



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 1

MÓDULO DE CONFIGURACIÓN PARA EL MECANISMO DE RASTREO DEL BUSCADOR ORIÓN

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

Autores:

Yulio Aleman Jimenez
Yoniel Jorge Thomas Sosa

Tutores:

Ing. Aylin Estrada Velazco
Ing. Eyeris Rodríguez Rueda

Co-Tutor:

Lic. Yanisel Gillard Vara

JUNIO DE 2014



“Mi confianza en el triunfo de lo que creo, es completa”

Ernesto Guevara de la Serna

DECLARACIÓN DE AUTORÍA

Nosotros, Yulio Aleman Jimenez y Yoniel Jorge Thomas Sosa, declaramos ser los únicos autores de este trabajo y autorizamos a la Facultad 1 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autor
Yulio Aleman Jimenez

Firma del Autor

Autor
Yoniel Jorge Thomas Sosa

Firma del Autor

Tutor
Ing. Aylin Estrada Velazco

Firma del Tutor

Tutor
Ing. Eyeris Rodríguez Rueda

Firma del Tutor

AGRADECIMIENTOS

Agradezco especialmente a mis viejos, por su amor, ejemplo, comprensión, consagración y apoyo en todo momento durante mis años de vida, especialmente en estos cinco años de estudio. Gracias por confiar en mí, por escucharme, por ayudarme a tomar las decisiones correctas y además, por hacer crecer en mí los valores que hoy defiendo con gran orgullo.

A mi hermana Yordy, mis sobrinos Yuramys y Pedro Eduardo, mi cuñado, y mis tías y tíos por su amor, confianza y apoyo incondicional.

A mi hermano Lázaro, por ejercer en mí el apoyo y el ejemplo que siempre necesité.

A Mauriciel, por ser mi amigo y hermano durante todos estos años, y acogerme en su familia como si fuese un hijo más.

A Liomar, Alexis y Hansel por su sincera amistad durante estos años de carrera universitaria.

A Betty, por ofrecerme su amor y su apoyo durante el tiempo que hemos estado juntos. Por hacerme tan feliz como nunca lo fui antes y demostrarme, que el amor está lleno de sorpresas inimaginables.

A Yoniel, por ser mi amigo y compañero de tesis, capaz de transmitir la audacia necesaria para enfrentar los problemas en cualquier circunstancia, y soportar mis regaños y meticulosidad para con las cosas.

Muchas Gracias, Yulio.

En especial a mis padres por su ejemplo, dedicación, comprensión y sacrificio durante todos estos años de estudio, gracias por confiar en mí y transmitirme la fuerza y la energía que necesitaba para seguir adelante cuando creía que no podía llegar hasta el final.

A mi hermano Yasser, tío y abuelos Lidia y Julio, por su amor, confianza y apoyo incondicional.

A Rubén y Ana por toda la ayuda que siempre me han brindado y acogerme como un hijo en su familia.

A mis amigos Raúl, Yosliet y en especial a Alejandro que ha sido como un hermano para mí, gracias por ofrecerme su amistad sincera.

A Julio, mi compañero de tesis por todo el apoyo y soportar mis caprichos aun cuando ya no quedaba tiempo para más.

Muchas Gracias, Yoniel

Agradecimientos compartidos:

A nuestros tutores Aylin, Eyeris y Yanisel por su preocupación, atención y dedicación.

A todos nuestros compañeros de aula que siempre estuvieron para ayudarnos en los momentos más difíciles en esta universidad.

A todos los profesores que durante estos cinco años influyeron no sólo en nuestra formación profesional, sino que nos enseñaron a crecer cada día como mejores personas.

A todos los que de una forma u otra han contribuido a la realización de este sueño.

DEDICATORIA

*A mis viejos, por darme la vida, ayudarme en todo momento, educarme
y convertirme en la persona que soy hoy.*

*A mi abuela Arcadia, por ser el nieto que más tiempo llevase en el
recuerdo, por ejercer tu ejemplo y gran amor hacia mí.*

Yulio

*A mis padres por el amor, la confianza y educación que me han brindado
desde que nací, y en especial a mis abuelos por su fe en mí y estar
siempre presentes en cada paso que doy en la vida.*

Yoniel

RESUMEN

El constante aumento del cúmulo de información en formato digital pública en las redes informáticas alrededor del mundo, ha propiciado la dificultad de los usuarios de encontrar lo que realmente necesitan en un momento dado. Para dar respuesta a la necesidad de localizar oportunamente la información requerida, se diseñaron los Sistemas de Recuperación de Información; cuyas funcionalidades, además de contribuir a su objetivo primordial, poseen un gran número de opciones de configuración, difíciles de administrar. En la Universidad de las Ciencias Informáticas se desarrolló el motor de búsqueda Orión, pero el mismo presenta numerosas desventajas que impiden un desempeño óptimo del proceso de configuración de su mecanismo de rastreo; por lo que la presente investigación permitió desarrollar un módulo para el buscador que mejora la usabilidad y facilita el trabajo de los administradores en las tareas de configuración. Para la implementación de la propuesta de solución, guiado por la metodología OpenUP, se seleccionaron como principales tecnologías: el marco de trabajo Symfony 2.2.6 para la programación en PHP, el Sistema de Gestión de Base de Datos no relacional MongoDB 1.4.5 y Visual Paradigm 8.0 como herramienta para el modelado. El módulo de configuración para el mecanismo de rastreo del buscador Orión implementado, posee un conjunto de características y funcionalidades que contribuyen, mediante la disponibilidad de interfaces web, al aumento del control de cambios en las configuraciones y a la agilización de dicho proceso; proporcionando además, información acerca de las configuraciones realizadas, que anteriormente era imposible o difícil de obtener.

Palabras clave: configuración, interfaz web, Nutch, Orión, Sistemas de Recuperación de Información.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN Y LAS APLICACIONES WEB	6
1.1 INTRODUCCIÓN	6
1.2 FUNDAMENTOS TEÓRICOS ASOCIADOS AL DOMINIO DEL PROBLEMA	6
1.2.1 <i>Recuperación de Información</i>	6
1.2.2 <i>Sistemas de Recuperación de Información</i>	7
1.2.3 <i>Spider o araña como mecanismo de rastreo de información</i>	9
1.2.4 <i>Apache Nutch</i>	9
1.2.5 <i>Sistemas para la configuración de spiders</i>	12
1.2.6 <i>Aplicaciones web</i>	16
1.3 TECNOLOGÍAS Y HERRAMIENTAS SELECCIONADAS.....	17
1.3.1 <i>Tecnologías del lado del Cliente</i>	17
1.3.2 <i>Tecnologías del lado del Servidor</i>	19
1.3.3 <i>Entorno de Desarrollo Integrado</i>	19
1.3.4 <i>Sistema de Gestión de Base de Datos</i>	20
1.3.5 <i>Servidor web</i>	21
1.3.6 <i>Herramienta CASE</i>	22
1.4 LENGUAJE DE MODELADO	22
1.5 METODOLOGÍA DE DESARROLLO	23
1.6 MARCO DE TRABAJO.....	23
1.7 CONCLUSIONES PARCIALES	25
CAPÍTULO 2. DISEÑO DE LA PROPUESTA DE SOLUCIÓN	26
2.1 INTRODUCCIÓN	26
2.2 MODELADO DEL DOMINIO	26
2.2.1 <i>Descripción de Clases del Modelo del Dominio</i>	26
2.2.2 <i>Diagrama de Clases del Modelo del Dominio</i>	27
2.3 ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE	27
2.3.1 <i>Requisitos funcionales</i>	27
2.3.2 <i>Requisitos no funcionales</i>	30
2.4 MODELO DE CASOS DE USO DEL SISTEMA.....	31
2.4.1 <i>Diagrama de Casos de Uso del Sistema</i>	31
2.4.2 <i>Patrones de casos de uso utilizados</i>	34
2.4.3 <i>Especificación de casos de uso</i>	35
2.5 ESTILO ARQUITECTÓNICO	38
2.6 PATRONES DE DISEÑO	39

2.7	MODELO DE DISEÑO.....	41
2.7.1	<i>Diagrama de clases del diseño con estereotipos web</i>	41
2.8	DIAGRAMA DE INTERACCIÓN	42
2.8.1	<i>Diagramas de secuencia</i>	42
2.9	DISEÑO DE LA BASE DE DATOS	43
2.10	MODELO DE DESPLIEGUE	44
2.11	CONCLUSIONES PARCIALES	45
CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL MÓDULO DE CONFIGURACIÓN WEB PARA EL MECANISMO DE RASTREO DEL BUSCADOR ORIÓN.....		46
3.1	INTRODUCCIÓN	46
3.2	MODELO DE COMPONENTES QUE INTEGRAN LA SOLUCIÓN INFORMÁTICA.....	46
3.2.1	<i>Diagrama de componentes</i>	46
3.3	ESTÁNDARES DE CODIFICACIÓN UTILIZADOS	48
3.4	PRINCIPALES PANTALLAS DEL MÓDULO DE CONFIGURACIÓN WEB PARA EL MECANISMO DE RASTREO DEL BUSCADOR ORIÓN 50	
3.5	VALIDACIÓN DEL MÓDULO DE CONFIGURACIÓN WEB IMPLEMENTADO	51
3.5.1	<i>Pruebas funcionales</i>	51
3.5.2	<i>Pruebas de integración</i>	54
3.5.3	<i>Pruebas de seguridad</i>	55
3.5.4	<i>Pruebas de usabilidad</i>	57
3.5.5	<i>Evaluación del tiempo del proceso de configuración</i>	60
3.6	CONCLUSIONES PARCIALES	60
CONCLUSIONES		61
RECOMENDACIONES		62
REFERENCIAS BIBLIOGRÁFICAS.....		63
GLOSARIO DE TÉRMINOS		68
ANEXOS.....		69
ANEXO # 1: COMPARACIÓN ENTRE LAS CLASIFICACIONES DE LOS SRI.....		69
ANEXO # 2: ESQUEMA DEL FUNCIONAMIENTO INTERNO DE NUTCH.		70
ANEXO # 3: ESTADÍSTICAS HISTÓRICAS DE USO DE SERVIDORES WEB EN EL PERÍODO 1995 - 2012.....		70
ANEXO # 4: ESPECIFICACIÓN DE LOS CASOS DE USO DEL SISTEMA.		71
ANEXO # 5: DIAGRAMAS DE CLASES DEL DISEÑO.		87
ANEXO # 6: DIAGRAMAS DE SECUENCIA.		88
ANEXO # 7: PRINCIPALES IMÁGENES DEL SISTEMA.		92
ANEXO # 8: DISEÑO DE CASOS DE PRUEBA BASADOS EN CASOS DE USO.....		95

ÍNDICE DE TABLAS

TABLA 1: COMPARACIÓN ENTRE <i>SPIDERS</i>	15
TABLA 2: DESCRIPCIÓN DE LAS CLASES DEL MODELO DEL DOMINIO.	26
TABLA 3: DESCRIPCIÓN DEL CU "GESTIONAR ARCHIVOS"	35
TABLA 4: FRAGMENTO DEL CASO DE PRUEBA BASADO EN EL CU "GESTIONAR ARCHIVOS"	52
TABLA 5: VARIABLES EMPLEADAS EN EL DISEÑO DEL CASO DE PRUEBA BASADO EN EL CU "GESTIONAR ARCHIVOS"	53
TABLA 6: CANTIDAD DE NO CONFORMIDADES POR CADA ITERACIÓN DE LAS PRUEBAS FUNCIONALES.	53
TABLA 7: RESULTADOS DE LA APLICACIÓN DE LA LISTA DE CHEQUEO PARA PRUEBAS DE SEGURIDAD DE NIVEL 1.....	56
TABLA 8: RESULTADOS DEL ESCANEADO DE VULNERABILIDADES CON ACUNETIX WEB VULNERABILITY SCANNER 8.0 PARA PRUEBAS DE SEGURIDAD DE NIVEL 2.	56
TABLA 9: RESULTADOS DE PRUEBA DE USABILIDAD UTILIZANDO LISTA DE CHEQUEO.	58
TABLA 10: RESULTADOS DEL PRE EXPERIMENTO.	60
TABLA 11: CARACTERÍSTICAS DE LOS SRI.....	69
TABLA 12: ESPECIFICACIÓN DEL CU "LISTAR ARCHIVOS"	71
TABLA 13: ESPECIFICACIÓN DEL CU "MOSTRAR DETALLES DE ARCHIVO".	72
TABLA 14: ESPECIFICACIÓN DEL CU "LISTAR INSTANCIAS".....	73
TABLA 15: ESPECIFICACIÓN DEL CU "GESTIONAR INSTANCIAS".....	74
TABLA 16: ESPECIFICACIÓN DEL CU "MOSTRAR DETALLES DE INSTANCIA".	76
TABLA 17: ESPECIFICACIÓN DEL CU "LISTAR <i>PLUGINS</i> ".....	77
TABLA 18: ESPECIFICACIÓN DEL CU "GESTIONAR <i>PLUGINS</i> ".....	78
TABLA 19: ESPECIFICACIÓN DEL CU "MOSTRAR DETALLES DE <i>PLUGIN</i> ".	82
TABLA 20: ESPECIFICACIÓN DEL CU "LISTAR MODIFICACIONES".	83
TABLA 21: ESPECIFICACIÓN DEL CU "GESTIONAR MODIFICACIONES".	84
TABLA 22: ESPECIFICACIÓN DEL CU "MOSTRAR DETALLES DE MODIFICACIÓN".	85
TABLA 23: SECCIÓN "MODIFICAR DATOS DE ARCHIVO" DEL CASO DE PRUEBA BASADO EN EL CU "GESTIONAR ARCHIVOS".	95
TABLA 24: SECCIÓN "ELIMINAR DATOS DE ARCHIVO" DEL CASO DE PRUEBA BASADO EN EL CU "GESTIONAR ARCHIVOS".	96
TABLA 25: SECCIÓN "REGISTRAR DATOS DE INSTANCIA" DEL CASO DE PRUEBA BASADO EN EL CU "GESTIONAR INSTANCIAS".	97
TABLA 26: SECCIÓN "MODIFICAR DATOS DE INSTANCIA" DEL CASO DE PRUEBA BASADO EN EL CU "GESTIONAR INSTANCIAS".	97
TABLA 27: SECCIÓN "ELIMINAR DATOS DE INSTANCIA" DEL CASO DE PRUEBA BASADO EN EL CU "GESTIONAR INSTANCIAS".....	98
TABLA 28: VARIABLES EMPLEADAS EN EL DISEÑO DEL CASO DE PRUEBA BASADO EN EL CU "GESTIONAR INSTANCIAS".	98

ÍNDICE DE FIGURAS

FIGURA 1: RESULTADOS DE LOS ESTUDIOS WEBMÉTRICOS REALIZADOS EN LA UCI EN EL PERÍODO 2008-2011.	2
FIGURA 2: ARQUITECTURA DE UN MOTOR DE BÚSQUEDA.	9
FIGURA 3: INTERFAZ DE CONFIGURACIÓN DE OPEN SEARCH SERVER.	13
FIGURA 4: INTERFAZ DE CONFIGURACIÓN DE MNOGoSEARCH.	14
FIGURA 5: INTERFAZ DE CONFIGURACIÓN DE APACHE LUCENE.	15
FIGURA 6: ANALOGÍA ENTRE MONGODB Y LOS SISTEMAS DE GESTIÓN DE BASES DE DATOS SQL.....	21
FIGURA 7: DIAGRAMA DE CLASES DEL MODELO DEL DOMINIO.	27
FIGURA 8: JERARQUÍA DE USUARIOS DEL SISTEMA.	31
FIGURA 9: CASO DE USO INICIALIZADO POR EL SÚPER ADMINISTRADOR.	31
FIGURA 10: CASOS DE USO INICIALIZADOS POR EL ADMINISTRADOR.....	32
FIGURA 11: CASOS DE USO INICIALIZADOS POR EL USUARIO COMÚN.	32
FIGURA 12: EJEMPLO DE USO DEL PATRÓN DE CU CONCORDANCIA – REUTILIZACIÓN.	34
FIGURA 13: EJEMPLO DE USO DEL PATRÓN DE CU CRUD PARCIAL.	34
FIGURA 14: EJEMPLO DE USO DEL PATRÓN DE CU SERVICIO OPCIONAL – ADICIÓN.	35
FIGURA 15: EJEMPLO DE USO DEL PATRÓN DE CU MÚLTIPLES ACTORES - ROL COMÚN.	35
FIGURA 16: PATRÓN ARQUITECTÓNICO MODELO - VISTA - CONTROLADOR.....	39
FIGURA 17: DIAGRAMA DE CLASES DEL DISEÑO CON ESTEREOTIPOS WEB DEL CU “GESTIONAR ARCHIVOS”.	41
FIGURA 18: DIAGRAMA DE SECUENCIA DEL ESCENARIO “REGISTRAR ARCHIVO” DEL CU “GESTIONAR ARCHIVOS”.	42
FIGURA 19: DIAGRAMA DE SECUENCIA DEL ESCENARIO “MODIFICAR ARCHIVO” DEL CU “GESTIONAR ARCHIVOS”.	42
FIGURA 20: DIAGRAMA DE SECUENCIA DEL ESCENARIO “ELIMINAR ARCHIVO” DEL CU “GESTIONAR ARCHIVOS”.	43
FIGURA 21: MODELO FÍSICO DE LA BASE DE DATOS.	44
FIGURA 22: DIAGRAMA DE DESPLIEGUE DEL MÓDULO DE CONFIGURACIÓN WEB.	45
FIGURA 23: DIAGRAMA DE COMPONENTES DEL MÓDULO DE CONFIGURACIÓN WEB PARA EL MECANISMO DE RASTREO DEL BUSCADOR ORIÓN.....	47
FIGURA 24: GESTIÓN DE INSTANCIAS DE NUTCH.	50
FIGURA 25: GESTIÓN DE ARCHIVOS DE CADA INSTANCIA.....	51
FIGURA 26: COMPORTAMIENTO DE LAS NO CONFORMIDADES POR CADA ITERACIÓN DE LAS PRUEBAS FUNCIONALES.....	54
FIGURA 27: CANTIDAD DE NO CONFORMIDADES DETECTADAS POR TIPO.	54
FIGURA 28: CUMPLIMIENTO DE INDICADORES DE LA LISTA DE CHEQUEO DE USABILIDAD APLICADA.	59
FIGURA 29: NIVEL DE USABILIDAD RESULTANTE DESPUÉS DE CORREGIDOS ALGUNOS INDICADORES DE USABILIDAD.	59
FIGURA 30: ESQUEMA DEL FUNCIONAMIENTO INTERNO DE NUTCH.	70
FIGURA 31: ESTADÍSTICAS HISTÓRICAS DE USO DE SERVIDORES WEB EN EL PERÍODO 1995 - 2012.	70
FIGURA 32: DIAGRAMA DE CLASES DEL DISEÑO CON ESTEREOTIPOS WEB DEL CU "GESTIONAR INSTANCIAS".	87
FIGURA 33: DIAGRAMA DE CLASES DEL DISEÑO CON ESTEREOTIPOS WEB DEL CU "GESTIONAR PLUGINS".	87
FIGURA 34: DIAGRAMA DE CLASES DEL DISEÑO CON ESTEREOTIPOS WEB DEL CU "GESTIONAR MODIFICACIONES".	88
FIGURA 35: DIAGRAMA DE SECUENCIA DEL ESCENARIO “REGISTRAR INSTANCIA” DEL CU “GESTIONAR INSTANCIAS”.	88
FIGURA 36: DIAGRAMA DE SECUENCIA DEL ESCENARIO “MODIFICAR INSTANCIA” DEL CU “GESTIONAR INSTANCIAS”.	89

FIGURA 37: DIAGRAMA DE SECUENCIA DEL ESCENARIO “ELIMINAR INSTANCIA” DEL CU “GESTIONAR INSTANCIAS”	89
FIGURA 38: DIAGRAMA DE SECUENCIA DEL ESCENARIO “REGISTRAR <i>PLUGIN</i> ” DEL CU “GESTIONAR <i>PLUGINS</i> ”	89
FIGURA 39: DIAGRAMA DE SECUENCIA DEL ESCENARIO “MODIFICAR <i>PLUGIN</i> ” DEL CU “GESTIONAR <i>PLUGINS</i> ”	90
FIGURA 40: DIAGRAMA DE SECUENCIA DEL ESCENARIO “ELIMINAR <i>PLUGIN</i> ” DEL CU “GESTIONAR <i>PLUGINS</i> ”	90
FIGURA 41: DIAGRAMA DE SECUENCIA DEL ESCENARIO “LISTAR MODIFICACIONES” DEL CU “GESTIONAR MODIFICACIONES”	90
FIGURA 42: DIAGRAMA DE SECUENCIA DEL ESCENARIO “CREAR PROPIEDAD DE PERFIL A PARTIR DE MODIFICACIÓN” DEL CU “GESTIONAR MODIFICACIONES”	91
FIGURA 43: DIAGRAMA DE SECUENCIA DEL ESCENARIO “ARCHIVAR MODIFICACIONES” DEL CU “GESTIONAR MODIFICACIONES”.	91
FIGURA 44: VISTA DE AUTENTICACIÓN DEL SISTEMA.....	92
FIGURA 45: VISTA DE GESTIÓN DE <i>PLUGINS</i>	92
FIGURA 46: VISTA DE GESTIÓN DE PERFILES DE CONFIGURACIÓN.....	93
FIGURA 47: VISTA DE GESTIÓN DE ARCHIVOS DE CONFIGURACIÓN FAVORITOS.....	93
FIGURA 48: VISTA DE GESTIÓN DE GRÁFICOS ESTADÍSTICOS RESPECTO A LAS MODIFICACIONES.....	94
FIGURA 49: VISTA DE GRÁFICO <i>SPLINE</i> DE HISTORIAL DE MODIFICACIONES.....	94

INTRODUCCIÓN

Desde 1969, con el nacimiento de la Red de la Agencia de Proyectos de Investigación Avanzada (ARPANET por sus siglas en inglés) a petición del Departamento de Defensa de los Estados Unidos, las comunicaciones a nivel mundial han trascendido de manera asombrosa. Debido a la importancia de la comunicación, como una actividad fundamental presente en cualquier proceso social, Internet¹ se ha convertido en un medio imprescindible para el desarrollo del mundo moderno. La creciente interacción de los usuarios con la gran red, ha propiciado el auge de nuevas estrategias para el intercambio de información digital, revolucionando la forma de hacer política, construir economías e influir positivamente en la cultura de millones de personas (Alarcón y Reyes, 2009).

El aumento del cúmulo de información publicada en Internet; producto del indiscriminado uso de las redes sociales, la generación constante de conocimientos y publicaciones científicas a nivel global y la necesidad de persistencia de datos personales en la nube²; ha provocado que la misma pierda facilidad de acceso o localización. En disímiles ocasiones los usuarios no saben cómo encontrar lo que realmente necesitan entre tantos documentos existentes en la Web. Por tal motivo se considera de vital importancia la disponibilidad de determinados sistemas que auxilien a las personas en esta tarea.

La Universidad de las Ciencias Informáticas (UCI), creada en el 2002 en el marco de la Batalla de Ideas y como parte de uno de los programas de la Revolución para la educación superior y la informatización de la sociedad cubana, cuenta con una infraestructura tecnológica privilegiada entre las demás instituciones nacionales, conectando en su red informática interna una gran cantidad de computadoras. Gracias al desarrollo tecnológico que posee la universidad y a la creación de sus centros de investigación y desarrollo de software, se llevan a cabo diversos trabajos de investigación y eventos de corte docente y científico que generan gran cantidad de documentación.

De acuerdo a un estudio de la evolución de la Web de la universidad a partir de análisis webmétricos en el período 2008 – 2011, la UCI contaba en ese entonces con una lista de hasta 300 direcciones en el dominio “.uci.cu” y más de dos millones de contenidos publicados entre los que se destacan páginas web con documentos portables, texto plano, imágenes, entre otros. (Mondelo, et. al., 2012)

¹ **Internet:** Conjunto descentralizado de redes de comunicación que se conectan entre si y utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

² **Nube:** Constituye un metáfora o término abreviado de Internet asociada a la computación en la nube, el cual es un paradigma que permite ofrecer servicios de computación a través de Internet y el almacenamiento de datos personales de los usuarios.

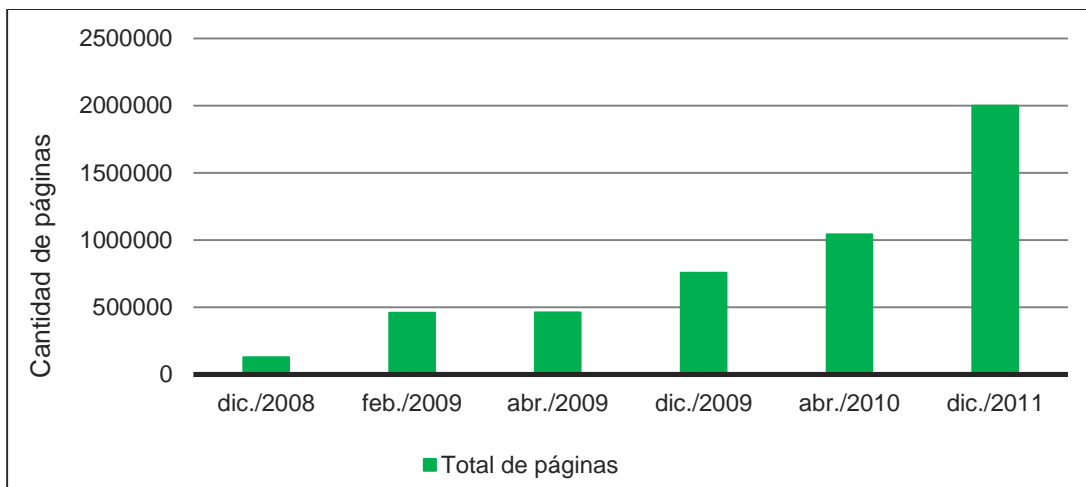


Figura 1: Resultados de los estudios webmétricos realizados en la UCI en el período 2008-2011.

Como se muestra en la figura 1 (Mondelo, et. al., 2012), la Web de la UCI estuvo creciendo aceleradamente durante esos 4 años a un nivel que cerca del 90% de las páginas publicadas fueron creadas o actualizadas en este período de tiempo. Como consecuencia de esto, existen dificultades que impiden a las personas encontrar cierta información en determinado momento.

En el centro de Ideoinformática (CIDI), perteneciente a la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI), se desarrolla el buscador Orión, el cual se encuentra en estos momentos desplegado en los servidores de la UCI y en el Ministerio de Educación Superior (MES). Este tipo de sistemas, brinda servicios de búsqueda y recuperación de información, permitiendo a los usuarios un rápido acceso a los recursos publicados en la intranet de la universidad, bajo el dominio “uci.cu” y en la red del MES respectivamente; además de tenerse como propuesta para el buscador cubano a desplegarse próximamente para el dominio de primer nivel “.cu”.

Orión cuenta con tres mecanismos principales, los que permiten acceder rápidamente a los documentos publicados en los espacios virtuales existentes (sitios web, repositorios, entre otros): una interfaz web de consulta para realizar las búsquedas, Solr como servidor de indexación de contenidos y Nutch como robot o araña. Especialmente este último, constituye el mecanismo de rastreo encargado de analizar todos los documentos publicados en la intranet a través su estructura hipertextual.

Actualmente el proceso de administración y configuración de Nutch se realiza a través de la consola del sistema operativo GNU/Linux, directamente en el servidor donde se encuentra desplegada la solución informática, lo que trae consigo un considerable gasto de tiempo y esfuerzo en dicho proceso. La conexión con el servidor anfitrión se realiza utilizando el protocolo SSH y editando cada fichero por separado, lo que implica riesgos en el surgimiento de amenazas que comprometan la estabilidad del sistema y violaciones de la política de seguridad informática de la universidad. No se tiene control de los cambios en la

configuración, por lo que las probabilidades de ocurrencia de un error crítico en el sistema aumentan considerablemente.

Este robot cuenta con más de 150 opciones de configuración y existen varias instancias del mismo, entre las que se encuentran: la instancia que permite la búsqueda general de documentos y la instancia de búsqueda de imágenes. Cada instancia desplegada duplica las opciones a tener en cuenta por los administradores; además, se deben manipular muchos ficheros distribuidos por todo el sistema operativo con numerosas propiedades en cada uno de ellos; las cuales tienen repercusión en la calidad de los recorridos y por ende, en las búsquedas realizadas por los usuarios.

