1.5.2 Lenguaje de modelado	12
1.5.3 Herramienta Case Visual Paradigm.....	13
1.5.4 Entorno Integrado de Desarrollo	13
1.5.5 Lenguaje de Desarrollo.....	14
1.5.6 WIRE	15
1.5.7 Marco de trabajo Grails.....	15
1.5.8 Sistema de gestión de base de datos <i>SQLite</i>	15
2 Capítulo II: Propuesta de la Herramienta para Análisis Webmétrico	18
2.1 Introducción.....	18
2.2 Características de la herramienta	18
2.3 Modelado del dominio.....	18
2.4 Especificación de requisitos de software.....	19
2.5 Diagrama de casos de uso del sistema	21
2.6 Especificación de casos de uso	22
2.7 Descripción de los patrones arquitectónicos y de diseño	25
2.7.1 Patrones arquitectónicos	25

2.7.2	Patrones de diseño.....	26
2.8	Diagrama de clases del diseño.....	27
2.9	Diagrama de interacción del diseño.....	28
2.10	Modelo de datos.....	29
3	Capítulo III. Vista previa de la Herramienta para Análisis Webmétrico.....	31
3.1	Introducción.....	31
3.2	Modelo de despliegue.....	31
3.3	Diagrama de componentes.....	32
3.4	Estándar de codificación.....	33
3.5	Estudio realizado a los sitios web en la UCI con la herramienta webmétrica.....	34
3.6	Realización de pruebas del sistema.....	37
3.6.1	Niveles de Prueba.....	38
3.6.2	Métodos de prueba.....	38
3.6.3	Pruebas de funcionalidades.....	39
4	Conclusiones generales.....	47
5	Recomendaciones.....	48
6	Referencias Bibliográficas.....	49
7	Bibliografía Consultada.....	50
8	Glosario de Términos.....	51
9	Anexos.....	52
9.1	Anexo 1. Descripción de los casos de uso del sistema.....	52
9.1.1	CU Reiniciar datos.....	52
9.1.2	CU Cambiar Contraseña.....	53
9.1.3	CU Realizar recorrido.....	55
9.1.4	CU Generar reporte de extensiones.....	57
9.1.5	CU Generar reporte de recorrido.....	58
9.1.6	CU Generar reporte de idiomas.....	60
9.1.7	CU Generar reporte de sitios.....	62

9.1.8	CU Generar reporte de enlaces	63
9.1.9	CU Generar reporte de documentos	65
9.1.10	CU Mostrar fichero de configuración.....	67
9.1.11	CU Mostrar fichero de urls iniciales.....	68
9.2	Anexo 2. Diagramas de clases del diseño.	71
9.3	Anexo 3. Diagramas de interacción del diseño.	77
9.4	Anexo 4. Diseño de casos de prueba a funcionalidades del sistema.	82
9.4.1	Caso de prueba del CU Reiniciar Datos	82
9.4.2	Caso de prueba del CU Cargar Urls.	83
9.4.3	Caso de prueba del CU Realizar Recorrido	83
9.4.4	Caso de prueba del CU Mostrar fichero de configuración	84
9.4.5	Caso de prueba del CU Mostrar fichero de urls iniciales.....	85
9.4.6	Caso de prueba del CU Generar reporte de extensiones.....	88
9.4.7	Caso de prueba del CU Generar reporte de sitios	88
9.4.8	Caso de prueba del CU Generar reporte de documentos	89
9.4.9	Caso de prueba del CU Generar reporte de enlaces	89
9.4.10	Caso de prueba del CU Generar reporte de idiomas	90
9.4.11	Caso de prueba del CU Generar reporte de recorrido	91
9.5	Anexo 5. Entrevista realizada en la Dirección de Informatización de la UCI.....	92

Índice de Figuras

Figura 1. Diagrama del modelado del dominio.	19
Figura 2. Diagrama de casos de uso del sistema.....	21
Figura 3. Diagrama de clases del diseño con estereotipos web: CU Realizar Recorrido.....	28
Figura 4. Diagrama de clases del diseño con estereotipos web: CU Cargar URL.	28
Figura 6. Diagrama de secuencia: CU Realizar Recorrido.	29
Figura 5. Diagrama de secuencia: CU Cargar URLs.....	29
Figura 7. Modelo de datos	30
Figura 8. Diagrama de despliegue de la herramienta webmétrica.....	31
Figura 9. Diagrama de componentes	32
Figura 10. Fichero de configuración.....	35
Figura 11. Estadísticas de los sitios.....	35
Figura 12. Estadísticas de extensiones multimedia.....	36
Figura 13. Estadísticas de los enlaces.....	36
Figura 14. Estadísticas de los documentos.....	37
Figura 15. Estadísticas de los idiomas.....	37

Índice de Tablas

Tabla 1. Indicadores webmétricos.....	7
Tabla 2. Requisitos funcionales	20
Tabla 3. Descripción del caso de uso autenticar usuario.....	23
Tabla 4. Descripción del caso de uso cargar urls.....	25
Tabla 5. Descripción de las variables para el caso de prueba del CU Cambiar Contraseña.	40
Tabla 6. Caso de prueba del CU Cambiar Contraseña	43
Tabla 7. Descripción de las variables para el CU Autenticar Usuario.....	44
Tabla 8. Caso de prueba del CU Autenticar Usuario.....	45

Introducción

En informática, la *World Wide Web* (WWW) o red informática mundial comúnmente conocida como la *Web*, es una interfaz simple y consistente a través de la cual se puede acceder a los disímiles recursos de Internet. De este modo, cuando un usuario solicita un determinado recurso almacenado en algún servidor web (contenedores de programas, archivos comprimidos, documentos de texto, archivos multimedia tales como videos, imágenes, sonidos y animaciones), es mostrado en forma de documento hipertexto.

Existen otras definiciones del término *Web*, expresadas por científicos reconocidos. Dichas definiciones aportan información veraz del tema. Por ejemplo, Tim Berners-Lee: describe la *Web* como "*una red que trajo como resultado a la sociedad poder para el individuo, la eficacia social, la comprensión, la armonía y el funcionamiento de la potencia de la informática en la vida real*" (Blázquez Ochando, 2012). En cambio, los científicos Cronin y McKim consideran que: "*La web funciona como un foro mundial, un espacio compartido que crea nuevas formas de interacción social*" (Cronin, y otros, 1999).

Como uno de los principales medios de difusión de la información en la actualidad, la *Web* es utilizada por 2 405 518 376 (Group, 2013) de usuarios aproximadamente con diferentes propósitos. La misma ofrece servicios para múltiples usos como la publicidad, la educación, el entretenimiento y el comercio electrónico, propiciando su dinamismo para que se encuentre en constante crecimiento. El estudio de sus características proporciona diversos tipos de información para entender su estructura y evolución. Debido a que ha estado sometida a constantes investigaciones, la importancia de conocer sus características, permite establecer parámetros sobre su desarrollo, así como observar tendencias de los usuarios y/o desarrolladores que interactúan con ella. En virtud de la información que proporciona, se han realizado varios trabajos relacionados con la caracterización de la *Web*; en los cuales se ha tenido en cuenta la utilización de indicadores webmétricos que permitan el conocimiento detallado acerca de su estructura y calidad.

Los indicadores webmétricos a los que se ha hecho alusión anteriormente son elementos que se tienen en cuenta para realizar estudios de la *Web*, los cuales ayudan a caracterizar su comportamiento. Entre los países que han incursionado en dichos estudios se encuentran: Argentina, Austria, Chile, Brasil, Corea, España, Hungría, Perú y Cuba.

Actualmente en Cuba, específicamente en la Universidad de las Ciencias Informáticas (UCI), los *webmasters*¹ no disponen de suficiente información sobre las características de los sitios web en dicha

¹ Webmaster: es el encargado de la programación de los sitios *web*, darles mantenimiento y actualizar su contenido.

universidad. No cuentan con una caracterización actualizada de los mismos, lo que implica la carencia de estudios de indicadores que permitan medirlos en cifras para la obtención de estadísticas rigurosas sobre el impacto de la red en distintas actividades de interés científico, tecnológico, económico, político, social y su trascendencia para el proceso de informatización de la sociedad cubana.

Además, no existe la manera de construir indicadores que permitan el estudio de aspectos tales como la evolución del tamaño de la *Web*, y la forma en que la misma pudiera hacer algún tipo de cambio; aunque no necesariamente en el tamaño sino en ámbitos como estructura y el contenido. En ocasiones a dichos *webmasters* se les dificulta la realización de labores de mantenimiento tales como: la detección y corrección de vínculos rotos; así como las clasificaciones de páginas por tipo de documentos, distribuciones de sitios por dominios, páginas según el tipo de idioma utilizado en las mismas, la manera de mostrar la información, o sencillamente por la edad de cada una de ellas.

De lo expuesto anteriormente se puede establecer como **problema de la investigación**, el siguiente: ¿Cómo contribuir a la toma de decisiones de los *webmasters* sobre la gestión de los sitios *web* en la UCI?

Por tanto el **objeto de estudio** donde estará enmarcada la investigación es la Webmetría, siendo el **campo de acción** el proceso de obtención de estadísticas webmétricas en la UCI.

Estableciendo como **objetivo general**: Desarrollar una herramienta de análisis webmétrico que permita contribuir a la toma de decisiones de los *webmasters*, sobre la gestión de los sitios web en la UCI para mejorar su estructura y composición.

El cual es desglosado en los siguientes **objetivos específicos**:

- Elaborar el marco teórico conceptual y el estado del arte respecto a las tecnologías actuales para realizar análisis webmétrico.
- Diseñar una herramienta de análisis webmétrico que permita contribuir a la toma de decisiones de los *webmasters*, sobre la gestión de los sitios web en la UCI.
- Implementar una herramienta de análisis webmétrico que permita contribuir a la toma de decisiones de los *webmasters*, sobre la gestión de los sitios web en la UCI.
- Validar el correcto funcionamiento de la herramienta webmétrica implementada.

Para dar cumplimiento a estos objetivos se proponen las siguientes **tareas de la investigación**:

- Estudio de la bibliografía necesaria para conocer el estado actual de las tecnologías y herramientas que se utilizan para desarrollar aplicaciones webmétricas.
- Análisis de los indicadores webmétricos existentes, para caracterizar la información de los sitios *web* en la UCI.
- Definición de la metodología a utilizar en el desarrollo de la herramienta de análisis webmétrico.
- Identificación de necesidades de información, requisitos funcionales y no funcionales.
- Documentación de los artefactos generados durante los procesos de análisis, diseño e implementación de la herramienta webmétrica según la metodología seleccionada.
- Implementación de una herramienta webmétrica que permita obtener estadísticas de los sitios web y generar reportes.
- Aplicación de las pruebas de funcionalidad a la herramienta de análisis webmétrico implementada.

Esperando como **posible resultado**:

Obtención de una herramienta para el análisis webmétrico en la Universidad de las Ciencias Informáticas.

Para la realización de la investigación relacionada con la webmetría se utilizarán los siguientes **métodos de investigación**:

Analítico-Sintético

Se aplicará para lograr el entendimiento a partir del análisis de caracterizaciones realizadas con anterioridad sobre la *Web* tanto en el mundo como en la Universidad de las Ciencias Informáticas (UCI) y los indicadores webmétricos utilizados en las mismas.

Histórico-Lógico

Permitirá tener mayor comprensión de los estudios webmétricos a través del análisis de su evolución cronológica a nivel mundial.

Entrevista

Se aplicará a webmasters, para obtener conocimiento acerca de las necesidades existentes con relación a la forma en la que es evaluado el comportamiento de los sitios web en la UCI.

Modelación:

Se utilizará en el modelamiento de diagramas, para la realización del diseño de la herramienta webmétrica a implementar.

El Trabajo de Diploma tiene una estructura de tres capítulos, los cuales están descritos de la siguiente forma:

Capítulo I: Aspectos Teóricos Generales sobre Webmetría: Comprende el estudio de los conceptos fundamentales relacionados con la webmetría, de sistemas homólogos en el ámbito internacional, nacional y en la Universidad de las Ciencias Informáticas (UCI), así como el análisis de las tecnologías, herramientas y metodología de desarrollo.

Capítulo II: Propuesta de la Herramienta para Análisis Webmétrico: Establece las características del sistema a desarrollar, realizando el diseño de la aplicación a través del modelado del dominio, la especificación de los requerimientos del sistema, la descripción de los patrones arquitectónicos y diseño, así como la modelación del diagrama de casos de uso y la especificación de los mismos, además de la modelación del diagrama de clases del diseño.

Capítulo III: Vista Previa de la Herramienta para Análisis Webmétrico: Muestra las pantallas principales del sistema, el estándar de codificación, el diagrama de componentes y modelo de despliegue, así como los resultados de las pruebas realizadas al mismo.

1 Capítulo I: Aspectos teóricos generales sobre webmetría.

1.1 Introducción

El presente capítulo comprenderá el estudio de los conceptos fundamentales relacionados con la webmetría. Además se realiza una caracterización de sistemas homólogos en el ámbito internacional y nacional, así como el análisis de las tecnologías, lenguajes y metodología de desarrollo a utilizar.

1.2 Conceptos Fundamentales

El presente epígrafe comprenderá el estudio de conceptos y definiciones relacionados con la webmetría e indicadores webmétricos. De los indicadores webmétricos se estudiarán las métricas clasificadas por categorías y se realizará una selección de las adecuadas para brindar estadísticas que sean de utilidad en el cumplimiento del objetivo propuesto.

1.2.1 Webmetría

Para realizar un estudio webmétrico se debe tener bien clara la definición del término webmetría², la cual ha sido abordada de diferentes formas por varios científicos reconocidos mundialmente, entre los que se pueden citar, se encuentra:

El científico Abraham, el cual desde su punto de vista establece que la Webmetría “estudia la relación entre los elementos principales de la *Web* (comparable a una red neuronal conformada por nodos y conexiones: los nodos se corresponden con los dominios, servidores y páginas que componen la *Web* y las conexiones con los enlaces que se establecen entre los mismos dando lugar a una matriz de conexiones que abarca todo el entorno WWW)” (Fernández Herrera, 2010).

También Björneborn, asegura que la Webmetría “es el estudio de los aspectos cuantitativos de la construcción y uso de los recursos de información, estructuras y tecnologías de una parte concreta de Internet, por regla general a una web o portal” (Björneborn, 2004).

En 1997 Tomas C. Almind y Meter Ingwerse sostienen que la Webmetría “utiliza informaciones y datos extraídos de motores de búsqueda, tales como el número y los tipos de hipervínculos, el número y el tipo de archivos, así como la tipología del sitio web” (Fernández Herrera, et al., 2010)

Luego de haber analizado estos conceptos expuestos anteriormente, se está de acuerdo con todos pero se escoge como el más completo al establecido por el autor Björneborn, el cual abarca en general, todos los aspectos que son estudiados de la *Web*.

² En los diferentes artículos relacionados con los estudios webmétricos se ha utilizado el término webmetría como traducción de *webometrics* (en inglés), por lo que dicho término será utilizado a todo lo largo de la investigación.

Cuando hablan de webmetría, hacen referencia a los distintos estudios que le realizan a la Web constantemente, para ello se hace necesario el establecimiento de indicadores webmétricos, los cuales permitirán la evaluación de la misma, contribuyendo así al apoyo en la toma de decisiones de variados aspectos.

1.2.2 Indicadores webmétricos

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

De los indicadores que se mostrarán en la tabla siguiente, puede que existan dudas acerca del significado de algunos de ellos, por lo que posteriormente se dará una breve descripción de los mismos:

- Factor de impacto: se basa en contar el número de enlaces que recibe una página web con el fin de determinar su relevancia en relación con otras.
- Porcentaje de enlaces válidos: se basa en el porcentaje de los vínculos que se encuentran en buen estado, es decir, se puede acceder a ellos sin problema alguno.
- Porcentaje de errores de enlaces: se basa en el porcentaje de la cantidad de enlaces que tienen error de acceso.

A continuación se muestra en la **Tabla 1**, una agrupación de los indicadores, por las categorías generales que evalúan:

Categorías	Indicador
Infraestructura	Número de <i>host</i> ³ , de servidores web, de usuarios, de dominios, de sitios y de sitios institucionales.

³ Host: (“anfitrión” en español). En informática se refiere a las computadoras u ordenadores que se encuentran conectados a una red, las cuales proveen y utilizan servicios de ella.

Tamaño	Número de páginas, de objetos, de objetos multimedia, de archivos ejecutables, tamaño de los archivos, distribución por lenguajes, evolución temporal, número de niveles y de enlaces por página.
Calidad	Porcentaje de enlaces válidos, de errores de enlace y apariencia.
Conectividad	Total de enlaces, número de enlaces internos y externos.
Visibilidad	Número de enlaces recibidos o externos, enlaces nacionales externos y enlaces internacionales externos.
Impacto	Factor de impacto.
Popularidad	Número de visitas.

Tabla 1. Indicadores webmétricos

A partir de las necesidades planteadas en la entrevista aplicada en el Departamento de Tecnología de la Dirección de Informatización en la UCI (Ver Anexo 5). Se realizó un estudio a los indicadores mostrados anteriormente por categorías, en el cual se hizo una búsqueda de cuáles de ellos eran factibles para la obtención de estadísticas que brindaran información de los sitios web a analizar. La misma estuvo centrada en los indicadores que deben cubrir las necesidades planteadas en la entrevista. A continuación se muestra la selección de dichos indicadores:

- Número de páginas
- Número de objetos multimedia
- Número de archivos ejecutables
- Porcentaje de errores de enlace
- Estadísticas de lenguaje

- *Ranking* de páginas
- Total de enlaces por página

Con el objetivo de realizar las mediciones pertinentes a los sitios web en la UCI, para ayudar a la toma de decisiones de los webmasters a la hora de obtener información actualizada de dichos sitios, se hace necesario realizar el análisis de herramientas que midan los indicadores webmétricos antes mencionados.

1.3 Estudio de sistemas homólogos en el ámbito nacional e internacional.

A continuación se realizará una breve descripción de los sistemas encontrados en la bibliografía consultada, en función a si utilizan los indicadores webmétricos descritos anteriormente, los cuales son de interés para la presente investigación.

PostRank Analytics es una herramienta de analítica de medios sociales que resulta de utilidad para un mejor conocimiento de parámetros como la participación, comunidad y el posicionamiento de un sitio web, trabaja con 20 de las redes sociales más importantes, entre las que se encuentran *Twitter*, *Linkedin*, *Delicious*, etc. Uno de los recursos que la misma brinda es *Top Posts Widget*, el cual es utilizado para destacar los *posts* (artículos) que más interés despiertan (Martínez Guaita).

El sistema anterior no mide los indicadores webmétricos que son de interés para la presente investigación, debido a que las estadísticas que brinda se encuentran enfocada en los artículos publicados en las redes sociales u otros sitios. Además es una herramienta propietaria, lo cual va en contra de las políticas de informatización de la universidad.

WooRank es una herramienta a la cual se le ingresa una URL de algún sitio web y de ella se obtiene un detallado y rápido informe del sitio web ingresado. Dentro del diagnóstico se destacan enlaces, velocidad de carga, palabras clave, información del dominio, software y estándares usados, últimas publicaciones en las redes sociales y hasta una vista de cómo se ve el sitio en dispositivos móviles (David Quiñónez, 2012). El sistema *WooRank* brinda estadísticas relacionadas con los indicadores webmétricos de interés para la presente investigación, pero tiene como inconveniente que solo analiza un único sitio web. Además es un sistema privativo, lo cual va en contra de las políticas de informatización de la universidad.

Google Analytics es una herramienta utilizada por millones de usuarios, la cual realiza análisis de visitas en tiempo real, comportamiento de los visitantes (mapa de calor), información de procedencia (país, sistema operativo, navegador, hora), contenido más popular, variaciones en gustos, estadísticas de conversiones, múltiples usuarios, reportes en PDF, etc. (David Quiñónez, 2012).

Google Analytics es una herramienta que brinda indicadores webmétricos relacionados con la presente investigación, pero se descarta debido a que es privativo. Esto impide se pueda personalizar de acuerdo a las necesidades de dicha investigación y su uso se le puede negar al país como otros de los servicios que presta la empresa *Google*.

Generador de Estudios Webmétricos (GEWEB v1.0) es un producto informático cuyo objetivo principal consiste en realizar un estudio webmétrico (algo muy similar a una fotografía estadística de la web), con múltiples resultados cuantitativos y cualitativos, para asistir a la toma de decisiones respecto a la web analizada. Mediante la utilización del Spider WIRE, el mismo permite extraer estadísticas específicas sobre la Web analizada tales como: estructura, tamaño y contenido de la Web, elabora una serie de distribuciones como conectividad, tecnologías utilizadas, idiomas presentes y edad de las páginas o sitios web (Mondelo, 2010).

El sistema brinda todas las estadísticas relacionadas con los indicadores webmétricos que son de interés para la presente investigación. No obstante, no se cuenta con la aplicación ni el código fuente no se encuentra disponible en la UCI.

Es de destacar que en el desarrollo de esta aplicación se utilizó el sistema WIRE, el cual es un *spider* encargado de recorrer los sitios web (Ver epígrafe 1.5.6).

WebSiteAnalyzer: es una herramienta webmétrica libre para el análisis personalizado de sitios web en la UCI, la cual permite el análisis de un sitio web, obteniendo información crucial para el desarrollo eficiente y con calidad de la *Web* universitaria. Los indicadores Tamaño y Calidad son los que mayor influencia tuvieron en la definición principal del desarrollo de dicha herramienta, donde contribuyen a la evaluación del número de páginas que tiene el sitio en cuestión, el número total de objetos (Ficheros), la cantidad de ficheros ricos, el número de ficheros media, ficheros dinámicos y ejecutables, así como el tamaño de los ficheros y las páginas bajo protocolos seguros, además se puede conocer la cifra exacta de todos los enlaces rotos, ya sean externos o internos, lo cual es de gran importancia, pues cuando el índice de estos problemas sea menor, influenciará grandemente en la mejora de la calidad del sitio (Casanova Hernández, y otros, 2012).

El sistema brinda la capacidad de extracción de información para una cantidad de sitios limitada, por lo que si se desea realizar el análisis de una cantidad de sitios que exceda el límite fijado en el sistema, los sitios restantes dejan de ser analizados completamente.