La configuración siempre se realiza a partir de la experiencia de los administradores y no se cuenta con un mecanismo que aproveche la experiencia adquirida, de manera que se transmita hacia otras personas que asuman la tarea. Resulta contradictorio que con los avances logrados por las Tecnologías de la Información y las Comunicaciones (TIC), no se realice este proceso de configuración de una mejor manera.

Considerando la situación problemática anteriormente descrita, se plantea: **¿Cómo gestionar de manera centralizada las configuraciones del mecanismo de rastreo de Orión, para reducir el tiempo y el insuficiente control de cambios en dicho proceso?**; interrogante que constituye el **problema de investigación** del presente trabajo. Para la realización de la investigación se define como **objeto de estudio**: los Sistemas de Recuperación de Información; enfocándose en el proceso de configuración del mecanismo de rastreo del buscador Orión como **campo de acción**.

Para dar solución al problema planteado, se define como **objetivo general**: Desarrollar un módulo de configuración que permita gestionar de manera centralizada, a través de una interfaz web, las configuraciones del mecanismo de rastreo del buscador Orión, para reducir el tiempo y el insuficiente control de cambios en dicho proceso.

Con el propósito de cumplimentar gradualmente el objetivo general antes mencionado, el mismo se ha descompuesto en los siguientes **objetivos específicos**:

- Analizar el marco teórico de la investigación para identificar las tendencias actuales en cuanto a Sistemas de Recuperación de Información y aplicaciones web.
- Diseñar un módulo de configuración web que permita gestionar de manera centralizada las configuraciones del mecanismo de rastreo de Orión.
- Implementar el módulo de configuración web diseñado haciendo uso correcto del lenguaje y las técnicas de programación a utilizar.
- Validar el correcto funcionamiento del módulo implementado.

Después de haber tratado los elementos fundamentales del área de la ciencia a incidir y los objetivos primordiales, se formula la siguiente **hipótesis de investigación**: El desarrollo de un módulo de configuración para gestionar de manera centralizada, a través de una interfaz web, las configuraciones del mecanismo de rastreo de Orión; contribuirá al perfeccionamiento del control de cambios y a la agilización de dicho proceso. Teniendo en cuenta la hipótesis planteada se define como **variable independiente**: el módulo de configuración. Este consiste en una aplicación informática con un conjunto de funcionalidades que permiten realizar las configuraciones del mecanismo de rastreo del buscador Orión por medio de interfaces web. Como **variable dependiente** se especifica: el control de cambios. Dicha variable hace alusión al registro de información relacionada con los cambios en la configuración del mecanismo de rastreo, de manera que pueda ser consultada en el tiempo para futuras referencias. La misma a su vez, se dimensiona en usabilidad y tiempo del proceso de configuración, constituyendo estos últimos los indicadores a medir.

A partir de lo anterior, se establecen un conjunto de **tareas de investigación** que guiarán la realización del trabajo:

- Realización del levantamiento bibliográfico de los conceptos fundamentales a tener en cuenta, referenciando la literatura y los autores más destacados en el tema.
- Investigación de las tendencias más actuales sobre aplicaciones web y el uso de interfaces de usuario para la administración de Sistemas de Recuperación de Información.
- Estudio del proceso de configuración de Nutch, como mecanismo de rastreo del buscador Orión.
- Diseño del modelo de datos que permita almacenar el histórico de configuraciones realizadas por medio del sistema.
- Diseño del sistema a desarrollar, generando los artefactos de Ingeniería de Software correspondientes de acuerdo a la metodología de desarrollo, patrones y arquitectura a utilizar.
- Obtención del código fuente de la aplicación web para la gestión ágil y centralizada de la configuración de Nutch.
- Ejecución de las pruebas de software para la correcta validación de la propuesta de solución.

Para cumplimentar las tareas de investigación, se utilizaron los siguientes métodos de investigación:

- Histórico – Lógico: Con el objetivo de constatar teóricamente cómo han evolucionado en el tiempo los Sistemas de Recuperación de Información, en especial los mecanismos de rastreo en la Web; así como las herramientas y tecnologías utilizadas en el desarrollo de aplicaciones web.
- Hipotético – deductivo: Puesto en práctica en la formulación de la hipótesis de investigación y su verificación a partir del análisis de los resultados obtenidos.

- Pre experimental: Utilizado para medir el tiempo del proceso de configuración, empleando la consola del sistema operativo y posteriormente mediante el uso del módulo implementado.
- Analítico – Sintético: Empleado para el análisis de los elementos esenciales referentes a las teorías, documentos y literatura en general relacionada con los Sistemas de Recuperación de Información.
- Análisis dinámico: Aplicado en la ejecución iterativa de las funcionalidades del módulo de configuración desarrollado para la medición de su funcionamiento, permitiendo el constante perfeccionamiento del mismo.
- Análisis estático: Puesto en práctica a través de la ejecución e inspección minuciosa del módulo de configuración o determinada parte de este, para la detección de posibles errores y su posterior corrección.
- Modelación: Utilizado en la representación, mediante el uso de diagramas, de las características del sistema, relaciones entre objetos; y las actividades que intervienen en el proceso de configuración del mecanismo de rastreo de Orión.

Estructura del documento

Capítulo 1. Fundamentación teórica de los Sistemas de Recuperación de Información y las aplicaciones web: Se realiza el levantamiento bibliográfico, el estudio de los fundamentos teóricos, las tendencias actuales y principales autores relacionados con los Sistemas de Recuperación de Información como objeto de estudio del presente trabajo. También se exponen las características más generales entre otros elementos a considerar, respecto a las tecnologías de desarrollo de aplicaciones web a utilizar en el presente trabajo.

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

Capítulo 3. Implementación y validación del módulo de configuración web para el mecanismo de rastreo del buscador Orión: Se exponen algunos aspectos asociados con la implementación de la solución informática, así como los componentes que la integran. Además, se presentan los diseños de casos de prueba a utilizar en la validación del sistema y se analizan los resultados de las pruebas realizadas que permiten evaluar la calidad de la propuesta de solución.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN Y LAS APLICACIONES WEB

1.1 Introducción

A lo largo de la existencia del hombre, la información ha adquirido variadas connotaciones en diferentes campos del saber y en distintas actividades sociales. De acuerdo a los planteamientos de Santos (2009), la información está presente en las palabras, imágenes, señales, documentos, sonidos, en todo lo que se percibe y de manera consciente o inconsciente contribuye a la formación de nuevos conocimientos.

La Revista General de Información y Documentación³ explica que en principio, existen dos tipos de información: *“la que existe con independencia de la intervención o interpretación del hombre, y la que se crea y se registra como consecuencia de la interpretación que hace el hombre de este primer tipo de información”* (Castro, 2010). En un sentido más amplio el autor resalta: *“información es lo que afecta o modifica el estado intelectual del hombre, lo que da lugar a ideas, lo que permite que se genere nueva información como resultado de un proceso de interacción entre el hombre y la información registrada”*.

1.2 Fundamentos teóricos asociados al dominio del problema

1.2.1 Recuperación de Información

A lo largo de los años el concepto de Recuperación de Información o *Information Retrieval* (RI), ha sido analizado por varios autores, entre ellos Baeza-Yates y Ribeiro-Neto (1999), Pérez-Carballo y Strzalkowski (2000), Tolosa y Bordignon (2008); debido a que se trata de un campo que ha involucrado a investigadores y estudiosos de diferentes ciencias como: la Informática, Ciencia de la Información, Documentación, Lingüística, entre otras.

Baeza-Yates y Ribeiro-Neto (1999), plantean que *“la Recuperación de Información trata con la representación, el almacenamiento, la organización y el acceso a ítems de información”*. Sin embargo un año después, Pérez-Carballo y Strzalkowski (2000) indican que *“la Recuperación de Información es traer documentos relevantes desde un gran archivo en respuesta a una pregunta formulada”*.

³ **Revista General de Información y Documentación:** Creada en 1991, recoge los trabajos generados por la investigación propia y ajena a la Escuela de Biblioteconomía y Documentación de la Universidad Complutense de Madrid. Dichas investigaciones están relacionadas con las llamadas Ciencias de la Documentación en su más amplia acepción. Esta ha sido seleccionada por Elsevier para ser indizada en Scopus a partir de 2009.

Aunque se evidencia con claridad que existe determinada variación en las definiciones de los clásicos, respecto a la evolución del término RI en el tiempo, se puede constatar que existe gran similitud con la definición que plantean años más tarde Tolosa y Bordignon (2008) en su libro *Introducción a la Recuperación de Información: “la RI consiste en encontrar y rankear documentos relevantes que satisfagan la necesidad de información de un usuario, expresada en un determinado lenguaje de consulta”*.

De acuerdo con la definición anterior, los autores del presente trabajo consideran que la Recuperación de Información consiste en la presentación de información relevante desde una gran colección de documentos, a un usuario que hace una petición sobre la base del lenguaje natural, mediante la utilización de métodos, técnicas y herramientas de búsqueda que logren satisfacer sus necesidades de información.

1.2.2 Sistemas de Recuperación de Información

La diversidad de información poco estructurada y dispersa disponible en la Web, dificulta el proceso de encontrar información útil que responda a las necesidades de los usuarios. Los Sistemas de Recuperación de Información (SRI) constituyen el mecanismo ideal para resolver este tipo de problemas. Estos permiten localizar y procesar cualquier contenido existente, tales como textos, imágenes, videos, archivos de sonido, entre otros.

De acuerdo con Baeza–Yates (2005) los SRI *“deben de alguna manera interpretar el contenido de la información dentro de una colección de documentos y establecer con ellos, un orden de acuerdo al grado de relevancia que estos posean para las consultas de los usuarios”*.

Por lo general los SRI comparten una misma arquitectura, la cual se describe a continuación (López, 2006):

Interfaz: Parte de un SRI mediante el cual un usuario introduce los criterios de búsqueda y obtiene un conjunto de resultados. La misma puede ser una interfaz web, una interfaz de escritorio o ambas.

Sistema de formulación de consultas: Realiza un pre-procesamiento de las consultas trasladando las consultas hechas en lenguaje natural a consultas entendibles por los sistemas de información.

Mecanismo de evaluación de consultas: Compara los documentos representados en el sistema de información con la consulta pre-procesada, para obtener un subconjunto de documentos relevantes que satisfagan la consulta introducida por el usuario, ordenados estos de acuerdo a un criterio de relevancia.

Años más tarde, Chowdhury (2010) plantea que: “*Un Sistema de Recuperación de Información está diseñado para analizar, procesar y almacenar las fuentes de información y recuperar aquellas que coincidan con las necesidades de un usuario en particular*”; siendo esta definición la que más se ajusta al objeto de estudio tratado en la presente investigación.

1.2.2.1 Clasificación de los Sistemas de Recuperación de Información

Los Sistemas de Recuperación de Información pueden clasificarse atendiendo a disímiles criterios de acuerdo a la función que realizan, la forma de operar, el alcance que poseen, los tipos de documentos que recuperan, entre otros. Entre las clasificaciones que más destacan se encuentran: los buscadores, los metabuscadores y los directorios. Atendiendo a que Orión se clasifica como un buscador, sólo se tratarán en este trabajo los aspectos relacionados con la arquitectura y funcionamiento de este tipo de SRI.

Sin embargo, para una mejor comprensión de las semejanzas y diferencias existentes entre las clasificaciones anteriores, se puede consultar una tabla comparativa en el **Anexo # 1**, presentada en la tesis de ingeniería en Ciencias Informáticas de Parrilla (2011).

Buscadores

Los buscadores o motores de búsqueda son Sistemas de Recuperación de Información que permiten obtener aquellos documentos de mayor relevancia, a partir de un criterio de búsqueda introducido por el usuario. Cuentan con un subsistema o programa informático denominado *spider* o araña que recorre la Web de forma automática y almacena la información recopilada indizándola en una base de datos.

Desde la perspectiva del usuario, los buscadores deben cumplir dos requisitos fundamentales: “un tiempo corto de respuesta y una gran colección de documentos web disponibles en su índice. La calidad de un buscador reside en lo abundante, relevante y actualizada que sea su colección” (Baeza–Yates, 2005).

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

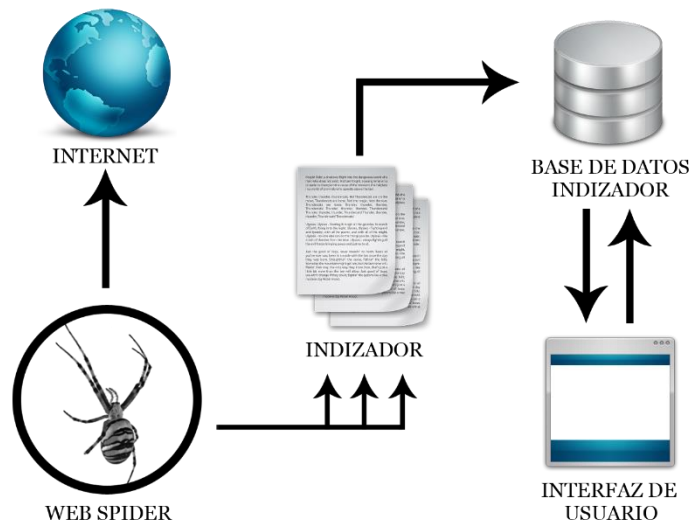


Figura 2: Arquitectura de un motor de búsqueda.

1.2.3 *Spider* o araña como mecanismo de rastreo de información

Heydon y Najork (1999), al escribir “*Mercator: A scalable, extensible Web crawler*”, describieron a los *spiders* como programas que descargan documentos de Internet de manera automatizada, los cuales también solían nombrarse con los términos: *web spiders*, *web crawler*, *bots*, robots, caminantes y vagabundos. Al mencionar cualquiera de los vocablos anteriores en el presente trabajo, se hará alusión al término *spider* o araña, por su significado en español.

Al considerar las valoraciones de otros autores de literaturas más recientes, un *spider* puede definirse como una aplicación que se conecta a Internet periódicamente y recorre la web en busca de información pública, la cual es almacenada en forma de índices para su posterior recuperación. Dicha información puede guardarse en una base de datos o en la memoria del sistema para obtener resultados más rápidos. Por último, los usuarios podrán realizar búsquedas utilizando una interfaz gráfica, la cual consultará el indizador para localizar la información requerida y devolverla en forma de lista ordenada (Kumar, et. al., 2010; Fuentes, 2011; Roclaw, 2012; Seker, 2012; Singh y Tyagi, 2013).

En entornos reales los motores de búsqueda suelen contar con varias instancias de su *spider*, distribuidos estratégicamente. De esta manera se garantiza mayor rapidez, concurrencia y distribución de las tareas de rastreo entre los mismos, así como la alta disponibilidad de los servicios de búsqueda de información.

1.2.4 Apache Nutch

Nutch es un *spider* o *web crawler* libre y de código abierto, desarrollado totalmente en Java por *The Apache Software Foundation*. Ha sido implementado sobre la base de Apache Lucene, una librería

de alto rendimiento para la búsqueda basada en texto y que utiliza una modificación del algoritmo “*Vector Space Model*” (Modelo de Espacio Vectorial en español), con un enfoque booleano que restringe las estimaciones de los resultados obtenidos (Mahecha, 2009; Apache Nutch, 2013).

Los argumentos expuestos por Mahecha (2009) demuestran que Nutch no altera los resultados obtenidos en las consultas realizadas por los usuarios, sino que trata de ofrecer los mejores resultados posibles de acuerdo al índice construido a partir de sus rastreos. Al meditar sobre este particular, y teniendo en cuenta las valoraciones de Khare y colaboradores (2004), se puede concluir que este *spider* constituye una alternativa transparente a los buscadores comerciales más difundidos; los cuales poseen fórmulas propietarias de ranking, evitando la explicación de por qué una página determinada ocupa un lugar relevante en los resultados.

1.2.4.1 Arquitectura y funcionamiento de Nutch

Para llevar a cabo las tareas de rastreo e indización de los documentos encontrados en la red, Nutch realiza una secuencia de procesos internos que garantizan la indización de cada uno de los enlaces a las páginas web y la extracción adecuada de los contenidos verdaderamente relevantes que serán consultados por las búsquedas de los usuarios.

Cada proceso es ejecutado por uno o varios de los componentes que integran la arquitectura del *web crawler*. De manera general, se pueden enunciar cuatro componentes fundamentales que le dan vida a Nutch: Indizador, Base de Datos Web, Selector y Buscador; a los cuales también se les suele llamar por sus respectivos términos en inglés: *Indexer*, *Crawler DB*, *Fetcher* y *Searcher* (Khare, 2004; Mahecha, 2009).

Julien Nioche, especialista en Ingeniería de Texto, Procesamiento del Lenguaje Natural y la Recuperación de Información; describe los procesos internos de Nutch de la forma siguiente (Nioche, 2013), véase además la figura del **Anexo # 2**:

1. **Inyección (*Inject* en inglés):** Obtención de las URLs contenidas en el semillero inicial de Nutch, para la inicialización de la base de datos web y el comienzo del rastreo.
2. **Generación (*Generate* en inglés):** Se genera una nueva lista de selección de URLs para el segmento. En el primer rastreo solo se toman los links del semillero inicial, y en los demás se agregan los nuevos enlaces encontradas en los rastreos anteriores. Además, el generador es el encargado de definir la frecuencia de visitas a las páginas.
3. **Selección (*Fetch* en inglés):** Selecciona una de las URLs de la lista generada, trata de establecer una conexión con la página para su descarga y la almacena en el segmento de rastreo.

4. **Análisis (*Parse en inglés*):** Una vez descargada la página, se procede al análisis de su contenido (texto y metadatos). Es aquí donde se descubren los nuevos enlaces en el contenido.
5. **Actualización de la Base de Datos (*Update CrawlDB and LinkDB en inglés*):** Al terminar el análisis, Nutch actualiza la base de datos web con el contenido encontrado y añade los nuevos enlaces a la base de datos de URLs.
6. **Indización (*Indexer en inglés*):** Al finalizar los procesos anteriores, Nutch envía toda la información a la aplicación que se encargará de indizar todo el contenido, para su posterior consulta.

1.2.4.2 Configuración de Nutch

Para realizar las operaciones pertinentes, Nutch almacena todos los documentos de forma organizada, atendiendo al contenido del archivo y la función que realiza en las tareas de rastreo. El *spider* posee ocho directorios principales: Bin, Crawl, Docs, Lib, Logs, Conf, Plugins y Urls, siendo los tres últimos los que contienen la mayoría de los archivos de configuración que más inciden en la calidad del funcionamiento del mismo.

Los archivos de configuración de Nutch, distribuidos por toda su estructura de directorios, se encuentran por lo general en formato XML, XSD, XLS, DTD o TXT. Cada tipo de archivo tiene una función diferente, aunque en muchas ocasiones existen algunos que responden a tareas homólogas y se encuentran en formatos distintos; esto se debe a que muchos de los *plugins* han sido desarrollados por la comunidad de desarrolladores de todo el mundo, lo cual dificulta el establecimiento de un estándar en cuanto a la codificación.

De los tipos de ficheros antes mencionados, sólo se trabajará en la presente investigación, con los archivos en formato XML, por ser los que intervienen directamente en el correcto funcionamiento de las tareas de rastreo más importantes del *spider*.

Cabe mencionar que actualmente, cada uno de estos archivos sólo pueden ser modificados utilizando un editor de texto, ya sea visual o mediante la consola del sistema operativo. Este hecho entorpece la calidad y rapidez del proceso de configuración de cada instancia del *web crawler*, teniendo en cuenta además, el excesivo abuso de la memoria humana de los administradores del sistema, los cuales, deben memorizar constantemente cuáles son las últimas modificaciones que se han realizado en cada archivo y el motivo del cambio. Además, también se dificulta el registro de las configuraciones históricas, de las que se pueden obtener la configuración más adecuada para cada instancia de acuerdo a los requerimientos del entorno de despliegue.

1.2.4.3 Interfaz de administración web para Nutch

La administración y configuración de forma manual y mediante la línea de comandos es una de las grandes desventajas que posee Nutch ante sus principales competidores. Esto provoca cierto desánimo y dificultades a los usuarios que se insertan en el mundo de la búsqueda y Recuperación de Información.

Por estas razones, en el 2006, Hanze, Bauhard y Groschupf comienzan el desarrollo de una interfaz de administración web para el *web crawler*. El propósito fundamental era ampliar las capacidades funcionales de Nutch con una confortable interfaz de usuario para el monitoreo, configuración y administración de una o más instancias del *spider* a través de una API-REST (Nutch Wiki, 2013).

Entre las principales funcionalidades propuestas se encontraban: el monitoreo del estado del sistema y sus funciones; la configuración de las instancias y parámetros que intervienen en los recorridos por la Web y la administración de las tareas de rastreo.

Debido a varios problemas de integración entre Nutch y las primeras versiones de la interfaz de administración, se discontinuó el desarrollo de la misma en el 2010, según comunican algunas de las conversaciones entre miembros de la comunidad y los autores, plasmadas en el sitio del proyecto y en GitHub⁴, donde reside la última actualización del código fuente (GitHub, 2010; The Apache Software Foundation, 2011).

Debido a las razones antes mencionadas y poseer un alcance mucho mayor que el presente trabajo, el proyecto descrito anteriormente no constituye una solución al problema de esta investigación. Sin embargo, se tuvieron en cuenta algunas de las funcionalidades y características presentes en los diseños de la interfaz. Entre ellas se destacan: la organización del sistema por pestañas, la gestión de las instancias y sus parámetros de configuración, el registro de cambios en la configuración y la gestión de las URLs iniciales.

1.2.5 Sistemas para la configuración de *spiders*

Alrededor del mundo son muchos los spiders que existen y que han sido creados para suplir diversas necesidades en particular. Sin embargo, una característica común entre ellos es su complejidad a la hora de configurar sus componentes internos para realizar las tareas de rastreo. Ello se debe fundamentalmente a la gran cantidad de opciones de configuración que presentan, las cuales responden a las necesidades específicas para las que se concibieron. Atendiendo a este trabajo,

⁴ **GitHub**: Repositorio online de código abierto.

sólo se analizarán algunos *spiders* que se encuentran liberados como software libre, y que los autores consideraron el oportuno análisis como parte de los fundamentos teóricos.

1.2.5.1 Open Search Server

Open Search Server es un motor de búsqueda de código abierto, multiplataforma, desarrollado en Java bajo la licencia GNU Public Licence versión 3 (GPLv3). Entre sus características principales destaca el uso de un conjunto de algoritmos de análisis de texto completo que permiten la extracción de información no sólo desde documentos en diferentes formatos, sino también desde varias bases de datos donde el contenido puede ser indizado en múltiples idiomas. La búsqueda fonética, la comprobación ortográfica y la personalización de la relevancia de las búsquedas mediante funciones algebraicas, son algunas de las funcionalidades avanzadas que pueden ser extendidas como parte de su arquitectura basada en plugins. El proceso de configuración se realiza por medio de una interfaz web de administración como se muestra en la figura 3 o modificando cada uno de los parámetros desde un conjunto de archivos XML (OpenSearchServer, 2014).

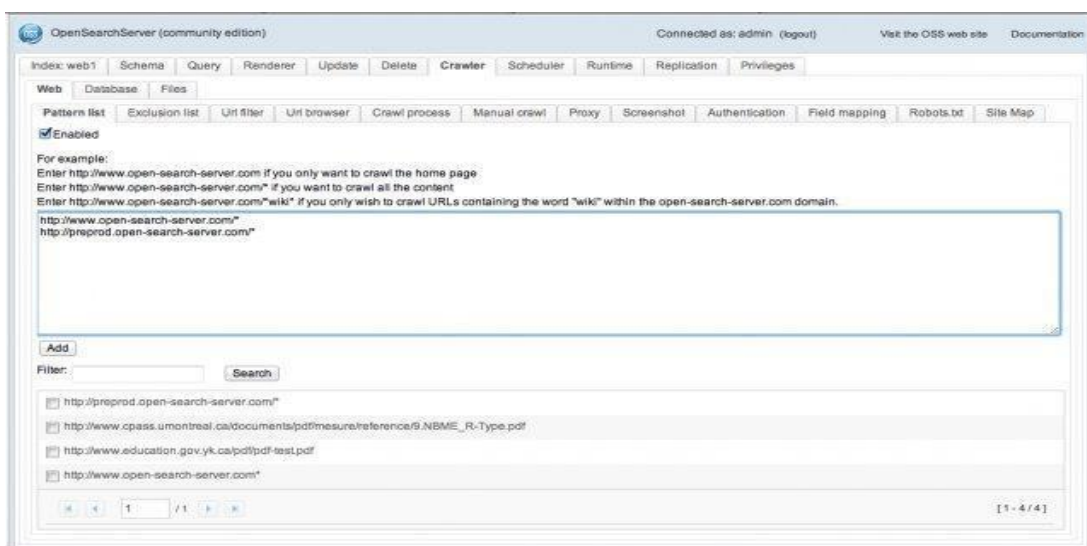


Figura 3: Interfaz de configuración de Open Search Server.

1.2.5.2 mnoGoSearch

Inicialmente denominado UDMSearch hasta la adquisición del proyecto por la compañía Lavtech.Com en el año 2000; es un motor de búsqueda de código abierto basado en SQL⁵ y desarrollado utilizando el lenguaje de programación C. En principio el sistema consiste de dos partes: un mecanismo de indización multihilo que utiliza un índice invertido de texto completo y se encarga de la recopilación de la información contenida en cada uno de los documentos encontrados a partir

⁵ **SQL**: Acrónimo de *Structured Query Language*, cuyo significado en español se le confiere a Lenguaje de Consulta Estructurado. Es un lenguaje utilizado para el trabajo con bases de datos relacionales.

de los enlaces hipertextuales y en segundo lugar, una interfaz web de consulta donde se muestran los resultados de la búsqueda al usuario. Entre sus características principales, además de ser multiplataforma, soporta gran variedad de formatos de documentos de texto, búsqueda booleana, búsqueda utilizando lógica difusa y conversión de caracteres así como la indización de documentos en diferentes idiomas dentro de una misma base de datos. En dependencia de las características incluidas el sistema se distribuye bajo las licencias GNU Public Licence versión 2 (GPLv2) y Mnogosearch Search Engine para Windows⁶. El proceso de configuración se realiza a partir de un conjunto de comandos predefinidos en un archivo de texto o mediante una interfaz de administración como se muestra en la figura 4, sólo disponible para distribuciones del sistema operativo Windows (Lavtech.Com Corp, 2014).

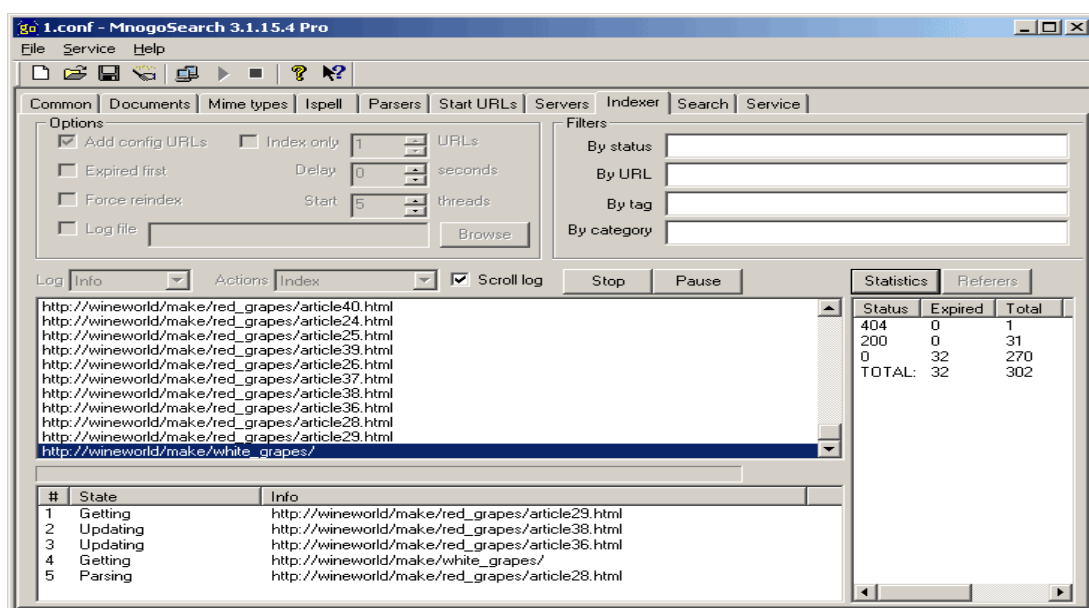


Figura 4: Interfaz de configuración de mnoGoSearch.

1.2.5.3 Lucene

Apache Lucene es una librería para motores de búsqueda basada en texto, escrito completamente en Java. Puesto que se trata de un software gratuito y multiplataforma, bajo la Licencia Apache 2.0, se presenta como una tecnología ideal para proyectos libres y aplicaciones que requieran la búsqueda de texto completo. Entre sus características más connotadas se pueden mencionar: bajo consumo de recursos, aplicación de múltiples algoritmos eficientes para la búsqueda e indización de texto (incluyendo el Modelo de Espacio Vectorial), permite la búsqueda de texto utilizando diferentes

⁶ **Mnogosearch Search Engine para Windows:** Se refiere a la licencia “Mnogosearch Search Engine For Windows License Agreement” dado su significado en inglés.

critérios, presenta una arquitectura extensible a través de *plugins* y posee una interfaz de configuración como se muestra en la figura 5 (Apache Lucene, 2014).

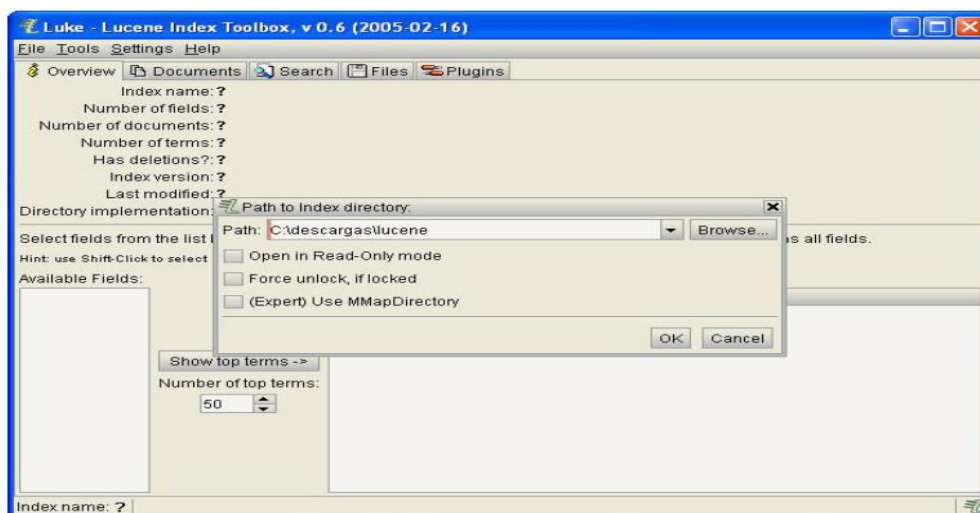


Figura 5: Interfaz de configuración de Apache Lucene.

1.2.5.4 Comparación entre Spiders

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

Tabla 1: Comparación entre *spiders*.

Spider	Lucene	Nutch	Open Search Server	mnoGoSearch
Configuración	Interfaz de configuración	Ficheros XML	Ficheros XML, Interfaz de configuración	Líneas de Comando (Fichero de Texto), Interfaz de configuración (Windows)
Lenguaje de programación	Java	Java	Java	C
Multihilo	Si	Si	Sí	Sí
Extensible (uso de <i>plugins</i>)	Si	Si	Sí	No
Plataformas	Windows, Mac OS, GNU/Linux, Unix	Windows, Mac OS, GNU/Linux, Unix	Windows, GNU/Linux, Mac OS, Solaris.	Windows, GNU/Linux, FreeBSD, Mac OS, Solaris.
Licencia	Licencia Apache 2.0	Licencia Apache 2.0	GNU Public Licence (GPLv3)	GNU Public Licence (GPLv2), Mnogosearch Search Engine For Windows License Agreement

Atendiendo al estudio y comparación de los *spiders* anteriores, se determinó que las interfaces de configuración desarrolladas para estos sistemas no constituyen soluciones potenciales a la problemática planteada en este trabajo, dado que cada uno responde a necesidades particulares del *web crawler* para el que fue concebido.

Sin embargo, se identificaron aspectos de interés que pueden ser utilizados en la concepción de la solución informática que se propone en este trabajo. Entre los elementos positivos a considerar se destacan: la distribución del contenido por pestañas, la arquitectura de información y las diferentes formas de interacción entre el usuario y el sistema que contribuyen a la usabilidad de la aplicación. Además, se identificaron como aspectos negativos: la disponibilidad de algunos de ellos para una única plataforma y el poco aprovechamiento de las potencialidades de las tecnologías web, obligando al usuario a usar interfaces poco atractivas visualmente.

1.2.6 Aplicaciones web

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

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

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

1.2.6.1 Interfaces web

Con el surgimiento de las aplicaciones web, las interfaces web marcaron un nuevo hito en la evolución de las interfaces gráficas de usuario (GUI) (del inglés *Graphical User Interface*), aportando elementos propios de presentación y organización de la información; los cuales, en muy poco tiempo, se convirtieron en estándares del diseño web (Hernández y Greguas, 2008). Este nuevo tipo de interfaces permitió romper los esquemas de interfaces de escritorios y líneas de comandos, captando la atención de muchas personas gracias a su gran atractivo y facilidad de personalización.

En el libro *“Programación de aplicaciones web: historia, principios básicos y clientes web”*, el autor se refiere de manera general a la interfaz web como una capa de presentación utilizada por las páginas web, que funcionan como intermediarias entre el usuario final y un software ejecutado en un servidor web remoto (Luján, 2002).

Este tipo de interfaces están soportadas generalmente por un lenguaje de marcas (usualmente HTML) utilizado por los navegadores o clientes web para mostrar la información proporcionada por el servidor. En su esencia, posibilitan a los usuarios inexpertos el uso de aplicaciones informáticas, la fácil navegación y localización de la información deseada, la comprensión adecuada de las funcionalidades que se ofrecen y la realización de las actividades específicas; garantizando la transparencia de todas las operaciones internas ante los usuarios.

1.3 Tecnologías y herramientas seleccionadas

Una de las características fundamentales de las aplicaciones web, es el uso de la arquitectura Cliente – Servidor, la cual, además de ser empleada comúnmente para el despliegue, describe con claridad la filosofía de funcionamiento de este tipo de aplicaciones. A continuación se describen las tecnologías seleccionadas para la implementación del módulo de configuración web. Estas tecnologías están clasificadas atendiendo a las tipologías que propone la arquitectura mencionada. Además, se enuncian otras herramientas necesarias para cumplimentar determinadas actividades dentro del proceso de desarrollo del software.

1.3.1 Tecnologías del lado del Cliente

Las operaciones que se realizan en el lado del cliente, generalmente son ejecutadas por mediación de un navegador web. Estas operaciones suelen ser muy rápidas y seguras, puesto que en la mayoría de los casos no es necesario el procesamiento complejo de los datos manipulados. Lo anterior permite reducir la carga de procesamiento que recibe el servidor y la implementación de funcionalidades que doten a las aplicaciones de una mayor usabilidad e interactividad.

HTML

HTML, acrónimo de *HyperText Markup Language*, es un lenguaje de publicación especificado como un estándar por el W3C (*World Wide Web Consortium*) que permite la creación de páginas web. Inicialmente fue presentado por Tim Berners-Lee que propuso un sistema basado en hipertexto como resultado de la aplicación de SGML para el intercambio de información en la Web. La aparición del lenguaje influyó notablemente en el crecimiento de Internet, dónde la información era distribuida mediante colecciones fragmentadas de textos, imágenes y sonidos. HTML es independiente de la plataforma utilizada y se basa fundamentalmente en el uso de etiquetas estructurales y semánticas, adecuadas para la creación de documentos relativamente simples que permiten simplificar su estructura (World Wide Web Consortium, 2014).

CSS

Con el crecimiento de Internet y la aparición del lenguaje HTML para la creación de páginas web, el W3C demostró la necesidad de un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos creados, de manera que pudieran visualizarse de igual forma en cualquier navegador web. A partir de entonces surgieron entre varias propuestas los lenguajes CHSS (*Cascading HTML Style Sheets*) y SSP (*Stream-based Style Sheet Proposal*), el primero realizado por Hakon Wium Lie y el segundo por Bert Bos, que a finales de 1994 y 1995 se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta denominado CSS (*Cascading Style Sheets*, por sus siglas en inglés).

Actualmente el equipo de desarrollo trabaja en una tercera versión del lenguaje (CSS3), del cual se han liberado algunos módulos y otros se encuentran en fases de prueba y revisión (World Wide Web Consortium, 2014).

JavaScript

Es un lenguaje de programación interpretado desarrollado inicialmente por *Netscape Communications Corporation*. Consiste en código que puede ser insertado dentro de un documento HTML, con el objetivo de proporcionarle cierto dinamismo a las páginas web. JavaScript no constituye en su concepción un lenguaje de programación multipropósito, pues no permite crear aplicaciones independientes. Sin embargo, es capaz de controlar determinados objetos que componen el entorno en el que se desenvuelve, en los documentos HTML se evidencia este hecho con la modificación del DOM utilizando código script. (Mozilla Project, 2013)

Uno de los aportes más significativos de JavaScript como lenguaje, son todas las librerías que han sido codificadas y que permiten agregarle mayor dinamismo, estética, funcionalidad y usabilidad a los sitios web; entre ellas se destacan: ExtJS, JQuery, Node.JS, entre otras.

AJAX

El desarrollo de *Aplicaciones de Internet Enriquecidas* o *Rich Internet Applications (RIA)*, revolucionó el modo en que la información era presentada al usuario por medio de las tradicionales páginas web estáticas. En este sentido, AJAX, acrónimo de *Asynchronous JavaScript And XML*, destaca entre las tecnologías utilizadas para el intercambio asíncrono de información entre cliente y servidor. En lo esencial, se trata de un conjunto de tecnologías, entre las que se encuentran XHTML, DOM, XML, XSLT, JSON, XMLHttpRequest y JavaScript; con las que se logra el desarrollo de aplicaciones web dinámicas e interactivas, evitando el envío de peticiones continuas al servidor y la recarga constante

de la página. El uso de AJAX mejora la usabilidad e interactividad de las aplicaciones web y potencialmente reduce el tráfico de red (Singh, 2012).

1.3.2 Tecnologías del lado del Servidor

De manera usual, las tecnologías en el lado del servidor se encargan básicamente de atender las solicitudes de los clientes. Es aquí donde se realiza la mayor parte del procesamiento de la información solicitada, generando una respuesta que puede ser mostrada en formato HTML, XML u otro, que facilite la comprensión por el usuario. Aunque generalmente desempeñan un papel pasivo en la comunicación que se establece en dicha arquitectura, pueden atender un gran número de peticiones al mismo tiempo procedentes de muchos clientes web y además, operar con disímiles tecnologías que se integran entre sí.

PHP

PHP es un lenguaje de alto nivel e interpretado, utilizado en su mayoría para el procesamiento dinámico de información en la Web. El mismo, cuyo significado se le confiere a “*PHP Hypertext Preprocessor*” por sus siglas en inglés, puede ser incrustado en documentos HTML, pero sólo puede ejecutarse en el lado del servidor (Bakken, 2013). Por ser de código abierto, posee una amplia comunidad internacional que colabora en el mejoramiento del código fuente, el desarrollo y actualización de librerías para el lenguaje y la traducción de la documentación del proyecto.

Su simplicidad y similitud en la sintaxis con la de C, Java y Perl, permite a los programadores aprenderlo y comenzar a usarlo rápidamente en el desarrollo de aplicaciones para la arquitectura cliente – servidor. Por tal motivo, se ha comportado como uno de los lenguajes de programación más utilizados a nivel mundial. Este hecho se encuentra aparejado al surgimiento de *frameworks* de desarrollo que proporcionan mayor comodidad y rapidez a la hora de codificar un proyecto y establecer su arquitectura. Entre los *frameworks* más populares se encuentran: Symfony2, CodeIgniter, Zend Framework, Yii y Silex.

Para la obtención del código fuente del módulo a desarrollar se usará la versión 5.4.3 de PHP. Dicha elección se basa fundamentalmente en las características y múltiples ventajas que proporciona; teniendo en cuenta además, que todos los servicios del buscador Orión, orientados al usuario final, han sido implementados utilizando en su mayoría este lenguaje, lo cual favorece en principio, el uso de una tecnología homogénea, sin dejar de aprovechar sus potencialidades.

1.3.3 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o IDE (siglas en inglés de *Integrated Development Environment*) consiste en un conjunto de herramientas de programación puestas a disposición como un programa

informático. En otras palabras, puede estar compuesto por un editor de código, un compilador, un intérprete, un sistema de apoyo al control de versiones, un constructor de interfaces gráficas, entre otras herramientas comúnmente utilizadas para el desarrollo de software.

NetBeans

Liberado bajo el licenciamiento dual de CDDL y GPL (versión 2), NetBeans es un potente IDE para programadores que proporciona una plataforma ideal para escribir, compilar, depurar y ejecutar programas informáticos (ORACLE, 2014). Aunque inicialmente fue ideado para Java, puede ser empleado para la codificación de aplicaciones en múltiples lenguajes de programación. Este, además de ser gratuito y sin restricciones de uso, posee versiones para los distintos sistemas operativos del mercado, convirtiéndolo en una alternativa con grandes ventajas para los desarrolladores.

La estructura modular de NetBeans le proporciona estabilidad y grandes posibilidades de ser extendido gradualmente por desarrollos comunitarios, permitiendo agregar continuamente nuevas funcionalidades. Su versatilidad lo ha convertido en el IDE por excelencia entre miles de programadores alrededor del mundo.

1.3.4 Sistema de Gestión de Base de Datos

Este tipo de sistema tiene la función principal de servir de interfaz de comunicación entre el usuario final, la base de datos y las aplicaciones que la utilizan. De esta manera, se garantizan requisitos indispensables en toda aplicación que requiera la persistencia de grandes volúmenes de datos, por ejemplo: la independencia de la lógica de programación respecto a la forma de almacenamiento, la seguridad de la información, la abstracción de las operaciones, la seguridad y la rapidez de las respuestas a consultas. Si bien existen varios tipos de bases de datos, atendiendo a su estructura y a la forma en que almacenan los datos, todas contribuyen al aumento de la calidad de los sistemas informáticos.

MongoDB

MongoDB es un sistema de código abierto escrito en C++, orientado al almacenamiento de datos al estilo JSON con esquemas dinámicos. La estructura se compone fundamentalmente por Colecciones de Documentos, conformándose estos últimos por Campos que almacenan los datos de interés (MongoDB, 2013). Esta estructura posee cierta analogía con los esquemas de bases de datos SQL, con la diferencia de que no se definen estrictamente los campos de cada documento, véase figura 6.

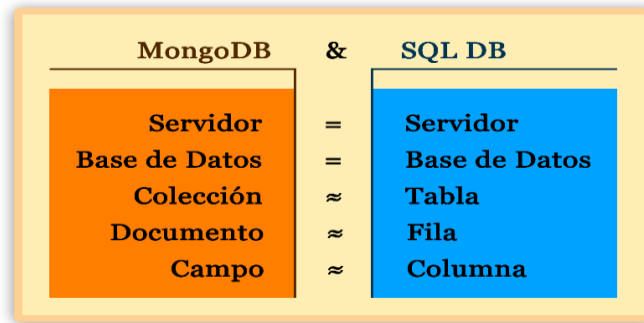


Figura 6: Analogía entre MongoDB y los Sistemas de Gestión de Bases de Datos SQL.

Se decide utilizar MongoDB como Sistema de Gestión de Base de Datos (SGBD) dado que se presenta como una solución escalable y de alto rendimiento en el manejo de información frente a grandes volúmenes de datos; pues almacenará el histórico de cambios en los archivos de configuración y otros datos referentes a perfiles de configuración. También cabe mencionar que el buscador Orión utiliza MongoDB para el almacenamiento de determinada información referente a las consultas de búsqueda, lo que proporciona ventaja en el uso de un mismo tipo de base de datos, evitando sobrecargar el servidor con futuras instalaciones de varios SGBD. Puede ser integrado fácilmente con el lenguaje de programación seleccionado, además de que el marco de trabajo a utilizar, Symfony2, presenta soporte para MongoDB.

1.3.5 Servidor web

Un servidor web, en términos de aplicaciones informáticas, constituye un programa que permite alojar aplicaciones que funcionan comúnmente bajo la filosofía de la arquitectura Cliente – Servidor. Estos sistemas se encargan de procesar las peticiones de los clientes web, para luego devolver una respuesta con los datos procesados, y mostrarlos mediante el uso de un lenguaje de marcas, usualmente HTML.

Apache 2

Apache 2 es un servidor web de código abierto que implementa el protocolo HTTP 1.1, caracterizado fundamentalmente por su alto nivel de configuración, modularidad, robustez y estabilidad. Desarrollado bajo la licencia ASF por *The Apache Software Foundation*, es considerado una de las mejores y más aceptadas creaciones del mundo del software libre (NetCraft, 2013).

Teniendo en cuenta las estadísticas históricas y uso diario proporcionadas por NetCraft (2013), este servidor llegó a usarse durante el 2005 en el 70% de los sitios web a nivel mundial, representando su cuota máxima en el mercado hasta la actualidad. Aunque ha sufrido un descenso en dicha cuota, aún se mantiene en el tope de los servidores web más utilizados, ver **Anexo # 3**.

Las estadísticas presentadas demuestran que Apache 2 ha sido considerado como el servidor web HTTP por excelencia, logrando que millones de servidores a nivel mundial ratifiquen su utilización.

Se decide utilizar Apache 2 como servidor web, ya que además de poseer grandes ventajas sobre sus competidores (INTECO – CERT, 2012; Apache HTTP Server, 2013), es el utilizado para el despliegue de la interfaz web del buscador Orión. Esto permitirá instaurar el módulo a implementar en el mismo servidor donde se encuentre el motor de búsqueda, aprovechando al máximo los recursos disponibles.

1.3.6 Herramienta CASE

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

Visual Paradigm

Visual Paradigm es una herramienta CASE multiplataforma, que soporta el ciclo completo de desarrollo de software: análisis, diseño, implementación y pruebas. Facilita la construcción de aplicaciones informáticas con un menor coste que destacan por su alta calidad y contribuye a mejorar la experiencia de usuario mediante el diseño de un gran número de artefactos de ingeniería de software. Permite la generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería directa e inversa, la gestión de requisitos de software y la modelación de procesos del negocio (Visual Paradigm, 2014).

1.4 Lenguaje de modelado

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

1.5 Metodología de desarrollo

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

OpenUp

OpenUp, acrónimo de *Open Unified Process*, es un modelo de desarrollo de software de código abierto, creado por la fundación Eclipse como parte del Framework de Modelo de Procesos de Eclipse (*Eclipse Process Framework*). Diseñado para el desarrollo de proyectos con un enfoque ágil mediante un proceso iterativo e incremental, que puede ser extendido en grandes o pequeñas formas para adicionar nuevos contenidos de desarrollo o personalizar el proceso para un entorno específico. Puede ser aplicado en equipos pequeños, por lo general de tres a seis integrantes, y se organiza dentro de cuatro áreas principales de contenido: Comunicación y Colaboración, Intención, Solución, y Administración. El ciclo de vida del proyecto se estructura en cuatro fases: Concepción, Elaboración, Construcción y Transición que podrán tener tantas iteraciones como se requiera dependiendo del grado de novedad del dominio de negocio, de la tecnología a ser utilizada, de la complejidad de la arquitectura de la solución y del tamaño del proyecto, entre otros factores (Eclipse Process Framework, 2014).

Las características del equipo de desarrollo de este trabajo propician una correcta adecuación con la metodología mencionada, dado que se está en presencia de un equipo pequeño que requiere la constante comunicación con el cliente y que deberá realizar el proceso de implementación de forma iterativa según las necesidades de trabajo y requisitos previos. Además, es necesario destacar que dicha metodología está contenida en la base tecnológica definida por el centro de desarrollo al cual pertenecen los desarrolladores. Para la obtención de algunos artefactos, se hará uso de determinadas plantillas que propone el expediente de proyecto definido por la universidad.

1.6 Marco de trabajo

Un marco de trabajo o “*framework*” como popularmente se le conoce, puede definirse como un conjunto bien estructurado y organizado de librerías que ponen a disposición determinadas funcionalidades básicas implementadas con anterioridad, y que permiten a los programadores

centrarse objetivamente en las nuevas funcionalidades a desarrollar. Usualmente cada *framework* posee una filosofía o metodología de trabajo estándar, así como uno o varios patrones de diseño e implementación que definen cómo utilizarlos de la manera más eficiente posible. Su estructura a menudo suele tratarse como el esqueleto o núcleo para la construcción de nuevas aplicaciones, posibilitando el desarrollo basado en módulos, garantizando así la futura extensibilidad de los sistemas.

Symfony2

Symfony2 es un *framework* PHP basado en componentes, ideado básicamente para ayudar a los programadores en el desarrollo ágil y la construcción de aplicaciones web más robustas y de alto rendimiento (SensioLabs, 2013). Su estructura modular, altamente flexible y adaptable, se compone de la reutilización de una amplia gama de tecnologías, librerías y buenas prácticas de la programación web que lo convierten en uno de los *frameworks* PHP con mejor rendimiento y mayor versatilidad.

Dentro de las características propias que lo distinguen entre la gran variedad de marcos de trabajo para PHP (Eguiluz, 2013) y que influyeron en su elección para la realización de este trabajo, se destaca su fácil integración con cualquier sistema gestor de base de datos, tanto para esquemas relacionales como no relacionales; gran extensibilidad, permitiendo la integración con librerías desarrolladas por la comunidad internacional; muy versátil, pues sirve para desarrollar cualquier tipo de aplicación web, ya sea grande, mediana o pequeña y además; posee una buena documentación que ayuda a las personas con experiencia y a las que deciden iniciarse con este *framework*.

La elección de Symfony2 (versión 2.2.6) para la implementación del módulo propuesto, propiciará una correcta integración con el motor de búsqueda Orión, logrando a su vez cierta homogeneidad con los demás componentes del sistema.

Doctrine

Doctrine (versión 2.3) es una potente librería para ORM (*Mapeo Relacional de Objetos en español*) diseñado especialmente para PHP 5.2 o superior y que posee una Capa de Abstracción de Base de Datos (DBAL por sus siglas en inglés) (SensioLabs, 2013).

Entre sus más reconocidas funcionalidades, se destaca la posibilidad de exportación de una base de datos existente a sus clases PHP correspondientes y también a la inversa, siguiendo las pautas del ORM. Dado que utiliza el patrón *Active Record*, el mismo permite la manipulación de la base de datos con el uso de la programación orientada a objetos (POO). Gracias a que posee su propio lenguaje de consulta, que consiste en tratar los datos como objetos; se garantiza la independencia

de las aplicaciones web hacia los sistemas gestores de bases de datos y un rendimiento óptimo. Además de esto, se han implementado múltiples extensiones que posibilitan la interacción con los diferentes tipos de bases de datos, facilitando así el trabajo de los programadores. Atendiendo a los aspectos antes mencionados, Doctrine ha llegado a popularizarse entre los desarrolladores web, permitiendo integrarse fácilmente con los principales *frameworks* de desarrollo.

1.7 Conclusiones parciales

Durante el análisis de los preceptos teóricos abordados en el capítulo, se pudo constatar que existe gran diversidad de Sistemas de Recuperación de Información, atendiendo a sus clasificaciones y uso. Los mismos constituyen sistemas complejos, con un gran número de opciones de configuración y muy difíciles de administrar.

En la evaluación de las tendencias de aplicaciones similares a la propuesta de solución presentada en este trabajo, fueron identificados sistemas con un alto nivel de aceptación; sin embargo, debido a que cada uno responde a las necesidades particulares del *spider* para el que fue concebido, no se consideran posibles soluciones para Nutch como mecanismo de rastreo. Aun así, fue posible identificar a partir de ellos, nuevos elementos que tributan al perfeccionamiento de la propuesta de solución.

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

CAPÍTULO 2. DISEÑO DE LA PROPUESTA DE SOLUCIÓN

2.1 Introducción

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

2.2 Modelado del Dominio

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

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

2.2.1 Descripción de Clases del Modelo del Dominio

La modelación del dominio constituye la herramienta fundamental para garantizar la comprensión y descripción de las clases o conceptos y sus relaciones más importantes dentro del contexto del problema. A continuación se presenta la descripción de los conceptos identificados.

Tabla 2: Descripción de las clases del modelo del dominio.

Concepto	Descripción
Administrador	Persona que realiza las labores de administración y configuración del mecanismo de rastreo (Nutch) de Orión.
Instancias Nutch	Constituyen cada una de las instancias de Nutch concebidas para rastrear determinados contenidos específicos en la Web.
Plugins	Se refiere a componentes externos que se le agregan a Nutch para extender su funcionamiento.

Archivos de Configuración	Son todos los archivos que contienen opciones de configuración, cuyo valor puede afectar positiva o negativamente el resultado de las tareas de rastreo. Pueden ser de diferentes formatos.
Archivo XML	Se refiere a los archivos de configuración escritos en formato XML.
Archivo TXT	Se refiere a los archivos de configuración escritos en formato TXT.

2.2.2 Diagrama de Clases del Modelo del Dominio

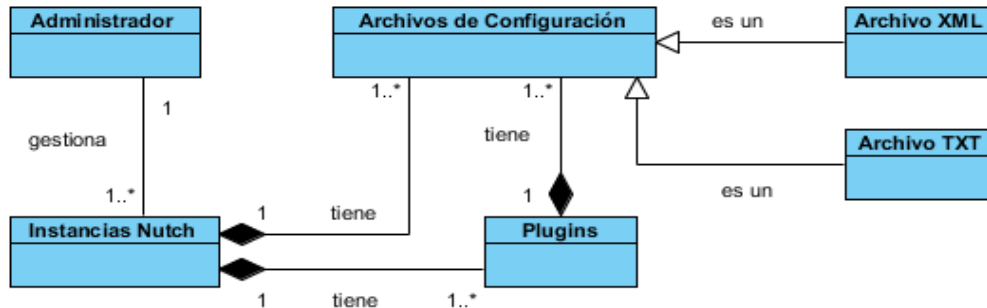


Figura 7: Diagrama de clases del modelo del dominio.

El modelo del dominio representado en la figura anterior, muestra la relación existente entre los objetos fundamentales que intervienen en la configuración de Nutch como *spider web*. Inicialmente se observa como el administrador puede gestionar una o más instancias de Nutch. Este último posee en su estructura interna cierta cantidad de *plugins* y archivos de configuración que definen el comportamiento del *spider* en las tareas de rastreo. Por último, los archivos de configuración pueden encontrarse en distintos formatos de escritura, según la función para la que se utilicen.

2.3 Especificación de los Requisitos del Software

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

Una vez definidos los conceptos principales relacionados con el dominio, se muestran a continuación los requerimientos funcionales y no funcionales, de acuerdo con el objetivo planteado al inicio de este trabajo.

2.3.1 Requisitos funcionales

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

Requisitos funcionales del módulo de configuración web para el mecanismo de rastreo del buscador Orión:

- RF 1. Gestionar instancias de Nutch.
 - RF 1.1. Mostrar listado de las instancias de Nutch registradas.
 - RF 1.2. Mostrar detalles de una instancia de Nutch.
 - RF 1.3. Registrar los datos de una nueva instancia de Nutch.
 - RF 1.4. Modificar los datos de una instancia de Nutch.
 - RF 1.5. Eliminar los datos de una instancia de Nutch.
- RF 2. Gestionar los *plugins* pertenecientes a las instancias de Nutch.
 - RF 2.1. Mostrar listado de los *plugins* registrados.
 - RF 2.2. Mostrar detalles de un *plugin*.
 - RF 2.3. Registrar los datos de un nuevo *plugin*.
 - RF 2.4. Modificar los datos de un *plugin*.
 - RF 2.5. Eliminar los datos de un *plugin*.
 - RF 2.6. Registrar automáticamente los *plugins* de Nutch activos.
 - RF 2.7. Activar *plugin*.
 - RF 2.8. Desactivar *plugin*.
- RF 3. Gestionar los archivos de configuración pertenecientes a los *plugins* e instancias de Nutch.
 - RF 3.1. Mostrar listado de los archivos de configuración registrados.
 - RF 3.2. Mostrar detalles de un archivo de configuración.
 - RF 3.3. Registrar los datos de un nuevo archivo de configuración.
 - RF 3.4. Modificar los datos de un archivo de configuración.
 - RF 3.5. Eliminar los datos de un archivo de configuración.
 - RF 3.6. Crear nuevo favorito a partir de los datos de un archivo de configuración.
- RF 4. Gestionar las modificaciones de los archivos de configuración.
 - RF 4.1. Mostrar listado de las modificaciones de archivo registradas.
 - RF 4.2. Mostrar detalles de una modificación de archivo.
 - RF 4.3. Registrar los datos de una nueva modificación de archivo.
 - RF 4.4. Crear nueva propiedad de perfil a partir de los datos de una modificación.
 - RF 4.5. Archivar los registros de modificaciones anteriores a una fecha dada.
- RF 5. Gestionar las etiquetas de propiedades de archivos de configuración.
 - RF 5.1. Mostrar listado de las etiquetas registradas.
 - RF 5.2. Mostrar detalles de una etiqueta.
 - RF 5.3. Registrar los datos de una nueva etiqueta.
 - RF 5.4. Modificar los datos de una etiqueta.