En el anterior estudio de sistemas homólogos se destaca que *Google Analytics*, *PostRank Analytics* y *WooRank* son propietarios, por lo cual su uso queda descartado como propuesta de solución en el

presente trabajo de diploma. La herramienta desarrollada en la UCI *WebSiteAnalyzer*, solo extrae información de un único sitio web y no puede obtener información de sitios que tienen protocolo HTTPS. GeWeb no se encuentra actualmente en desarrollo ni se dispone del código fuente y ni de la herramienta en la universidad, tarda en ejecutar el código fuente y consume gran cantidad recursos

De este análisis queda evidenciada la necesidad de implementar una nueva herramienta que sea capaz de brindar estadísticas que ayuden a la toma de decisiones de los webmasters, teniendo en cuenta la medición de los indicadores webmétricos antes mencionados. Para el desarrollo de esta herramienta se reutilizará el *spider WIRE* utilizado previamente en el sistema GEWeb, por lo cual se hace necesario analizar algunos temas teóricos sobre los *webcrawlers* o *spiders* y su funcionamiento.

1.4 Webcrawler

Un *webcrawler* es un programa que cumple múltiples propósitos de análisis y extracción de información de la web. Constituye el instrumento de investigación principal con el que se realizan los estudios cibernéticos y webmétricos, lo que implica una estrecha relación entre la información que es capaz de recuperar y las técnicas de análisis, tabulación y medición de la metría. Se trata de un programa especializado en el desarrollo de análisis webmétricos para una determinada área del conocimiento en la web, del cual posteriormente se analizará su funcionamiento (Blázquez Ochando, 2012).

Funcionamiento de un webcrawler

Primeramente se hace necesario elaborar un archivo txt, el cual contendrá las URL iniciales que se pretenden analizar, luego se realiza un proceso de configuración del webcrawler en el que se determina la profundidad del análisis según los niveles de enlazamiento de los sitios y páginas definidas en el archivo txt. Esto es analizar los vínculos de los sitios y páginas de la semilla de forma sucesiva, hasta finalizar el proceso. Cada salto de una página a otra, se denomina nivel de profundidad, de tal manera que es posible navegar de una página a través de sus vínculos, determinando un recorrido que puede ser trazado y reflejado en un sistema de información, como un *webcrawler*. Pero también pueden y deben configurarse otros ajustes de importancia, como por ejemplo el buffer (que permite retener la información del proceso de extracción de datos), el tiempo en cache (para determinar el número de segundos que el sistema mantiene las entradas DNS en memoria), el tiempo de conexión (define el número de milisegundos que el sistema espera cuando está intentando conectar con la dirección URL especificada) y el tiempo de ejecución máximo por URL. Finalmente también es relevante determinar filtros y extensiones para que el análisis del *webcrawler* sea más especializado y rápido. Por ejemplo aplicar restricciones por sitio web, extensión o por patrones, permite diferenciar fácilmente el tipo de enlace se pretende recuperar en el análisis webmétrico y obviar o no aceptar aquel que fue especificado.

El proceso de *webcrawling*, consiste en la extracción de toda la información de la dirección URL de una página o sitio web objetivo que de forma secuencial presenta el programa *Webcrawler*. Para realizar la extracción de la información de una página web y dividir sus correspondientes elementos, se emplean programas de tipo *parser*⁴ dentro del *webcrawler*, capaces de reconocer las etiquetas e instrucciones de HTML y con ello desarrollar el proceso de extracción de los datos contenidos en ellas. De la misma forma que existen *parsers* capaces de analizar archivos de metadatos en XML, existen *parsers* capaces de recopilar los enlaces de una página, los párrafos, metadatos en HTML, canales de sindicación, imágenes, documentos y archivos multimedia (Blázquez Ochando, 2012).

1.5 Tecnologías, lenguajes y metodología de desarrollo

Para la construcción de un software o aplicación informática, se hace necesario la utilización de una metodología de desarrollo que sirva de guía para el diseño de la misma. A partir de la selección de dicha metodología, se definen los procedimientos, herramientas a utilizar y los artefactos que deben ser generados en el proceso. A continuación se realizará un breve estudio acerca de las metodologías existentes según su clasificación y se seleccionará la apropiada para llevar a cabo el objetivo propuesto.

1.5.1 Metodología de desarrollo

Las metodologías de desarrollo surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un software, las mismas se encuentran clasificadas en dos grupos: tradicionales y ágiles.

Las metodologías tradicionales se centran en la definición detallada de los procesos, tareas a realizar y herramientas a utilizar, requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida. Entre las metodologías tradicionales se cita a continuación RUP (Rational Unified Process) que divide el desarrollo en 4 fases que definen su ciclo de vida: Inicio, Elaboración, Construcción y Transición. Sin embargo las metodologías ágiles se encargan de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados, en ellas se hace mucho más importante crear un producto software que funcione que escribir mucha documentación, cliente está en todo momento colaborando en el proyecto y es más importante la capacidad de respuesta ante un cambio. Como ejemplo vivo de las mismas, las cuales se encuentran entre las más utilizadas, están: la Programación Extrema (XP por sus siglas en inglés), Scrum y OpenUp.

⁴ Parser: es un programa que analiza un texto para determinar su estructura sintáctica.

Luego de haber realizado el análisis de las diferentes metodologías de desarrollo de software que constituyen un elemento muy importante en la construcción de un producto, se seleccionó OpenUp como metodología a utilizar en el desarrollo de la herramienta para el análisis webmétrico en la Web universitaria, ya que la misma es apropiada para proyectos pequeños y de bajos recursos, permitiendo disminuir las probabilidades de fracaso en dichos proyectos e incrementar las de éxito, permite detectar errores tempranos a través de un ciclo iterativo, evita la elaboración de documentación, diagramas e iteraciones innecesarias requeridas en la metodología RUP y además, tiene un enfoque centrado al cliente con iteraciones cortas y se encuentra en la base tecnológica del centro de desarrollo al cual se le va a desarrollar la herramienta webmétrica.

OpenUp

Es un proceso extensible, dirigido a la gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental apropiado para proyectos pequeños y de bajos recursos; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo. Tiene como principios colaborar para sincronizar intereses y compartir conocimiento, equilibrar las prioridades para maximizar el beneficio obtenido por los interesados en el proyecto, centrarse en la Arquitectura de forma temprana para minimizar el riesgo y organizar el desarrollo y realizar un desarrollo evolutivo para obtener retroalimentación y mejoramiento continuo.

Una vez seleccionada la metodología de desarrollo a utilizar, es necesario definir el lenguaje de modelado con el cual se especificarán y visualizarán los artefactos de la aplicación informática.

1.5.2 Lenguaje de modelado

Los lenguajes de modelado, en su mayoría visuales, intentan representar un sistema de software a un nivel más alto que los lenguajes de programación, representándolo en forma intuitiva para personas sin especialización en informática. Existen distintos tipos de lenguajes de modelado que han sido utilizados en diversos proyectos, tomados a consideración por el equipo de desarrollo en conveniencia de lo que es factible para ellos a la hora de lograr un entendimiento entre lo que quiere el cliente y lo que ellos implementarán, entre ellos se encuentran: la Notación para el Modelado de Procesos de Negocio (BPMN por sus siglas en inglés), el cual es un lenguaje de modelado que está planeado para dar soporte únicamente a aquellos procesos que sean aplicables a procesos de negocios, lo cual significa que cualquier otro tipo de modelado, con fines distintos a los del negocio, no estará en su ámbito; también está el Lenguaje Unificado de Modelado (UML por sus siglas en inglés), siendo este el más conocido y utilizado en la actualidad para especificar, visualizar, construir y documentar artefactos de un sistema de software, captura decisiones y conocimiento sobre los sistemas que se deben construir, y además, está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y

medios. Debido a estas características, este último ha sido seleccionado para la modelación del desarrollo de la herramienta webométrica ya que se ajusta a la idea que se tiene para visualizar lo que se desea desarrollar.

Con la selección del lenguaje de modelado, se hace necesario tener en cuenta la herramienta donde se modelarán los diagramas de clases que proporcionarán la visión de cómo estará estructurada la aplicación informática a desarrollar.

1.5.3 Herramienta Case Visual Paradigm

Para la realización del diseño de la herramienta webométrica se utilizará Visual Paradigm la cual es una herramienta para el Lenguaje de Modelado Unificado (UML del inglés *Unified Modeling Language*) profesional que soporta el ciclo de vida completo del desarrollo de software, contribuye a una rápida construcción de aplicaciones con calidad a un menor costo, además permite construir todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases, permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación.

Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto. Incorpora el soporte para trabajo en equipo, donde permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

Visual Paradigm es una herramienta que cumple con las políticas actuales de migración a software libre, siendo multiplataforma, de forma tal que facilita la modelación del software independientemente del sistema operativo que se emplee.

Para la implementación de las clases modeladas, se hace necesario la utilización de un entorno integrado de desarrollo, en el cual se utilice un lenguaje de desarrollo acorde con el objetivo que se desea lograr.

1.5.4 Entorno Integrado de Desarrollo

Un Entorno Integrado de Desarrollo (IDE por sus siglas en inglés) es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, compilador, depurador y constructor de interfaz gráfica. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

El IDE seleccionado para la implementación de la herramienta webométrica fue IntelliJIDEA, debido a la facilidad que brinda para trabajar con el lenguaje de desarrollo Groovy. Además, el mismo brinda soporte

para la realización de pruebas unitarias de integración de *frameworks*⁵ con un conveniente compilador de pruebas. Otras características que hacen a este IDE atractivo:

- Facilidad de realizar completamiento de código inteligente.
- En la marcha del análisis del código, garantiza la calidad del mismo a medida que se escribe y sugiere soluciones rápidas.
- Se encarga de aplicar los cambios complicados para el código de forma segura.
- Ofrece un conjunto de herramientas integradas que hacen el desarrollo más productivo, tales como herramientas de base de datos incluyendo un editor de base de datos con todas las funciones y soporte SQL.
- Ofrece herramientas para el control de versiones con una interfaz unificada para Git, SVN y Mercurial.
- Tiene soporte para los lenguajes basados en JVM (*Java Virtual Machine*) más populares, incluyendo Java, Scala, Groovy, PHP, Python y Ruby.
- Despliega las aplicaciones rápidamente, directamente desde el IDE.
- Utiliza gestión de tareas y contexto para la integración con gestores de fallos y el cambio de contexto.

Luego de estar seleccionado el IDE con el que se va a trabajar en el desarrollo de la herramienta webmétrica, se hace necesario la selección del lenguaje de desarrollo, la cual se verá reflejada a continuación.

1.5.5 Lenguaje de Desarrollo

Un lenguaje de desarrollo o programación en informática es cualquier lenguaje que pueda utilizarse para definir una secuencia de instrucciones para su procesamiento en una computadora. Generalmente los lenguajes de programación son usados para crear programas que controlen el comportamiento físico y lógico de una máquina. Para el desarrollo de la herramienta para realizar el análisis webmétrico en la UCI, ha sido seleccionados varios lenguajes, entre los que se pueden mencionar: HTML como lenguaje de marcado hipertexto para el diseño de la interfaz *web*. También fue seleccionado el lenguaje XML para el trabajo con el fichero de configuración del sistema WIRE, en el momento de establecer los parámetros para la realización del recorrido por los sitios web de la UCI. Además, se seleccionó Groovy para la implementación de la interfaz web de la herramienta webmétrica. Groovy es un lenguaje orientado a objeto para la plataforma Java, dinámico y ágil, similar a Python, Perl y Ruby. El mismo, resulta ser cómodo para trabajar en Entornos Integrados de Desarrollo (IDE) para crear aplicaciones web. Entre las ventajas que posee el lenguaje, está la forma en la que es escrito el código, debido a que las sintaxis son

⁵ Framework: marcos de trabajo.

compactas, tiene paquetes importados de forma predeterminada, lo que elimina la utilización en el código de la palabra clave *import* en el caso de la utilización de los mismos, así como las clases y los métodos son públicos de forma predeterminada también (Judd, y otros, 2008).

1.5.6 WIRE

El sistema WIRE es un esfuerzo iniciado por el Centro de Búsquedas en la Web (*Center For Web Research*). Actualmente incluye un formato simple para almacenar una colección de documentos web, herramientas para la extracción de estadísticas desde la colección de documentos extraídos y la generación de reportes acerca de dicha colección.

Entre las principales características de este software se encuentran:

- Es un software escalable debido a que fue diseñado para trabajar con grandes volúmenes de documentos.
- Se encuentra escrito en C/C++ para tener un alto rendimiento.
- Todos los parámetros para la indexación y rastreo pueden ser configurados a través de un archivo XML.
- Su código fuente se encuentra libremente disponible.

1.5.7 Marco de trabajo Grails

Grails es un nuevo *framework* web para la plataforma Java, basado en el lenguaje dinámico Groovy. Hace uso de un ecosistema de *PlugIns* que mejora la productividad en un conjunto cada vez más amplio de escenarios. Grails se ha hecho inmensamente popular, y un motor de cambio en el espacio Java. Grails permite crear aplicaciones en días, en lugar de semanas. En comparación con el típico *framework* de Java, con Grails se necesita menos código para obtener el mismo resultado. Menos código significa menos errores y menos líneas de código de mantener (Rocher, 2009). Grails se encuentra bajo la licencia de *software* Apache 2.0, pretende ser un marco de trabajo altamente productivo siguiendo paradigmas tales como: convención sobre configuración o no te repitas, proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador.

1.5.8 Sistema de gestión de base de datos *SQLite*

En el desarrollo de la herramienta webométrica fue utilizado el sistema de gestión de base de datos *SQLite*, debido a que no se manejan grandes cantidades de datos porque solo se almacenan los datos de un único usuario. Los mismos son utilizados en la autenticación del sistema.

SQLite es un sistema completo de bases de datos que soporta múltiples tablas, índices, triggers y vistas. No necesita un proceso separado funcionando como servidor ya que lee y escribe directamente sobre

archivos que se encuentran en el disco duro. El formato de la base de datos es multiplataforma e indistintamente se puede utilizar el mismo archivo en sistemas de 32 y 64 bits. La base de datos se almacena en un único fichero a diferencia de otros sistemas de bases de datos que hacen uso de varios archivos. *SQLite* emplea registros de tamaño variable de forma tal que se utiliza el espacio en disco que es realmente necesario en cada momento. El código fuente está pensado para que sea entendido y accesible por programadores promedio. Todas las funciones y estructuras están bien documentadas. Existe un programa independiente de nombre *sqlite* que puede ser utilizado para consultar y gestionar los ficheros de base de datos *SQLite*. También sirve como ejemplo para la escritura de aplicaciones utilizando la biblioteca *SQLite*. Además se encuentra preparada para bases de datos de sitios web de pequeño y mediano tamaño (Jeon, y otros, 2011).

A continuación se muestran algunos beneficios que brinda *SQLite* como sistema de gestión (Jeon, y otros, 2011):

- Tamaño: *SQLite* tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- Rendimiento de base de datos: *SQLite* realiza operaciones de manera eficiente y es más rápido que *MySQL* y *PostgreSQL*.
- Portabilidad: se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- Estabilidad: *SQLite* es compatible con ACID, reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.
- SQL: implementa un gran subconjunto de la ANSI – 92 SQL estándar, incluyendo sub-consultas, generación de usuarios, vistas y triggers.
- Interfaces: cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, *Perl*, *Python*, *Ruby*, *Groovy* y *Qt*, ofreciendo el *plugin sqlite*.
- Costo: *SQLite* es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.

Conclusiones parciales del capítulo

- El análisis de indicadores webmétricos utilizados a nivel mundial, permitió realizar una selección de los indicadores que son de interés para la presente investigación.
- Al realizar el estudio de los sistemas homólogos se llegó a la conclusión que los sistemas analizados no miden los indicadores webmétricos de interés para la presente investigación. Por lo

que es necesario implementar una nueva herramienta capaz de medir los indicadores antes mencionados.

- Para realizar el recorrido de los sitios web se reutilizará el sistema WIRE como *spider*.
- Para el desarrollo de la herramienta propuesta se utilizará Visual Paradigm 8.0 para el diseño de los diagramas de clases, UML como lenguaje de modelado, OpenUp como metodología de desarrollo, IntelliJIDEA como entorno integrado de desarrollo y Groovy como lenguaje de programación.

2 Capítulo II: Propuesta de la Herramienta para Análisis Webmétrico

2.1 Introducción

En el presente capítulo se establece las características del sistema a desarrollar, realizando el diseño de la aplicación a través del modelado del domino, la especificación de los requerimientos del sistema, la descripción de los patrones arquitectónicos y diseño, así como la modelación del diagrama de casos de uso y la especificación de los mismos, además de la modelación del diagrama de clases del diseño y la presentación del modelo de despliegue.

2.2 Características de la herramienta

Dada la necesidad planteada en la situación problemática, se decide desarrollar una herramienta para el análisis webmétrico en la UCI. La misma será accedida por un administrador de red o webmaster a través de una interfaz web, mediante la cual podrá obtener varias estadísticas de otros sitios web. Las estadísticas son mostradas a través de los reportes. Dichos reportes brindan variada información acerca de los sitios que van siendo analizados, tales como: el tamaño, la calidad y estructura de los mismos, así como el lenguaje contenido en ellos. Además el administrador puede configurar el *spider* para modificar los parámetros de recorrido y el tiempo estimado para ello.

2.3 Modelado del dominio

Luego de haber realizado un análisis del problema, se decide realizar el modelo de dominio debido a que el negocio no se encuentra bien definido. El modelo de dominio es la representación conceptual de las clases más importantes contenidas en el sistema. Su objetivo principal se enmarca en la comprensión de los conceptos y descripción de dichas clases, teniendo una visión estática de las mismas y sus interacciones.

En el modelo se puede apreciar un **webmaster**, siendo este el actor principal en el proceso. El mismo se encarga de **modificar** las **configuraciones** y **generar** los **reportes** donde se **muestran** las **estadísticas** que **pertenecen** al **sitio web**, el cual es **administrado** el mismo **webmaster** (Ver Figura 1).

Descripción de las clases del dominio

Webmaster: Persona encargada del acceso al sitio web para la realización del análisis webmétrico.

Configuración: Son los diferentes parámetros a los que se le hacen modificaciones para adaptar el sistema a las necesidades del webmaster.

Reporte: Informe estructurado y resumido donde se muestran las estadísticas.

Estadísticas: Información que guarda relación con la estructura calidad y contenido de los sitios web analizados.

Sitio web: Interfaz web con la que interactúa el webmaster.

Diagrama de clases del modelo del dominio

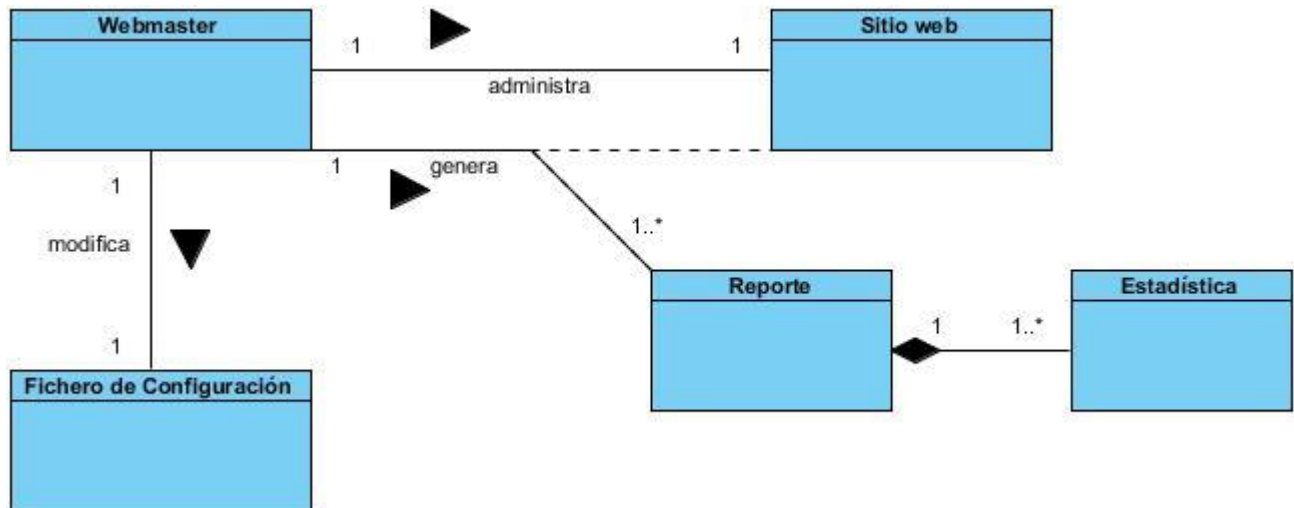


Figura 1. Diagrama del modelado del dominio.

2.4 Especificación de requisitos de software

Los requisitos son tareas que determinan las necesidades o condiciones a satisfacer de un sistema determinado, los mismos deben ser medibles, comprobables, sin ambigüedades o contradicciones. Estos son clasificados en funcionales y no funcionales, siendo estos últimos propiedades o cualidades que el sistema debe tener.

A continuación, en la **Tabla 2** se mostrarán los requisitos que debe satisfacer la herramienta para análisis webmétrico:

Requisitos funcionales

RF-1	Autenticar usuario	RF-2	Cambiar contraseña
RF-3	Cargar urls	RF-4	Reiniciar datos
RF-5	Realizar recorrido	RF-6	Generar reporte de extensiones
RF-7	Generar reporte de recorrido	RF-8	Generar reporte de idiomas
RF-9	Generar reporte de sitios	RF-10	Generar reporte de enlaces
RF-11	Generar reporte de documentos	RF-12	Mostrar fichero de configuración
RF-13	Mostrar fichero de urls iniciales		

Tabla 2. Requisitos funcionales

Requisitos no funcionales

Los requisitos no funcionales son cualidades que el sistema debe tener, los mismos se basan en la apariencia, seguridad, las condiciones que deben tener el software y el hardware donde será desplegado el sistema para que el mismo pueda funcionar correctamente. Además, en los requisitos no funcionales, se dejan esclarecidos los acuerdos entre el cliente y el equipo de desarrollo en cuanto a las actividades de soporte que debe ofrecer este último para garantizar la satisfacción del cliente. Para el desarrollo de la herramienta webmétrica se establecieron los siguientes requisitos no funcionales:

Apariencia de interfaz

RNF-1 La aplicación debe contar con una interfaz amigable y sencilla en la que se representen los datos con colores claros.

Seguridad

RNF-2 La herramienta debe contar con un único rol, perteneciente al webmaster que se encarga de realizar las configuraciones y generar los reportes estadísticos.

RNF-3 La herramienta no debe permitir la autenticación del webmaster en dos estaciones de trabajo distintas.

Hardware

RNF-4 La computadora donde se despliegue la herramienta debe tener como mínimo 5GB de espacio disponible en disco duro.

RNF-5 Se debe utilizar como servidor una estación con 2GB de RAM como mínimo.

Software

RNF-6 Utilizar sistema operativo GNU/Linux.

RNF-7 Utilizar como servidor web Apache Tomcat versión 7.

RNF-8 Debe estar instalado y configurado el sistema WIRE en el servidor.