- RF 5.5. Eliminar los datos de una etiqueta.
- RF 6. Gestionar los perfiles de configuración de Nutch.
 - RF 6.1. Mostrar listado de los perfiles de configuración registradas.
 - RF 6.2. Mostrar detalles de un perfil de configuración.
 - RF 6.3. Registrar los datos de un nuevo perfil de configuración.
 - RF 6.4. Modificar los datos de un perfil de configuración.
 - RF 6.5. Eliminar los datos de un perfil de configuración.
 - RF 6.6. Aplicar datos de perfil de configuración.
- RF 7. Gestionar las propiedades de los perfiles de configuración.
 - RF 7.1. Mostrar listado de las propiedades de perfil de configuración registradas.
 - RF 7.2. Mostrar detalles de una propiedad de perfil de configuración.
 - RF 7.3. Registrar los datos de una nueva propiedad de perfil de configuración.
 - RF 7.4. Modificar los datos de una propiedad de perfil de configuración.
 - RF 7.5. Eliminar los datos de una propiedad de perfil de configuración.
 - RF 7.6. Aplicar datos de propiedad de perfil de configuración.
- RF 8. Gestionar las configuraciones favoritas de Nutch.
 - RF 8.1. Mostrar listado de las configuraciones favoritas registradas.
 - RF 8.2. Mostrar detalles de una configuración favorita.
 - RF 8.3. Registrar los datos de una nueva configuración favorita.
 - RF 8.4. Modificar los datos de una configuración favorita.
 - RF 8.5. Eliminar los datos de una configuración favorita.
 - RF 8.6. Exportar archivos de configuraciones favoritas.
 - RF 8.7. Importar archivos de configuraciones favoritas.
- RF 9. Gestionar los usuarios con acceso al sistema.
 - RF 9.1. Mostrar listado de los usuarios registrados.
 - RF 9.2. Mostrar detalles de un usuario.
 - RF 9.3. Registrar los datos de un nuevo usuario.
 - RF 9.4. Modificar los datos de un usuario.
 - RF 9.5. Eliminar los datos de un usuario.
- RF 10. Permitir a los usuarios autenticarse en el sistema.
- RF 11. Generar reportes de modificaciones registradas.
 - RF 11.1. Mostrar reporte de modificaciones filtrado por cantidad de modificaciones, archivos de configuración y/o rango de tiempo.
 - RF 11.2. Exportar reportes a PDF y HTML.
- RF 12. Generar gráficos de estadísticas de modificaciones registradas.
 - RF 12.1. Mostrar gráfico estadístico de modificaciones filtrado por cantidad de modificaciones, archivos de configuración y/o rango de tiempo.

- RF 12.2. Mostrar gráfico estadístico tipo *Spline* con historial de modificaciones filtrado por rango de tiempo.
- RF 13. Gestionar propiedades del sistema.
 - RF 13.1. Mostrar listado de las propiedades del sistema registradas.
 - RF 13.2. Mostrar detalles de una propiedad del sistema.
 - RF 13.3. Modificar los datos de una propiedad del sistema.
- RF 14. Permitir al sistema reconocer modificaciones de los archivos de configuración realizadas manualmente.

2.3.2 Requisitos no funcionales

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

Requerimientos de software

- RNF 1. Se requiere una distribución GNU/Linux como Sistema Operativo.

Requerimientos de diseño e implementación

- RNF 2. Se requiere el uso del paradigma de Programación Orientada a Objetos (POO).
- RNF 3. Se requiere el uso de Symfony 2.0 o superior como marco de trabajo.
- RNF 4. Se requiere MongoDB como Sistema Gestor de base de Datos.

Requerimientos de interfaz de usuario

- RNF 5. Se requiere una interfaz visualmente agradable y que posibilite al usuario una navegabilidad intuitiva por las funcionalidades del sistema.
- RNF 6. Para la realización del diseño se requiere homogeneidad con la identidad del motor de búsqueda Orión.

Requerimientos de seguridad

- RNF 7. Se requiere un mecanismo que permita el control de acceso a la aplicación, para contribuir a la confidencialidad y no divulgación de la información manejada por el sistema.
- RNF 8. Se requiere la encriptación de la contraseña de los usuarios del sistema, en el trayecto desde la PC Cliente al Servidor.

Requerimientos de usabilidad

RNF 9. Se requiere que la aplicación pueda ser utilizada por usuarios con conocimientos mínimos de computación.

Requerimientos de licencia

RNF 10. Se requiere el uso de herramientas y recursos de software libre, las cuales se podrán usar, modificar y distribuir libremente.

2.4 Modelo de Casos de Uso del Sistema

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

2.4.1 Diagrama de Casos de Uso del Sistema

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

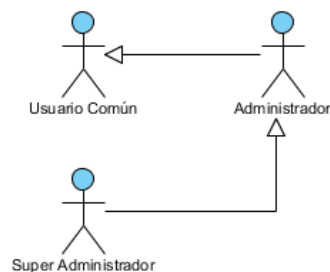


Figura 8: Jerarquía de usuarios del sistema.

Con el objetivo de favorecer la organización y el entendimiento de los casos de uso representados, se muestran cada uno de estos, inicializados por el actor correspondiente, en diagramas separados.

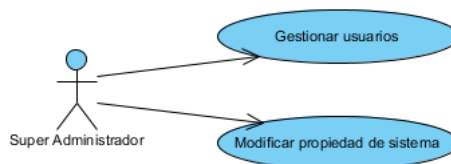


Figura 9: Caso de uso inicializado por el Súper Administrador.

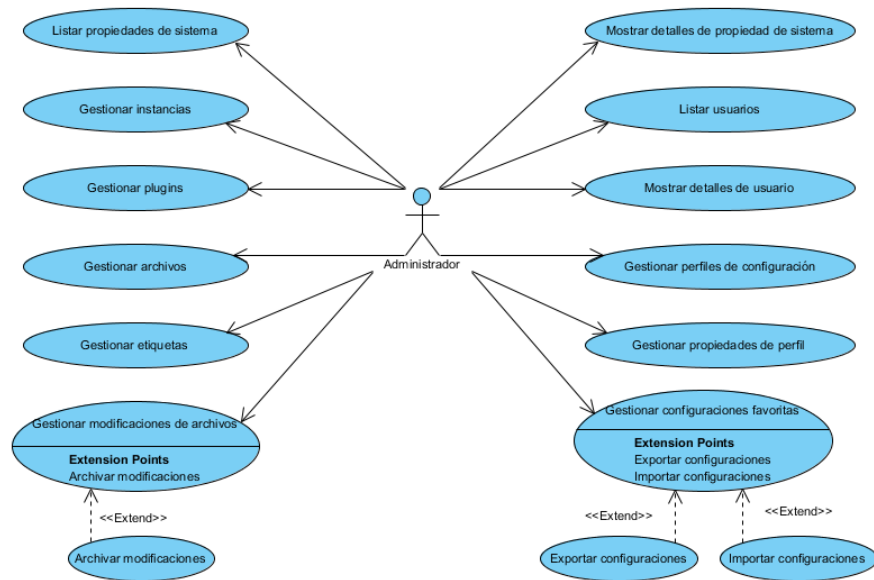


Figura 10: Casos de uso inicializados por el Administrador.

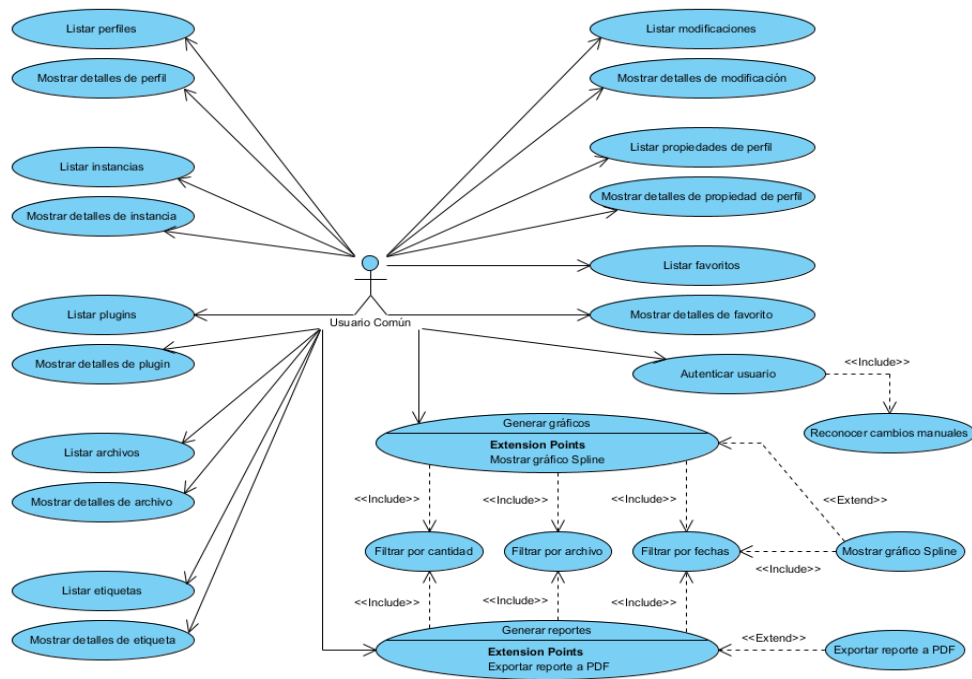


Figura 11: Casos de uso inicializados por el Usuario Común.

En los diagramas anteriores se encuentran representados en total, 42 casos de uso y la jerarquía de actores que interactúan con cada uno de ellos, teniendo en cuenta los permisos y roles correspondientes. Se identificaron varios CU que se consideran críticos para el correcto funcionamiento de la aplicación en general:

- **CU Gestionar archivos:** Se encarga de todos los procesos asociados a la gestión de archivos

de configuración; entre ellos el registro de los archivos en el sistema, la organización atendiendo a la instancia o *plugin* al que pertenece, la edición del contenido interno, entre otros.

- **CU Gestionar modificaciones de archivos:** Engloba todo el proceso de edición y registro de cambios del contenido de los archivos de configuración. Utilizando estos registros, la aplicación será capaz de realizar otras operaciones, incluyendo la gestión de perfiles de configuración y la generación de reportes y gráficos estadísticos.
- **CU Gestionar perfiles de configuración:** Se refiere a la gestión de propiedades de configuración de archivos que son predefinidas por el administrador o definidas a través de registros de modificaciones, contribuyendo en gran medida a la agilización del proceso de configuración de Nutch.

Con el propósito de lograr una mayor reutilización y mejor representación en los diagramas, existen varios casos de uso incluidos, ejemplo de ello son:

- **CU Filtrar por fecha:** Representa la opción de filtrar los resultados de los reportes y gráficos estadísticos de modificaciones por rangos de fechas.
- **CU Filtrar por archivo:** Representa la opción de filtrar los resultados de los reportes y gráficos estadísticos de modificaciones por archivos seleccionados.
- **CU Filtrar por cantidad:** Representa la opción de filtrar los resultados de los reportes y gráficos estadísticos de modificaciones por cantidad de registros de modificaciones.
- **CU Reconocer cambios manuales:** Simboliza la acción asociada al reconocimiento de modificaciones manuales de archivos de configuración sin hacer uso del sistema. Dicha acción se desencadena en el momento de la autenticación de un usuario.

También se muestran varios casos de uso extendidos que posibilitaron representar funcionalidades que no se ejecutan siempre, sino bajo determinadas circunstancias dentro de otros casos de uso:

- **CU Archivar modificaciones:** Se refiere a la opción de archivar de forma externa, en formato XML, todos los registros de modificaciones anteriores a una fecha definida por el actor.
- **CU Exportar configuraciones:** Se encarga de exportar los archivos definidos como favoritos en un paquete de archivos externo, con el fin de utilizarlos en otras instancias desplegadas en estaciones de trabajo diferentes.
- **CU Importar configuraciones:** Se refiere a importar un paquete de archivos externo, el cual contiene archivos de configuración pre-configurados o definidos como favoritos.
- **CU Mostrar gráfico Spline:** Abarca la funcionalidad de Mostrar un gráfico tipo Spline con el historial de modificaciones registrado en el sistema, filtrado por un rango de fecha.
- **CU Exportar reporte a PDF:** Representa el proceso de exportar un reporte generado por el sistema a un archivo externo en formato PDF.

2.4.2 Patrones de casos de uso utilizados

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

Concordancia - Reutilización: Extrae una sub-secuencia de acciones que aparecen en diferentes lugares como parte del flujo de varios casos de uso. Este patrón incluye al menos tres casos de uso, uno de ellos llamado “sub-secuencia común”, que modela una secuencia de acciones que aparecen en múltiples casos de uso en el modelo; mientras que el resto describe el comportamiento del sistema que comparte la sub-secuencia común de acciones.

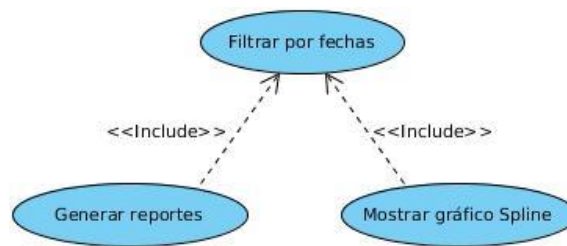


Figura 12: Ejemplo de uso del patrón de CU Concordancia – Reutilización.

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

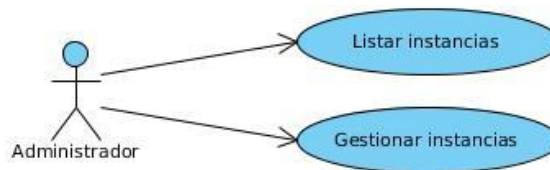


Figura 13: Ejemplo de uso del patrón de CU CRUD Parcial.

Servicio Opcional – Adición: Consiste de dos casos de uso vinculados mediante una relación de extensión, donde uno de ellos modela una secuencia obligatoria de acciones y el segundo una secuencia adicional de acciones que permite extender el comportamiento del primero. Es importante señalar que el caso de uso extendido debe ser un caso de uso abstracto, lo que significa que no podrá ser instanciado de forma independiente.

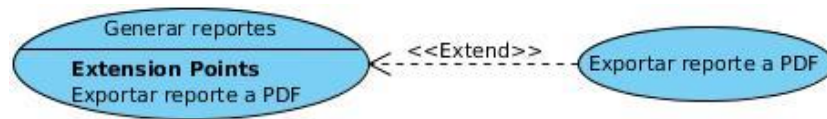


Figura 14: Ejemplo de uso del patrón de CU Servicio Opcional – Adición.

Múltiples Actores - Rol Común: Este patrón es utilizado cuando existe una sola entidad externa que interactúa con cada instancia de un caso de uso; por lo general esta situación ocurre cuando dos o más personas con roles diferentes dentro del negocio interactúan de la misma forma con un caso de uso.

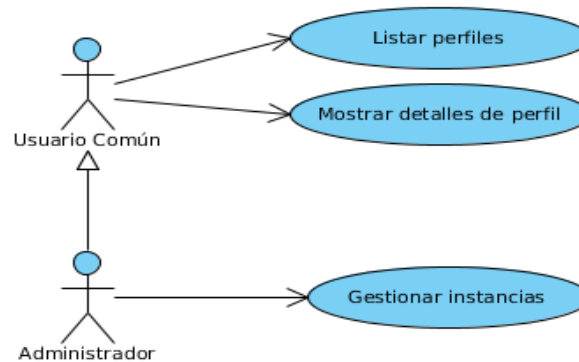


Figura 15: Ejemplo de uso del patrón de CU Múltiples Actores - Rol Común.

2.4.3 Especificación de casos de uso

A continuación se realiza la descripción de uno de los CU críticos del sistema, el resto de las descripciones se encuentran en el **Anexo # 4**.

Descripción del CU “Gestionar archivos”.

Tabla 3: Descripción del CU “Gestionar archivos”.

Objetivo	Gestionar los archivos de configuración de cada instancia de Nutch.
Actores	Administrador
Resumen	<p>El caso de uso se inicia cuando el actor (en lo adelante usuario) pulsa el botón asociado a una de las siguientes acciones desde la lista de archivos de configuración correspondientes a una instancia o <i>plugin</i>.</p> <ul style="list-style-type: none"> • Mostrar detalles de archivo • Registrar datos de archivo • Modificar datos de archivo • Eliminar datos de archivo • Crear favorito a partir de archivo
Complejidad	Media
Prioridad	Alta
Precondiciones	<p>El usuario ha sido autenticado en el sistema.</p> <p>El usuario ha accedido a la lista archivos de configuración. <u>Ver CU Listar archivos</u></p>

Postcondiciones Se realizó exitosamente cualquier acción desde la vista "Listar archivos".

Prototipo



Flujo de eventos

Flujo básico Gestionar archivos

	Actor	Sistema
1.	<p>Pulsa un botón asociado a alguna acción:</p> <ul style="list-style-type: none"> Mostrar detalles de archivo Registrar datos de archivo Modificar datos de archivo Crear favorito a partir de archivo 	
2.		<p>Muestra la vista asociada a la acción:</p> <ul style="list-style-type: none"> Mostrar detalles de archivo. <u>Ver CU Mostrar detalles de archivo</u>.(Prototipo - 2) Registrar datos de archivo. Ver Sección 1: "Registrar datos de archivo".(Prototipo - 4) Modificar datos de archivo. Ver Sección 2: "Modificar datos de archivo".(Prototipo - 3) Eliminar datos de archivo. Ver Sección 3: "Eliminar datos de archivo". Crear favorito a partir de archivo. Ver Sección 4: "Crear favorito a partir de archivo".(Prototipo - 1)
3.		Termina el CU

Sección 1: "Registrar datos de archivo" (Prototipo - 4)

Flujo básico Registrar datos de archivo

	Actor	Sistema
1.	<p>Introduce los datos del archivo en los campos correspondientes:</p> <ul style="list-style-type: none"> Nombre Ruta Formato Contenido 	

	<ul style="list-style-type: none"> Descripción Pulsa el botón "Guardar".	
2.		Valida los datos entrados por el usuario.
3.		Registra los datos del archivo. Muestra el mensaje "El archivo se ha registrado satisfactoriamente".
4.		Redirecciona a la vista "Mostrar detalles". <u>Ver CU Mostrar detalles de archivo</u>
5.		Termina Sección 1.
Flujos alternos		
1ª El usuario pulsa el botón "Cancelar".		
	Actor	Sistema
1.1		Redirecciona a la vista "Lista de archivos".
		Vuelve al paso 1 del flujo básico del CU.
2ª Los datos entrados son incorrectos.		
	Actor	Sistema
3.1		Muestra el mensaje "Error: Algunos datos son incorrectos".
3.2	Corrige los datos incorrectos y pulsa el botón "Guardar"	
3.3		Vuelve al paso 2 del flujo básico de la Sección 1.
Sección 2: "Modificar datos de archivo" (Prototipo - 3)		
Flujo básico Modificar datos de archivo		
	Actor	Sistema
1.	Modifica los datos de los campos correspondientes: <ul style="list-style-type: none"> Nombre Ruta Formato Contenido Descripción Pulsa el botón "Guardar".	
2.		Valida los datos modificados por el usuario.
3.		Registra los datos modificados. Muestra el mensaje "El archivo se ha modificado satisfactoriamente".
4.		Redirecciona a la vista "Mostrar detalles". <u>Ver CU Mostrar detalles de archivo</u>
5.		Termina Sección 2.
Flujos alternos		
1ª El usuario cancela la edición del archivo.		
	Actor	Sistema
1.1		Redirecciona a la vista "Lista de archivos".
		Vuelve al paso 1 del flujo básico del CU.
2ª El usuario elige eliminar datos de archivo.		
	Actor	Sistema
1.1		Ver sección 3 del CU.
1.2		Termina Sección 2.
3ª Los datos modificados son incorrectos.		
	Actor	Sistema
3.1		Muestra el mensaje "Error: Algunos datos son incorrectos".
3.2	Corrige los datos incorrectos y pulsa el botón "Guardar".	

3.3		Vuelve al paso 2 del flujo básico de la Sección 2.
Sección 3: “Eliminar datos de archivo”		
Flujo básico Eliminar datos de archivo		
	Actor	Sistema
1.	Pulsa el botón “Eliminar”.	
2.		Muestra un cuadro de diálogo pidiendo confirmar la acción con el mensaje: “¿Realmente desea eliminar este dato?”.
3.	Pulsa el botón “Si”.	
4.		Elimina los datos del archivo y redirecciona a la vista “Lista de archivos”. Muestra el mensaje “El archivo se ha eliminado satisfactoriamente”.
5.		Termina Sección 3.
Flujos alternos		
1ª El usuario cancela la eliminación.		
	Actor	Sistema
2.1		Quita el cuadro de diálogo de verificación y no elimina los datos del archivo.
2.2		Termina Sección 3.
Sección 4: “Crear favorito a partir de archivo” (Prototipo - 1)		
Flujo básico Crear favorito a partir de archivo		
	Actor	Sistema
1.		Redirecciona a la vista “Crear favorito” y pone en los campos, los datos del archivo por defecto. <u>Ver Sección 1: Crear favorito en el CU Gestionar favoritos</u>
2.		Termina Sección 4.

2.5 Estilo arquitectónico

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

Para el desarrollo de la aplicación se decide emplear el patrón arquitectónico Modelo - Vista - Controlador (MVC), ya que el marco de trabajo Symfony2 brinda ciertas facilidades que permiten la construcción de aplicaciones web basadas en la estructura y funcionamiento que propone este patrón.

El patrón MVC pertenece a la familia de los estilos de llamada y retorno, la cual enfatiza su trabajo en aumentar la capacidad de modificación y la escalabilidad de las aplicaciones. El mismo propone

el diseño de arquitecturas de software, donde la lógica de negocio se encuentre separada de la interfaz gráfica de usuario, de manera tal que cualquier cambio en esta última no afecte la lógica de negocio y viceversa (Reynoso, 2004b).

En la figura que se muestra debajo, se representa la estructura que propone el patrón MVC teniendo en cuenta los componentes que ofrece el marco de trabajo Symfony2 para la implementación de aplicaciones.

Entre las ventajas más apreciadas de este patrón se pueden destacar (Welicki, 2010):

- Facilita la representación de los datos del modelo en vistas diferentes a la misma vez.
- Garantiza una correcta adaptación a los cambios de interfaz gráfica de usuario.
- Crea independencia en el funcionamiento de cada componente.
- Agiliza las labores de mantenimiento.
- Permite escalar el sistema mediante la comunicación de otras vistas o componentes externos a la capa de dominio existente, sin necesidad de afectar su funcionamiento.

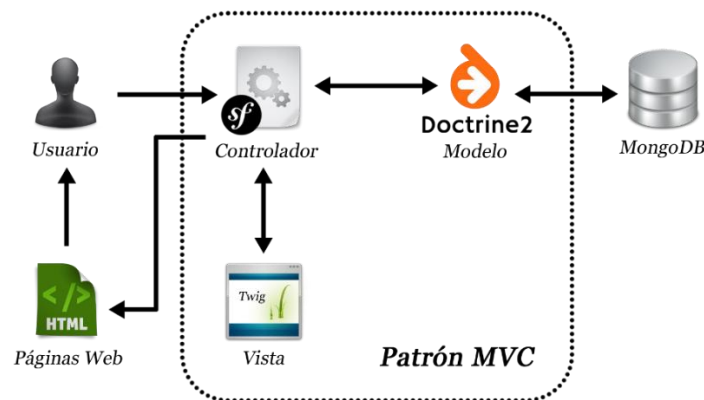


Figura 16: Patrón arquitectónico Modelo - Vista - Controlador.

2.6 Patrones de diseño

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

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

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

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

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

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

La aplicación de estos patrones de diseño se manifiesta, por ejemplo, en la implementación de las clases *“NutchSite”* y *“Property”*, creadas con el objetivo de facilitar el manejo de la información registrada en el archivo principal de configuración de Nutch. La clase *“NutchSite”* es la encargada de realizar las operaciones de lectura y escritura de los parámetros de configuración en este archivo, mientras que la clase *“Property”* representa cada una de las propiedades de configuración.

Controlador: Este patrón tiene como objetivo asignar la responsabilidad a una clase de recibir o manejar un mensaje de evento del sistema generado por un actor externo, por lo general a través de una interfaz gráfica de usuario a la que accede una persona para realizar ciertas operaciones en el sistema.

La utilización de este patrón se evidencia en cada una de las clases controladoras, como por ejemplo la clase *“archivoController”*, que se encarga de atender y ofrecer respuesta a cada una de las solicitudes de un usuario relacionadas con la gestión de archivos de configuración.

2.7 Modelo de Diseño

El modelo de diseño es aquel que se encarga de describir la realización de los casos de uso del sistema, y se utiliza como medio de abstracción del modelo de implementación y el código fuente del software. Su objetivo fundamental es transmitir, a través de la representación mediante diagramas, una comprensión en profundidad de los aspectos relacionados con los requerimientos no funcionales y restricciones concernientes a los lenguajes de programación.

2.7.1 Diagrama de clases del diseño con estereotipos web

A continuación se relaciona el diagrama de clases del diseño correspondiente al CU descrito anteriormente. En el mismo se muestran las relaciones entre las clases participantes, teniendo como característica particular sobre los diagramas de clases tradicionales, el uso de estereotipos UML especiales que son específicos para los diagramas de clases web, los cuales permiten modelar de forma certera la arquitectura y funcionamiento de aplicaciones de este tipo. El resto de los diagramas de clases del diseño se encuentran en el **Anexo # 5**.

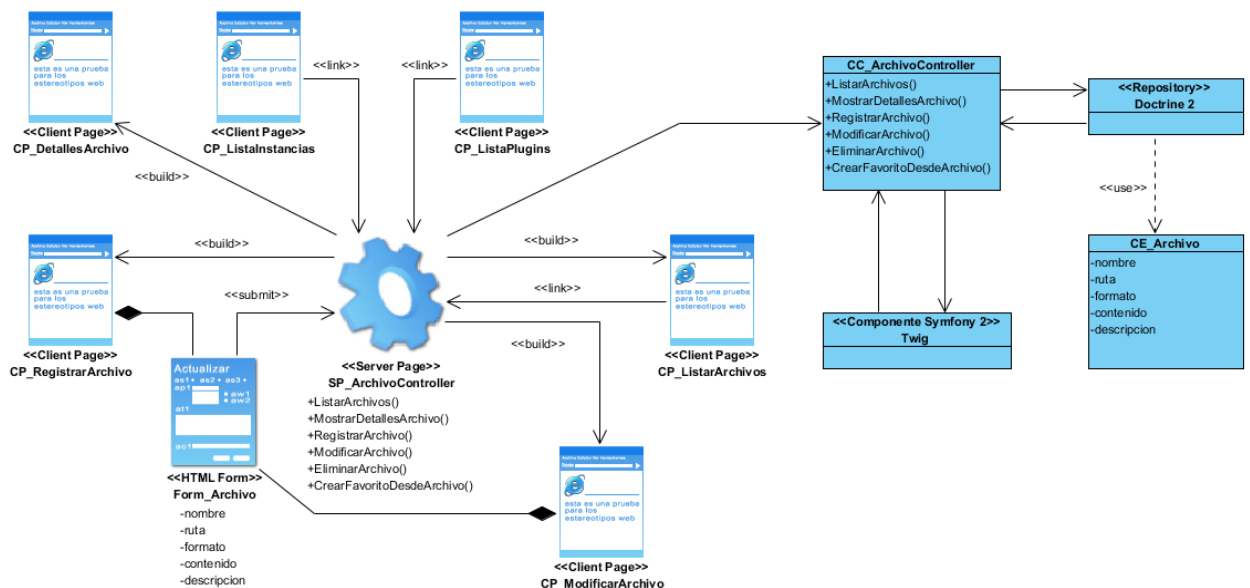


Figura 17: Diagrama de clases del diseño con estereotipos web del CU "Gestionar archivos".

El diagrama anterior modela la relación existente entre las clases que componen el caso de uso "Gestionar archivos". El mismo ilustra la relación que se establece entre las clases clientes, encargadas de enviar peticiones al servidor de acuerdo a las acciones del usuario; y las clases controladoras, encargadas de procesar la información utilizando un conjunto de servicios que ofrece el marco de trabajo Symfony2 y que permiten mostrar los resultados en las vistas correspondientes utilizando el motor de plantillas Twig.