2.5 Diagrama de casos de uso del sistema

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. Este demuestra la interacción de los mismos con cada caso de uso, modelando así el comportamiento del sistema percibido por el usuario externo. El propósito de la vista de casos de uso es enumerar a los actores y los casos de uso, y demostrar qué actores participan en cada caso de uso (Rumbaugh, y otros).

Definición de los actores del sistema

Un actor caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema. Cada actor participa en uno o más casos de uso e interactúa con el caso de uso (y por lo tanto con el sistema o la clase que posee el caso de uso) intercambiando mensajes. El mismo puede ser un ser humano, un sistema informático o un proceso ejecutable (Rumbaugh, y otros).

El sistema cuenta con un único actor, el Webmaster, el cual es el usuario que interactúa con el sistema modificando las configuraciones del WIRE para lograr su objetivo, así como generar los reportes que contiene las estadísticas informativas.

La **Figura 2**, corresponde al diagrama de casos de uso del sistema:

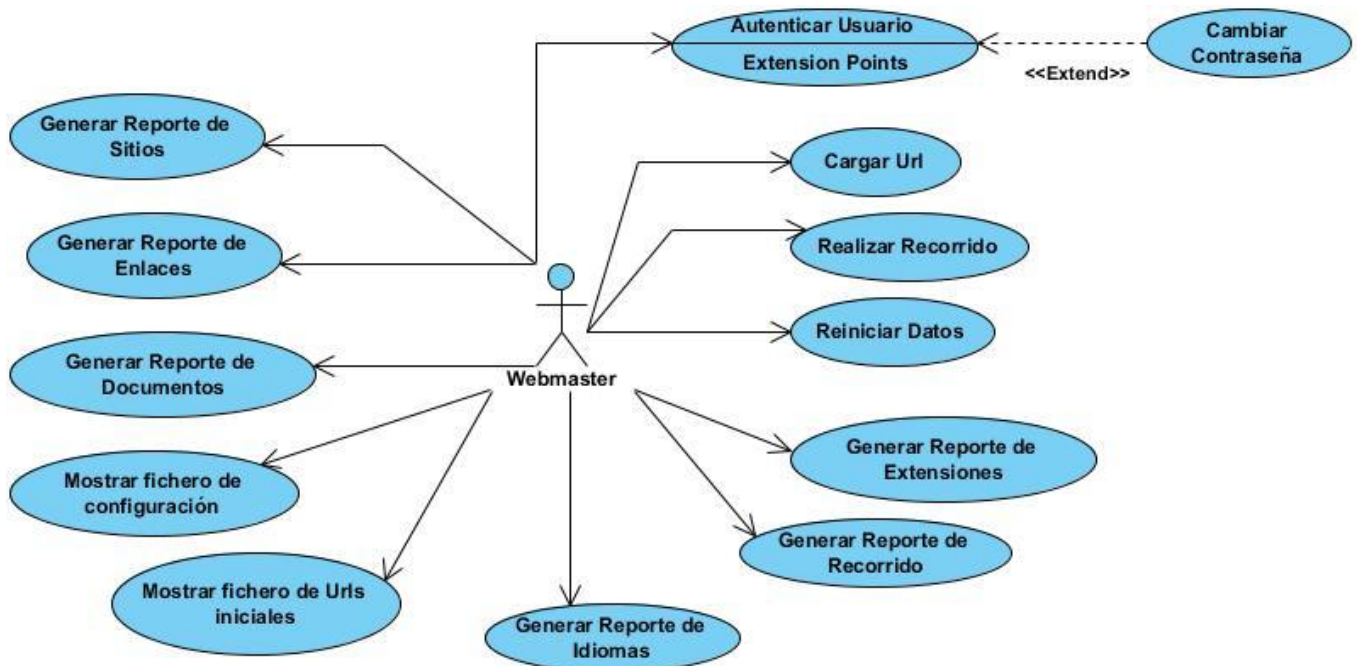


Figura 2. Diagrama de casos de uso del sistema

2.6 Especificación de casos de uso

Para entender la funcionalidad asociada a cada caso de uso no es suficiente con la representación gráfica del Diagrama de casos de uso. La especificación de los casos de uso es una breve o extendida descripción realizada a los casos de uso. La misma incluye el nombre del caso de uso, precondiciones que deben cumplirse para que este sea ejecutado, los actores que interactúan, su complejidad, prioridad y el flujo de eventos.

Las descripciones mostradas a continuación en las **Tablas 3 y 4**, pertenecen a los casos de uso Autenticar Usuario y Cargar *URLs* (*Uniform Resource Locator* por sus siglas en inglés). Las descripciones de los restantes casos de uso serán mostradas en el Anexo 1.

CU: Autenticar usuario

Caso de Uso	Autenticar usuario.
Actores	<i>Webmaster</i> .
Resumen	El CU inicia cuando el <i>webmaster</i> desea autenticarse y accede a la interfaz de autenticación para realizar esta operación, finaliza cuando el <i>webmaster</i> se ha autenticado correctamente.
Complejidad	Media
Prioridad	Alta
Precondiciones	
Poscondiciones	El <i>webmaster</i> luego de haberse autenticado, adquiere todos los privilegios en el sistema.
Prototipo de Interfaz	
 El prototipo de interfaz muestra un formulario de inicio de sesión con el título "Iniciar Sesión" y un ícono de llave dorada con el número "15". El formulario contiene dos campos de entrada: "Usuario" y "Password", y un botón "Aceptar" con una flecha a la izquierda.	
Flujo de eventos	

Flujo básico autenticar usuario		
	Actor	Sistema
1		El sistema muestra la interfaz de autenticación que contiene dos campos: Usuario y Contraseña.
2	El <i>webmaster</i> introduce su usuario y contraseña en el sistema y los envía a la aplicación dando clic en el botón Aceptar.	
3		Comprueba que el usuario esté registrado previamente en el sistema.
4		Comprueba que los datos del usuario sean correctos.
5		Direcciona el usuario hacia la interfaz principal según el rol que posea con los permisos autorizados.
6		Termina el CU.
Flujos alternos		
1. El usuario no se encuentra registrado en el sistema		
	Actor	Sistema
5		Muestra un mensaje de error informando que el usuario no está registrado en el sistema.
2. Los datos del usuario no son correctos		
	Actor	Sistema
6		Muestra un mensaje de error informando que los datos son incorrectos.
Relaciones	CU Incluidos	
	CU Extendidos	CU Cambiar Contraseña.
Requisitos		

Tabla 3. Descripción del caso de uso autenticar usuario

CU: Cargar URLs

Caso de Uso	Cargar URLs.
Actores	Webmaster.
Ejecución	El CU inicia cuando el webmaster accede al menú “Ejecución” en la interfaz,

	selecciona la opción “Cargar Urls”, y el sistema carga las Urls válidas contenidas dentro de un fichero de texto que se encuentra localmente almacenado en el ordenador donde está desplegado el sistema.
Complejidad	Media
Prioridad	Alta
Precondiciones	El fichero txt debe contener Urls.
Poscondiciones	

Prototipo de Interfaz



Flujo de eventos

Flujo básico cargar urls

	Actor	Sistema
1	El webmaster en la interfaz, accede a la opción Cargar Urls que se encuentra en el menú “Ejecución”.	
2		El sistema comprueba que el fichero cargado contenga URLs.
3		El sistema carga las URLs y muestra un mensaje informando la cantidad de URLs válidas que han sido cargadas satisfactoriamente.
4		Termina el CU.

Flujos alternos

2. El fichero cargado está vacío

	Actor	Sistema
--	-------	---------

		El sistema muestra un mensaje informando que el fichero se encuentra vacío.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

Tabla 4. Descripción del caso de uso cargar urls

2.7 Descripción de los patrones arquitectónicos y de diseño

Los patrones arquitectónicos y de diseño permiten tener una estructura de código común a todos los proyectos que implementen una funcionalidad genérica. La utilización de patrones, permite ahorrar grandes cantidades de tiempo en la construcción de software. Los mismos se centran en proporcionar modelos y métodos re-utilizables específicamente para la arquitectura general de los sistemas de información (Larman, 2002).

2.7.1 Patrones arquitectónicos

Los patrones generalmente describen cómo resolver un problema abstracto y la tarea del usuario consiste en modificar los elementos del patrón para cumplir las demandas del compromiso actual. En el desarrollo de la herramienta para análisis webmétrico se utilizará el patrón Modelo-Vista-Controlador (MVC). El mismo propone la separación de la interfaz, los datos y la lógica de control en componentes diferentes:

Modelo: Es la representación de la información con la que el sistema trabaja. La lógica de datos permite derivar nuevos datos y asegura la integridad de estos.

En el sistema, la clase que trabaja con los datos del webmaster es la clase Usuario, la misma guarda información relacionada con los datos de autenticación en dicho sistema.

Vista: Muestra la interfaz de usuario. Maneja la presentación visual de los datos representados por el Modelo. Decide **cómo** se mostrará la información suministrada por el controlador.

Las vistas que muestran la información suministrada por las clases controladoras pertenecientes al sistema son las enfocadas al cambio de contraseña, la autenticación del webmaster, la muestra de las urls iniciales a analizar y la muestra de los informes de reportes estadísticos.

Controlador: Responde a las acciones realizadas por el usuario a través de la interfaz, e invoca cambios en el modelo. Controla a través de funciones el paso de información entre la interfaz de usuario y el modelo. Recibe las peticiones y decide **qué** hacer o **cuándo**.

Las clases controladoras encargadas de realizar las acciones solicitadas por el usuario a través de las vistas, en la herramienta para el análisis webmétrico en la UCI son las siguientes: *RunningController*, *ConfigController*, *LoginController* y *AnalysisController*.

Este tipo de patrón proporciona la facilidad para desarrollar prototipos rápidos, además de facilitar el mantenimiento de la aplicación en caso de errores y crear independencia de funcionamiento.

2.7.2 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada a resolver un problema de diseño general en un contexto particular (Gamma , y otros, 1995). Para el diseño de la herramienta de análisis webmétrico se tienen en cuenta los patrones GRASP (*General Responsibility Assignment Software Patterns* por sus siglas en inglés) y GOF (*Gang of Four* por sus siglas en inglés), su traducción en español: Patrones de Software para la Asignación General de Responsabilidad y Banda de los Cuatro respectivamente. A continuación se muestra una selección de estos patrones:

Experto: este principio básico suele utilizarse en el diseño orientado a objetos, donde siempre se le asigna una responsabilidad al experto en información, es decir, la clase que tiene la información para llevar a cabo la funcionalidad. En este caso las clases expertas serían las clases controladoras: *RunningController*, *ConfigController*, *AnalysisController* y *LoginController*. Por ejemplo esta última clase, funciona como experta en información ya que es la encargada de llevar a cabo el trabajo con el usuario, a la hora de la autenticación del mismo.

Alta Cohesión: la cohesión es una medida de la fuerza con la que se relacionan los objetos y el grado de focalización de las responsabilidades de un elemento. Este elemento puede ser una clase o un subsistema, el cual debe tener poco trabajo y sus responsabilidades deben de estar altamente relacionadas. En la herramienta de análisis webmétrico, las clases controladoras son de alta cohesión ya que las mismas contienen funciones relacionadas entre ellas, teniendo un propósito único, donde se encargan de controlar las acciones de la entidad a la que pertenecen. Estas clases controladoras pertenecen a la capa Controlador dentro de la arquitectura Modelo-Vista-Controlador, esto permite que el software sea flexible a cambios.

Controlador: este patrón se pone de manifiesto en todo el sistema, puesto que cada acción que realiza el usuario, es redirigida a una clase controladora que ejecuta las operaciones solicitadas. Por ejemplo, la clase *RunningController*, es la encargada de la ejecución de las operaciones de todo el flujo de generación de reportes estadísticos.

Decorador: este patrón se pone de manifiesto cuando las vistas son separadas, formando una plantilla base y varias plantillas que heredan de esta. La plantilla base es la que contiene el código HTML global y

común de todas las páginas de la aplicación, para no tener que repetirlo una y otra vez en a una de las páginas. Esto sería entonces una implementación del patrón Decorador.

2.8 Diagrama de clases del diseño

El diagrama de clases del diseño está conformado por las clases pertenecientes a una aplicación web determinada, las mismas son representadas en forma de estereotipos, así como las relaciones existentes entre ellas.

Algunos de los estereotipos más comunes que se pueden asociar a las clases y las relaciones entre ellas son (Colectivo de profesores, 2010):

- *Server Page*: Representa una página Web que tiene scripts ejecutados por el servidor. Estos scripts interactúan con los recursos que se encuentran al alcance del servidor. Sólo puede mantener relaciones con objetos que se encuentren en el servidor.
- *Client Page*: Representan páginas que son dibujadas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado.
- *Form*: Representa una colección de campos de entrada que forman parte con una página del lado cliente (*Client Page*). Tiene una correspondencia directa con la etiqueta <FORM> de XHTML.
- *Link*: Representa un apuntador desde una “*client page*” hacia una “*client page*” o “*server page*”. Corresponde directamente con una etiqueta <a> (ancla) de HTML.
- *Submit*: Esta relación siempre se da entre una “*form*” y una “*server page*”, por supuesto, la “*server page*” procesa los datos que la “*form*” le envía (*submits*).
- *Build*: Sirve para identificar cuales “*server page*” son responsables de la creación de una “*client page*”.
- *Redirect*: Esta es también una relación unidireccional que indica que una página Web redirige hacia otra.

A continuación, en la **Figura 3 y 4** se muestra la representación de los diagramas de clases del diseño de algunos casos de uso críticos dentro del sistema. El resto de los diagramas de clases del diseño pertenecientes a los demás casos de uso pueden ser visualizados en el Anexo 2:

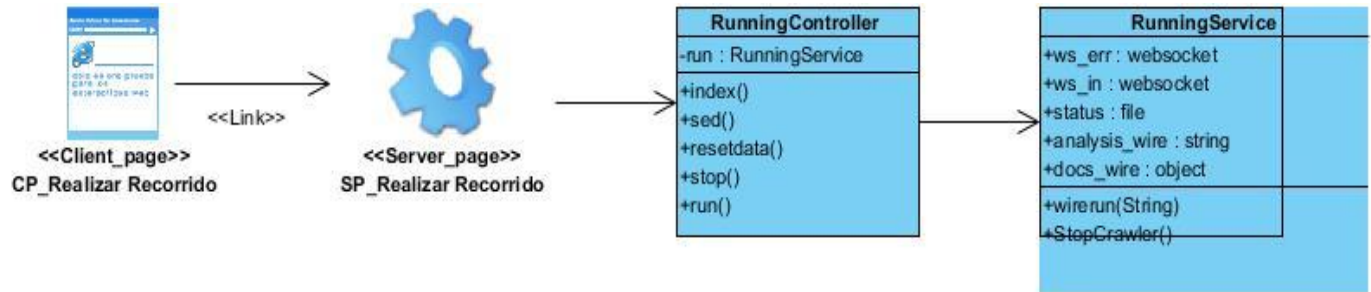


Figura 3. Diagrama de clases del diseño con estereotipos web: CU Realizar Recorrido.

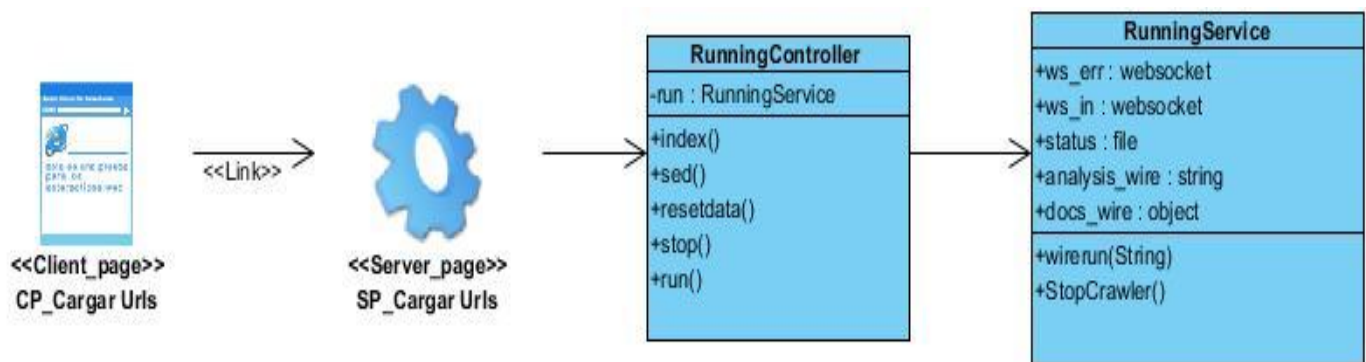


Figura 4. Diagrama de clases del diseño con estereotipos web: CU Cargar URL.

2.9 Diagrama de interacción del diseño

Los diagramas de interacción muestran cómo interactúan los objetos. Estos objetos interactúan para realizar colectivamente los servicios ofrecidos por las aplicaciones. En esencia, la misión de los diagramas de interacción es localizar el comportamiento de los objetos. Existen dos tipos de clasificaciones de diagramas de interacción: diagrama de colaboración y diagrama de secuencia.

En este caso para representar la interacción entre los objetos de la herramienta webmétrica, se tomó la decisión de utilizar el diagrama de secuencia, debido a que muestra la secuencia cronológica de mensajes entre objetos durante un escenario completo. A continuación, en la **Figura 5 y 6** se muestra el diagrama de secuencia de algunos casos de uso críticos dentro del sistema:

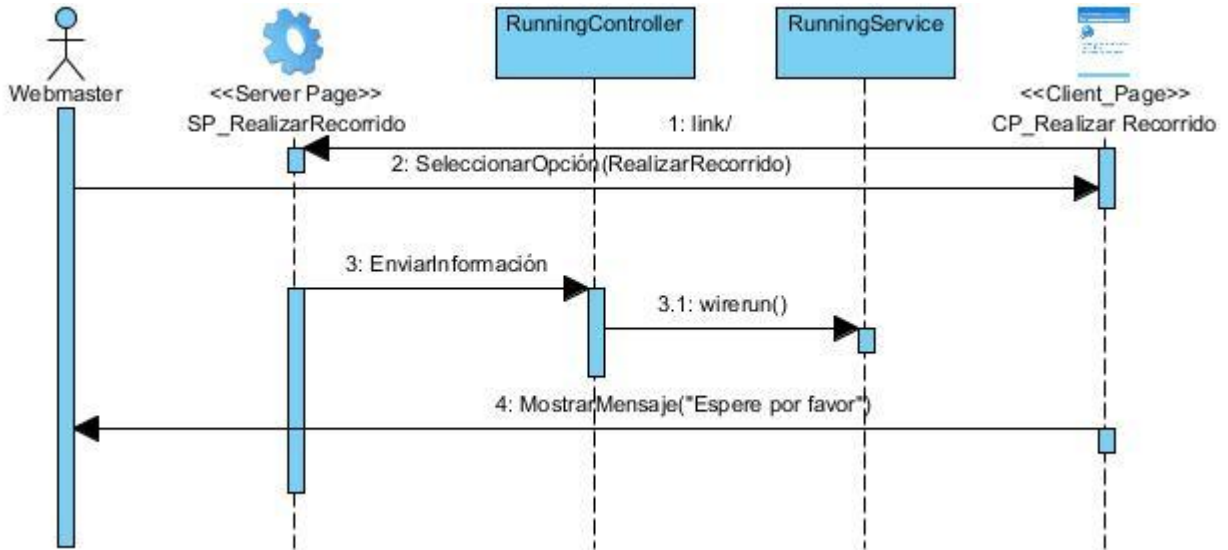


Figura 5. Diagrama de secuencia: CU Realizar Recorrido.

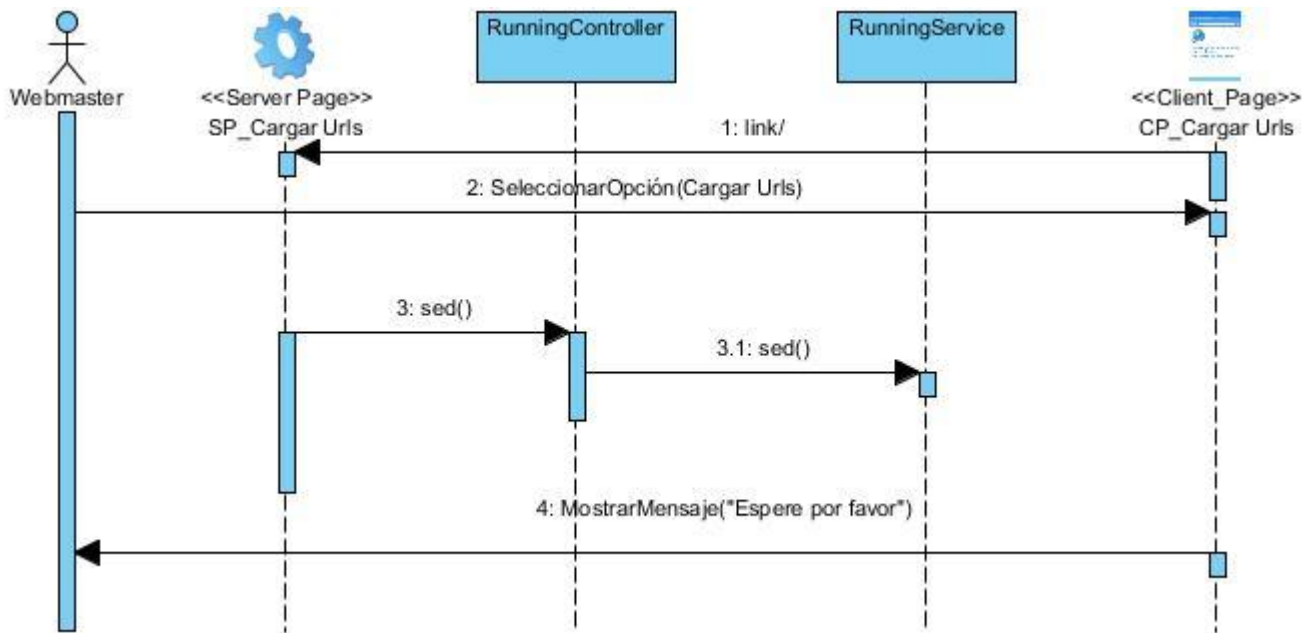


Figura 6. Diagrama de secuencia: CU Cargar URLs.

2.10 Modelo de datos

El modelo de datos tiene como objetivo definir formalmente las estructuras permitidas en la representación de datos de un sistema informático. Funcionalmente, debe captar tanto las propiedades estáticas como las dinámicas. En el caso de la herramienta para el análisis webmétrico en la UCI, se realizó una pequeña

base de datos, contenedora de una única tabla, la cual contiene los datos del usuario que se autentique en el sistema. Con esos datos guardados, se pueden realizar acciones como cambiar la contraseña del usuario, permitiendo así dar un poco de seguridad al sistema en caso que la misma sea robada. A continuación se muestra en la **Figura 7**, el modelo que representa los datos de la base de datos de la herramienta para análisis webmétrico.

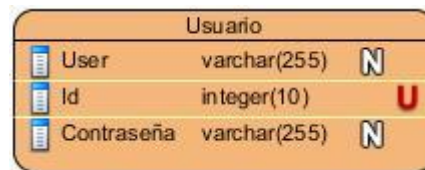


Figura 7. Modelo de datos

Conclusiones parciales del capítulo

Con el estudio de las características de la herramienta de análisis webmétrico y la realización del diseño de la misma, se tuvieron en cuenta varios aspectos, tales como: la idea de implementar una interfaz web, a través de la cual el webmaster pudiera interactuar fácilmente con la herramienta de análisis webmétrico. Además, la descripción detallada de los casos de uso, dió una visión de cómo deberán estar organizadas las funciones que se realizan en el sistema y la forma en la que el webmaster deberá interactuar con las mismas.

3 Capítulo III. Vista previa de la Herramienta para Análisis Webmétrico

3.1 Introducción

En el presente capítulo se muestran las pantallas principales de la herramienta de análisis webmétrico, logrando así tener una breve visión de lo que se va a tener en la entrega del producto o despliegue del mismo. También se dará a conocer el estándar de codificación que ha servido de guía para la implementación de la interfaz con la que va a interactuar el webmaster. Además se mostrará el diagrama de componentes, el modelo de despliegue y los resultados de las pruebas realizadas a la herramienta webmétrica, con motivo de realizar una entrega del producto de forma satisfactoria.

3.2 Modelo de despliegue

El modelo de despliegue permite describir la arquitectura física del sistema durante la ejecución, en términos de procesadores, dispositivos y componentes de software. Describe la tipología del sistema, tal como la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. En el mismo los nodos representan objetos físicos que existen en tiempo de ejecución y que representan algún tipo de recurso computacional. Estos recursos pueden ser impresoras, computadoras, dispositivos de comunicación, lectoras de códigos de barras entre otros dispositivos. La relación que existe entre los nodos se denomina relación de asociación, las cuales indican algún tipo de ruta de comunicación a través de un protocolo o red (Guerrero, 2010).

El diagrama de despliegue propuesto está conformado por dos nodos, mostrando cómo será desplegada la herramienta para el análisis webmétrico en la UCI. La PC_Cliente representa la estación de trabajo desde donde accederá el webmaster a través del navegador utilizando protocolo HTTPS para conectarse al servidor donde estará desplegada la herramienta webmétrica. La herramienta realizará el recorrido de la web desde el servidor y brindará al webmaster la información que este necesite a través de reportes estadísticos solicitados por él.

A continuación, en la **Figura 8** se muestra una representación del diagrama de despliegue perteneciente a la herramienta para el análisis webmétrico en la UCI:



Figura 8. Diagrama de despliegue de la herramienta webmétrica.

3.3 Diagrama de componentes

El diagrama de componentes es la representación de un sistema de software dividido en componentes y la dependencia existente entre estos. Los mismos son equivalentes a librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema; modelando de esa forma su vista estática y dinámica.

En un diagrama de componentes se tiene la posibilidad del agrupamiento de estos por paquetes, cuando se sienta la necesidad de mostrar que ciertas clases son parte de un subsistema en particular. En la representación de los componentes de la herramienta webmétrica se muestran las interfaces, clases de servicio y controladoras que forman parte de la misma, estableciendo además las dependencias que existen en cada una de ellas. A continuación, en la **Figura 9** se encuentra disponible el diagrama de componentes de dicha herramienta:

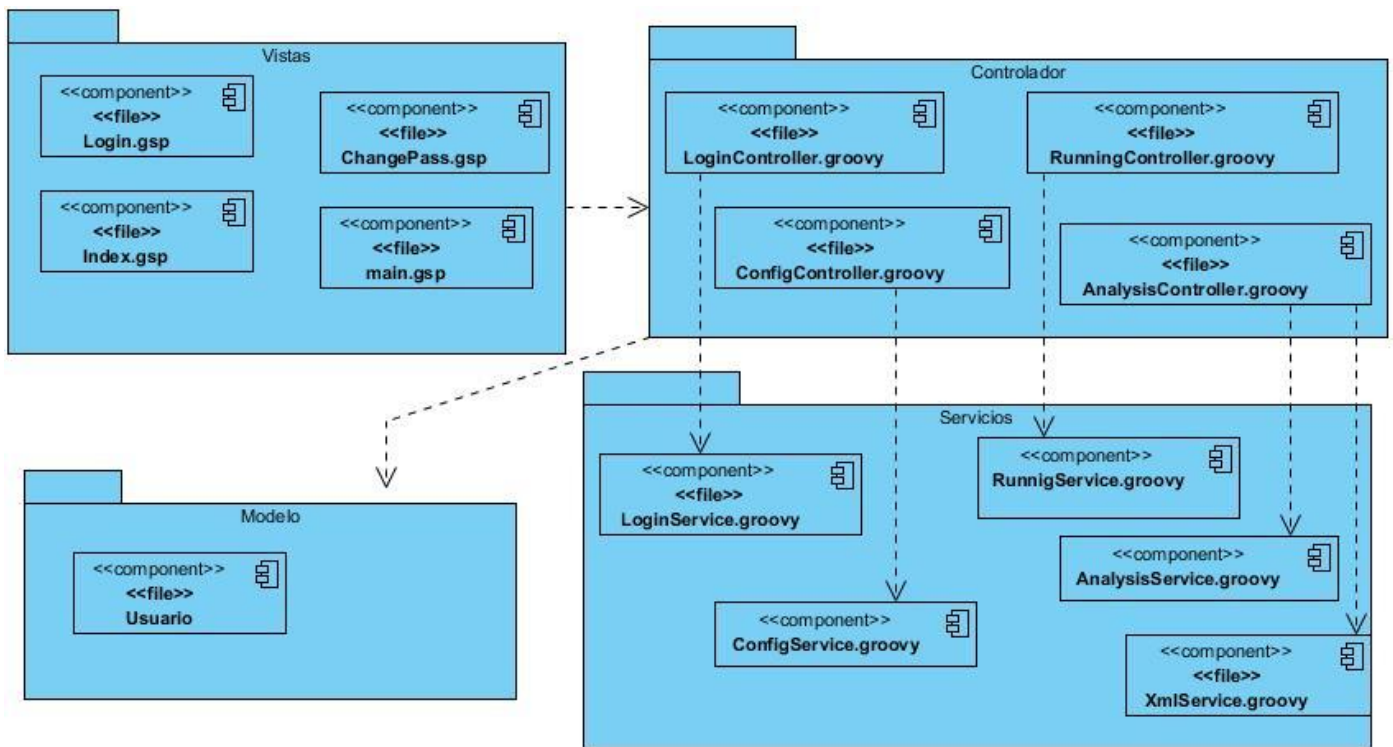


Figura 9. Diagrama de componentes

3.4 Estándar de codificación

Los estándares de codificación son una documentación valiosa en cualquier sistema que se esté desarrollando o vaya a ser desarrollado, ya que permite el mismo tenga una arquitectura y un estilo consistente. Además, hace el código fuente del sistema fácil de entender y mantener por otros desarrolladores. Es por ello que un estándar de codificación debe incorporar las mejores prácticas de codificación y permitir que los programas se acerquen lo mejor posible al lenguaje natural.

En la implementación de la herramienta webmétrica se tuvo en cuenta la utilización de algunas reglas de codificación para el lenguaje de programación Groovy. Todo esto proporciona que el código fuente de dicha herramienta sea fácil de entender por otros desarrolladores en caso que el mismo fuera a ser modificado. A continuación se muestran varios ejemplos de la utilización de estas reglas en la implementación.

Definición de clases y métodos

En la definición de clases y métodos, el lenguaje groovy considera los mismos por defecto como públicos, por lo que la palabra clave *public*, no es necesaria usarla. En caso de no ser pública alguna clase o método, entonces si se hace necesario especificar el estado de visibilidad. Un ejemplo claro de la definición de clases y métodos en la implementación de la herramienta webmétrica se muestra a continuación:

```
class RunningController {}
```

Nombres

Para nombrar las clases se utiliza la notación UpperCamelCase, la misma se evidencia cuando la primera letra de cada una de las palabras es mayúscula, por ejemplo: **RunningService**. Los nombres de las funciones dentro de las clases en minúscula, se encuentran en minúscula, para brindarle al usuario una factible visibilidad en la barra de direcciones de los navegadores, debido a la utilización de los mismos en las urls, por ejemplo: **urls()**, **run()** y **sed()**.

Formato

En el código se utiliza cuatro espacios para la sangría, esto proporciona la facilidad de lectura del mismo y minimiza los problemas de interpretación de código con otras herramientas. Se tiene definido, añadir un solo espacio alrededor de los operadores (==, &&,...) y usar llaves para indicar el control de la estructura sin tener en cuenta el número de declaraciones que el grupo pueda contener.

Ejemplo:

```
def urls(){
```

```
....def url = config.loadUrls()
....if (url.size() > 0) {
.....flash.message = "Cargadas correctamente "+url.size () +" urls"
    } else {
.....flash.message = "El archivo no contenía urls válidas"
    }
.... [urls:url]
}
```

Novedades en la implementación

Teniendo en cuenta que en el desarrollo de la herramienta webmétrica se hizo necesaria la utilización del *webcrawler* WIRE para realizar el recorrido de los sitios web en la UCI, el mismo tenía el inconveniente de no analizar los sitios que se encontraban bajo el protocolo https. En la Universidad de las Ciencias Informáticas, todos los sitios web se encuentran funcionando bajo el protocolo https. Por la posibilidad que brinda el sistema WIRE de estar basado en los preceptos del software libre, el equipo de desarrollo tuvo la necesidad de modificar el código fuente de dicho sistema e implementó una función para que el mismo pudiera analizar los sitios que se encontraran bajo el protocolo https.

3.5 Estudio realizado a los sitios web en la UCI con la herramienta webmétrica

Con la herramienta para el análisis webmétrico en la Universidad de las Ciencias Informáticas mostrada en la **Figura 10**, se han realizado varios recorridos a los sitios web de la universidad. Dichos recorridos han servido de pruebas para comprobar el funcionamiento correcto de la misma. A continuación se mostrarán ejemplos de las estadísticas obtenidas en los recorridos.

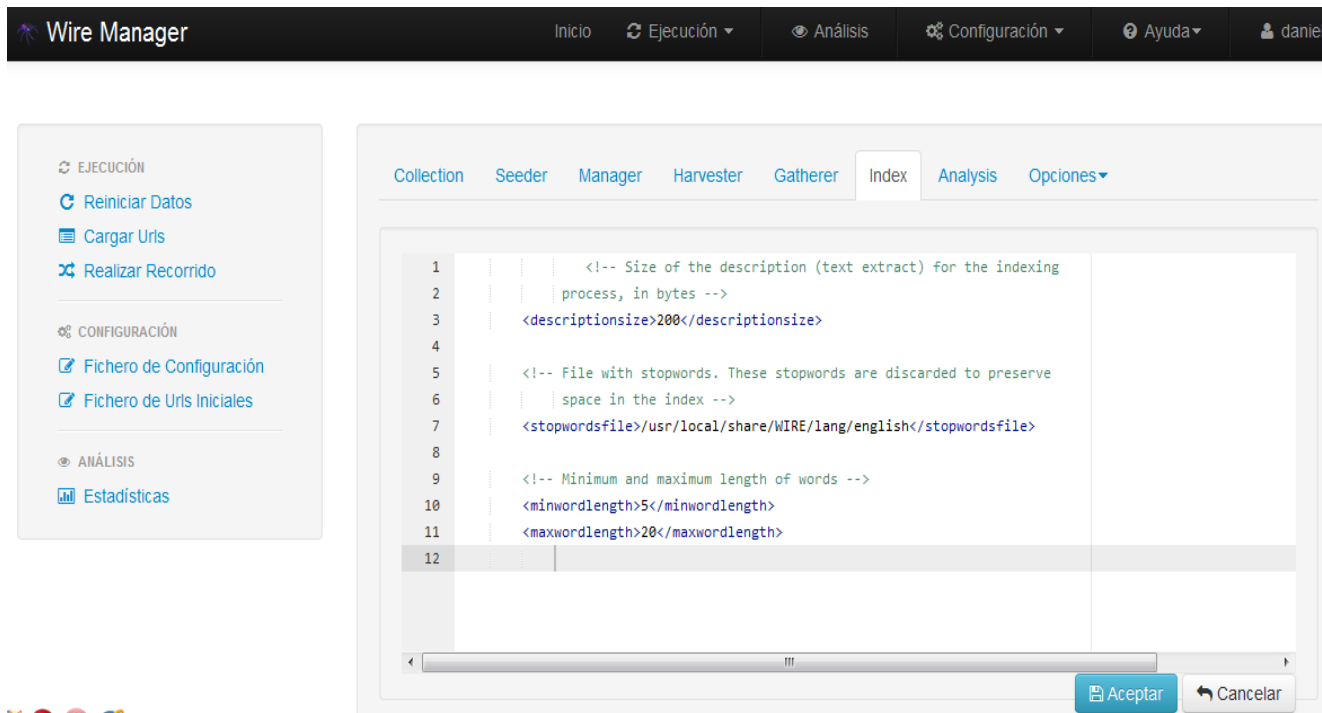


Figura 10. Fichero de configuración.

```

HOST=enoc
BASE=/var/tmp/wiredata
CONF=/usr/local/etc/wire.conf
    
```

Number of sites ok	98
Average internal links	8,291.16
Average pages per site	1,504.16
Average static pages per site	983.91
Average dynamic pages per site	520.26
Number of sites with valid page age	65
Average of age of oldest page in months	31.06
Average of age of average page in months	19.02
Average site size in MB	13.81
Average of age of newest page in months	10.11
Average site max depth	6.24
Average in-degree	4.79
Average out-degree	4.79

Table 1: Site summary

Figura 11. Estadísticas de los sitios.

Filename extension	Number of links found	Percent
webm	969	74.54%
rmvb	168	12.92%
mp4	76	5.85%
wmv	42	3.23%
flv	37	2.85%
ogv	7	0.54%
mpg	1	0.08%

Table 5: Extensions video

Figura 12. Estadísticas de extensiones multimedia.

```
HOST=enoc  
BASE=/var/tmp/wiredata  
CONF=/usr/local/etc/wire.conf
```

Total number of site names known	265
Sites with at least one page ok	98
Sites without in links (but at least one page ok)	2
Sites without out links (but at least one page ok)	33
Size of largest SCC	61
SCC-id of largest SCC	170
Number of SCCs with one site only (singletons)	37

Table 1: Stats

Figura 13. Estadísticas de los enlaces.

```
HOST=enoc  
BASE=/var/tmp/wiredata  
CONF=/usr/local/etc/wire.conf
```

Documents sampled	124
Documents with more than 50 words	113
Documents that were not identified	70

Table 1: Summary

Figura 14. Estadísticas de los documentos.

Document language	Number of documents	Percent
spanish	50	92.59%
english	4	7.41%
catala	0	0%
danish	0	0%
dutch	0	0%
french	0	0%
german	0	0%
greek	0	0%
italian	0	0%
norwegian	0	0%
portugues	0	0%
swedish	0	0%
turkish	0	0%

Table 2: Count

Figura 15. Estadísticas de los idiomas.

3.6 Realización de pruebas del sistema

Normalmente un software es desarrollado por seres humanos, y estos desafortunadamente no son perfectos. Por lo que en la creación de un sistema de software, además de las actividades realizadas para la implementación del mismo, se deben realizar otras actividades que corroboren y garanticen la calidad con la que este ha sido construido. La manera más adecuada en la que se puede determinar el estado de calidad de un producto es con la realización de un proceso de pruebas.

3.6.1 Niveles de Prueba

El proceso de prueba es realizado por un equipo de prueba, el cual estructura las pruebas a realizar según los objetivos propuestos y los distintos escenarios. Las mismas se distinguen por niveles, entre los que se pueden encontrar:

Unidad: Estas pruebas están enfocadas en ejecutar cada unidad mínima a ser probada, buscan que el código funcione de acuerdo con las especificaciones y sea válido. Dentro de las pruebas en este nivel se encuentran: prueba de interfaz, estructura de datos locales, condiciones límites y de caminos independientes o básicos.

Sistema: El propósito de las pruebas de sistema es ejercitar profundamente el sistema, donde se trabaja para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas. Las pruebas que se realizan en este nivel son: pruebas de recuperación, seguridad, resistencia, rendimiento y pruebas funcionales.

Integración: Es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados en unidad y construir una estructura del programa que esté de acuerdo con lo que dicta el diseño. En este nivel existen dos tipos de pruebas: prueba de integración no incremental y de integración incremental, donde en esta última utiliza como estrategias de integración: descendente y ascendente.

Aceptación: Estas pruebas son realizadas por el cliente, básicamente son pruebas funcionales realizadas al sistema completo. Las mismas, buscan una cobertura de la especificación de requisitos y son realizadas sobre el producto terminado.

Para la realización de las pruebas enfocadas a la herramienta de análisis webométrico. El equipo de trabajo decidió utilizar el nivel **Unidad**, por la necesidad de comprobar el funcionamiento del código de acuerdo a las especificaciones. También será utilizado el nivel **Sistema**, para verificar el funcionamiento del programa final, asegurando que este funcione correctamente en todo su conjunto. Y por último se utilizará el nivel **Aceptación**, donde el cliente, que en este caso serían los *webmasters*, probaría las funcionalidades del sistema completo teniendo en cuenta las especificaciones de los requisitos planteados.

3.6.2 Métodos de prueba

Las principales estrategias utilizadas para la realización de las pruebas son los métodos de caja negra y blanca. En el primero de ellos, las pruebas se realizan sobre la interfaz del software, pretendiendo probar que las funciones del mismo son operativas, produciendo un resultado correcto. Sin embargo, el método

de caja blanca, comprueba los caminos lógicos del software, proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones.

Pruebas de caja negra

Estas pruebas verifican las especificaciones funcionales del sistema y no se considera la estructura interna del programa. En este método de prueba son utilizadas las siguientes técnicas: particionamiento equivalente y análisis de valores límites.

Pruebas de caja blanca

Las técnicas que comúnmente son utilizadas en este método son: camino básico, condición, flujo de datos y bucles. Con estas pruebas el ingeniero de software puede obtener casos de prueba que garanticen se ejerciten al menos una vez todos los caminos independientes para cada módulo programado.

Para la realización de las pruebas se selecciona el método de caja negra, por lo que de las técnicas comúnmente utilizadas en este método se utilizará la técnica de partición equivalente, la cual está dirigida a una definición de casos de prueba que descubran clases de errores. El diseño de casos de prueba según esta técnica estará dividido en dos pasos:

1. Identificar las clases de equivalencia: la clase de equivalencia representa el conjunto de estados válidos y no válidos para las condiciones de entrada de un programa.
2. Identificar los casos de prueba: su objetivo es minimizar el número de casos de prueba, así cada caso de prueba debe considerar tantas condiciones de entrada como sea posible.

3.6.3 Pruebas de funcionalidades

Las pruebas de funcionalidades son realizadas con el objetivo de comprobar que el sistema cumpla con las funciones específicas para las cuales ha sido creado. Además, se evalúa el sistema tratando de encontrar las fallas que pudiera tener el mismo a la hora de interactuar con los usuarios.

Se definieron casos de prueba para los casos de uso del sistema Autenticar Usuario y Cambiar Contraseña:

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Contraseña Anterior	Campo de texto	No	Se debe especificar la contraseña perteneciente al usuario que está iniciada su sesión.
2	Contraseña Nueva	Campo de texto	No	Se debe especificar una secuencia de caracteres que sea igual a la de la variable 3.
3	Repetir Contraseña Nueva	Campo de texto	No	Se debe especificar una secuencia de caracteres que sea igual a la de la variable 2.

Tabla 5. Descripción de las variables para el caso de prueba del CU Cambiar Contraseña.

En la **Tabla 6**, las celdas que contienen los valores V, significa válido, el valor I indica inválido y el valor N/A significa que no es necesario proporcionar un valor del dato en este caso.

Escenario	Descripción	Variable 1	Variable 2	Variable 3	Respuesta del sistema	Flujo central
EC 1.1 Datos correctos	El usuario introduce todos los valores de manera correcta en el sistema.	V	V	V	El sistema verifica que los datos son correctos, cambia la contraseña del usuario en el sistema y muestra un mensaje informando: "Contraseña cambiada correctamente".	1-El usuario accede a la parte superior derecha en el ícono de usuario. 2- Selecciona la opción Cambiar Contraseña. 3-Llena todos los campos de

						manera correcta y le da clic en Aceptar o presiona la tecla Enter.
		I	V	V		
EC 1.2 Datos incorrecto.	El usuario introduce un valor incorrecto en el sistema.				El sistema verifica si los datos son correctos, de no serlo muestra el siguiente mensaje: "Contraseña anterior incorrecta".	El usuario introduce un dato incorrecto y da clic en Aceptar o presiona la tecla Enter.
		V	I	V		
					El sistema verifica si los datos son correctos, de no serlo muestra el siguiente mensaje: "Las contraseñas no coinciden".	El usuario introduce un dato incorrecto y da clic en Aceptar o presiona la tecla Enter.
		V	V	I		

					El sistema verifica si los datos son correctos, de no serlo muestra el siguiente mensaje: " Las contraseñas no coinciden".	El usuario llena todos los campos de manera correcta pero no coinciden las dos contraseñas que ingresó (variable 2 y 3) y le da clic en Aceptar o se presiona la tecla Enter.
		V	V	I		
					Si el navegador es Firefox a partir de la versión 5 el sistema muestra un mensaje que dice: "Por favor, rellene este campo" o "Please fill out this field" depende del	El usuario llena todos los campos de manera correcta pero no coinciden las dos contraseñas que ingresó (variable 2 y 3) y le da clic en Cambiar

					idioma que tenga el navegador en el caso que deje un campo vacío, en el caso de que las contraseñas no sean las mismas muestra el siguiente mensaje: "Las contraseñas no coinciden".	Contraseña o se presiona la tecla Enter.
EC 1.3 Campos vacíos	El usuario deja algún campo sin llenar	vacío	V	V	El sistema muestra un mensaje informando que hay un campo sin llenar.	El usuario inserta los datos pero deja un campo sin llenar.
		V	vacío	V	El sistema muestra un mensaje informando que hay un campo sin llenar.	El usuario inserta los datos pero deja un campo sin llenar.
		V	V	vacío	El sistema muestra un mensaje informando que hay un campo sin llenar.	El usuario inserta los datos pero deja un campo sin llenar.