2.8 Diagrama de Interacción

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

2.8.1 Diagramas de secuencia

La función que realizan los diagramas de secuencia es ilustrar cómo se comunican los objetos a través del paso de mensajes intercambiados entre las clases.

Con el propósito de modelar los comportamientos dinámicos del sistema, se presentan a continuación los diagramas de secuencia por cada escenario del CU descrito con anterioridad. El resto de los diagramas de secuencia pueden consultarse en el **Anexo # 6**.

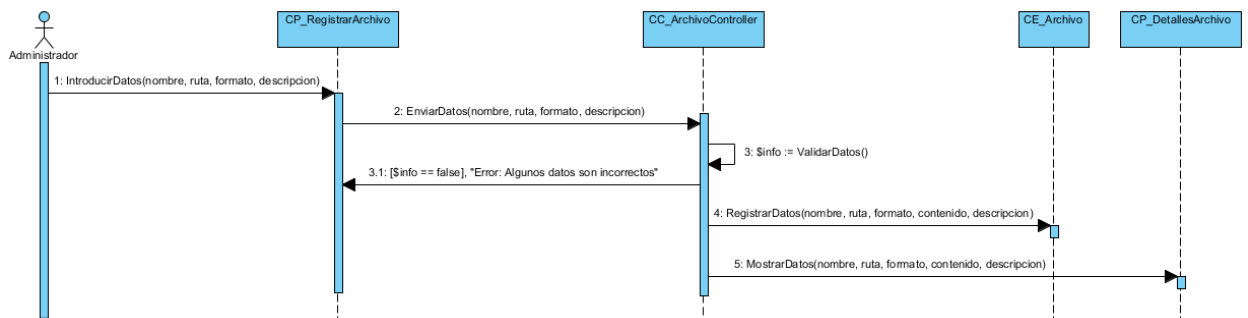


Figura 18: Diagrama de secuencia del escenario “Registrar archivo” del CU “Gestionar archivos”.

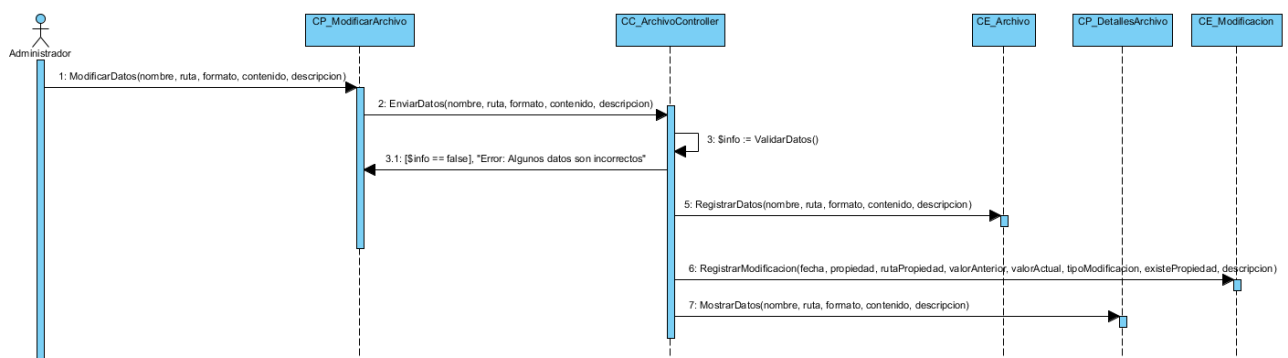


Figura 19: Diagrama de secuencia del escenario “Modificar archivo” del CU “Gestionar archivos”.

A continuación se muestra el diagrama de secuencia de este escenario para cuando el usuario pulsa el botón eliminar desde la interfaz “Detalles de archivo”.

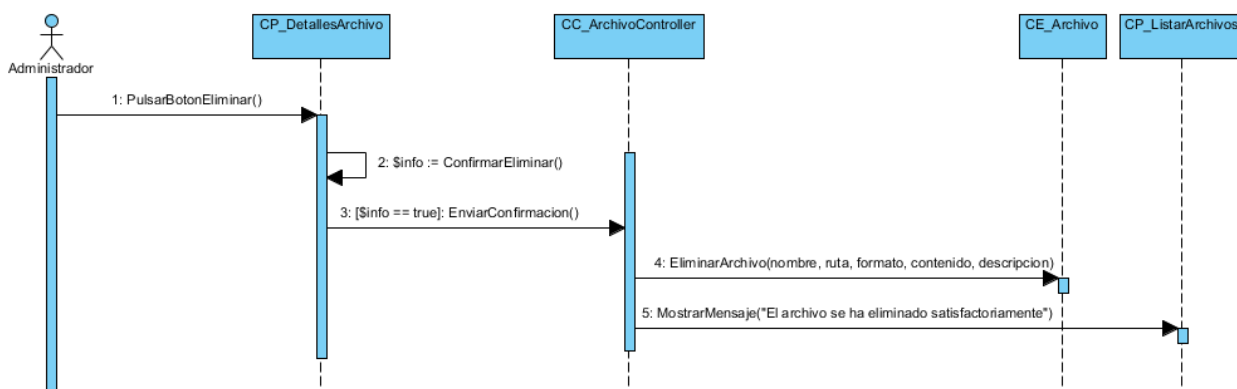


Figura 20: Diagrama de secuencia del escenario “Eliminar archivo” del CU “Gestionar archivos”.

2.9 Diseño de la Base de Datos

La persistencia de la información es uno de los requerimientos claves a tener en cuenta a la hora de diseñar la arquitectura de una aplicación informática. En este sentido, es de vital importancia la realización de un diseño de base de datos coherente con las necesidades puntuales del sistema a desarrollar. El objetivo del diseño de una base de datos, es la representación de las clases entidades y sus relaciones, que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información.

La aplicación que se propone hace uso de una base de datos, puesto que se hace imprescindible el almacenamiento de una gran cantidad de información relacionada con el historial de cambios en los archivos de configuración de Nutch. Como ya se ha definido anteriormente, la información manipulada por la aplicación será almacenada por MongoDB, un gestor de base de datos documentales escalable y de alto rendimiento.

A continuación se describen las colecciones de la base de datos, que se ilustran en la figura 21:

- **Colección instancia:** Almacena la información referente a las instancias de Nutch desplegadas en un mismo servidor.
- **Colección plugin:** Alberga los datos correspondientes a los *plugins* de cada instancia de Nutch.
- **Colección archivo:** En esta colección residen los datos correspondientes a los archivos de configuración que pertenecen a los *plugins* e instancias de Nutch.
- **Colección modificación:** Acumula el historial de cambios de los archivos de configuración registrados en el sistema.
- **Colección perfil:** Posee la información relacionada a los perfiles de configuración gestionados por el administrador del sistema.
- **Colección propiedadPerfil:** Contiene los datos correspondientes a cada propiedad de un perfil determinado.

- **Colección etiqueta:** Registra breves descripciones de propiedades de los archivos de configuración.
- **Colección favorito:** Almacena los datos de los archivos de configuración definidos como favoritos en un momento determinado.
- **Colección sistema:** Guarda la información referente a determinadas propiedades o valores utilizados por el sistema que pueden ser variables en el tiempo.

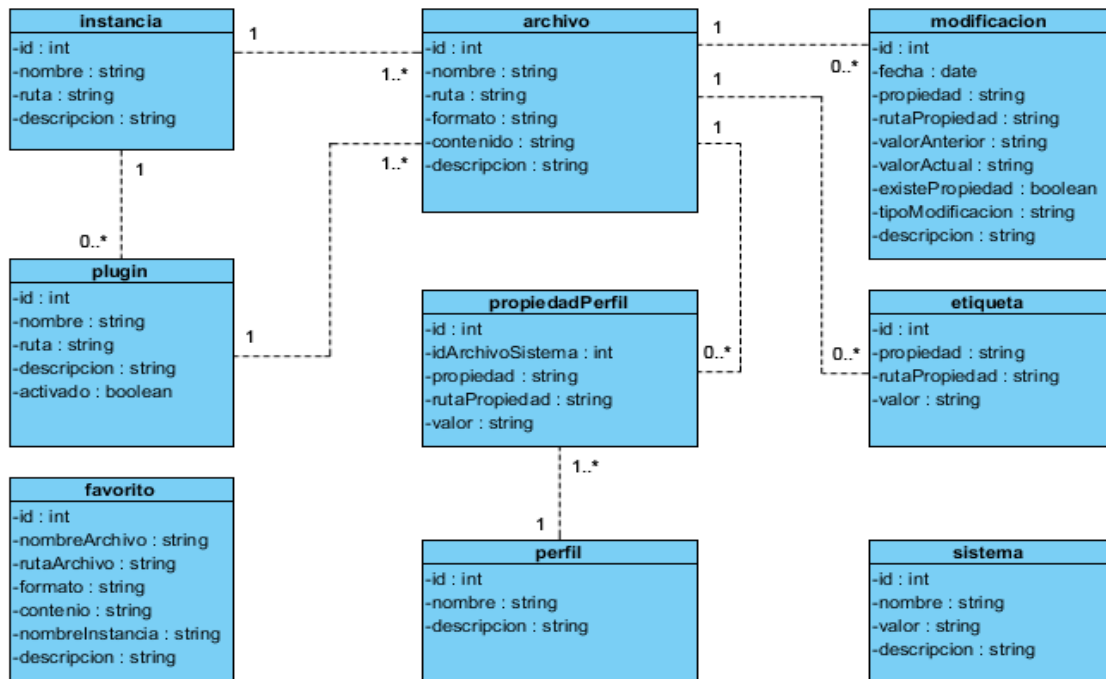


Figura 21: Modelo físico de la base de datos.

2.10 Modelo de Despliegue

Un modelo de despliegue representa la relación física que se establece entre los distintos componentes o nodos que describen la topología de un sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y la información relacionada con la forma en la que los componentes se comunicarán a lo largo de la infraestructura del sistema.

Como se puede apreciar en el siguiente diagrama, el nodo “PC Cliente” representa un ordenador desde el cual se podrá acceder al módulo de configuración, a través del protocolo HTTPS, haciendo uso de un navegador web.

El nodo “Servidor Web Apache” es el encargado de atender y ofrecer respuesta a cada una de las solicitudes del cliente. En éste deberá ser instaurado el módulo de configuración, utilizando el servidor web Apache 2, así como también cada una de las instancias de Nutch sobre las que se tendrá un control de los cambios realizados en la configuración.

La comunicación entre el servidor web y el servidor de base de datos se realizará por medio del protocolo TCP/IP. Este último, representado por el nodo “Servidor de Base de Datos MongoDB”, es el encargado de administrar la información que maneja el módulo de configuración, utilizando el sistema de gestión de base de datos MongoDB.

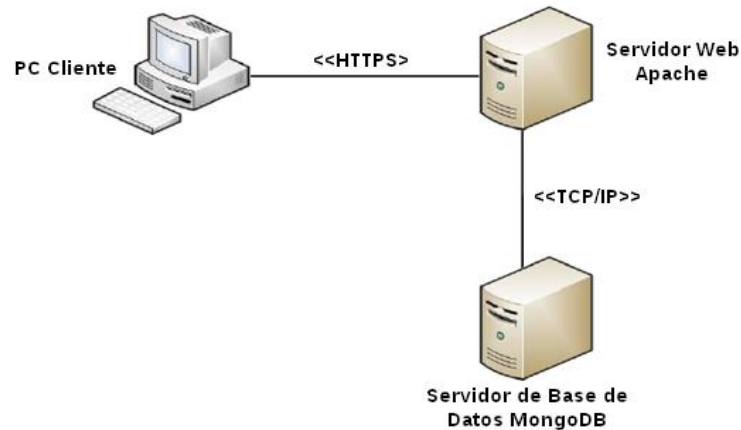


Figura 22: Diagrama de despliegue del módulo de configuración web.

2.11 Conclusiones parciales

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

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL MÓDULO DE CONFIGURACIÓN PARA EL MECANISMO DE RASTREO DEL BUSCADOR ORIÓN

3.1 Introducción

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

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

3.2 Modelo de componentes que integran la solución informática

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

3.2.1 Diagrama de componentes

Este tipo de diagrama ilustra la relación que existe entre componentes de software, así como la ubicación de cada uno de ellos dentro del sistema. A continuación se muestra el diagrama de componentes del módulo de configuración web para el mecanismo de rastreo del buscador Orión; cuya organización se encuentra acorde con la arquitectura MVC propuesto por Symfony2 y descrita en el capítulo anterior de este trabajo.

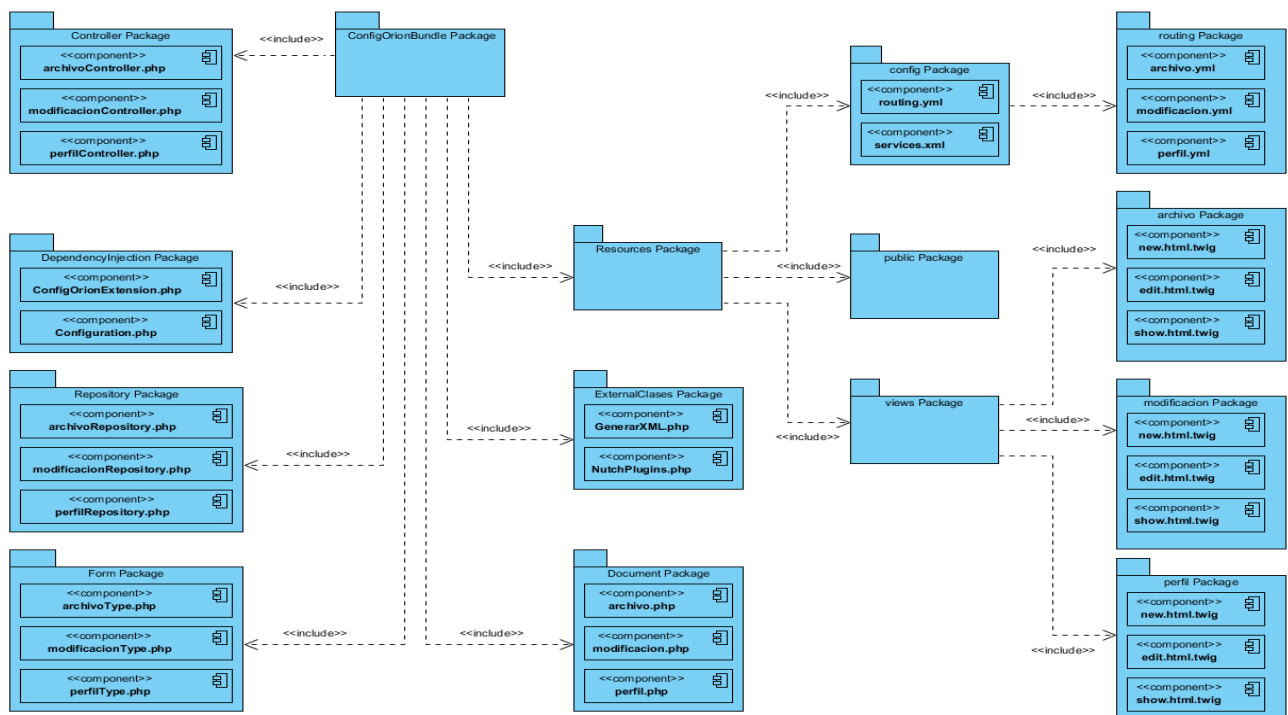


Figura 23: Diagrama de componentes del módulo de configuración web para el mecanismo de rastreo del buscador Orión.

Descripción de los principales paquetes que componen la propuesta de solución:

- **ConfigOrionBundle:** Agrupa en su interior todos los componentes, estableciendo una estructura organizativa acorde al patrón de arquitectura MVC.
- **Controller:** Contiene las clases controladoras encargadas de procesar las peticiones de las páginas cliente y devolver las respuestas con la información requerida.
- **Form:** Alberga los formularios HTML utilizados para el envío de información por parte de los usuarios.
- **Document:** En este paquete residen las clases documentos que se corresponden con las colecciones de la base de datos en MongoDB.
- **Repository:** Concentra aquellas clases que auxilian a las clases documentos, y que contienen las funciones especiales para la obtención de información desde la base de datos.
- **ExternalClasses:** Reúne las clases externas a Symfony2 que contienen funcionalidades específicas.
- **Config:** Contiene en su mayoría los archivos donde se definen las rutas de la aplicación.

- **Public:** En su interior contiene las diferentes clases de apoyo a las vistas de la aplicación entre otros recursos. Dentro de este paquete se encuentran los archivos CSS, los archivos JavaScripts y las imágenes.
- **Views:** Agrupa las páginas referentes a las vistas de la aplicación, así como las plantillas bases.

3.3 Estándares de codificación utilizados

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

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

Identificadores

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

Ejemplos de uso:

- Clase: `class` archivoController
- Función: `private function` getDocumentManager()
- Variable: `$deleteForm`;

Indentación

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

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

Llaves

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

Líneas en blanco

Una de las mejores prácticas de codificación que ayuda a garantizar la capacidad de mantenimiento de los sistemas es la inclusión de líneas en blanco entre secciones de código, funciones, clases, sentencias, declaraciones y comentarios. En este trabajo se ha definido el uso de líneas en blanco para separar funciones de una misma clase, así como secciones de código dentro de una misma función.

Es prudente señalar que este estilo agrega más líneas de código al programa, pero a su vez se gana en legibilidad y limpieza en el código.

Comentarios

Los comentarios en el código representan la documentación interna más precisa de un software. Estos garantizan el entendimiento de lo que realmente realiza un determinado bloque de código, evitando confusiones y agilizando considerablemente las tareas de revisión y mantenimiento. Para la inclusión de comentarios es necesario respetar algunas reglas básicas, entre ellas se encuentran:

- Abreviar el contenido de los comentarios.
- Usar un lenguaje técnico y entendible, evitando el uso de vocablos rebuscados.
- Evadir las descripciones paso a paso de un bloque de código, abogando por expresar un resumen de su propósito.
- Emplear un estilo uniforme de comentarios estándares para todo el equipo.

Para la implementación de módulo de configuración web se decidieron las siguientes conveniencias:

- Deberá incluirse un espacio simple entre los caracteres “//” y el texto del comentario.
- Si el comentario se aplica antes de un bloque de código dentro de funciones, deberá escribirse antes de la primera sentencia. Por el contrario, dentro de sentencias anidadas o bloques de código donde se puede confundir a cuál de las líneas pertenece el comentario, este deberá colocarse a continuación de la línea a la que se hace referencia.
- Para las funciones se deberá redactar una pequeña descripción encima de la cabecera utilizando los comodines de comentario “/* */”, especificando el nombre de la función, los parámetros que recibe y su tipo de dato, así como el valor que retorna.

3.4 Principales pantallas del módulo de configuración web para el mecanismo de rastreo del buscador Orión

Para ilustrar de una mejor manera las funcionalidades implementadas, se muestran a continuación varias imágenes del módulo de configuración web para el mecanismo de rastreo del buscador Orión, así como una breve descripción de cada una de ellas. El resto de las imágenes del sistema se encuentran en el **Anexo # 7**.

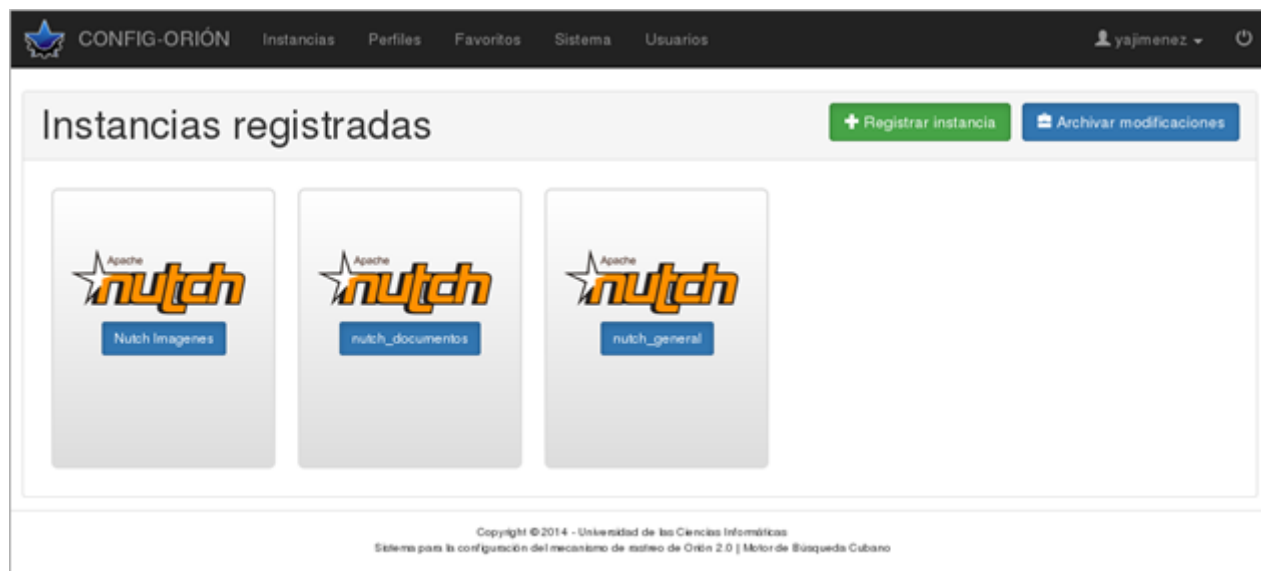


Figura 24: Gestión de instancias de Nutch.

En la figura anterior se ilustra la vista de gestión de instancias de Nutch, en la cual se pueden registrar nuevas instancias, así como ver los detalles de cada una y realizar modificaciones en sus datos.

Además, en esta vista es posible archivar las modificaciones anteriores a una fecha entrada por el administrador, de los archivos de todas las instancias registradas.

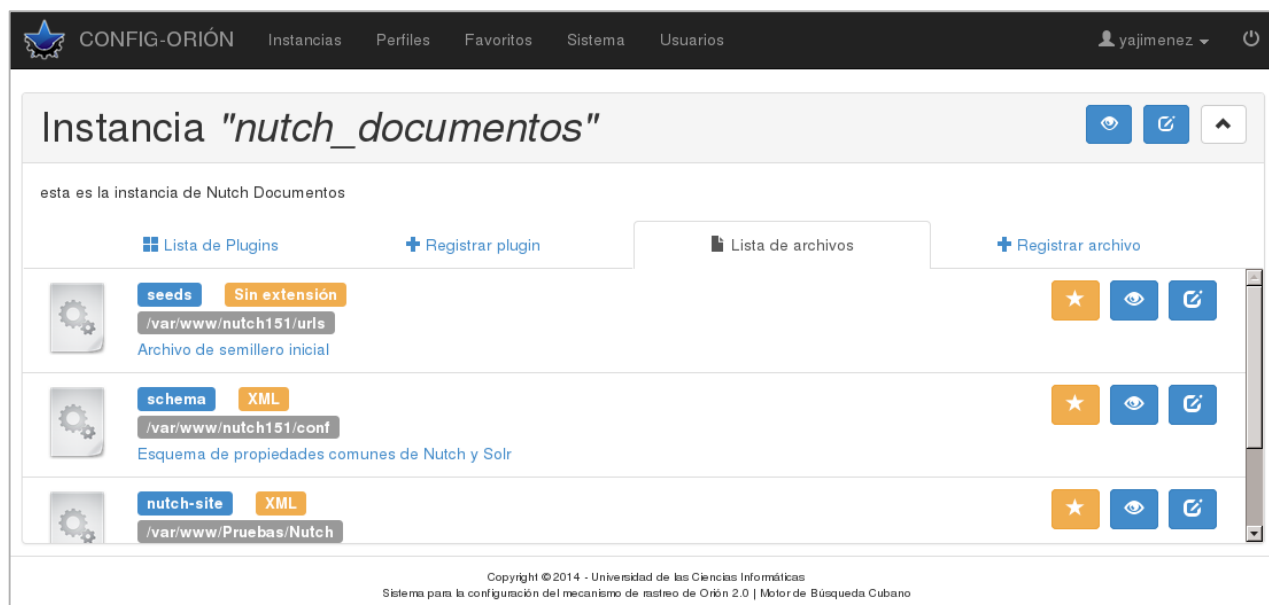


Figura 25: Gestión de archivos de cada instancia.

En la vista de la figura precedente se pueden gestionar los archivos correspondientes a una instancia de Nutch. Además de la edición y visualización de los datos y el contenido de los archivos, es posible definir los archivos como favoritos para su posterior aplicación o exportarlos a otros servidores que posean alguna instancia de Nutch semejante.

3.5 Validación del módulo de configuración web implementado

A continuación se detallan los tipos de pruebas de software aplicados al módulo de configuración web implementado. Las mismas persiguen como objetivo fundamental, la detección de las no conformidades respecto a las funcionalidades de la aplicación, las vulnerabilidades que atentan contra la seguridad de la información que se manipula con el software, la medición del grado de usabilidad de las funcionalidades implementadas, así como también la correcta integración entre los diferentes componentes de la arquitectura del sistema.

3.5.1 Pruebas funcionales

Las pruebas funcionales son aquellas que se aplican a un software determinado, con el objetivo de validar que las funcionalidades implementadas funcionen de acuerdo a las especificaciones de los requisitos definidos con anterioridad. Para la ejecución de este tipo de pruebas, suelen emplearse dos métodos fundamentales: el método de Caja Blanca y el método de Caja Negra. El primero se centra en las pruebas al código de las aplicaciones; mientras que el segundo permite a los

probadores enfocar su atención en el funcionamiento de la interfaz, a través del análisis de los datos de entrada y los de salida.

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

Diseño de los casos de prueba basados en casos de uso

A continuación se muestra un fragmento del diseño del caso de prueba basado en caso de uso de la sección "Registrar datos de archivo" del CU "Gestionar archivos". El resto de los diseños de casos de prueba se pueden encontrar en el **Anexo # 8**.

Caso de uso: Gestionar archivos.

Sección: Registrar datos de archivo.

Escenarios: 1.1 y 1.2.

Condiciones de ejecución:

- El usuario ha sido autenticado en el sistema.
- El usuario autenticado es un Administrador.
- El usuario ha accedido a la lista de archivos de configuración. **Ver CU Listar archivos**

Tabla 4: Fragmento del caso de prueba basado en el CU "Gestionar archivos".

Escenario	Descripción	Nombre	Ruta	Formato	Contenido	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar datos de archivo de forma exitosa.	El sistema introduce los datos del archivo de forma correcta.	V	V	V	N/A	V	El sistema registra los datos del archivo de forma correcta y muestra el mensaje "El archivo se ha registrado satisfactoriamente" .	1 - El usuario se autentica en el sistema con rol de Administrador. 2 - El sistema muestra la vista "Administrar instancias" .
EC 1.2 Registrar datos de archivo de forma incorrecta.	El sistema introduce los datos del archivo de forma incorrecta.	I	V	V	NA	V	El sistema no registra los datos del archivo y muestra el mensaje "Error: El archivo especificado no existe" .	3 - El usuario pulsa en el botón de una Instancia. 4 - El sistema muestra la vista "Administrar archivos" .
		V	I	V	NA	V	El sistema no registra los datos del archivo y muestra el mensaje "Error: La ruta no puede contener dos caracteres '/' seguidos" .	5 - El usuario pulsa el botón "Registrar archivo" . 6 - El sistema

		V	I	V	NA	V	El sistema no registra los datos del archivo y muestra el mensaje "Error: La ruta no puede terminar con el caracter '/' ".	muestra la vista "Registrar archivo". 7 - El administrador llena los datos de los campos y pulsa el botón "Guardar".
		V	V	I	NA	V	El sistema no registra los datos del archivo y muestra el mensaje "Error: El archivo no es realmente de tipo 'TXT' o 'Texto plano' ".	
		V	V	I	NA	V	El sistema no registra los datos del archivo y muestra el mensaje "Error: El archivo no es realmente de tipo 'XML' ".	

Tabla 5: Variables empleadas en el diseño del caso de prueba basado en el CU "Gestionar archivos".

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Sólo debe contener el nombre del archivo. Sólo se incluye el formato en caso de que no exista en la lista de los formatos. (Obligatorio)
2	Ruta	Campo de texto	No	Debe ser una ruta de archivo existente. El archivo real debe estar dentro del plugin o instancia al que se está especificando en la ruta. (Obligatorio)
3	Formato	Campo de selección	No	Debe seleccionarse uno de los valores de la lista. Sólo se aceptan los valores "XML", "TXT", "Sin extensión" (Texto plano). (Obligatorio)
4	Contenido	Área de texto	No	
5	Descripción	Área de texto	Si	

Resultados de las pruebas funcionales

Para llevar a cabo la detección de las no conformidades presentes en la aplicación desarrollada, se realizaron 3 iteraciones de pruebas funcionales. En la siguiente tabla se muestran los resultados obtenidos en cada iteración de prueba al módulo de configuración para el mecanismo de rastreo del buscador Orión, así como la corrección de cada uno de los errores.