Tabla 6. Caso de prueba del CU Cambiar Contraseña

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Usuario	Campo de texto	No	Se debe especificar la contraseña perteneciente al usuario que está iniciada su sesión.
2	Contraseña	Campo de texto	No	Se debe especificar una secuencia de caracteres, que exceda la cantidad de 8 caracteres.

Tabla 7. Descripción de las variables para el CU Autenticar Usuario

Escenario	Descripción	Variable 1	Variable 2	Respuestas del sistema	Flujo central
EC 1: Datos correctos	El usuario introduce los datos correctamente.	V	V	El sistema autentica al usuario.	El usuario llena los campos de forma correcta y da clic en el botón Aceptar.
		user	pass		
EC 2: Campos vacíos	El Usuario deja uno o los dos campos vacíos.	I	I	El sistema muestra un mensaje: "Debe llenar los campos vacíos".	El usuario llena los campos y deja un de ellos vacío y presiona el botón Aceptar.
		vacío	vacío		
		I	V		
		vacío	pass		
		V	I		
EC 3: Datos incorrectos	El Usuario introduce algún valor incorrecto.	I	V	El sistema muestra un mensaje: "Datos incorrectos"	El usuario introducen un dato
		pepe	pass		

		V	I		incorrecto y presiona el botón Aceptar.
		user	Pepe1		
		I	I		
		pepe	Pepe1		
EC 4: Sesión iniciada.	Un usuario intenta iniciar sesión desde otro cliente, encontrándose ya iniciada la sesión previamente.	V	V	El sistema no permite la entrada de otro usuario, si la sesión se encuentra iniciada.	1-El usuario introduce el nombre de usuario y la contraseña, luego presiona el botón aceptar. 2-Si la sesión se encuentra iniciada, el sistema no permite la entrada de otro usuario y muestra el mensaje "El usuario ya tiene una sesión iniciada".
		user	pass		

Tabla 8. Caso de prueba del CU Autenticar Usuario

Conclusiones parciales del capítulo

En el presente capítulo, con la realización del diagrama de componentes se pudo obtener una visión clara acerca de las clases que conforman la aplicación, así como la relación existente entre ellas, proporcionando de esta forma, una organización estructural del sistema. Además, con la realización del diagrama de despliegue, se obtuvo una definición concreta de los dispositivos de hardware necesarios que intervendrán en el proceso de despliegue de la herramienta.

El estudio del estándar de codificación, conllevó al uso de una buena práctica de codificación, acercándose lo mejor posible al lenguaje natural; para que el código fuente del mismo sea entendible y manejable por futuros desarrolladores, que pretendan realizar modificaciones en él u obtener algunas de las funcionalidades que este posee.

La realización de las pruebas al sistema, posibilitaron la captura de errores existentes en la interfaz, permitiendo esto el arreglo de los mismos para proporcionarle al cliente una aplicación con todas sus funcionalidades trabajando correctamente.

4 Conclusiones generales

- La inexistencia de herramientas que permitan realizar estudios webmétricos en la UCI, provocó la necesidad de implementar una herramienta para análisis webmétrico que brinde estadísticas de los sitios web de la UCI.
- La identificación de los requerimientos del sistema y la generación de artefactos y esquemas, permitió definir la propuesta de solución, posibilitando a su vez obtener una visión del funcionamiento del sistema y claridad a la hora de implementar la herramienta webmétrica.
- La herramienta para análisis webmétrico, ayuda a los *webmasters* en la toma de decisiones con respecto a los sitios web existentes en la UCI, a través de las estadísticas de los reportes generados.
- Las pruebas realizadas a la herramienta para análisis webmétrico, permitieron la validación del comportamiento de manera correcta del sistema y detectar errores que fueron corregidos.

Por tanto, se puede concluir que el cumplimiento de los objetivos propuestos para el desarrollo del presente trabajo de diploma, fue logrado satisfactoriamente.

5 Recomendaciones

Al haber finalizado el presente trabajo de diploma se recomienda:

- La realización de edición del fichero de configuración del WIRE, a partir de formularios.
- Implementar el soporte multiusuario para el sistema WIRE.

6 Referencias Bibliográficas

Björneborn, Lennart. 2004. *Small-world link structures across an academic web space.* s.l. : Department of Information Studies, Royal School of Library and Information Science., 2004.

Casanova Hernández, Ing. Madisleidy y Rodríguez Martínez, Ing. Yenisleydis. 2012. *WEBSITEANALYZER: HERRAMIENTA WEMÉTRICA LIBRE PARA EL ANÁLISIS PERSONALIZADO DE SITIOS WEBS DE LA UCI.* La Habana : s.n., 2012.

Colectivo de profesores, UCI. 2010. *Diseño Web.* Ingeniería de Software, UCI. La Habana : s.n., 2010.

Cronin, Blaise y Mckim, Geoffrey. 1999. *Internet.In: A Informação: tendências para o novo milênio.* Brasília : Brasília, 1999. IBICT.

David Quiñónez, Juan. 2012. 10 espectaculares herramientas de estadísticas y análisis para tu sitio web. [En línea] Diego Polo, Juan, 3 de Abril de 2012. [Citado el: 10 de Diciembre de 2013.] <http://www.whatsnew.com/2012/04/03/10-espectaculares-herramientas-de-estadisticas-y-analisis-para-tu-sitio-web/>.

Fernández Herrera, Danny y Blanco Sanchez, Fidel Enrique. 2010. *Los indicadores webmétricos y su influencia en la calidad de los sitios web.* 2010.

Guerrero, Luis A. 2010. *Taller de UML.* 2010.

Larman, Graig. 2002. *UML y Patrones.* s.l. : PEARSON EDUCACION, 2002. ISBN 9788420534381.

Manuel, Blázquez Ochando Prof. Dr. 2012. Sistemas de Recuperación e Internet. [En línea] 26 de noviembre de 2012. [Citado el: 23 de noviembre de 2013.] <http://ccdoc-sistemasrecuperacioninternet.blogspot.com/2012/11/webmetria-y-analisis-de-paginas-web.html>.

Mondelo Hernández, Yonny. 2010. *Generador de estudios webmétricos (GEWeb v 1.0).* 2010.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. *El Lenguaje Unificado de Modelado. Manual de Referencia.* s.l. : Wesley, Addison. págs. 24-55. ISBN 978-84-7829-087-1.

7 Bibliografía Consultada

- Gamma , Erich, y otros. 1995.** *Design Patterns: Elements of Reusable Object-Oriented Software.* s.l. : Addison Wesley, 1995. ISBN:0-201-63361-2.
- Garzás, Javier.** *Sistemas de Información. Tema 10. Diagramas de Interacción.*
- González Barahona, Jesús M.** Introducción al Software Libre. [En línea] [Citado el: 11 de Diciembre de 2013.] <http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/sec-ide.html>.
- Group, Miniwatts Marketing. 2013.** Internet World Stats. [En línea] 2013. <http://www.internetworldstats.com/stats.htm>.
- Hernández Rodríguez, Nodelvis y Acosta Perez, Tomás. 2012.** *Módulo de reportes webmétricos para el motor de búsqueda Orión.* La Habana : s.n., 2012.
- Jeon, Sangjun, y otros. 2011.** *A recovery method of deleted record for SQLite database.* s.l. : Springer-Verlag London Limited, 2011. s00779-011-0428-7.
- JetBrains. 2000-2014.** IntelliJIDEA Features. [En línea] 2000-2014. [Citado el: 06 de Mayo de 2014.] <http://www.jetbrains.com/idea/features/>.
- Judd, Christopher M, Faisal Nusairat, Joseph y Shingler, James. 2008.** *Beginning Groovy and Grails: From Novice to Professional.* s.l. : Apress, 2008. págs. 1- 45. ISBN-13 (pbk): 978-1-4302-1045-0.
- Koenig, Dierk, y otros. 2006.** *Groovy in Action.* s.l. : Manning, 2006. ISBN 1-932394-84-2.
- Larman, Graig. 2002.** *UML y Patronos.* s.l. : PEARSON EDUCACION, 2002. ISBN 9788420534381.
- Lopez, Sergio F, y otros. 2007.** *Design and description of a clasification system framework for easier reuse.* s.l. : IEEE Computer Society, 2007.
- Pivotal. 2013.** Grails. [En línea] 2013. <https://grails.org>.
- Pressman, Roger S. 2005.** *Ingeniería de software. Un enfoque de práctico.* s.l. : Mc Graw Hill, 2005. ISBN: 97-0105-473-3.
- Profesores, UCI. 2011.** *Introduccion_a_la_Disciplina_de_Requisitos.* 2011.
- Rocher, Graeme. 2009.** *Manual de desarrollo web con Grails.* 2009. pág. 6. ISBN 978-84613-2651.
- Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* s.l. : Wesley, Addison. págs. 24-55. ISBN 978-84-7829-087-1.
- Urgellés Lafita, Reysel. 2007.** *Modelo de Prueba al Producto Multimedia: Educación Física Cubana.* UCI. La Habana : s.n., 2007. págs. 10-18, Trabajo de Diploma.

8 Glosario de Términos

Spider o Crawlers: son estimados arañas, que se encargan de descubrir nuevos contenidos en la Web, analizando los documentos hipertextos contenidos en ella de forma automática para luego estos ser procesados.

Interfaz: es el conjunto de elementos de la pantalla que permiten al usuario realizar acciones sobre el Sitio Web que está visitando.

Casos de Uso (CU): es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

URL: (*Uniform Resource Locator* por sus siglas en inglés) es un localizador uniforme de recurso. Es una secuencia de caracteres de acuerdo a un formato estándar, usado para nombrar recursos, documentos e imágenes en Internet, por su localización.

Internet: red de ordenadores de ámbito global y públicamente accesible, que proporciona gran cantidad de servicios de comunicación de diferentes tipos, incluyendo la *World Wide Web*, correo electrónico y muchos otros.

Hipertexto: tecnología que organiza una base de información en bloques distintos de contenidos, conectados a través de una serie de enlaces, cuya activación o selección provoca la recuperación de información.

SQL: *Structured Query Language* por sus siglas en inglés. Es un lenguaje de acceso a base de datos que permite efectuar consultas, con el fin de recuperar información de forma sencilla en las bases de datos, así como realizar cambios en ellas.

HTTPS: (*Hypertext Transfer Protocol Secure* por sus siglas en inglés). *Protocolo seguro de transferencia de hipertexto.*

DNS: (*Domain Name System* por sus siglas en inglés). Es un sistema de nombres de dominios, es decir, es un sistema para nomenclatura jerárquica para computadoras.

RF: estas siglas hacen referencia a los Requisitos Funcionales.

RNF: estas siglas hacen referencia a los Requisitos No Funcionales

9 Anexos

9.1 Anexo 1. Descripción de los casos de uso del sistema

9.1.1 CU Reiniciar datos

Caso de Uso	Reiniciar datos.
Actores	Webmaster.
Resumen	El caso de uso inicia cuando el usuario accede a la opción “Reiniciar Datos” del menú “Ejecución” en la interfaz, da clic en dicha opción y el sistema comienza a reiniciar los datos encontrados anteriormente.
Complejidad	Media
Prioridad	Alta
Precondiciones	
Poscondiciones	Los datos encontrados anteriormente deben ser borrados para posteriormente volver a realizar el recorrido de las páginas.
Prototipo de Interfaz	



Flujo de eventos		
Flujo básico reiniciar datos		
	Actor	Sistema
1	El usuario accede al menú "Ejecución" en la interfaz, da clic en la opción "Reiniciar datos".	
2		El sistema borra los datos de las urls cargadas anteriormente.
3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.1.2 CU Cambiar Contraseña

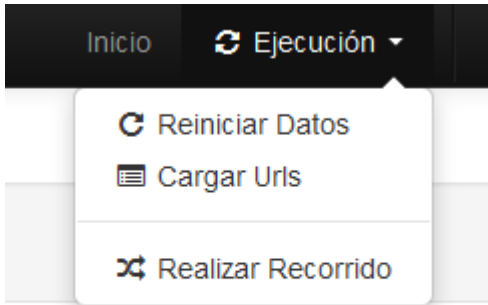
Caso de Uso	Cambiar Contraseña.
Actores	Webmaster.

Resumen	El caso de uso inicia cuando el usuario accede a su perfil y elige la opción "Cambiar Contraseña", inserta los datos en los campos requeridos y el sistema le cambia la contraseña.	
Complejidad	Media	
Prioridad	Media	
Precondiciones		
Poscondiciones		
Prototipo de Interfaz		
 <p>El prototipo de interfaz muestra un formulario con el título "Cambiar Contraseña". El formulario contiene tres campos de texto: "Contraseña Anterior", "Contraseña Nueva" y "Repetir Contraseña Anterior". Debajo de los campos hay un botón "Aceptar" con una flecha a la izquierda.</p>		
Flujo de eventos		
Flujo básico cambiar contraseña		
	Actor	Sistema

1	El usuario elige la opción “Cambiar Contraseña” y llena los campos requeridos: Contraseña Anterior , Contraseña Nueva y Repetir Contraseña Anterior , luego presiona el botón Aceptar .	
2		El sistema verifica que los datos estén correctos y cambia la contraseña.
3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.1.3 CU Realizar recorrido

Caso de Uso	Realizar recorrido.
Actores	Webmaster.
Resumen	El caso de uso inicia cuando el usuario accede a la opción “Realizar recorrido” del menú “Ejecución” en la interfaz, da clic en dicha opción y el sistema comienza a realizar el análisis de las Urls cargadas.

Complejidad	Media	
Prioridad	Alta	
Precondiciones	Se deben haber cargado las Urls en el sistema.	
Poscondiciones		
Prototipo de Interfaz		
		
Flujo de eventos		
Flujo básico realizar recorrido		
	Actor	Sistema
1	El usuario accede al menú “Ejecución” en la interfaz, da clic en la opción “Realizar recorrido”.	
2		El sistema comienza a hacer el análisis de las Urls cargadas.

3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.1.4 CU Generar reporte de extensiones

Caso de Uso	Generar reporte de extensiones.
Actores	Webmaster.
Resumen	El caso de uso inicia cuando el usuario accede a la opción "Extensiones" del menú "Análisis" en la interfaz, da clic en dicha opción y el sistema genera un reporte estadístico de la cantidad y porcentaje de las extensiones u objetos encontrados en las páginas recorridas.
Complejidad	Media
Prioridad	Alta
Precondiciones	
Poscondiciones	

Flujo de eventos		
Flujo básico generar reporte de extensiones		
	Actor	Sistema
1	El usuario accede al menú "Análisis" en la interfaz, da clic en la opción "Extensiones".	
2		El sistema genera un reporte estadístico con la cantidad y porcentaje de las extensiones u objetos encontrados en las páginas recorridas.
3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.1.5 CU Generar reporte de recorrido

Caso de Uso	Generar reporte de recorrido.
--------------------	-------------------------------

Actores	Webmaster.	
Resumen	El caso de uso inicia cuando el usuario accede a la opción "Recorrido" del menú "Análisis" en la interfaz, da clic en dicha opción y el sistema genera un reporte estadístico acerca del recorrido realizado a la web donde da información acerca de la cantidad de sitios activos, número de documentos y número de documentos que se encuentran en buen estado de enlace.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo básico generar reporte de recorrido		
	Actor	Sistema
1	El usuario accede al menú "Análisis" en la interfaz, da clic en la opción "Recorrido".	
2		El sistema genera un reporte estadístico de la cantidad de sitios, documentos y documentos en buen estado de enlace encontrados.
3		Termina el CU.

Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.1.6 CU Generar reporte de idiomas

Caso de Uso	Generar reporte de idiomas.
Actores	Webmaster.
Resumen	El caso de uso inicia cuando el usuario accede a la opción "Idiomas" del menú "Análisis" en la interfaz, da clic en dicha opción y el sistema genera un reporte estadístico con la cantidad de documentos según el tipo de lenguaje y el porcentaje que representa.
Complejidad	Media
Prioridad	Alta
Precondiciones	
Poscondiciones	

Flujo de eventos		
Flujo básico generar reporte de idiomas		
	Actor	Sistema
1	El usuario accede al menú "Análisis" en la interfaz, da clic en la opción "Idiomas".	
2		El sistema genera un reporte estadístico con la cantidad de documentos según el tipo de lenguaje y el porcentaje que representa.
3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.1.7 CU Generar reporte de sitios

Caso de Uso	Generar reporte de sitios.	
Actores	Webmaster.	
Resumen	El caso de uso inicia cuando el usuario accede a la opción "Sitios" del menú "Análisis" en la interfaz, da clic en dicha opción y el sistema genera un reporte estadístico con el ranking de los sitios según la cantidad de documentos, vínculos internos y <i>pagerank</i> .	
Complejidad	Media	
Prioridad	Alta	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo básico generar reporte de sitios		
	Actor	Sistema
1	El usuario accede al menú "Análisis" en la interfaz, da clic en la opción "Sitios".	
2		El sistema genera un reporte estadístico con el ranking de los sitios según la cantidad de documentos, vínculos internos y <i>pagerank</i> .

3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.1.8 CU Generar reporte de enlaces

Caso de Uso	Generar reporte de enlaces.
Actores	Webmaster.
Resumen	El caso de uso inicia cuando el usuario accede a la opción "Enlaces" del menú "Análisis" en la interfaz, da clic en dicha opción y el sistema genera un reporte estadístico con la cantidad de sitios con al menos una página en buen estado y el número de nombres de sitios conocidos.
Complejidad	Media
Prioridad	Alta
Precondiciones	

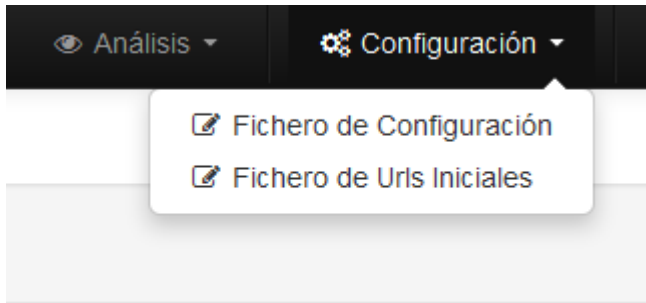
Poscondiciones		
Flujo de eventos		
Flujo básico generar reporte de enlaces		
	Actor	Sistema
1	El usuario accede al menú "Análisis" en la interfaz, da clic en la opción "Total de enlaces por páginas".	
2		El sistema genera un reporte estadístico con la cantidad de sitios con al menos una página en buen estado y el número de nombres de sitios conocidos.
3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.1.9 CU Generar reporte de documentos

Caso de Uso	Generar reporte de documentos.	
Actores	Webmaster.	
Resumen	El caso de uso inicia cuando el usuario accede a la opción "Documentos" del menú "Análisis" en la interfaz, da clic en dicha opción y el sistema genera un reporte estadístico con la cantidad de documentos en los sitios según el tipo, la edad y el estado del protocolo Http, así como el porcentaje que representan dichas cantidades.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo básico generar reporte de documentos		
	Actor	Sistema
1	El usuario accede al menú "Análisis" en la interfaz, da clic en la opción "Documentos".	
2		El sistema genera un reporte estadístico con la cantidad de

		documentos en los sitios según el tipo, la edad y el estado del protocolo Http, así como el porcentaje que representan dichas cantidades.
3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

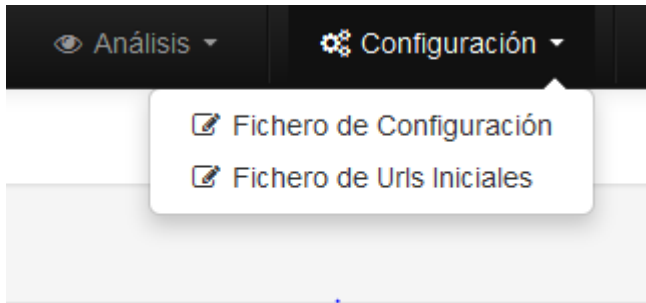
9.1.10 CU Mostrar fichero de configuración

Caso de Uso	Mostrar fichero de configuración.
Actores	Webmaster.
Resumen	El caso de uso inicia cuando el usuario accede a la opción “Fichero de Configuración” del menú “Configuración” en la interfaz, da clic en dicha opción y el sistema muestra el fichero XML de configuración de la herramienta WIRE.
Complejidad	Media
Prioridad	Media
Precondiciones	
Poscondiciones	
Prototipo de Interfaz	
 A screenshot of a software interface showing a dark navigation bar with two menu items: 'Análisis' (with an eye icon) and 'Configuración' (with a gear icon). The 'Configuración' menu is open, displaying a white dropdown list with two options: 'Fichero de Configuración' and 'Fichero de Urls Iniciales', each preceded by a document icon.	
Flujo de eventos	
Flujo básico mostrar fichero de configuración	

	Actor	Sistema
1	El usuario accede a la opción "Fichero de Configuración" del menú "Configuración" en la interfaz y da clic en dicha opción.	
2		El sistema muestra el fichero XML de configuración de la herramienta WIRE.
3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.1.11 CU Mostrar fichero de urls iniciales

Caso de Uso	Mostrar fichero de urls iniciales.
Actores	Webmaster.
Resumen	El caso de uso inicia cuando el usuario accede a la opción "Fichero de Urls Iniciales" del menú "Configuración" en la interfaz, da clic en dicha opción y el sistema muestra las Urls cargadas en el mismo, con las opciones de cargar

	nuevamente el fichero con nuevas Urls, descargarlas al ordenador, editarlas y eliminarlas.	
Complejidad	Media	
Prioridad	Media	
Precondiciones		
Poscondiciones		
Prototipo de Interfaz		
		
Flujo de eventos		
Flujo básico mostrar fichero de urls iniciales		
	Actor	Sistema
1	El usuario accede a la opción "Fichero de Urls Iniciales" del menú "Configuración" en la interfaz y da clic en dicha opción.	

2		El sistema muestra las Urls cargadas en el mismo, con las opciones de cargar nuevamente el fichero con nuevas Urls, descargarlas al ordenador, editarlas y eliminarlas.
3		Termina el CU.
Flujos alternos		
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos		

9.2 Anexo 2. Diagramas de clases del diseño.

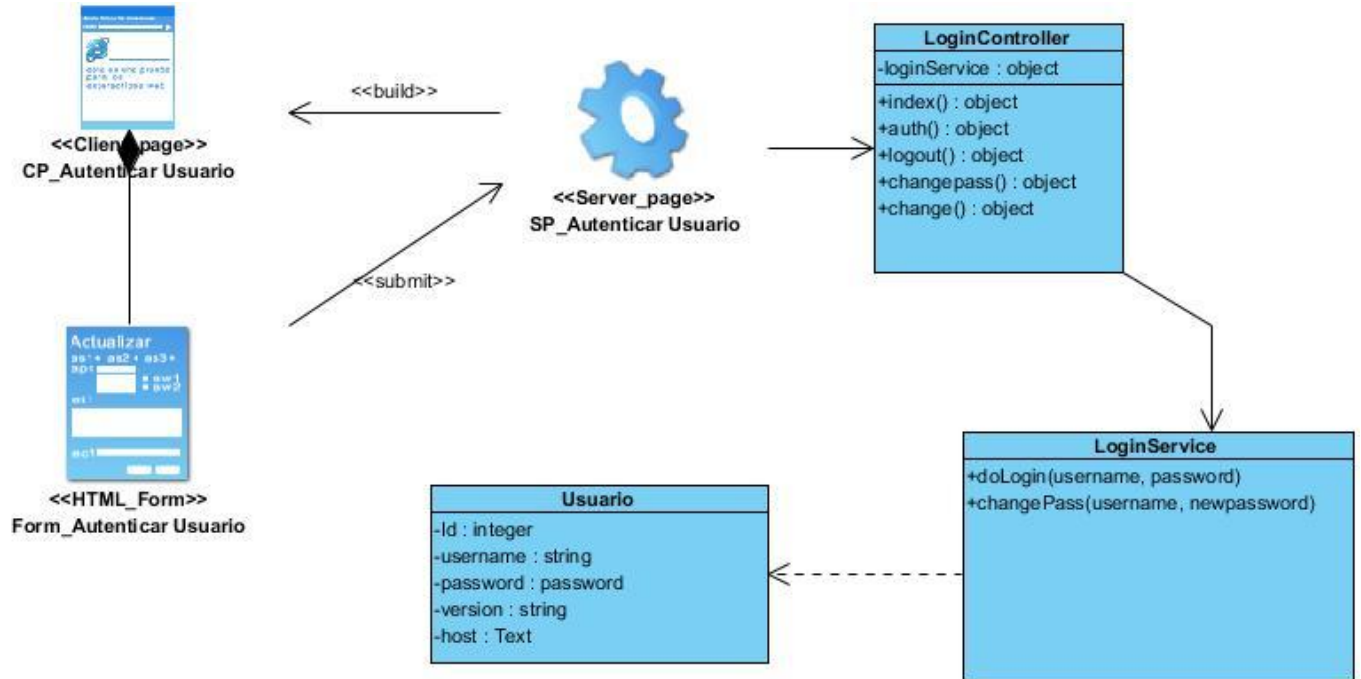


Figura 8. Diagrama de clases del diseño con estereotipos web: CU Autenticar Usuario.

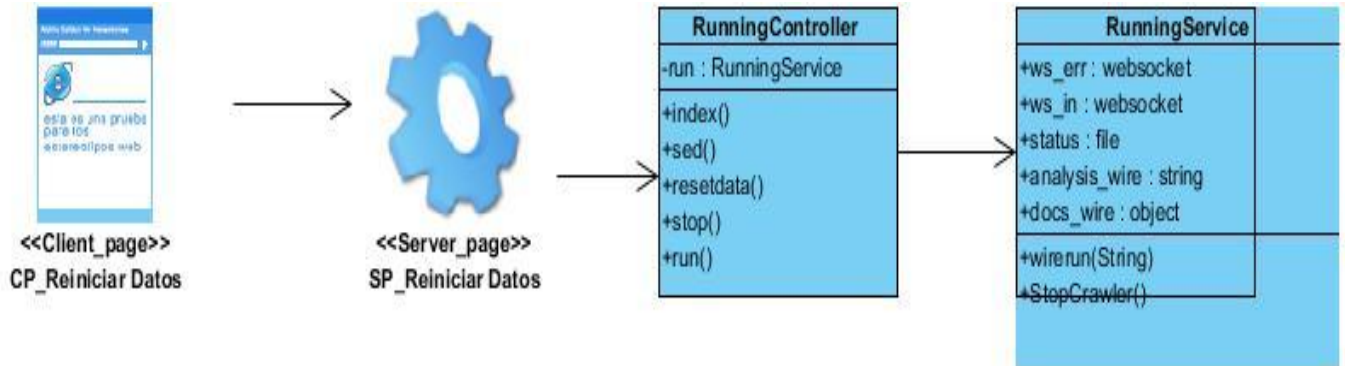


Figura 9. Diagrama de clases del diseño con estereotipos web: CU Reiniciar datos.

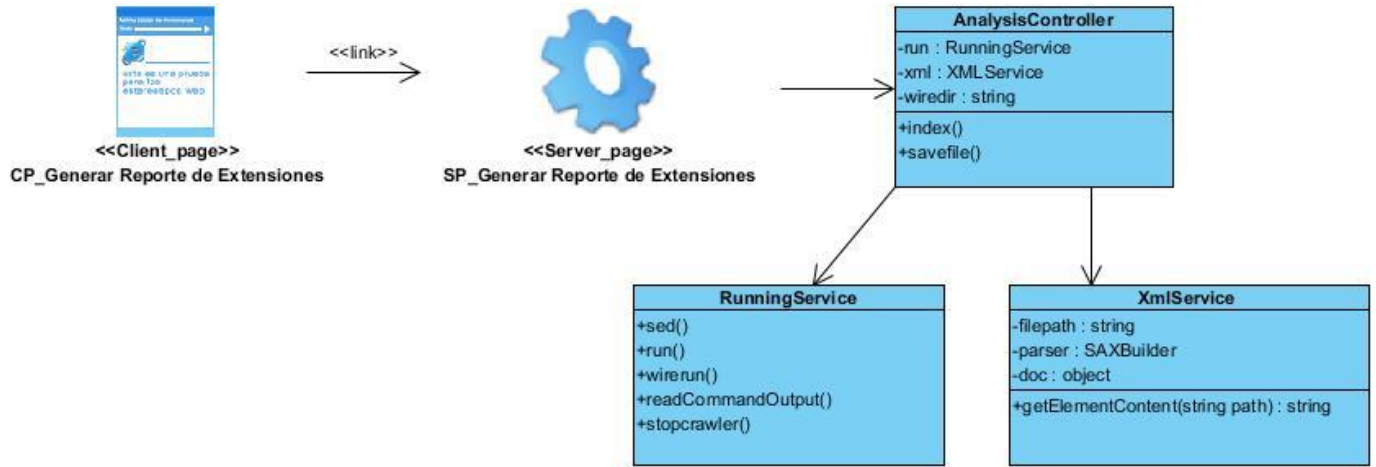


Figura 10. Diagrama de clases del diseño con estereotipos web: CU Generar reporte de extensiones.

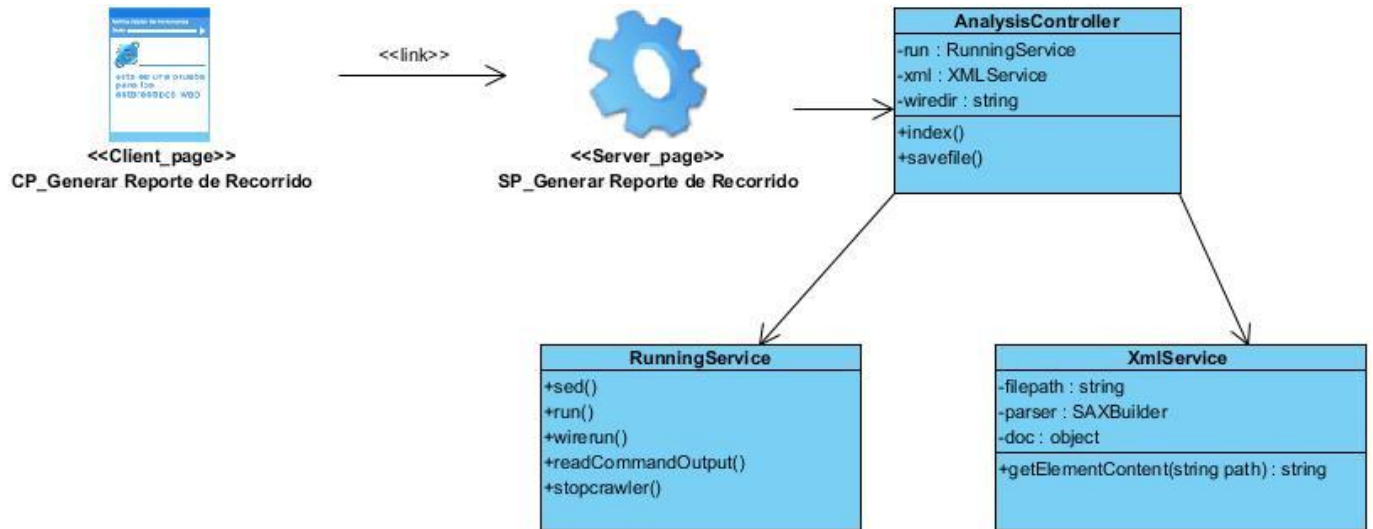


Figura 11. Diagrama de clases del diseño con estereotipos web: CU Generar reporte de recorrido.

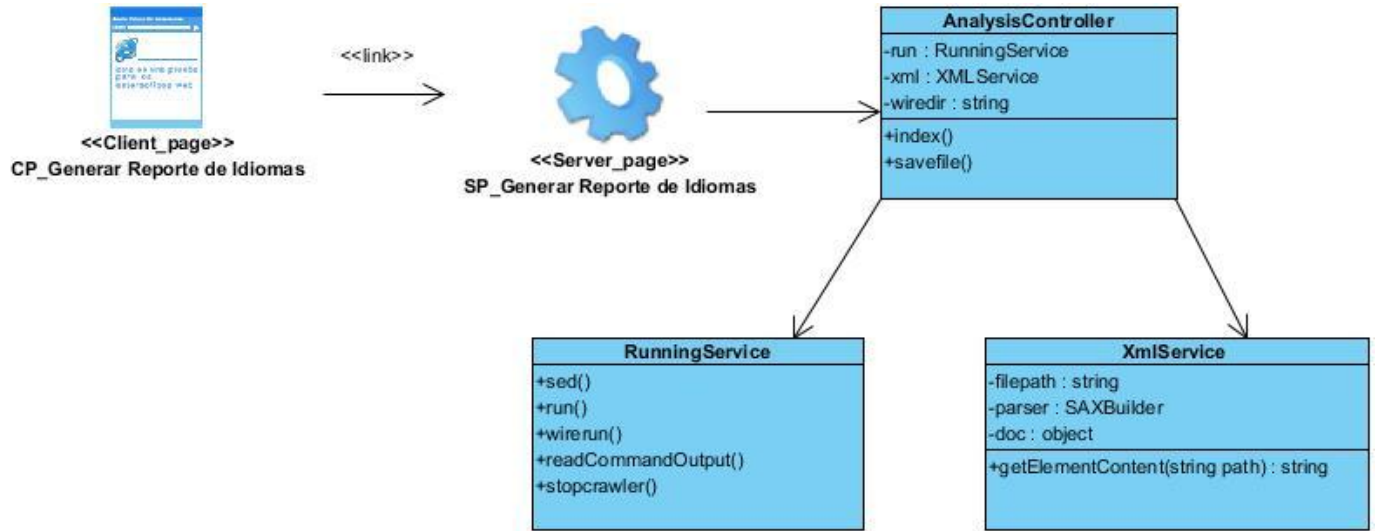


Figura 12. Diagrama de clases del diseño con estereotipos web: CU Generar reporte de idiomas.

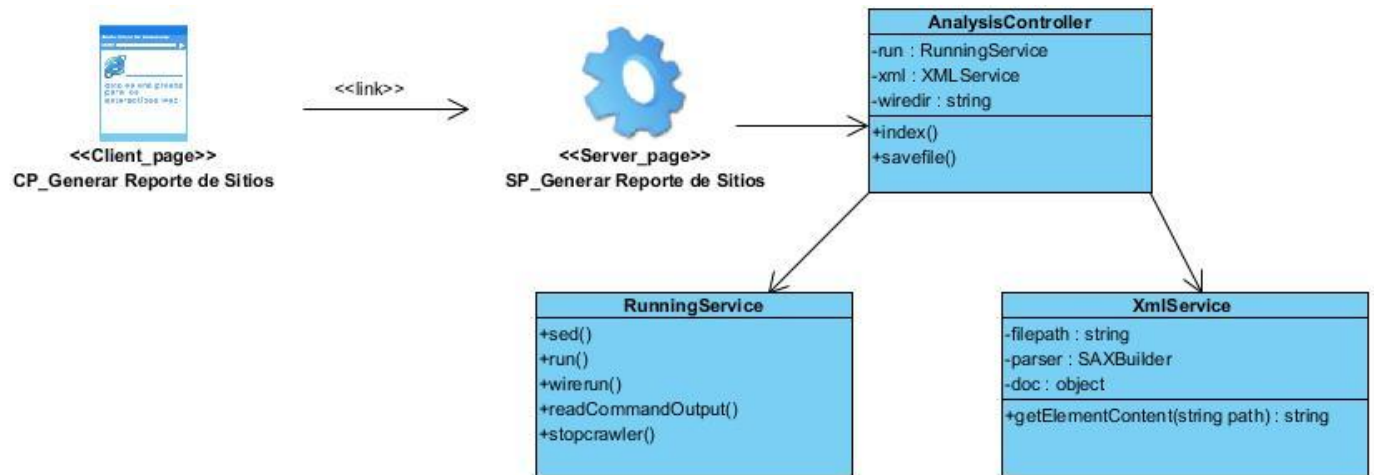


Figura 13. Diagrama de clases del diseño con estereotipos web: CU Generar reporte de sitios.

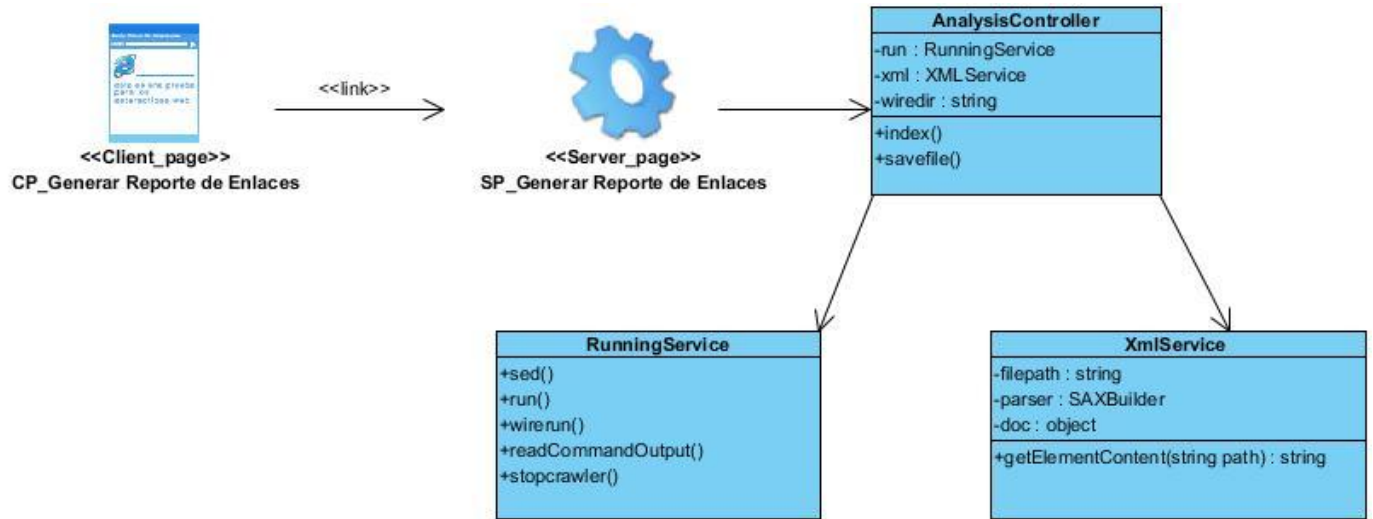


Figura 14. Diagrama de clases del diseño con estereotipos web: CU Generar reporte de enlaces.

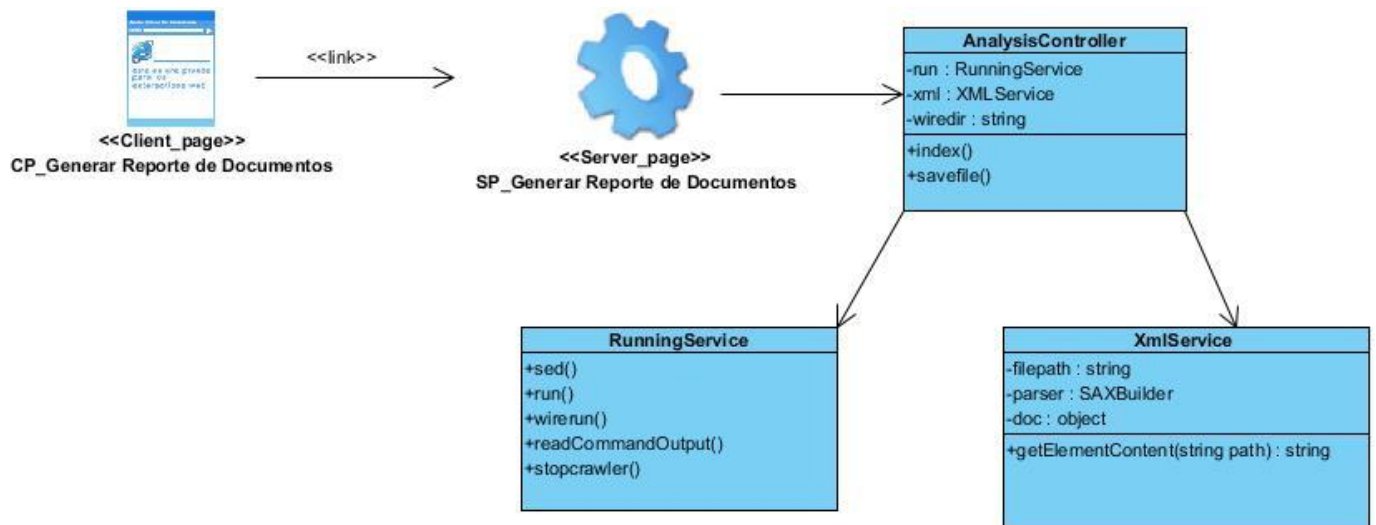


Figura 15. Diagrama de clases del diseño con estereotipos web: CU Generar reporte de documentos.

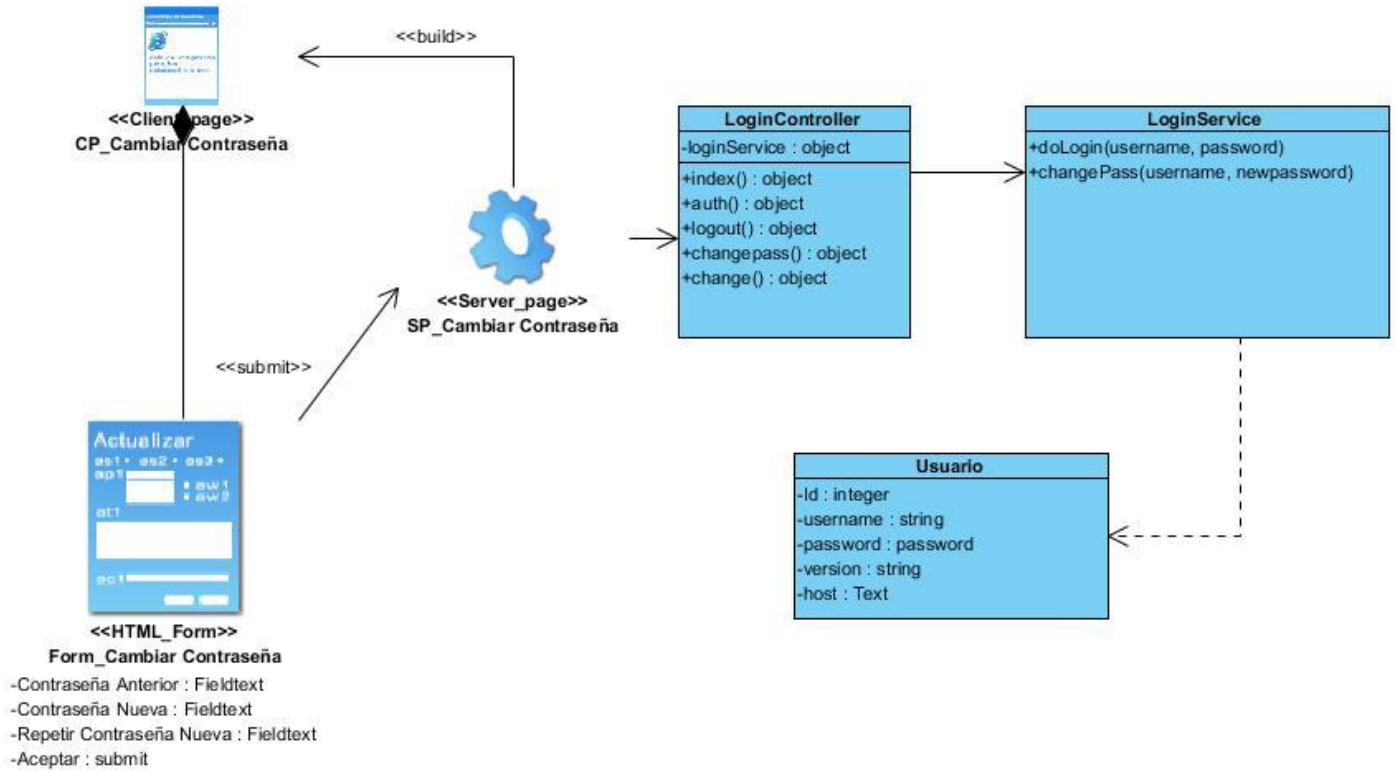


Figura 18. Diagrama de clases del diseño con estereotipos web: CU Cambiar contraseña.