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

No conformidades	1ra Iteración	2da Iteración	3ra Iteración
Detectadas	21	17	6
Resueltas	15	13	6
Pendientes	6	4	0

En la figura 26 se puede apreciar de forma más ilustrativa el comportamiento de las no conformidades por cada iteración de las pruebas funcionales ejecutadas.

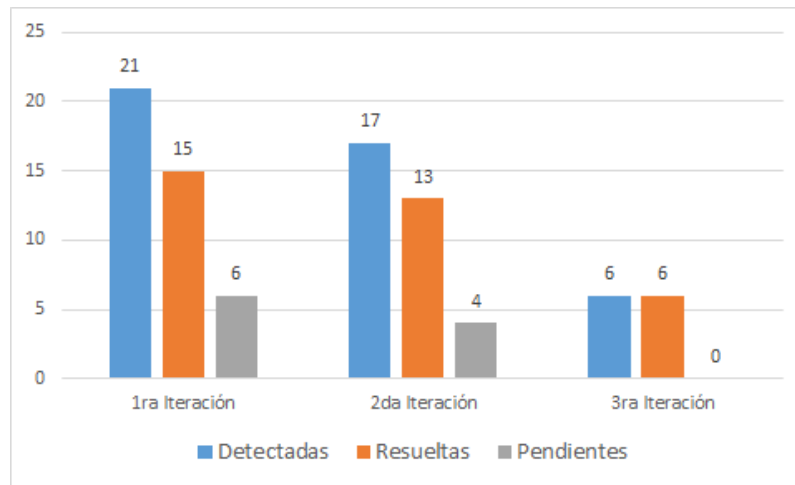


Figura 26: Comportamiento de las no conformidades por cada iteración de las pruebas funcionales.

Las no conformidades detectadas pueden ser agrupadas atendiendo a diferentes tipologías. En la figura 27 se muestra una gráfica con la cantidad de no conformidades detectadas por cada tipo.

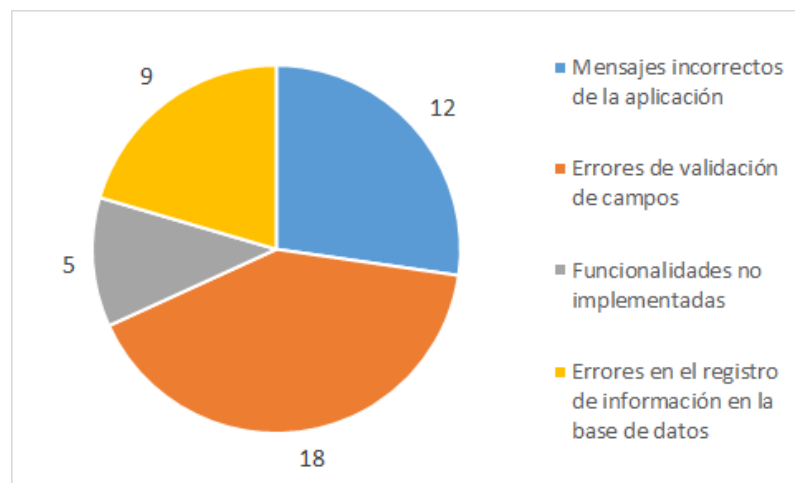


Figura 27: Cantidad de no conformidades detectadas por tipo.

3.5.2 Pruebas de integración

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático. Las mismas validan que estos componentes realmente funcionan juntos, son llamados correctamente y además, transfieren los datos correctos en el tiempo preciso y por las vías de comunicación establecidas (Sommerville, 2005).

Una vez realizadas las pruebas funcionales a cada componente interno de manera independiente, y verificado que las funcionalidades implementadas se corresponden de acuerdo a los requisitos funcionales y no funcionales establecidos; se pudo comprobar el correcto funcionamiento de los componentes mediante el estudio del flujo de datos entre ellos. Posterior a estas pruebas, se hace

necesaria la realización de pruebas de integración, con la finalidad de validar la compatibilidad y el funcionamiento de las interfaces que comunican las diferentes partes que componen la solución informática.

Para la realización de las pruebas de integración se llevaron a cabo diferentes acciones, a continuación se mencionan las más fundamentales:

- Comprobación del funcionamiento del enlace entre el motor de búsqueda Orión y el módulo de configuración web implementado.
- Verificación de la conexión entre el sistema y la base de datos en MongoDB.
- Correspondencia entre los datos manejados por los usuarios en el sistema y los existentes en las respectivas instancias de Nutch.
- Validación de las rutas de acceso, en el sistema de ficheros, a los archivos de configuración de Nutch desde la aplicación.
- Verificación de la correcta lectura y escritura de los archivos de configuración en formatos XML y TXT, comprobando que los símbolos y caracteres extraños sean codificados y decodificados adecuadamente.

La ejecución de las pruebas de integración permitió verificar el trabajo conjunto de los componentes del módulo en cuestión, junto a los componentes que conforman el buscador Orión y el sistema de ficheros que conforman las instancias de Nutch. Se hizo énfasis en la interacción entre estos componentes, lo que posibilitó la detección de incoherencias en el funcionamiento de la aplicación. Se pudieron identificar algunas deficiencias en el manejo de la información ya que, en ocasiones, los datos leídos y escritos en los ficheros de configuración de Nutch no eran codificados adecuadamente y la información de los perfiles de configuraciones no era enlazada con las propiedades de configuración correctas debido a actualizaciones en las estructuras XML. Estas deficiencias fueron corregidas, logrando así, una correcta integración de los componentes internos del módulo configuración web entre sí, con el sistema de ficheros de las instancias de Nutch y el motor de búsqueda Orión.

3.5.3 Pruebas de seguridad

La seguridad informática comprende la puesta en práctica de un conjunto de medidas preventivas y reactivas en los sistemas informáticos y tecnológicos, que posibilitan la protección de la información, persiguiendo como objetivo principal la integridad, confidencialidad y disponibilidad de la misma (INTECO-CERT, 2014).

La realización de pruebas de seguridad contribuye a la detección temprana de vulnerabilidades y la toma de medidas para la disminución de amenazas de ataque, y con ello proveer sistemas de cómputo más seguros y confiables. En este sentido, los especialistas del grupo de Seguridad del Departamento de Evaluación de Productos de Software (DEPSW), perteneciente al Centro Nacional de Calidad de Software (CALISOFT) radicado en la UCI, han establecido dos niveles principales para la realización de las pruebas de seguridad. Con el propósito de evaluar la seguridad de los sistemas en un primer nivel, se definió una lista de chequeo que establece 14 indicadores categorizados en 4 tipos de pruebas.

Después de aplicar la lista de chequeo al módulo de configuración web, se obtuvieron los siguientes resultados para los 4 tipos de pruebas:

Tabla 7: Resultados de la aplicación de la lista de chequeo para pruebas de seguridad de nivel 1.

Tipo de prueba	Indicadores	Correctos
Pruebas de autorización	2	2
Pruebas de gestión de sesiones	2	1
Comprobación del sistema de autenticación	6	3
Validación de datos	4	4

Los indicadores de la lista de chequeo aplicada que fueron evaluados como incorrectos, fueron corregidos oportunamente según se describe en cada indicador; proporcionando de esta forma mayor seguridad al módulo de configuración web. Entre las acciones realizadas para mejorar la seguridad en el nivel 1 se encuentran:

- Desactivación de la caché de Symfony2 para evitar el acceso al sistema luego de que el usuario haya terminado la sesión, pulsando el botón “Atrás” del navegador.
- Establecimiento de un tiempo límite de inactividad para las sesiones de usuarios autenticados.

Para la evaluación de la seguridad en un segundo nivel se utilizó la herramienta *Acunetix Web Vulnerability Scanner 8.0*, la cual es utilizada en la universidad para la detección de vulnerabilidades en sitios y aplicaciones web. Los resultados obtenidos con la herramienta son los siguientes:

Tabla 8: Resultados del escaneo de vulnerabilidades con Acunetix Web Vulnerability Scanner 8.0 para pruebas de seguridad de nivel 2.

Categorías de vulnerabilidades	Cantidad
Ataque a ciegas por inyección SQL (Blind SQL)	0
Formularios HTML sin protección contra ataques CSRF	24
Envío de contraseñas por canal de conexión no encriptado (HTTPS)	1

Campos de contraseña con auto completamiento activado	1
Enlaces rotos	0
Secuencias de comandos en sitios cruzados (XSS)	0
Total de vulnerabilidades	26

Después de analizar los resultados obtenidos en las pruebas, se procedió a corregir las deficiencias encontradas. Para ello se llevaron a cabo un conjunto de acciones en el código fuente de la aplicación, que permitió reforzar la seguridad del módulo de configuración web. A continuación se expone una lista con algunas de las acciones realizadas:

- Inclusión de campos con identificadores únicos en los formularios HTML, para prevenir los ataques CSRF.
- Intercambio de datos sensibles entre cliente y servidor mediante un canal de comunicación encriptado utilizando el protocolo criptográfico SSL.
- Desactivación de la opción de auto completamiento de los campos de formularios para usuario y contraseña.

3.5.4 Pruebas de usabilidad

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

Anteriormente no se contaba con una interfaz de usuario que permitiera llevar a cabo el proceso de configuración de Nutch, este requería no solo amplios conocimientos de administración del *spider*, sino también el aprendizaje de líneas de comando. Debido a lo anterior, se hace difícil la determinación del nivel de usabilidad presente en dicho proceso antes de la concepción del módulo de configuración web desarrollado.

La realización de pruebas de usabilidad contribuye, en cierta medida, a la adquisición de aplicaciones de alta calidad y gran facilidad de uso por parte de los usuarios finales. Para ello, especialistas del DEPSW han establecido una lista de chequeo que proporciona un conjunto de preguntas clasificadas en 12 categorías (Alonso y Fortún, 2014). A continuación se muestran los resultados de la aplicación

de la lista de chequeo al módulo de configuración web para el mecanismo de rastreo del buscador Orión.

Tabla 9: Resultados de prueba de usabilidad utilizando lista de chequeo.

Categorías de Indicadores	Indicadores	Correctos
Visibilidad del sistema	7	5
Adecuación del sistema	16	14
Control y libertad de usuarios	14	10
Consistencia y estándares	17	17
Prevención de errores	3	3
Reconocimiento	9	8
Flexibilidad y eficiencia de uso	7	7
Estética	21	20
Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores	3	2
Ayuda y documentación	3	2
Accesibilidad	35	26
Comprobaciones técnicas	9	8
TOTAL	144	122

En la tabla anterior se puede apreciar que de un total de 144 indicadores de usabilidad, el módulo de configuración web implementado cumple con 122 indicadores, cifra que representa el 84,72 %. Para una mejor comprensión de los resultados obtenidos se han representado los mismos en la siguiente gráfica.

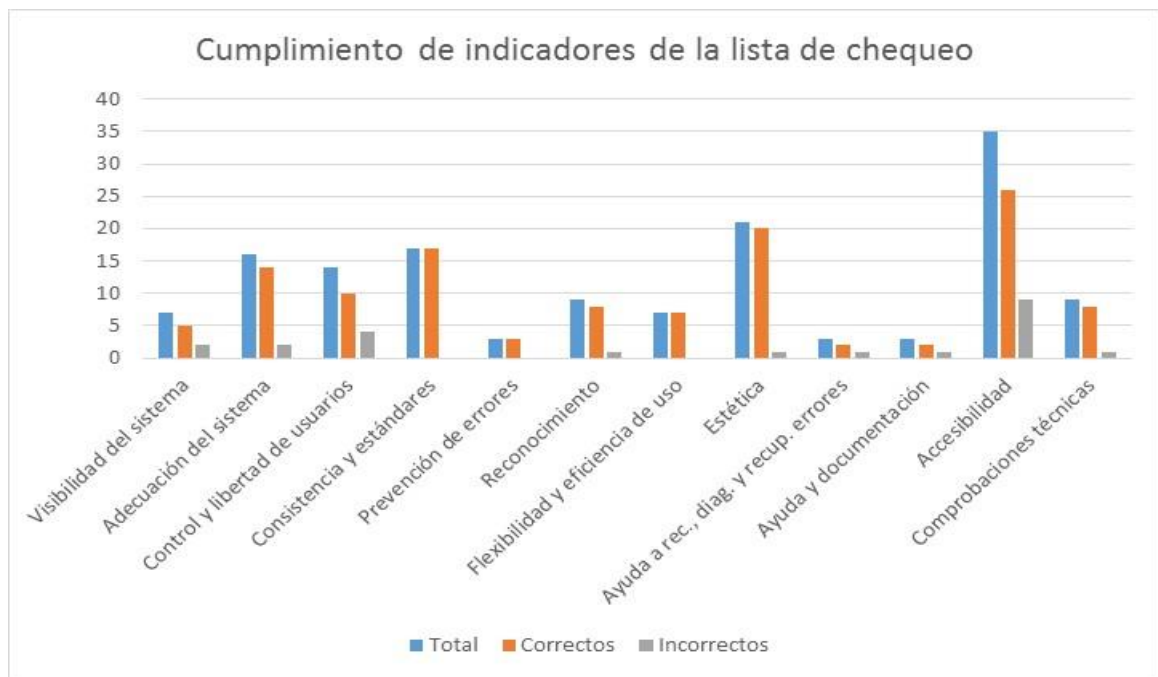


Figura 28: Cumplimiento de indicadores de la lista de chequeo de usabilidad aplicada.

Después de analizar los resultados obtenidos en las pruebas de usabilidad, se identificaron 8 indicadores con posibles mejoras de acuerdo al alcance del presente trabajo. Estos fueron corregidos en función de elevar el nivel de facilidad de uso e interacción entre el usuario y las operaciones que se realizan en el módulo de configuración web; incrementando el nivel de usabilidad hasta el 90,28%. En la gráfica siguiente se representa el estado del nivel de usabilidad resultante.

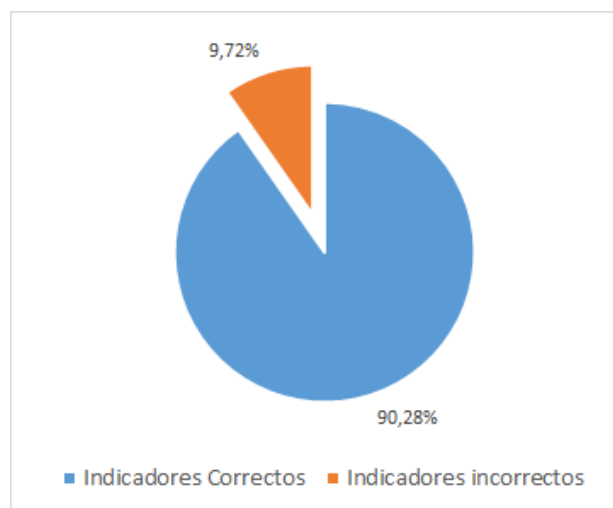


Figura 29: Nivel de usabilidad resultante después de corregidos algunos indicadores de usabilidad.

De acuerdo a la usabilidad resultante, se puede constatar que el módulo implementado favorece el perfeccionamiento del control de cambios en las tareas de configuración del mecanismo de rastreo

del buscador Orión; constituyendo este último aspecto, una ventaja sobre la forma en la que se realiza el proceso de configuración por medio de la consola del sistema operativo.

3.5.5 Evaluación del tiempo del proceso de configuración

Con el propósito de evaluar el indicador referente al tiempo del proceso de configuración, correspondiente a la variable dependiente definida como parte de la hipótesis de investigación, se realizó un pre experimento donde se comparan los resultados obtenidos teniendo en cuenta el uso de la consola del sistema operativo y en segundo lugar el módulo implementado.

Para esta tarea se tomó una muestra de 5 personas que poseen las habilidades esenciales en el trabajo con la consola del sistema operativo y además, se encuentran familiarizadas con el uso de aplicaciones web.

La ejecución de cinco tareas de configuración básicas arrojó los siguientes resultados en minutos:

Tabla 10: Resultados del pre experimento.

	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5	Media	Desviación estándar
Consola	0:04:36	0:04:36	0:03:36	0:01:12	0:04:48	0:18:48	0,003622
Módulo	0:01:12	0:03:12	0:02:00	0:01:48	0:01:24	0:09:36	0,001162

A partir del análisis de la comparación de medias y desviaciones estándar obtenidas en las pruebas, se evidencia una reducción significativa del tiempo del proceso de configuración mediante el uso del módulo implementado. Además, mediante la aplicación de una prueba T de Student, se puede afirmar con una certeza del 95% la veracidad de los resultados antes mencionados; obteniéndose 0,004 como resultado de esta prueba, lo que verifica la hipótesis de investigación anteriormente planteada.

3.6 Conclusiones parciales

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

La aplicación del método pre experimental y la realización de mediciones y cálculos estadísticos, aportaron elementos sustanciales que permitieron validar la hipótesis de investigación planteada con anterioridad, y con ello la factibilidad del módulo de configuración implementado.

CONCLUSIONES

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

- A partir de los fundamentos teóricos analizados en la presente investigación, se determinó que existe gran diversidad en cuanto a los Sistemas de Recuperación de Información. Estos, constituyen sistemas informáticos complejos y con un gran número de opciones de configuración que requieren la integración de determinadas aplicaciones que auxilien a sus administradores en las tareas de configuración, garantizando un mayor control de los cambios y el enfoque en otras tareas de mayor prioridad.
- Una vez estudiados los elementos que intervienen en el proceso de configuración del mecanismo de rastreo de Orión, fue posible la modelación de los artefactos que contribuyeron al diseño de la propuesta de solución. Se identificaron los patrones de diseño y arquitectura, que por sus características, ofrecen mayor soporte a la implementación de los requisitos previamente expresados por el cliente; garantizando la estructura base para la organización lógica del código fuente y la disminución del impacto ante futuras modificaciones en la aplicación.
- La implementación de clases PHP para el procesamiento de los archivos en formato XML y el uso de las rutas internas en este tipo de archivos, permitió definir funcionalidades más independientes y menos susceptibles a futuros cambios en el código fuente; así como la correcta asociación de etiquetas XML específicas con otros contenidos gestionados. Por otra parte, el uso de expresiones regulares en las validaciones, contribuyó a la obtención de un módulo de configuración mucho más seguro ante la entrada de datos no válidos.
- Se obtuvo un módulo de configuración web que permitirá gestionar las instancias del mecanismo de rastreo del buscador Orión y los archivos que estas contienen. La aplicación de parámetros de configuración preestablecidos de forma distribuida, puede realizarse gracias a la definición de archivos favoritos y perfiles de configuración. Además, es posible la obtención de determinada información acerca de las configuraciones realizadas, que anteriormente era imposible o muy difícil de obtener y que contribuye el análisis estadístico de apoyo a la toma de decisiones.
- La evaluación de las pruebas de software realizadas permitió erradicar las insuficiencias detectadas en el módulo desarrollado, que comprometían la seguridad de los archivos de configuración, la comunicación entre los diferentes componentes que integran el sistema y la facilidad de uso de las funcionalidades presentes.

RECOMENDACIONES

Una vez concluida la investigación y el desarrollo de la propuesta de solución, los autores del presente trabajo recomiendan:

- Implementar funcionalidades para el reconocimiento, búsqueda y registro automático de *plugins* y archivos de configuración de las instancias de Nutch.
- Implementar funcionalidades para la detección de diferencias entre los archivos de las instancias de Nutch y los registrados en el módulo de configuración, para permitir la combinación entre los contenidos de los archivos.
- Desarrollar, sobre la base de este trabajo, un módulo que permita la administración y el monitoreo del mecanismo de rastreo del buscador Orión.

REFERENCIAS BIBLIOGRÁFICAS

- ALARCÓN N., S., M. y REYES P., Y.** *Sistema para la toma de decisiones del Webmaster en el ámbito del Posicionamiento Web*. Tesis de Ingeniería Informática. Universidad de las Ciencias Informáticas. La Habana, Cuba, 2009.
- ALEMAN, Y. J.; VALDÉS, H. L. P.** *Estado de optimización para motores de búsqueda en portales de la prensa cubana*. Serie Científica de la Universidad de las Ciencias Informáticas, 2013, 6 (1): p. 20-28.
- ALONSO, Y. H.; FORTÚN, L. L.** *Indicadores para evaluar la usabilidad en aplicaciones web*. En: 1ra Conferencia Científica Internacional UCIENCIA 2014. La Habana, Cuba: Universidad de las Ciencias Informáticas, 2014, 10 p. ISBN: 978-959-286-026-1.
- ÁLVAREZ, C. Z.** *Metodología de la investigación científica*. Universidad de Oriente. Santiago de Cuba, Cuba. 1995.
- APACHE HADOOP.** *Sitio oficial de Apache Hadoop*. [En línea]. Apache Software Foundation, 2013. [Citado el: 4 de Diciembre de 2013]. Disponible en: [<http://hadoop.apache.org>].
- APACHE HTTP SERVER.** *Sitio oficial del servidor web HTTP Apache 2*. [En línea]. Apache Software Foundation, 2013. [Citado el: 3 de Enero de 2014]. Disponible en: [<http://httpd.apache.org/>].
- APACHE LUCENE.** *Apache Lucene - Apache Lucene Core*. [En línea]. 2014. [Consultado el: 27 de Febrero de 2014]. Disponible en: [<https://lucene.apache.org/core/>].
- APACHE NUTCH.** *Sitio oficial de Apache Nutch*. [En línea]. The Apache Software Foundation, 2013. [Citado el: 3 de Diciembre de 2013]. Disponible en: [<http://nutch.apache.org/index.html>].
- BAEZA-YATES, R. AND RIBEIRO-NETO, B.** *Modern information retrieval*. New York: ACM Press; Harlow [etc.]: Addison-Wesley, 1999. ISBN 0-201-39829-X.
- BAEZA-YATES, R., CASTILLO, C., MARÍN, M. y RODRÍGUEZ, A.** *Crawling a Country: Better Strategies than Breadth-First for Web Page Ordering*. 2005.
- BAKKEN, S. S. et. al.** *Manual de PHP*. The PHP Documentation Group, 2013. 1063 p.
- BALLAGAS, B. M. F. Y PLASENCIA, A. J. S.** *Herramientas para el desarrollo de motores de búsquedas especializados de contenidos en la web con procesamiento paralelo (grid) y distribuidos*. [En línea]. INTEMPRES, 2012. [Citado el: 4 de Diciembre de 2013]. Disponible en: [http://vigitec.uci.cu/wp-content/uploads/2012/06/Ballagas-Flores-Barbarade-los-Milagros_INTEMPRES-2012.pdf].
- BARCHINI, G. E.** *Métodos "I + D" de la informática*. Revista de Informática Educativa y Medios Audiovisuales, 2005, 2 (5): p. 16-24.
- CASTRO, P. M.** *Información y Documentación: fundamentación teórica y coyuntura académica*. Revista General de Información y Documentación, 2010, vol. 20, p. 25-43.
- CHOWDHURY, G.** *Introduction to modern information retrieval*. Facet publishing, 2010. 488 p.
- CUTTING, D.** *Scaling Nutch*. En: Proceedings of the 5th International Web Archiving Workshop (IWAW05). 5th International Web Archiving Workshop (IWAW05). Vienna, Austria, 2005.

DICCIONARIO INFORMÁTICO. *Definición de interfaz gráfica de usuario.* [En línea]. SOFTZone, 2013. [Citado el: 7 de Diciembre de 2013]. Disponible es: [<http://www.softzone.es/glosario/g-h-e-i/>].

ECLIPSE PROCESS FRAMEWORK. *Introduction to OpenUP (Open Unified Process).* [En línea]. Consultado el: [25 de Enero de 2014] Disponible en: [http://epf.eclipse.org/wikis/openupsp/openup_basic/customcategories/introduction_to_openup_basic,_BT_J_YMXwEduywMSzPTUUwA.html].

EGUILUZ, J. *Desarrollo web ágil con Symfony2.* 2013. 618 p.

FUENTES, M. *Buscadores en Internet: concepto, análisis y servicios.* [En línea] 2011. [Citado el: 1 de Diciembre de 2013] Disponible en: [<http://suite101.net/article/buscadores-en-internet-concepto-analisis-y-servicios-a68330>].

GITHUB. *Nutch GUI.* [En línea]. 2010. [Citado el: 26 de Abril de 2014]. Disponible en: [<https://github.com/101tec/nutch>].

GONZÁLEZ, L. C.; TORRES, E. R. P. *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información.* Serie Científica, 2012, vol. 5, no 10. Disponible en: [<http://publicaciones.uci.cu/index.php/SC/article/view/1032>].

HASSAN, Y.; MARTÍN F. J. Y LAZZA, G. *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información.* [En línea]. "Hipertext.net", núm. 2, 2004. [Citado el: 11 de Noviembre de 2013.] Disponible en: [<http://www.hipertext.net>].

HELM, R., et al. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, 2002.

HERNÁNDEZ, R. A.; COELLO, S. *El proceso de investigación científica.* Editorial Universitaria. 2011. ISBN: 978-959-16-1307-3

HERNÁNDEZ, R. C. Y GREGUAS, D. N. *Desarrollo de Interfaces Web para el SIGEP.* Tesis de diploma de Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, La Habana, Cuba, 2008.

HEYDON, A. Y MARC, N. *Mercator: A scalable, extensible web crawler.* World Wide Web. Vol. 2, No. 4, 1999. ISSN: 1573-1413. P. 219-229.

INTECO - CERT. *Guía básica para la securización del servidor web Apache.* España, INTECO – CERT, 2012. 35 p.

INTECO-CERT. *Conceptos de seguridad.* [En línea]. Formación en el Instituto Nacional de Tecnologías de la Comunicación, 2014. [Consultado el: 7 de mayo de 2014]. Disponible en: [https://www.inteco.es/Formacion/Conceptos_de_seguridad/].

KHARE, R., et. al. *Nutch: A flexible and scalable open-source web search engine.* Oregon State University, 2004. p. 32.

KUMAR, S. P.; et. al. *Integration of Web mining and web crawler: Relevance and State of Art.* International Journal on Computer Science and Engineering, Vol. 2, No. 3, 2010. ISSN: 0975-3397. P. 772-776.

LARMAN, CRAIG. *UML y Patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado.* 2da. Pearson, 2003. 520 p.

LAVTECH.COM CORP. *mnoGoSearch 3.3.15 reference manual.* [En línea]. 2014. [Consultado el: 20 de Febrero de 2014]. Disponible en: [<http://www.mnogosearch.org/doc33/>].

- LAVTECH.COM CORP.** *mnoGoSearch features*. [En línea]. 2014. [Consultado el: 24 de Febrero de 2014]. Disponible en: [<http://www.mnogosearch.org/features.html>].
- LÓPEZ H., A. G.**, *Modelos de Sistemas de Recuperación de Información Lingüística Difusa*, 2006.
- LUJÁN, S. M.** *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Editorial Club Universitario, 2002. ISBN: 978-84-8454-206-3, 321 p.
- MÆRSK-MØLLER, H. M.; JØRGENSEN, B. N.** *An evaluation of the NetBeans Module System as a Product Line Implementation Technology*. 2011.
- MAHECHA, I. A. N.** *Buscador web open-source: Nutch*. [En línea]. Universidad Nacional de Colombia, 2009. [Citado el: 3 de Diciembre de 2013]. Disponible en: [<http://dis.unal.edu.co/profesores/eleon/cursos/tamd/presentaciones/nutch.pdf>].
- MARTÍNEZ, J. A. C.** *Cómo instalar Nutch 1.3 y Solr 3.3 en Ubuntu 10.04 (Lucid)*. [En línea]. 2011. [Citado el: 4 de Diciembre de 2013]. Disponible en: [http://comeche.es/documents/Software/como_instalar_nutch_y_solr_ubuntu_10_04.pdf].
- MONDELO, Y. H.; et. al.** *Estudio de evolución de la Web de la Universidad de las Ciencias Informáticas, a partir de los estudios Webmétricos del proyecto productivo Geweb en el periodo 2008 - 2011*. En: Proceedings of the XII Congreso Internacional de Información. XII Congreso Internacional de Información. La Habana, Cuba: INFO2012, 2012, p. 11.
- MONGODB.** *Características de MongoDB*. [En línea]. Sitio oficial del Gestor de Base de Datos MongoDB, 2013. [Consultado el: 24 de Diciembre de 2013]. Disponible en: [<http://www.mongodb.com>].
- MOZILLA PROJECT.** *JavaScript Overview*. [En línea]. Mozilla Developer Network, 2013. [Consultado el: 18 de Diciembre de 2013]. Disponible en: [<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Overview>].
- NETCRAFT.** *Servidores web más usados desde 1995*. [En línea]. Sitio web de NetCraft, 2013. [Consultado el: 25 de Diciembre de 2013]. Disponible en: [<http://www.netcraft.com>].
- NIELSEN, J.** *Usability Engineering*. San Francisco, Editorial Morgan Kaufman, 2010. 362 p. ISBN: 978-0-12--51-8406-9.
- NIOCHE, J.** *Large Scale Crawling with Apache Nutch and Friends*. En: Proceedings of Lucene/Solr Revolution 2013. Lucene/Solr Revolution 2013. Dublin: LucidWorks, 2013. P. 43.
- NUTCH WIKI.** *Nutch Administration User Interface - Nutch Wiki*. [En línea]. 2013. [Citado el: 25 de Abril de 2014]. Disponible en: [<http://wiki.apache.org/nutch/NutchAdministrationUserInterface>].
- OBJECT MANAGEMENT GROUP.** *Introduction to OMG's Unified Modeling Language*. [En línea]. 2013. Consultado el: [25 de Enero de 2013]. Disponible en: [http://www.omg.org/gettingstarted/what_is_uml.htm].
- OPENSEARCHSERVER.** *Open Source Search Engine | OpenSearchServer*. [En línea]. 2014. [Consultado el: 21 de Febrero de 2014]. Disponible en: [<http://www.open-search-server.com/open-source-search-engine/>].
- ORACLE.** *Características del IDE NetBeans*. [En línea]. Sitio oficial del IDE NetBeans, 2014. [Consultado el: 25 de Enero de 2014]. Disponible en: [<https://netbeans.org/features/index.html>].

ORENSE, M.; ROJAS, O. I. *SEO Cómo triunfar en Buscadores*. [En línea]. ESIC Editorial, 2010. Disponible en:

[http://books.google.es/books?hl=es&lr=&id=rJ0bIFsEcjC&oi=fnd&pg=PA15&dq=SEO+C%C3%B3mo+triunfar+en+buscadores&ots=b2_Sv4b97b&sig=uZIUuMQ30BkbCH-WN0LxYKxRH_Q].

OVERGAARD, G., PALMKVIST, K. *Use Cases: patterns and blueprints*. Addison Wesley Professional, 2004.

PARRILLA MONROCLE, M. *La Internet que no aparece en los buscadores*. Tesis de grado en Ingeniería en Ciencias Informáticas. Universidad Carlos III, Madrid, España, 2011.

PÉREZ, J. E. *Introducción a AJAX*. 2009a. Consultado el: [8 de Enero de 2014] Disponible en: [<http://librosweb.es/ajax/>].

PÉREZ, J. E. *Introducción a CSS*. 2009b. Consultado el: [8 de Enero de 2014] Disponible en: [<http://librosweb.es/css/>]

PÉREZ-CARBALLO, J. AND STRZALKOWSKI, T. *Natural language information retrieval: progress report*. Information Processing and Management 36, 2000. p. 155-178.

PRESSMAN, R. S. *Ingeniería de software*. 7 ed. McGraw-Hill Interamericana de España S.L., 2010. 959 p. ISBN 9786071503145.

REAL ACADEMIA ESPAÑOLA (RAE). *Concepto de “configuración” en el Diccionario de la Real Academia Española*. [En línea]. RAE, 2013. [Citado el: 1 de Diciembre de 2013]. Disponible en [<http://lema.rae.es/drae/>].

REYNOSO, C. B. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. [En línea]. 2004b. [Consultado el: 22 de marzo de 2014]. Disponible en: [http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp].

REYNOSO, C. B. *Introducción a la Arquitectura de Software*. Universidad de Buenos Aires. [En línea]. Vol. 33, 2004a. [Consultado el: 22 de marzo de 2014]. Disponible en: [<http://carlosreynoso.com.ar/archivos/arquitectura/Arquitectura-software.pdf>].

RÍOS, S. S.; HINOJOSA, C. R.; DELGADO, R. R. *Desarrollo de un sistema de difusión de gestión del conocimiento de la especialidad, aplicando la metodología OpenUp y el framework Ruby On Rails*. [En línea]. 2013. Consultado el: [23 de Enero de 2014]. Disponible en: [<http://repositorio.espe.edu.ec/handle/21000/6316>].

ROCLAW, E. *Buscadores en Internet, o la búsqueda de lo imposible*. [En línea] 2012. [Citado el: 1 de Diciembre de 2013]. Disponible en: [http://www.educared.org/c/document_library/get_file?p_l_id=11383545&folderId=11518247&name=DLFE-23564.pdf]

RODRIGUEZ, A. P. POLANCO, J. HERNÁNDEZ, D. *OpenUp - Open Unified Process*. [En línea]. 2010. Consultado el: [23 de Enero de 2014]. Disponible en: [<http://es.scribd.com/doc/37116717/Open-Up>].

SAMPIERI, R. H.; et. al. *Metodología de la Investigación*. México D. F., McGraw-Hill/Interamericana, 2006. 882 p.

SANTOS, R. O. *Introducción a los Sistemas de Recuperación de Información*. Habana: Departamento de Ciencia de la Computación UH, 2009.

SEKER, S. E. *Performance Evaluation of a Regular Expression Crawler and Indexer*. [En línea]. CSREA Pres, 2012. [Citado el: 30 de Noviembre de 2013]. Disponible en: [http://www.researchgate.net/publication/236622946_Performance_Evaluation_of_a_Regular_Expression_Crawler_and_Indexer/file/e0b4951877458dc6a2.pdf].

SENSIOLABS. *The Book for Symphony 2.4*. SensioLabs, 2013. 254 p.

SINGH, A. *Ajax Complexity*. [En línea]. International Journal of Computer & organization Trends (IJCOT), 2012. [Consultado el: 8 de Enero de 2014]. Disponible en: [<http://ijesonline.com/Published%20Paper/Volume%2001/Issue%2001/IJES%2006/Ajax%20Complexity.pdf>].

SINGH, S. Y TYAGI, N. *A Novel Architecture of Mercator: A Scalable, Extensible Web Crawler with Focused Web Crawler*. IJCSMC, Vol. 2, No. 6, 2013. ISSN: 2320-088X p. 244-250.

SOMMERVILLE, I. *Ingeniería del software*. 7 ed. Pearson Educación, 2005. 712 p. ISBN 84-7829-074-5.

SOURCEFORGE. *OpenSearchServer search engine*. [En línea]. 2014. [Consultado el: 21 de Febrero de 2014]. Disponible en: [<http://sourceforge.net/projects/opensearchserve/>].

THE APACHE SOFTWARE FOUNDATION. *[NUTCH-251] Administration GUI - ASF JIRA*. [En línea]. 2011. [Citado el: 26 de Abril de 2014]. Disponible en: [<https://issues.apache.org/jira/browse/NUTCH-251>].

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

UML, OMG; MOF, IDL. *The Unified Modeling Language UML*. [En línea]. 2011. Disponible en: [http://www-sop.inria.fr/oasis/SAFA/slides09/KEYNOTE_SAFA_2009_Mallet.pdf].

VISUAL PARADIGM. *Visual Paradigm for UML - Software design tools for agile software development*. [En línea]. 2014. Consultado el: [22 de Enero de 2014] Disponible en: [<http://www.visual-paradigm.com/product/vpuml/>].

WELICKI, L. *Patrones y Antipatrones: una introducción*. [En línea]. Microsoft Corporation, 2010. [Consultado el: 22 de marzo de 2014]. Disponible en: [<http://msdn.microsoft.com/es-es/library/bb972242.aspx#M2>].

WORLD WIDE WEB CONSORTIUM (W3C). *CSS Specifications*. [En línea]. Consultado el: [10 de Enero de 2014] Disponible en: [<http://www.w3.org/Style/CSS/current-work>].

WORLD WIDE WEB CONSORTIUM (W3C). *HTML/Specifications*. [En línea]. Consultado el: [12 de Enero de 2014]. Disponible en: [<http://www.w3.org/community/webed/wiki/HTML/Specifications#HTML>].

WORLD WIDE WEB CONSORTIUM (W3C). *Web Style Sheets*. [En línea]. Consultado el: [15 de Enero de 2014] Disponible en: [<http://www.w3.org/Style/>].

WORLD WIDE WEB CONSORTIUM (W3C). *What is CSS?* [En línea]. Consultado el: [13 de Enero de 2014] Disponible en: [<http://www.w3.org/Style/CSS/>].

WORLD WIDE WEB CONSORTIUM (W3C). *What is HTML?* [En línea]. Consultado el: [13 de Enero de 2014] Disponible en: [<http://www.w3.org/html/>].

GLOSARIO DE TÉRMINOS

API-REST: Conjunto de funciones y procedimientos que ofrece cierta librería o biblioteca para ser utilizado por otro software como una capa de abstracción.

CASE: Acrónimo de *Computer Aided Software Engineering*, lo que se traduce en español a Ingeniería de Software Asistida por Computadoras.

CDDL: Acrónimo de *Common Development and Distribution License*, licencia basada en la *Mozilla Public License* (MPL).

CSRF (*Cross-Site Request Forgery*): Hace alusión a la falsificación de peticiones en sitios cruzados. Consiste en un tipo de *exploit* malicioso que transmite comandos no autorizados por un usuario en el cual un sitio web confía.

DOM: Acrónimo de *Document Object Model*, es un estándar definido por la W3C, que modela la estructura de los documentos HTML como objetos y que permite ser modificada utilizando JavaScript u otro lenguaje de programación.

Extensiones: Son librerías externas al núcleo de PHP, y que pueden ser utilizadas para múltiples propósitos.

GPL: Acrónimo de *General Public Licence*, licencia de software libre.

Internet: Conjunto descentralizado de redes de comunicación que se conectan entre si y utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

Servidor web: Programa informático que espera permanentemente las solicitudes realizadas por clientes web, mediante el protocolo HTTP, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente; generando respuestas en cualquier lenguaje o Aplicación del lado del cliente (Luján; 2002).

SGML: Acrónimo de *Standard Generalized Markup Language*, sistema para definir lenguajes para dar formato a documentos que se utilizan en el intercambio electrónico de documentos, gestión de documentos y la publicación de documentos.

SSH: Acrónimo de *Secure Shell*, que significa en español Intérprete de Órdenes Segura. Es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red

ANEXOS

Anexo # 1: Comparación entre las clasificaciones de los SRI.

Tabla 11: Características de los SRI.

criterio	Directorios	Motores de búsqueda	Metabuscadores
Base de datos	Bases de datos más pequeñas, menos actualizadas, y más elaboradas gracias a la presencia del factor humano.	Bases de datos más amplias y actualizadas.	No tienen bases de datos propias, sino que reenvían las consultas interrogando a varios buscadores a la vez.
Índices	Almacenan la información por temas y categorías, una vez recopilada, de forma manual en sus índices.	Colocan la información, que sean capaces de recoger en la red, en sus índices sin ordenarlas por temas, de manera automática y periódica.	No almacenan información porque no dependen de bases de datos propias.
Forma de búsqueda	No realizan las búsquedas en Internet de manera directa, sino que almacenan una breve descripción de las páginas y ofrecen enlace a éstas.	No realizan las búsquedas en Internet de manera directa, sino en las copias de las páginas que almacenan en sus índices.	Envían las consultas a varios motores de búsqueda, sus resultados dependen de que estos estén disponibles en el momento de la búsqueda, o se descarguen en el período de tiempo permisible.
Facilidad de uso	Son fáciles de usar, permiten, en primer lugar, ubicar la búsqueda en un tema determinado.	Son más difíciles de usar, se requiere explotar al máximo las opciones de búsqueda porque contienen más información.	Son difíciles de usar para búsquedas muy precisas, porque tienen menos control de la búsqueda al interrogar varias bases de datos con interfaces diferentes.
Utilidad	Son convenientes para buscar información general, institucional porque devuelven resultados a las páginas principales.	Se utilizan para buscar información más escasa, especializada, actualizada o incluida en páginas personales.	Se recomienda para temas difíciles de encontrar.

Anexo # 2: Esquema del funcionamiento interno de Nutch.

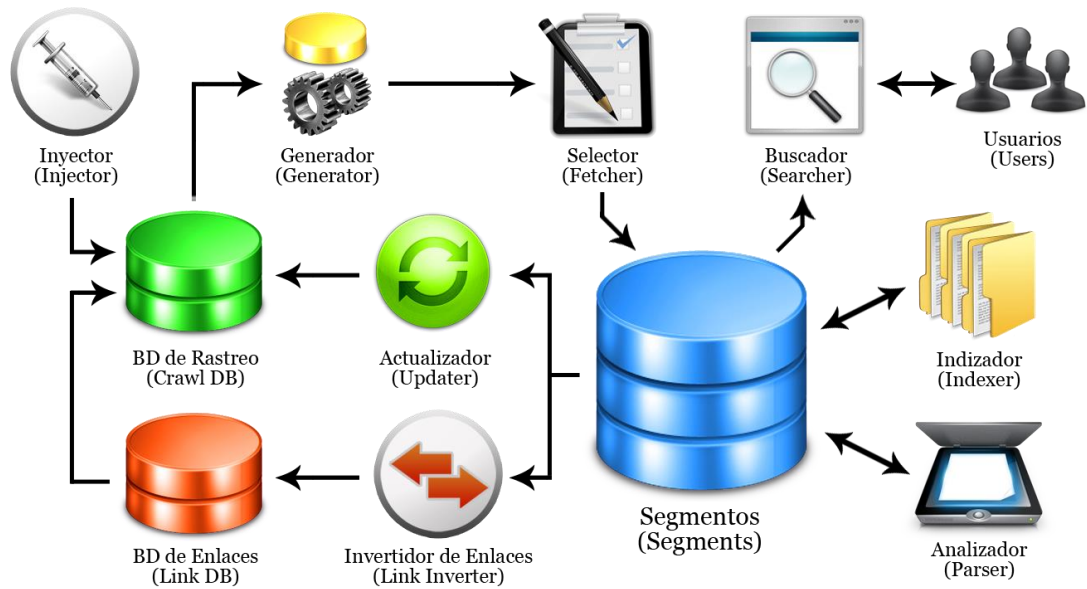
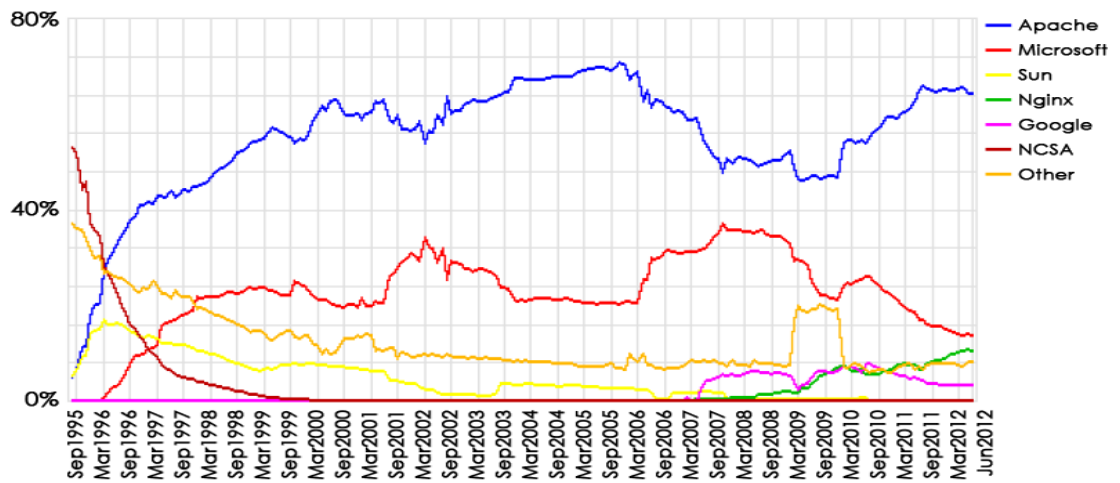


Figura 30: Esquema del funcionamiento interno de Nutch.

Anexo # 3: Estadísticas históricas de uso de servidores web en el período 1995 - 2012.




Servidor de	Mayo 2012	Porcentaje	Junio 2012	Porcentaje	Cambio
Apache	425,631,721	64.20%	448,452,703	64.33%	0.13
Microsoft	92,406,480	13.94%	95,891,537	13.76%	-0.18
Nginx	70,764,248	10.67%	72,881,755	10.46%	-0.22
Google	21,264,616	3.21%	22,464,345	3.22%	0.22

Figura 31: Estadísticas históricas de uso de servidores web en el período 1995 - 2012.


Anexo # 4: Especificación de los casos de uso del sistema.

Tabla 12: Especificación del CU "Listar archivos".

Objetivo	Listar los archivos de configuración de cada instancia de Nutch.	
Actores	Usuario Común	
Resumen	<p>El caso de uso se inicia cuando el actor (en lo adelante usuario) consulta la lista de archivos de configuración de una instancia o <i>plugin</i>. A partir de aquí el actor puede realizar otras operaciones relacionadas con los archivos. Las operaciones son:</p> <ul style="list-style-type: none"> • Mostrar detalles de archivo • Registrar datos de archivo • Modificar datos de archivo • Crear favorito a partir de archivo 	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones	El usuario ha sido autenticado en el sistema.	
Postcondiciones	Se listaron los archivos de configuración asociados a la instancia o <i>plugin</i> seleccionado.	
Prototipo		
Flujo de eventos	Flujo básico Gestionar archivos	
	Actor	Sistema
	Pulsa uno de los enlaces correspondientes a una instancia o <i>plugin</i> , dentro de la lista de instancias o <i>plugins</i> .	<p>Muestra una lista con los archivos correspondientes y algunos datos de los mismos.</p> <p>Permite realizar varias acciones con un archivo:</p> <p>Mostrar detalles de archivo. <u>Ver CU Mostrar detalles de archivo.</u>(Prototipo - 2)</p> <p>Registrar datos de archivo. <u>Ver Sección 1: Registrar datos de archivo del CU Gestionar archivos.</u>(Prototipo - 4)</p> <p>Modificar datos de archivo. <u>Ver Sección 2: Modificar datos de archivo del CU Gestionar archivos.</u>(Prototipo - 3)</p>

		Crear favorito a partir de archivo. <u>Ver Sección 4: Crear favorito a partir de archivo del CU Gestionar archivos</u> .(Prototipo - 1)
		Termina el CU
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 13: Especificación del CU "Mostrar detalles de archivo".

Objetivo	Mostrar detalladamente los datos de un archivo de configuración.	
Actores	Usuario Común	
Resumen	El caso de uso se inicia cuando el actor (en lo adelante usuario) pulsa el botón "Mostrar detalles" de un archivo determinado en la vista "Listar archivos". A partir de aquí el actor puede realizar otras operaciones relacionadas con el archivo. Las operaciones son: <ul style="list-style-type: none"> • Listar archivos • Modificar datos de archivo • Eliminar datos de archivo 	
Complejidad	Baja	
Prioridad	Baja	
Precondiciones	El usuario ha sido autenticado en el sistema.	
Postcondiciones	Se mostraron detalladamente los datos del archivo seleccionado.	
Prototipo		
Flujo de eventos		
Flujo básico Gestionar archivos		
	Actor	Sistema
7.	Pulsa el botón "Mostrar detalles" de un archivo determinado en la vista "Listar archivos".	
8.		Muestra de forma detallada los datos del archivo correspondiente. Permite realizar varias acciones: <ul style="list-style-type: none"> - Listar archivos. <u>Ver CU Listar archivos</u>.(Prototipo - 1) - Modificar datos de archivo. <u>Ver Sección 2: Modificar datos de archivo del CU Gestionar archivos</u>.(Prototipo - 2)

		- Eliminar datos de archivo. <u>Ver Sección 3: Eliminar datos de archivo del CU Gestionar archivos.</u> (Prototipo - 3)
9.		Termina el CU
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 14: Especificación del CU "Listar Instancias".

Objetivo	Listar las instancias de Nutch.	
Actores	Usuario Común	
Resumen	<p>El caso de uso se inicia cuando el actor (en lo adelante usuario) consulta la lista de instancias registradas en el sistema. A partir de aquí el actor puede realizar otras operaciones relacionadas con las instancias. Las operaciones son:</p> <ul style="list-style-type: none"> • Mostrar detalles de instancia • Registrar datos de instancia • Modificar datos de instancia 	
Complejidad	Baja	
Prioridad	Alta	
Precondiciones	El usuario ha sido autenticado en el sistema.	
Postcondiciones	Se listaron las instancias registradas en el sistema.	
Prototipo		
		
Flujo de eventos		
Flujo básico Gestionar instancias		
	Actor	Sistema
10.	Pulsa el enlace "Instancias" del menú principal de navegación.	
11.		<p>Muestra una lista con las instancias correspondientes y algunos datos de las mismas.</p> <p>Permite realizar varias acciones con una instancia:</p> <ul style="list-style-type: none"> - Mostrar detalles de instancia. <u>Ver CU Mostrar detalles de instancia.</u>(Prototipo - 2)

		<ul style="list-style-type: none"> - Registrar datos de instancia. Ver Sección 1: Registrar datos de instancia del CU Gestionar instancias.(Prototipo - 1) - Modificar datos de instancia. Ver Sección 2: Modificar datos de instancia del CU Gestionar instancias.(Prototipo - 3)
12.		Termina el CU
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 15: Especificación del CU "Gestionar instancias".

Objetivo	Gestionar las instancias de Nutch.	
Actores	Administrador	
Resumen	<p>El caso de uso se inicia cuando el actor (en lo adelante usuario) pulsa el botón asociado a una de las siguientes acciones desde la lista de instancias.</p> <ul style="list-style-type: none"> • Mostrar detalles de instancia • Registrar datos de instancia • Modificar datos de instancia • Eliminar datos de instancia 	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	<p>El usuario ha sido autenticado en el sistema. El usuario ha accedido a la lista instancias. Ver CU Listar instancias</p>	
Postcondiciones	Se realizó exitosamente cualquier acción desde la vista "Listar instancias".	
Prototipo		
Flujo de eventos		
Flujo básico Gestionar instancias		
	Actor	Sistema
13.	Pulsa un botón asociado a alguna acción: <ul style="list-style-type: none"> • Mostrar detalles de instancia • Registrar datos de instancia • Modificar datos de instancia 	
14.		Muestra la vista asociada a la acción: <ul style="list-style-type: none"> - Mostrar detalles de instancia. Ver CU <u>Mostrar detalles de instancia.</u>(Prototipo -

		2) - Registrar datos de instancia. Ver Sección 1: "Registrar datos de instancia".(Prototipo - 1) - Modificar datos de instancia. Ver Sección 2: "Modificar datos de instancia".(Prototipo - 3) - Eliminar datos de instancia. Ver Sección 3: "Eliminar datos de instancia".
15.		Termina el CU
Sección 1: "Registrar datos de instancia" (Prototipo - 1)		
Flujo básico Registrar datos de instancia		
	Actor	Sistema
6.	Introduce los datos de la instancia en los campos correspondientes: <ul style="list-style-type: none"> • Nombre • Ruta • Descripción Pulsa el botón "Guardar".	
7.		Valida los datos entrados por el usuario.
8.		Registra los datos de la instancia. Muestra el mensaje "La instancia se ha registrado satisfactoriamente".
9.		Redirecciona a la vista "Mostrar detalles". <u>Ver CU Mostrar detalles de instancia</u>
10.		Termina Sección 1.
Flujos alternos		
1ª El usuario pulsa el botón "Cancelar".		
	Actor	Sistema
1.1		Redirecciona a la vista "Lista de instancias". Vuelve al paso 1 del flujo básico del CU.
2ª Los datos entrados son incorrectos.		
	Actor	Sistema
3.1		Muestra el mensaje "Error: Algunos datos son incorrectos".
3.2	Corrige los datos incorrectos y pulsa el botón "Guardar"	
3.3		Vuelve al paso 2 del flujo básico de la Sección 1.
Sección 2: "Modificar datos de instancia" (Prototipo - 3)		
Flujo básico Modificar datos de instancia		
	Actor	Sistema
6.	Modifica los datos de los campos correspondientes: <ul style="list-style-type: none"> • Nombre • Ruta • Descripción Pulsa el botón "Guardar".	
7.		Valida los datos modificados por el usuario.
8.		Registra los datos modificados. Muestra el mensaje "La instancia se ha modificado satisfactoriamente".
9.		Redirecciona a la vista "Mostrar detalles". <u>Ver CU Mostrar detalles de instancia</u>
10.		Termina Sección 2.
Flujos alternos		
1ª El usuario cancela la edición de la instancia.		
	Actor	Sistema
1.1		Redirecciona a la vista "Lista de instancias". Vuelve al paso 1 del flujo básico del CU.
2ª El usuario elige eliminar datos de instancia.		
	Actor	Sistema
1.1		Ver sección 3 del CU.

1.2		Termina Sección 2.
3ª Los datos modificados son incorrectos.		
	Actor	Sistema
3.1		Muestra el mensaje "Error: Algunos datos son incorrectos".
3.2	Corrige los datos incorrectos y pulsa el botón "Guardar".	
3.3		Vuelve al paso 2 del flujo básico de la Sección 2.
Sección 3: "Eliminar datos de instancia"		
Flujo básico Eliminar datos de instancia		
	Actor	Sistema
6.	Pulsa el botón "Eliminar".	
7.		Muestra un cuadro de diálogo pidiendo confirmar la acción con el mensaje: "¿Realmente desea eliminar este dato?".
8.	Pulsa el botón "Si".	
9.		Elimina los datos de la instancia y redirecciona a la vista "Lista de instancias". Muestra el mensaje "La instancia se ha eliminado satisfactoriamente".
10.		Termina Sección 3.
Flujos alternos		
1ª El usuario cancela la eliminación.		
	Actor	Sistema
2.1		Quita el cuadro de diálogo de verificación y no elimina los datos de la instancia.
2.2		Termina Sección 3.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 16: Especificación del CU "Mostrar detalles de instancia".

Objetivo	Mostrar detalladamente los datos de una instancia.
Actores	Usuario Común
Resumen	El caso de uso se inicia cuando el actor (en lo adelante usuario) pulsa el botón "Mostrar detalles" de una instancia determinada en la vista "Listar instancias". A partir de aquí el actor puede realizar otras operaciones relacionadas con la instancia. Las operaciones son: <ul style="list-style-type: none"> • Listar instancias • Modificar datos de instancia • Eliminar datos de instancia
Complejidad	Baja
Prioridad	Baja
Precondiciones	El usuario ha sido autenticado en el sistema.
Postcondiciones	Se mostraron detalladamente los datos de la instancia seleccionada.
Prototipo	

ORIÓN Configuración Instancias Perfiles Favoritos Etiquetas Administrador

Detalles de instancia "Nutch_Imagenes"

Nombre: Nutch_Imagenes

Ruta: /var/www/nutch_imagenes

Descripción: Instancia para la recuperación de imágenes

Lista de instancias
Editar
Eliminar

Flujo de eventos					
Flujo básico Gestionar instancias					
Actor	Sistema				
16.	Pulsa el botón "Mostrar detalles" de una instancia determinada en la vista "Listar instancias".				
17.	<p>Muestra de forma detallada los datos de la instancia correspondiente.</p> <p>Permite realizar varias acciones:</p> <ul style="list-style-type: none"> - Listar instancias. Ver CU Listar instancias. (Prototipo - 1) - Modificar datos de instancia. Ver Sección 2: Modificar datos de instancia del CU Gestionar instancias. (Prototipo - 2) - Eliminar datos de instancia. Ver Sección 3: Eliminar datos de instancia del CU Gestionar instancias. (Prototipo - 3) 				
18.	Termina el CU				
Relaciones	<table border="1"> <tr> <td>CU Incluidos</td> <td></td> </tr> <tr> <td>CU Extendidos</td> <td></td> </tr> </table>	CU Incluidos		CU Extendidos	
CU Incluidos					
CU Extendidos					
Requisitos no funcionales					
Asuntos pendientes					

Tabla 17: Especificación del CU "Listar plugins".