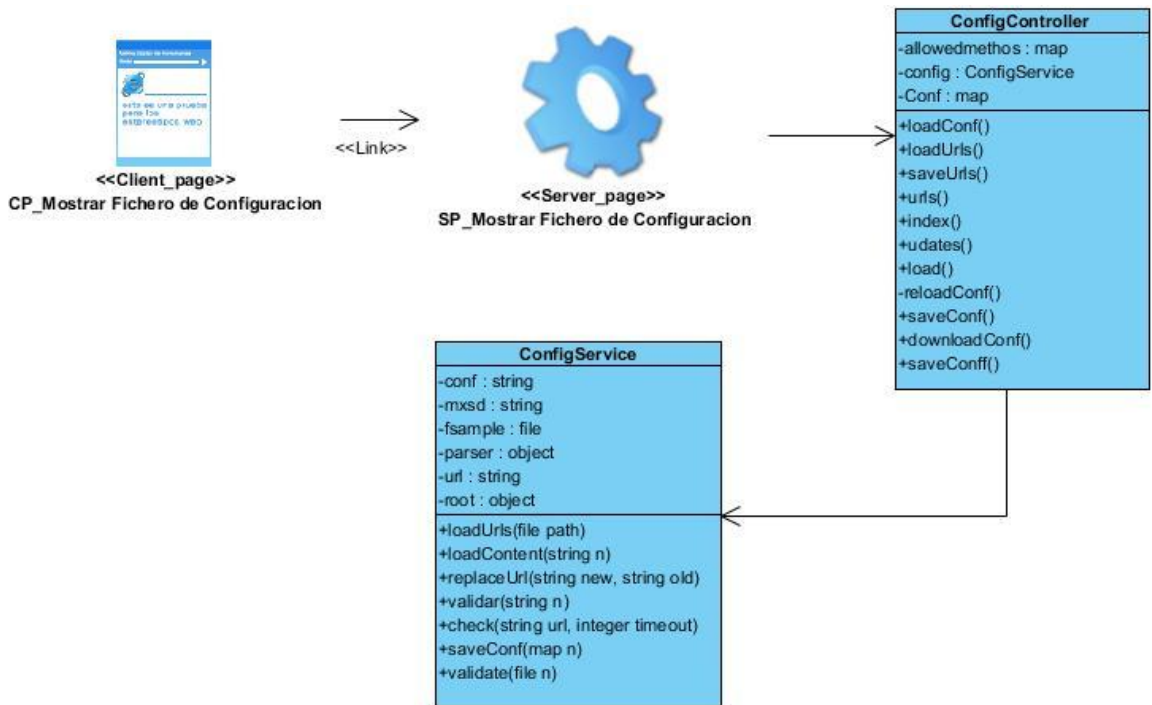


Figura 19. Diagrama de clases del diseño con estereotipos web: CU Mostrar fichero de configuración.

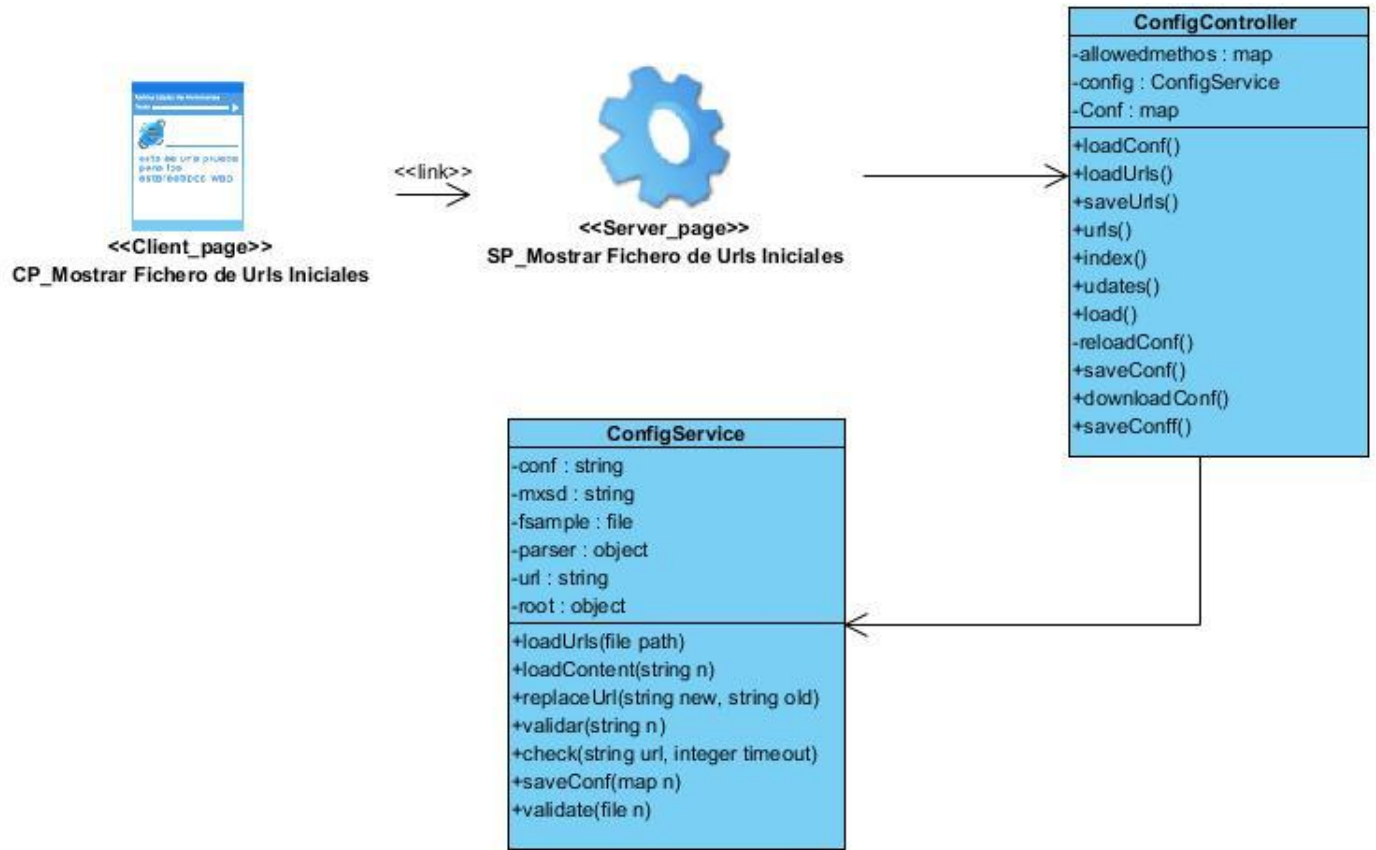


Figura 20. Diagrama de clases del diseño con estereotipos web: CU Mostrar fichero de Urls iniciales.

9.3 Anexo 3. Diagramas de interacción del diseño.

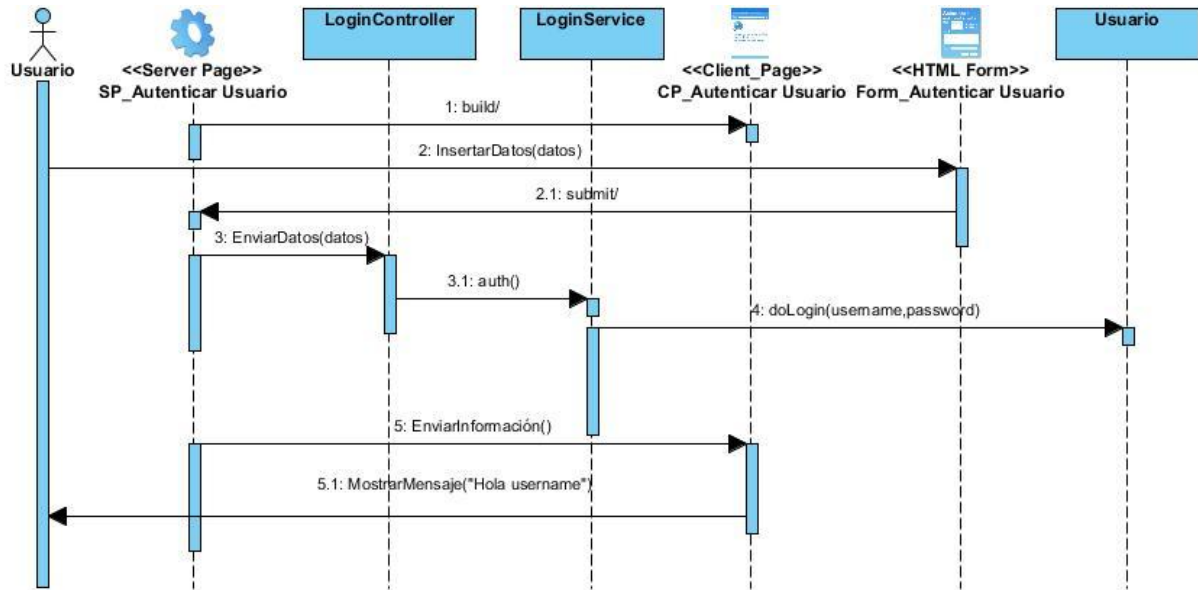


Figura 21. Diagrama de secuencia: CU Autenticar Usuario.

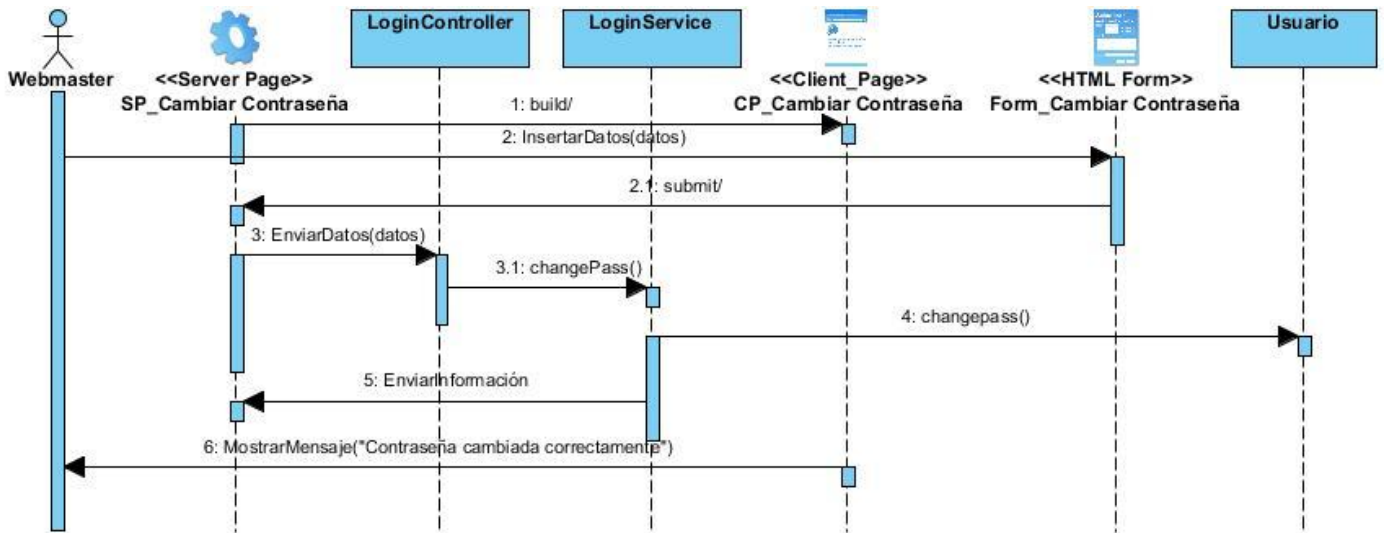


Figura 22. Diagrama de secuencia: CU Cambiar Contraseña.

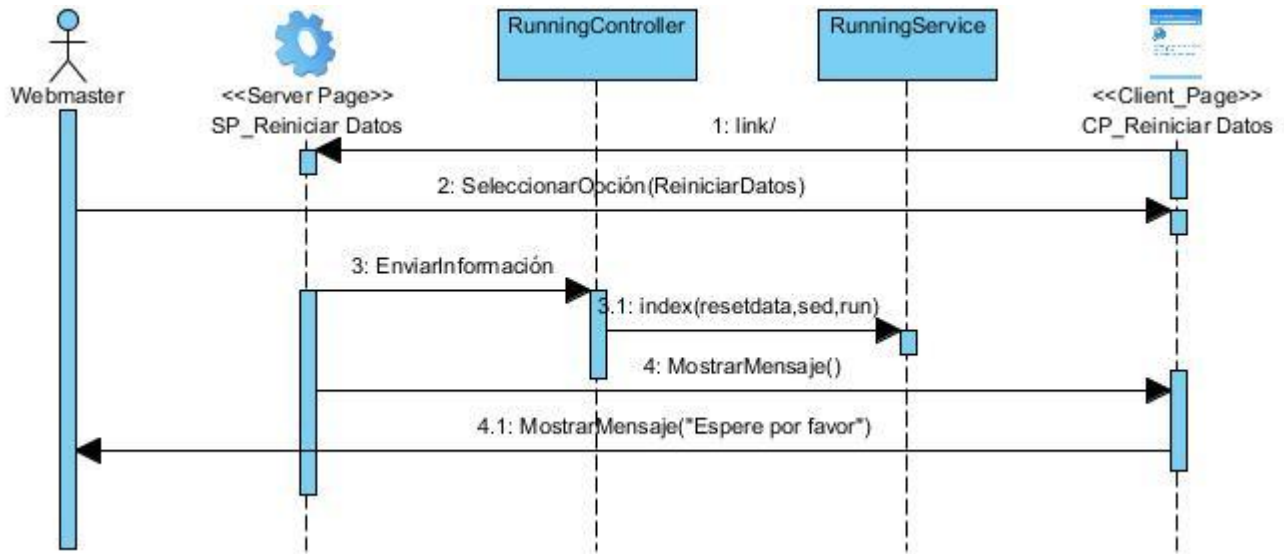


Figura 23. Diagrama de secuencia: CU Reiniciar Datos.

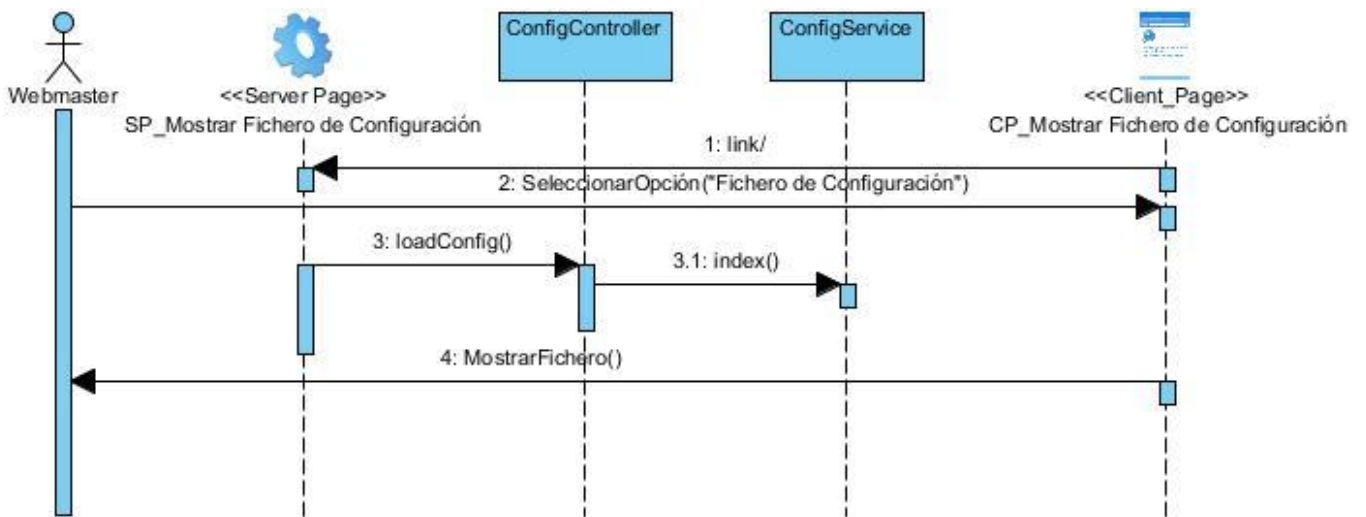


Figura 24. Diagrama de secuencia: CU Mostrar Fichero de Configuración.

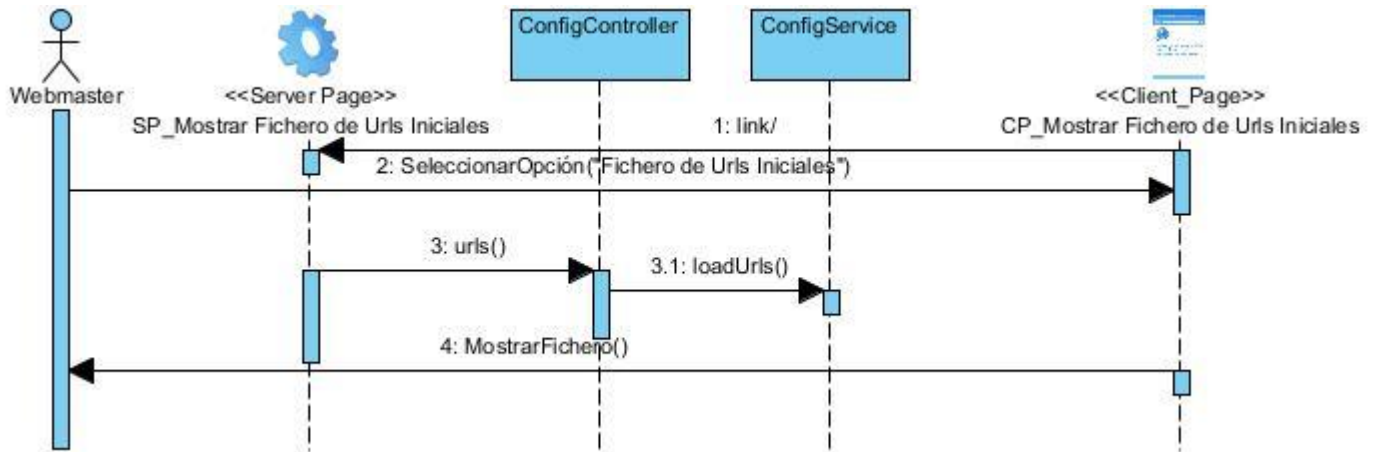


Figura 25. Diagrama de secuencia: CU Mostrar Fichero de Urls Iniciales.

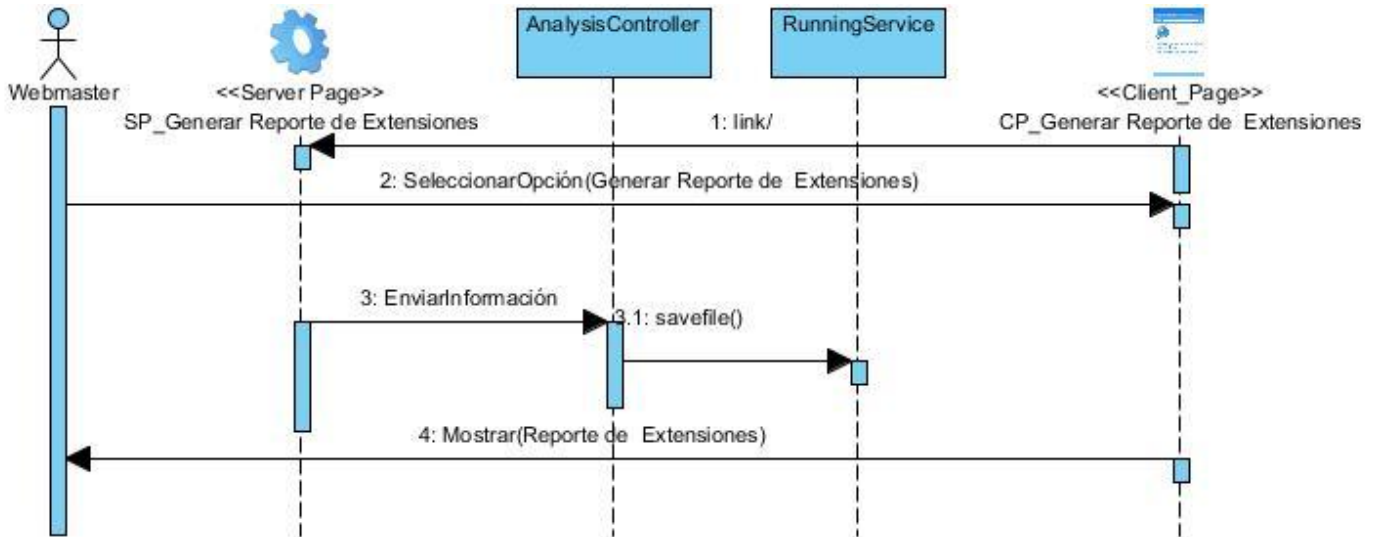


Figura 26. Diagrama de secuencia: CU Generar reporte de extensiones.

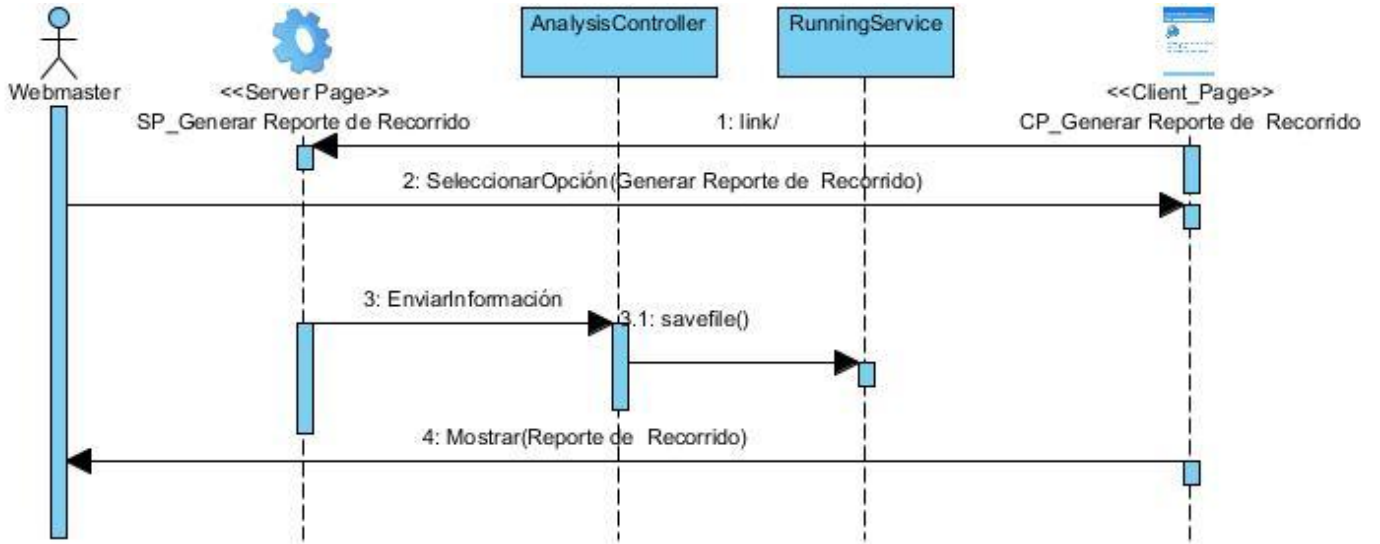


Figura 27. Diagrama de secuencia: CU Generar reporte de recorrido.

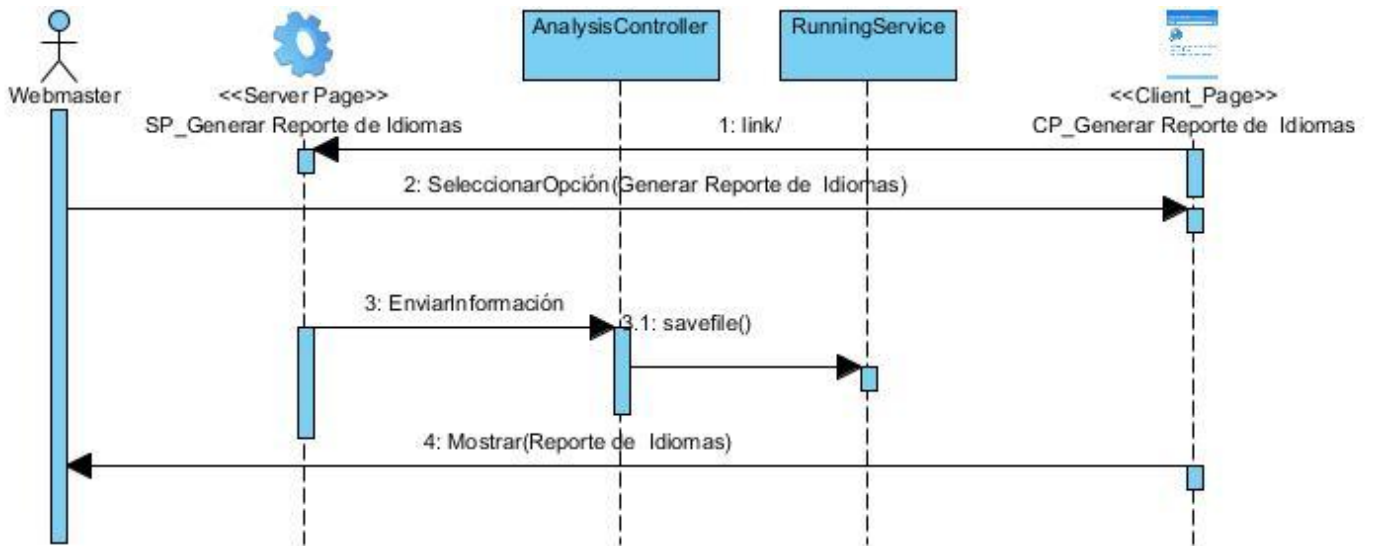


Figura 28. Diagrama de secuencia: CU Generar reporte de idiomas.

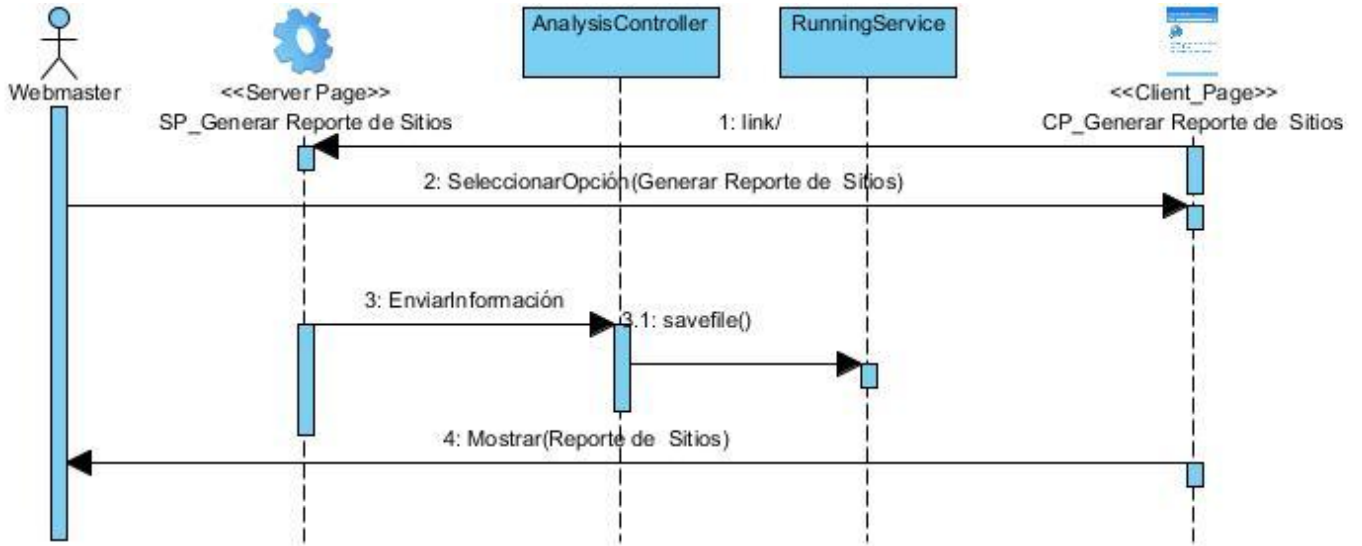


Figura 29. Diagrama de secuencia: CU Generar reporte de sitios.

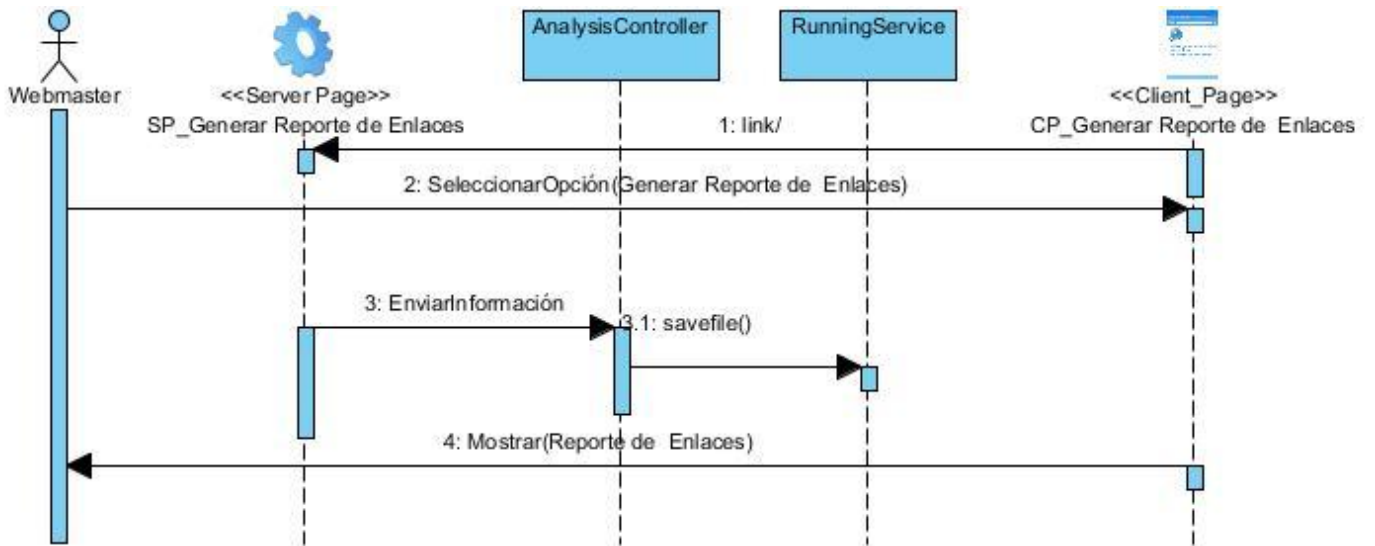


Figura 30. Diagrama de secuencia: CU Generar reporte de enlaces.

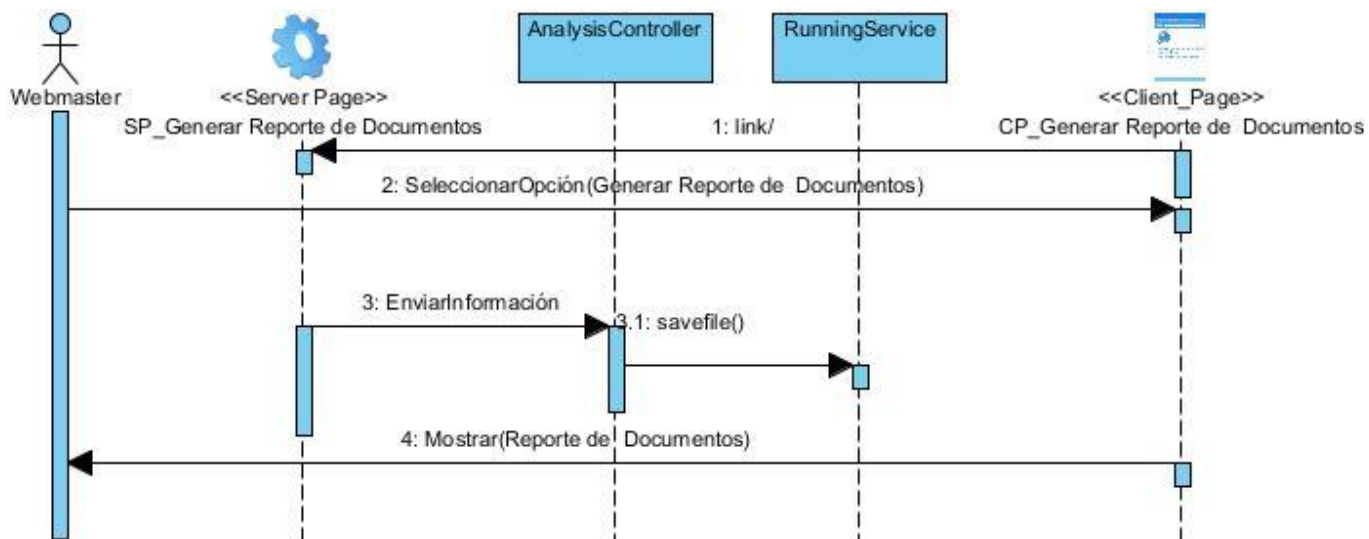


Figura 31 Diagrama de secuencia: CU Generar reporte de documentos.

9.4 Anexo 4. Diseño de casos de prueba a funcionalidades del sistema.

9.4.1 Caso de prueba del CU Reiniciar Datos

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar opción "Reiniciar Datos".	El usuario selecciona la opción "Reiniciar Datos".	El sistema reinicia los datos.	1-El usuario selecciona la opción "Reiniciar datos". 2-El sistema reinicia los datos y muestra los comandos de salida en la pestaña "stdErr".
EC 1.2 Seleccionar otra opción mientras el sistema está reiniciando los datos.	El usuario selecciona otra opción mientras el sistema se encuentra reiniciando los datos.	El sistema muestra un mensaje informando que hay un programa corriendo.	1-El usuario da clic en otra opción mientras el sistema se encuentra reiniciando

			<p>los datos.</p> <p>2-El sistema muestra el mensaje “Hay un programa corriendo, espere”.</p>
--	--	--	---

9.4.2 Caso de prueba del CU Cargar Urls.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar opción “Cargar Urls”.	El usuario selecciona la opción “Cargar Urls”.	El sistema carga las urls.	<p>1-El usuario selecciona la opción “Cargar Urls”.</p> <p>2-El sistema muestra el mensaje “Espere por favor”, mientras carga las urls.</p> <p>3-El sistema muestra los comandos de salida en la pestaña “stdErr” al finalizar la carga de las urls.</p>
EC 1.2 Seleccionar otra opción mientras el sistema está cargando las urls.	El usuario selecciona otra opción mientras el sistema se encuentra cargando las urls.	El sistema muestra un mensaje informando que hay un programa corriendo.	<p>1-El usuario da clic en otra opción mientras el sistema se encuentra cargando las urls.</p> <p>2-El sistema muestra el mensaje “Hay un programa corriendo, espere”.</p>

9.4.3 Caso de prueba del CU Realizar Recorrido

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Realizar recorrido de los sitios <i>web</i> .	El usuario selecciona la opción “Realizar Recorrido”.	El sistema realiza el recorrido de los sitios <i>web</i> .	1-El usuario selecciona la opción “Realizar Recorrido”.

			<p>2-El sistema muestra el mensaje “Espere por favor”, mientras realiza el recorrido de los sitios.</p> <p>3-El sistema muestra los comandos de salida en la pestaña “stdErr” al finalizar el recorrido.</p>
EC 1.2 Seleccionar otra opción mientras el sistema está realizando el recorrido de los sitios.	El usuario selecciona otra opción mientras el sistema se encuentra realizando el recorrido de los sitios.	El sistema muestra un mensaje informando que hay un programa corriendo.	<p>1-El usuario da clic en otra opción mientras el sistema se encuentra realizando el recorrido de los sitios.</p> <p>2-El sistema muestra el mensaje “Hay un programa corriendo, espere”.</p>

9.4.4 Caso de prueba del CU Mostrar fichero de configuración

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar fichero de configuración.	El usuario selecciona la opción “Fichero de Configuración”.	El sistema muestra el fichero de configuración.	<p>1-El usuario selecciona la opción “Fichero de Configuración”.</p> <p>2-El sistema muestra el fichero de configuración del WIRE para que el usuario posteriormente lo pueda editar.</p>
EC 1.2 Introducir	El usuario introduce comandos	El sistema muestra un	1-El usuario

comandos inválidos en el fichero de configuración.	inválidos en el fichero de configuración del WIRE.	mensaje informando que el comando escrito, no está correcto.	selecciona la opción "Fichero de Configuración". 2-El sistema muestra el fichero de configuración del WIRE para que el usuario posteriormente lo pueda editar. 3-El usuario introduce un comando incorrecto y presiona el botón "Aceptar". 4-El sistema muestra un mensaje informando que el comando escrito, no está correcto.
--	--	--	--

9.4.5 Caso de prueba del CU Mostrar fichero de urls iniciales.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar fichero de urls iniciales.	El usuario selecciona la opción "Fichero de Urls Iniciales".	El sistema muestra el fichero de urls iniciales.	1-El usuario selecciona la opción "Fichero de Urls Iniciales". 2-El sistema muestra en campos editables las urls cargadas inicialmente.
EC 1.2 Descargar fichero de urls iniciales.	El usuario accede a descargar el fichero de urls iniciales para el ordenador.	El sistema descarga el fichero de urls iniciales para el ordenador.	1- El usuario accede al ícono de descarga de archivos que se

			<p>encuentra en la parte superior derecha y da clic en él.</p> <p>2- El archivo de urls iniciales se descarga, en la dirección física del ordenador especificada por el usuario.</p> <p>3-El usuario edita el fichero de urls iniciales añadiéndole nuevas urls.</p>
EC 1.3 Cargar fichero de urls iniciales.	El usuario accede a cargar el fichero de urls iniciales.	El sistema carga el fichero de urls iniciales.	<p>1- El usuario accede al ícono de carga de archivos que se encuentra en la parte superior derecha y da clic en él.</p> <p>2- El usuario selecciona el fichero editado anteriormente y lo carga.</p> <p>3-El sistema carga las urls que se encuentren escritas en un formato válido.</p> <p>4-El sistema muestra un mensaje informando la cantidad de urls que han sido cargadas satisfactoriamente.</p>

<p>EC 1.4 Comprobar fichero de urls iniciales.</p>	<p>Se valida que las urls que se encuentran en el fichero, tengan el formato correcto.</p>	<p>El sistema valida que las urls que se encuentran en el fichero, estén en formato correcto, caso contrario, muestra un mensaje informando que existen urls inválidas en el fichero.</p>	<p>1-Al cargar el fichero de urls iniciales, el usuario marca la opción “comprobar el fichero”. 2- El sistema valida que las urls que se encuentran en el fichero, estén en formato correcto, caso contrario, muestra un mensaje informando que existen urls inválidas en el fichero.</p>
<p>EC 1.5 Eliminar url.</p>	<p>Se elimina una url seleccionada.</p>	<p>El sistema elimina la url seleccionada por el usuario.</p>	<p>1-El usuario selecciona una url a eliminar, en la parte derecha del campo donde se encuentra dicha url está el menú “URL” y selecciona la opción “Eliminar” de dicho menú. 2-El sistema muestra un mensaje al usuario un mensaje de confirmación, “Está seguro que desea eliminar esta url?”. 3-Si el usuario presiona el botón “Si”, el sistema elimina la url, caso contrario, regresa a la interfaz de las urls nuevamente.</p>

9.4.6 Caso de prueba del CU Generar reporte de extensiones.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de extensiones.	El usuario selecciona la opción “Estadísticas” y accede al vínculo “Extensiones”.	El sistema muestra el reporte de extensiones.	1- El usuario selecciona la opción “Estadísticas” y accede al vínculo “Extensiones”. 2- El sistema muestra el reporte de extensiones.
EC 1.2 Reporte de extensiones no disponible.	El usuario selecciona la opción “Estadísticas”, accede al vínculo “Extensiones” y el reporte no se encuentra disponible.	El sistema muestra un mensaje informando que el reporte extensiones no se encuentra disponible.	1- El usuario selecciona la opción “Estadísticas” y accede al vínculo “Extensiones”. 2- El sistema muestra el mensaje, “El reporte de extensiones no se encuentra disponible”.

9.4.7 Caso de prueba del CU Generar reporte de sitios

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de sitios.	El usuario selecciona la opción “Estadísticas” y accede al vínculo “Sitios”.	El sistema muestra el reporte de sitios.	1- El usuario selecciona la opción “Estadísticas” y accede al vínculo “Sitios”. 2- El sistema muestra el reporte de sitios.
EC 1.2 Reporte de sitios no disponible.	El usuario selecciona la opción “Estadísticas”, accede al vínculo “Sitios” y el reporte no se encuentra disponible.	El sistema muestra un mensaje informando que el reporte sitios no se encuentra disponible.	1- El usuario selecciona la opción “Estadísticas” y accede al vínculo “Sitios”. 2- El sistema muestra

			el mensaje, “El reporte de sitios no se encuentra disponible”.
--	--	--	--

9.4.8 Caso de prueba del CU Generar reporte de documentos

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de documentos.	El usuario selecciona la opción “Estadísticas” y accede al vínculo “Documentos”.	El sistema muestra el reporte de documentos.	1- El usuario selecciona la opción “Estadísticas” y accede al vínculo “Documentos”. 2- El sistema muestra el reporte de documentos.
EC 1.2 Reporte de documentos no disponible.	El usuario selecciona la opción “Estadísticas”, accede al vínculo “Documentos” y el reporte no se encuentra disponible.	El sistema muestra un mensaje informando que el reporte documentos no se encuentra disponible.	1- El usuario selecciona la opción “Estadísticas” y accede al vínculo “Documentos”. 2- El sistema muestra el mensaje, “El reporte de documentos no se encuentra disponible”.

9.4.9 Caso de prueba del CU Generar reporte de enlaces

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de enlaces.	El usuario selecciona la opción “Estadísticas” y accede al vínculo “Enlaces”.	El sistema muestra el reporte de enlaces.	1- El usuario selecciona la opción “Estadísticas” y accede al vínculo “Enlaces”. 2- El sistema muestra el reporte de enlaces.

EC 1.2 Reporte de enlaces no disponible.	El usuario selecciona la opción "Estadísticas", accede al vínculo "Enlaces" y el reporte no se encuentra disponible.	El sistema muestra un mensaje informando que el reporte enlaces no se encuentra disponible.	1- El usuario selecciona la opción "Estadísticas" y accede al vínculo "Enlaces". 2- El sistema muestra el mensaje, "El reporte de enlaces no se encuentra disponible".
--	--	---	---

9.4.10 Caso de prueba del CU Generar reporte de idiomas

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de idiomas.	El usuario selecciona la opción "Estadísticas" y accede al vínculo "Idiomas".	El sistema muestra el reporte de idiomas.	1- El usuario selecciona la opción "Estadísticas" y accede al vínculo "Idiomas". 2- El sistema muestra el reporte de idiomas.
EC 1.2 Reporte de idiomas no disponible.	El usuario selecciona la opción "Estadísticas", accede al vínculo "Idiomas" y el reporte no se encuentra disponible.	El sistema muestra un mensaje informando que el reporte idiomas no se encuentra disponible.	1- El usuario selecciona la opción "Estadísticas" y accede al vínculo "Idiomas". 2- El sistema muestra el mensaje, "El reporte de idiomas no se encuentra disponible".

9.4.11 Caso de prueba del CU Generar reporte de recorrido

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de recorrido.	El usuario selecciona la opción "Estadísticas" y accede al vínculo "Recorrido".	El sistema muestra el reporte de recorrido.	1- El usuario selecciona la opción "Estadísticas" y accede al vínculo "Recorrido". 2- El sistema muestra el reporte de recorrido.
EC 1.2 Reporte de recorrido no disponible.	El usuario selecciona la opción "Estadísticas", accede al vínculo "Recorrido" y el reporte no se encuentra disponible.	El sistema muestra un mensaje informando que el reporte recorrido no se encuentra disponible.	1- El usuario selecciona la opción "Estadísticas" y accede al vínculo "Recorrido". 2- El sistema muestra el mensaje, "El reporte de recorrido no se encuentra disponible".

9.5 Anexo 5. Entrevista realizada en la Dirección de Informatización de la UCI.

A continuación se muestran las interrogantes realizadas en el Departamento de Tecnología de la Dirección de Informatización:

- ¿Actualmente existe alguna forma de saber las características de la Web universitaria con respecto al tamaño de la misma, % de enlaces válidos, % de enlaces rotos y % de sitios en un determinado lenguaje?
- ¿De qué forma detectan las imágenes o videos que se encuentran activos o no en la red para ser accedidos por los usuarios?
- ¿Existe algún mecanismo o herramienta de análisis para la Web universitaria?
- ¿Qué indicadores quisieras conocer de la Web universitaria?
- ¿Qué necesidades o problemas tienen identificadas que al resolverlas se le facilitaría su trabajo diario?

A las interrogantes realizadas anteriormente, se dieron las siguientes respuestas:

- No, Actualmente no se tiene ningún tipo de información con respecto a las estadísticas de la web universitaria, sería importante tener estadísticas que permitan observar su comportamiento, en cuanto a tamaño, la cantidad de enlaces que sean accesibles.
- No tenemos como detectar los archivos multimedia que se encuentran activos o no, salvo que al acceder a las páginas web e intentar hacer uso de ellos, pues surja el inconveniente que no se encuentren accesibles.
- Sería bueno conocer estadísticas acerca de la cantidad de enlaces que se encuentran accesibles, la cantidad de archivos multimedia, la página que mayor cantidad de enlaces entrantes tenga, es decir, que tenga el mayor número de referencia por los usuarios.
- Es necesario tener información acerca de cómo se encuentra distribuido el contenido en los sitios web, es decir, cómo saber si un sitio se encuentra muy cargado o no de algún contenido en específico, como videos, imágenes animadas, la cantidad de javascript y css utilizado, lo cual permite que el tiempo de respuesta al usuario sea óptimo o no.