Objetivo	Listar los <i>plugins</i> de cada instancia de Nutch.
Actores	Usuario Común
Resumen	<p>El caso de uso se inicia cuando el actor (en lo adelante usuario) consulta la lista de <i>plugins</i> de una instancia. A partir de aquí el actor puede realizar otras operaciones relacionadas con los <i>plugins</i>. Las operaciones son:</p> <ul style="list-style-type: none"> • Mostrar detalles de <i>plugin</i> • Registrar datos de <i>plugin</i> • Modificar datos de <i>plugin</i> • Registrar automáticamente los <i>plugins</i> activos • Activar <i>plugin</i> • Desactivar <i>plugin</i>
Complejidad	Baja
Prioridad	Alta
Precondiciones	El usuario ha sido autenticado en el sistema.
Postcondiciones	Se listaron los <i>plugins</i> asociados a la instancia seleccionada.
Prototipo	

ORIÓN Configuración Instancias Perfiles Favoritos Etiquetas Administrador

Plugins de Instancia "Nutch-Imágenes"

 <p>Nombre: link-extractor Ruta: /var/www/Nutch-Imágenes/plugins</p> <p>Plugin para la extracción de metadatos de los enlaces web.</p>	<input checked="" type="checkbox"/> Activar (1) <input type="checkbox"/> (2) <input type="checkbox"/> (3)
 <p>Nombre: url-filter-domain Ruta: /var/www/Nutch-Imágenes/plugins</p> <p>Plugin para el reconocimiento de dominios en Internet.</p>	<input checked="" type="checkbox"/> Desactivar (4) <input type="checkbox"/> <input type="checkbox"/>
 <p>Nombre: images-metadata Ruta: /var/www/Nutch-Imágenes/plugins</p> <p>Plugin para la extracción de metadatos de las imágenes.</p>	<input checked="" type="checkbox"/> Activar <input type="checkbox"/> <input type="checkbox"/>

+ Registrar plugin (5) + Registrar plugins automáticamente (6)

Flujo de eventos		
Flujo básico Gestionar archivos		
	Actor	Sistema
19.	Pulsa uno de los enlaces correspondientes a una instancia, dentro de la lista de instancias.	
20.		<p>Muestra una lista con los <i>plugins</i> correspondientes y algunos datos de los mismos.</p> <p>Permite realizar varias acciones:</p> <ul style="list-style-type: none"> - Mostrar detalles de plugin. Ver CU <u>Mostrar detalles de plugin</u>.(Prototipo - 2) - Registrar datos de plugin. Ver Sección 1: <u>Registrar datos de plugin del CU Gestionar plugins</u>.(Prototipo - 5) - Modificar datos de plugin. Ver Sección 2: <u>Modificar datos de plugin del CU Gestionar plugins</u>.(Prototipo - 3) - Registrar automáticamente los <i>plugins</i> activos. Ver Sección 4: <u>Registrar automáticamente los plugins activos del CU Gestionar plugins</u>.(Prototipo - 6) - Activar plugin. Ver Sección 5: <u>Activar plugin del CU Gestionar plugins</u>.(Prototipo - 1) - Desactivar plugin. Ver Sección 6: <u>Desactivar plugin del CU Gestionar plugins</u>.(Prototipo - 4)
21.		Termina el CU
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 18: Especificación del CU "Gestionar *plugins*".

Objetivo	Gestionar los <i>plugins</i> de cada instancia de Nutch.
Actores	Administrador

Resumen	<p>El caso de uso se inicia cuando el actor (en lo adelante usuario) pulsa el botón asociado a una de las siguientes acciones desde la lista de <i>plugins</i> correspondientes a una instancia.</p> <ul style="list-style-type: none"> • Mostrar detalles de <i>plugin</i> • Registrar datos de <i>plugin</i> • Modificar datos de <i>plugin</i> • Eliminar datos de <i>plugin</i> • Registrar automáticamente los <i>plugins</i> activos • Activar <i>plugin</i>. • Desactivar <i>plugin</i>.
Complejidad	Media
Prioridad	Alta
Precondiciones	El usuario ha sido autenticado en el sistema. El usuario ha accedido a la lista <i>plugins</i> de una instancia. Ver CU Listar plugins
Postcondiciones	Se realizó exitosamente cualquier acción desde la vista "Listar <i>plugins</i> ".

Prototipo



Flujo de eventos

Flujo básico Gestionar plugins


	Actor	Sistema
22.	<p>Pulsa un botón asociado a alguna acción:</p> <ul style="list-style-type: none"> • Mostrar detalles de <i>plugin</i> • Registrar datos de <i>plugin</i> • Modificar datos de <i>plugin</i> • Registrar automáticamente los <i>plugins</i> activos • Activar/Desactivar <i>plugin</i> 	
23.		<p>Muestra la vista asociada a la acción:</p> <ul style="list-style-type: none"> - Mostrar detalles de <i>plugin</i>. Ver CU <u>Mostrar detalles de plugin</u>. (Prototipo - 2) - Registrar datos de <i>plugin</i>. Ver Sección 1: "Registrar datos de <i>plugin</i>". (Prototipo - 5) - Modificar datos de <i>plugin</i>. Ver Sección 2: "Modificar datos de <i>plugin</i>". (Prototipo - 3) - Eliminar datos de <i>plugin</i>. Ver Sección 3: "Eliminar datos de <i>plugin</i>". - Registrar automáticamente los <i>plugins</i> activos. Ver Sección 4: "Registrar automáticamente los <i>plugins</i> activos". (Prototipo - 6) - Activar <i>plugin</i>. Ver Sección 5: Activar <i>plugin</i>. (Prototipo - 1)

		- Desactivar <i>plugin</i> . Ver Sección 6: Desactivar <i>plugin</i> . (Prototipo - 4)
24.		Termina el CU
Sección 1: “Registrar datos de <i>plugin</i>” (Prototipo - 5)		
Flujo básico Registrar datos de <i>plugin</i>		
	Actor	Sistema
11.	Introduce los datos del <i>plugin</i> en los campos correspondientes: <ul style="list-style-type: none"> • Nombre • Ruta • Descripción Pulsa el botón “Guardar”.	
12.		Valida los datos entrados por el usuario.
13.		Registra los datos del <i>plugin</i> . Muestra el mensaje “El <i>plugin</i> se ha registrado satisfactoriamente”.
14.		Redirecciona a la vista “Mostrar detalles”. <u>Ver CU Mostrar detalles de <i>plugin</i></u>
15.		Termina Sección 1.
Flujos alternos		
1ª El usuario pulsa el botón “Cancelar”.		
	Actor	Sistema
1.1		Redirecciona a la vista “Lista de <i>plugins</i> ”.
		Vuelve al paso 1 del flujo básico del CU.
2ª Los datos entrados son incorrectos.		
	Actor	Sistema
3.1		Muestra el mensaje “Error: Algunos datos son incorrectos”.
3.2	Corrige los datos incorrectos y pulsa el botón “Guardar”	
3.3		Vuelve al paso 2 del flujo básico de la Sección 1.
Sección 2: “Modificar datos de <i>plugin</i>” (Prototipo - 3)		
Flujo básico Modificar datos de <i>plugin</i>		
	Actor	Sistema
11.	Modifica los datos de los campos correspondientes: <ul style="list-style-type: none"> • Nombre • Ruta • Descripción Pulsa el botón “Guardar”.	
12.		Valida los datos modificados por el usuario.
13.		Registra los datos modificados. Muestra el mensaje “El <i>plugin</i> se ha modificado satisfactoriamente”.
14.		Redirecciona a la vista “Mostrar detalles”. <u>Ver CU Mostrar detalles de <i>plugin</i></u>
15.		Termina Sección 2.
Flujos alternos		
1ª El usuario cancela la edición del <i>plugin</i> .		
	Actor	Sistema
1.1		Redirecciona a la vista “Lista de <i>plugins</i> ”.
		Vuelve al paso 1 del flujo básico del CU.
2ª El usuario elige eliminar datos de <i>plugin</i> .		
	Actor	Sistema
1.1		Ver sección 3 del CU.
1.2		Termina Sección 2.
3ª Los datos modificados son incorrectos.		
	Actor	Sistema
3.1		Muestra el mensaje “Error: Algunos datos son incorrectos”.
3.2	Corrige los datos incorrectos y pulsa el botón “Guardar”.	

3.3		Vuelve al paso 2 del flujo básico de la Sección 2.
Sección 3: “Eliminar datos de <i>plugin</i>”		
Flujo básico Eliminar datos de <i>plugin</i>		
	Actor	Sistema
11.	Pulsa el botón “Eliminar”.	
12.		Muestra un cuadro de diálogo pidiendo confirmar la acción con el mensaje: “¿Realmente desea eliminar este dato?”.
13.	Pulsa el botón “Si”.	
14.		Elimina los datos del <i>plugin</i> y redirecciona a la vista “Lista de <i>plugins</i> ”. Muestra el mensaje “El <i>plugin</i> se ha eliminado satisfactoriamente”.
15.		Termina Sección 3.
Flujos alternos		
1ª El usuario cancela la eliminación.		
	Actor	Sistema
2.1		Quita el cuadro de diálogo de verificación y no elimina los datos del <i>plugin</i> .
2.2		Termina Sección 3.
Sección 4: “Registrar automáticamente los <i>plugins</i> activos” (Prototipo - 6)		
Flujo básico Registrar automáticamente los <i>plugins</i> activos		
	Actor	Sistema
3.		Determina cuáles son los <i>plugins</i> activos en el archivo principal definido en el sistema (nutch-site.xml).
4.		Verifica que cada <i>plugin</i> activo exista en el directorio especificado en el archivo principal definido en el sistema (Nutch-site.xml).
5.		Registra automáticamente en el sistema los datos de cada <i>plugin</i> .
6.		Termina Sección 4.
Flujos alternos		
1ª Al menos un <i>plugin</i> no existe en el directorio definido.		
	Actor	Sistema
2.1		Registra los <i>plugins</i> existentes. Muestra el mensaje “Algunos <i>plugins</i> no han podido registrarse”.
2.2		Termina Sección 4.
Sección 5: “Activar <i>plugin</i>” (Prototipo - 1)		
Flujo básico Activar <i>plugin</i>		
	Actor	Sistema
1.		Verifica que el <i>plugin</i> exista en el directorio especificado en el archivo principal definido en el sistema (Nutch-site.xml).
2.		Activa el <i>plugin</i> . Muestra el mensaje “El <i>plugin</i> ha sido activado satisfactoriamente”.
3.		Termina Sección 5.
Flujos alternos		
1ª El <i>plugin</i> no existe en el directorio definido.		
	Actor	Sistema
2.1		No activa el <i>plugin</i> . Muestra el mensaje “El <i>plugin</i> no se pudo activar porque no existe”.
2.2		Termina Sección 5.
Sección 6: “Desactivar <i>plugin</i>” (Prototipo - 4)		
Flujo básico Desactivar <i>plugin</i>		
	Actor	Sistema
1.		Desactiva el <i>plugin</i> . Muestra el mensaje “El <i>plugin</i> ha sido desactivado satisfactoriamente”.

2.	Termina Sección 4.	
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 19: Especificación del CU "Mostrar detalles de *plugin*".

Objetivo	Mostrar detalladamente los datos de un <i>plugin</i> .	
Actores	Usuario Común	
Resumen	<p>El caso de uso se inicia cuando el actor (en lo adelante usuario) pulsa el botón "Mostrar detalles" de un <i>plugin</i> determinado en la vista "Listar <i>plugins</i>". A partir de aquí el actor puede realizar otras operaciones relacionadas con el <i>plugin</i>. Las operaciones son:</p> <ul style="list-style-type: none"> • Listar <i>plugins</i> • Modificar datos de <i>plugin</i> • Eliminar datos de <i>plugin</i> 	
Complejidad	Baja	
Prioridad	Baja	
Precondiciones	El usuario ha sido autenticado en el sistema.	
Postcondiciones	Se mostraron detalladamente los datos del <i>plugin</i> seleccionado.	
Prototipo		
Flujo de eventos	Flujo básico Gestionar archivos	
	Actor	Sistema
25.	Pulsa el botón "Mostrar detalles" de un <i>plugin</i> determinado en la vista "Listar <i>plugins</i> ".	
26.		<p>Muestra de forma detallada los datos del <i>plugin</i> correspondiente.</p> <p>Permite realizar varias acciones:</p> <ul style="list-style-type: none"> - Listar <i>plugins</i>. Ver CU Listar <i>plugins</i>. (Prototipo - 1) - Modificar datos de <i>plugin</i>. Ver Sección 2: <u>Modificar datos de plugin del CU Gestionar plugins</u>. (Prototipo - 2) - Eliminar datos de <i>plugin</i>. Ver Sección 3: <u>Eliminar datos de plugin del CU Gestionar plugins</u>. (Prototipo - 3)
27.		Termina el CU
Relaciones	CU Incluidos	
	CU Extendidos	

Requisitos no funcionales	
Asuntos pendientes	

Tabla 20: Especificación del CU "Listar modificaciones".

Objetivo	Listar las modificaciones de un archivo de configuración de Nutch.
Actores	Usuario Común
Resumen	El caso de uso se inicia cuando el actor (en lo adelante usuario) consulta la lista de modificaciones de un archivo de configuración. A partir de aquí el actor puede realizar otras operaciones relacionadas con las modificaciones. Las operaciones son: <ul style="list-style-type: none"> • Mostrar detalles de modificación • Crear propiedad de perfil a partir de modificación • Archivar modificaciones
Complejidad	Media
Prioridad	Alta
Precondiciones	El usuario ha sido autenticado en el sistema.
Postcondiciones	Se listaron las modificaciones asociadas al archivo de configuración seleccionado.

Prototipo

Flujo de eventos		
Flujo básico Gestionar modificaciones de archivos		
	Actor	Sistema
1.	Pulsa uno de los enlaces de un archivo de la lista de archivos de la instancia o <i>plugin</i> actual.	
2.		<p>Muestra una lista con las modificaciones del archivo correspondiente y algunos datos de las mismas.</p> <p>Permite realizar varias acciones con una modificación:</p> <ul style="list-style-type: none"> - Mostrar detalles de modificación. <u>Ver CU <i>Mostrar detalles de modificación.</i></u> (Prototipo - 2) - Crear propiedad de perfil a partir de modificación. <u>Ver Sección 1: Crear</u>

		<p><u>propiedad de perfil a partir de modificación del CU Gestionar modificaciones de archivos.</u> (Prototipo - 1)</p> <p>- Archivar modificaciones. <u>Ver CU Archivar modificaciones.</u> (Prototipo - 3)</p>
3.		Termina el CU
Relaciones	CU Incluidos	
	CU Extendidos	Archivar modificaciones en el CU Archivar modificaciones.
Requisitos no funcionales		
Asuntos pendientes		

Tabla 21: Especificación del CU "Gestionar modificaciones".

Objetivo	Gestionar el historial de modificaciones de un archivo de configuración de Nutch.
Actores	Administrador
Resumen	<p>El caso de uso se inicia cuando el actor (en lo adelante usuario) pulsa el botón asociado a una de las siguientes acciones desde la lista de modificaciones de un archivo de configuración.</p> <ul style="list-style-type: none"> Mostrar detalles de modificación Crear propiedad de perfil a partir de modificación Archivar modificaciones
Complejidad	Media
Prioridad	Alta
Precondiciones	<p>El usuario ha sido autenticado en el sistema.</p> <p>El usuario ha accedido a la lista de modificaciones de un archivo de configuración. <u>Ver CU Listar modificaciones</u></p>
Postcondiciones	Se realizó exitosamente cualquier otra acción desde la vista "Listar modificaciones".

Prototipo



Flujo de eventos		
Flujo básico Gestionar modificaciones de archivos		
	Actor	Sistema
4.	Pulsa un botón asociado a alguna acción:	

	<ul style="list-style-type: none"> Mostrar detalles de modificación Crear propiedad de perfil a partir de modificación Archivar modificaciones. 	
5.		Muestra la vista asociada a la acción: <ul style="list-style-type: none"> Mostrar detalles de modificación. <u>Ver CU <i>Mostrar detalles de modificación.</i></u> (Prototipo - 2) Crear propiedad de perfil a partir de modificación. Ver Sección 1: "Crear propiedad de perfil a partir de modificación". (Prototipo - 1) Archivar modificaciones. Ver Sección 2: "Archivar modificaciones". (Prototipo - 3)
6.		Termina el CU
Sección 1: "Crear propiedad de perfil a partir de modificación" (Prototipo - 1)		
Flujo básico Crear propiedad de perfil a partir de modificación		
	Actor	Sistema
1.		Muestra un cuadro de diálogo, dando la opción de especificar el perfil asociado.
2.	Selecciona el perfil. Pulsa el botón "Crear"	
3.		Registra los datos de la propiedad de perfil y muestra el mensaje "La propiedad de perfil se ha creado satisfactoriamente".
4.		Termina Sección 2.
Flujos alternos		
1ª El usuario cancela la creación de propiedad de perfil.		
	Actor	Sistema
2.1		Quita el cuadro de diálogo y no registra los datos de la propiedad de perfil.
		Termina Sección 2.
Sección 2: "Archivar modificaciones" (Prototipo - 3)		
Flujo básico Archivar modificaciones		
	Actor	Sistema
1.		Archivar modificaciones en el CU Archivar modificaciones.
2.		Termina Sección 3.
Relaciones	CU Incluidos	
	CU Extendidos	Archivar modificaciones en el CU Archivar modificaciones.
Requisitos no funcionales		
Asuntos pendientes		

Tabla 22: Especificación del CU "Mostrar detalles de modificación".

Objetivo	Mostrar detalladamente los datos de una modificación de un archivo de configuración.
Actores	Usuario Común
Resumen	El caso de uso se inicia cuando el actor (en lo adelante usuario) pulsa el botón "Mostrar detalles" de una modificación en la vista "Listar modificaciones". A partir de aquí el actor puede también dirigirse a la vista "Listar modificaciones".
Complejidad	Baja
Prioridad	Baja

Precondiciones	El usuario ha sido autenticado en el sistema.
Postcondiciones	Se mostraron detalladamente los datos de la modificación del archivo de configuración.

Prototipo

Flujo de eventos		
Flujo básico Gestionar archivos		
	Actor	Sistema
28.	Pulsa el botón "Mostrar detalles" de una modificación en la vista "Listar modificaciones".	
29.		Muestra de forma detallada los datos de la modificación correspondiente. Permite listar las modificaciones del archivo asociado. <u>Ver CU Listar modificaciones.</u> (Prototipo - 1)
30.		Termina el CU
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Anexo # 5: Diagramas de clases del diseño.

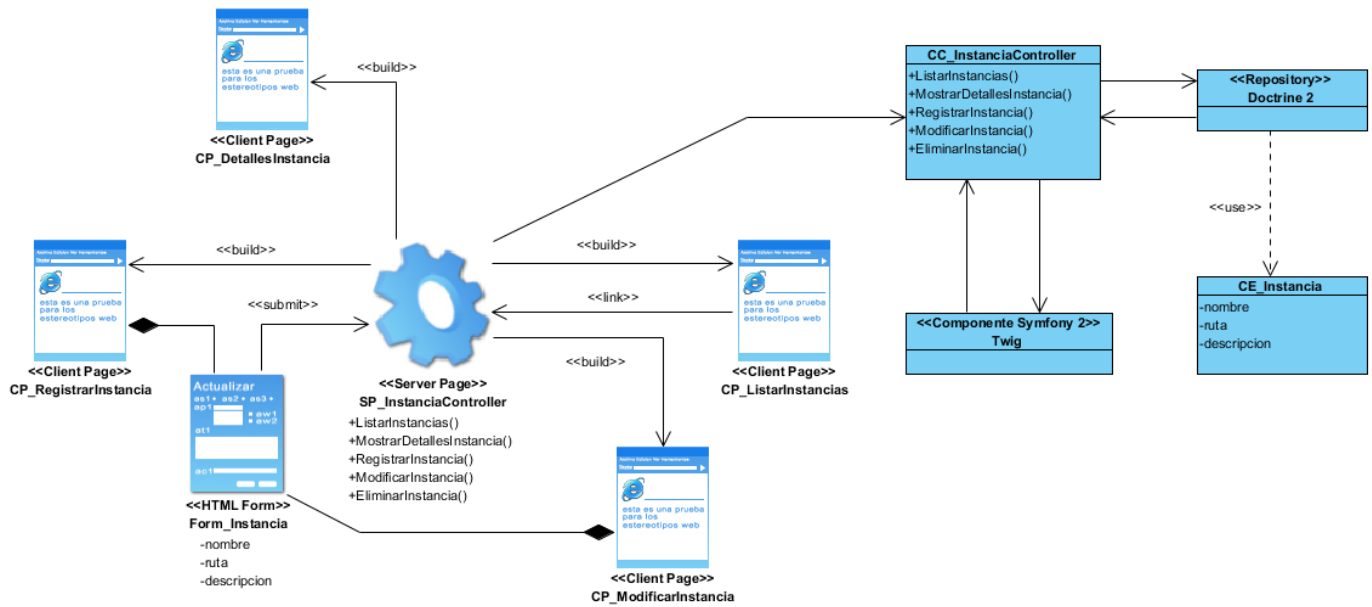


Figura 32: Diagrama de clases del diseño con estereotipos web del CU "Gestionar instancias".

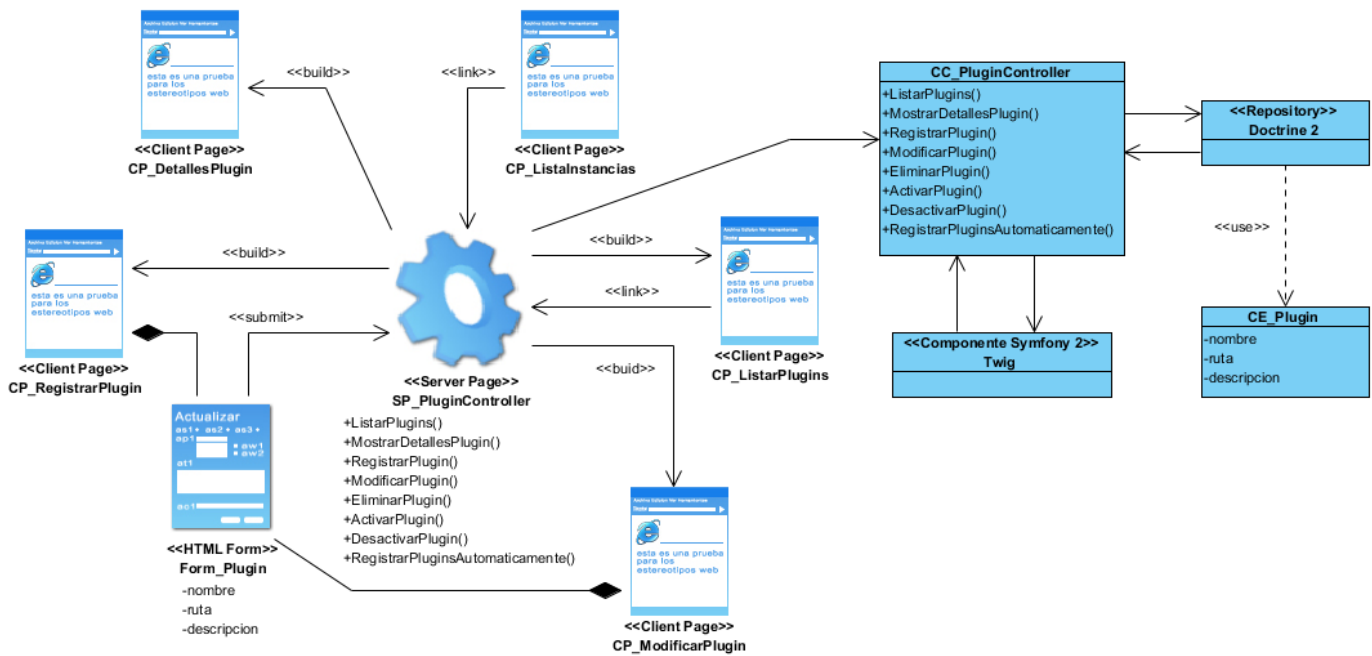


Figura 33: Diagrama de clases del diseño con estereotipos web del CU "Gestionar plugins".

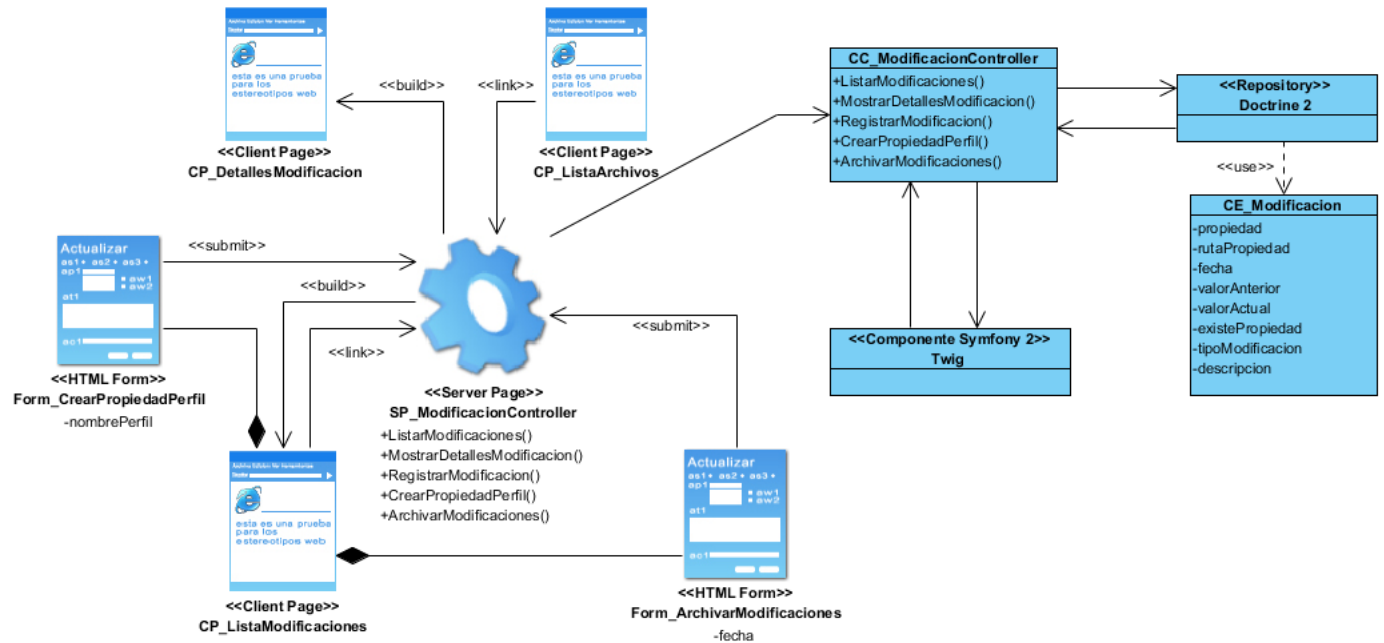


Figura 34: Diagrama de clases del diseño con estereotipos web del CU "Gestionar modificaciones".

Anexo # 6: Diagramas de secuencia.

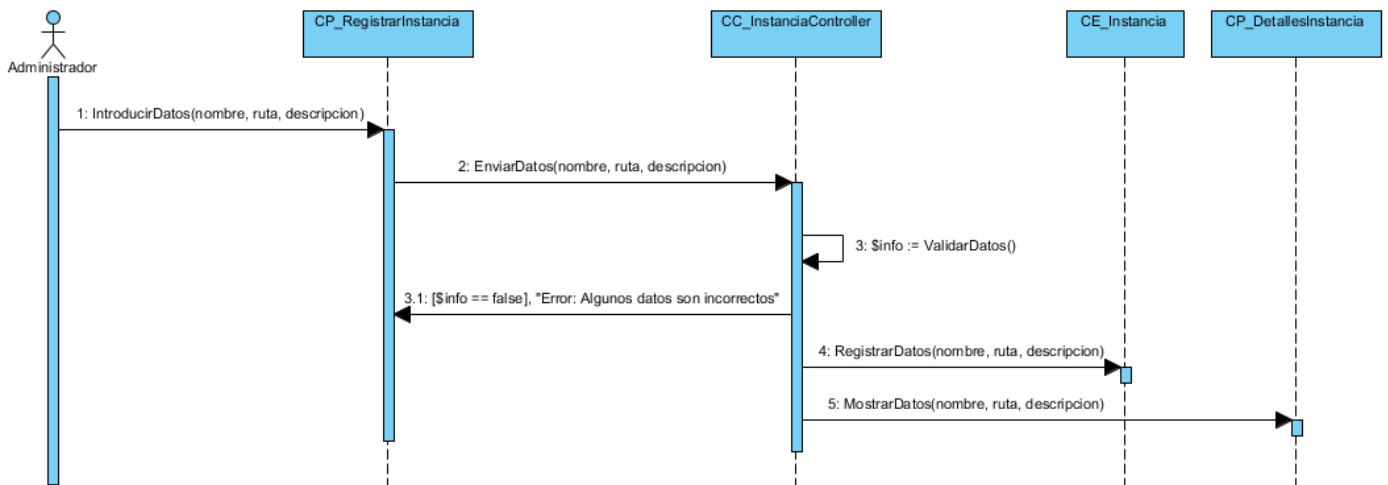


Figura 35: Diagrama de secuencia del escenario "Registrar instancia" del CU "Gestionar instancias".

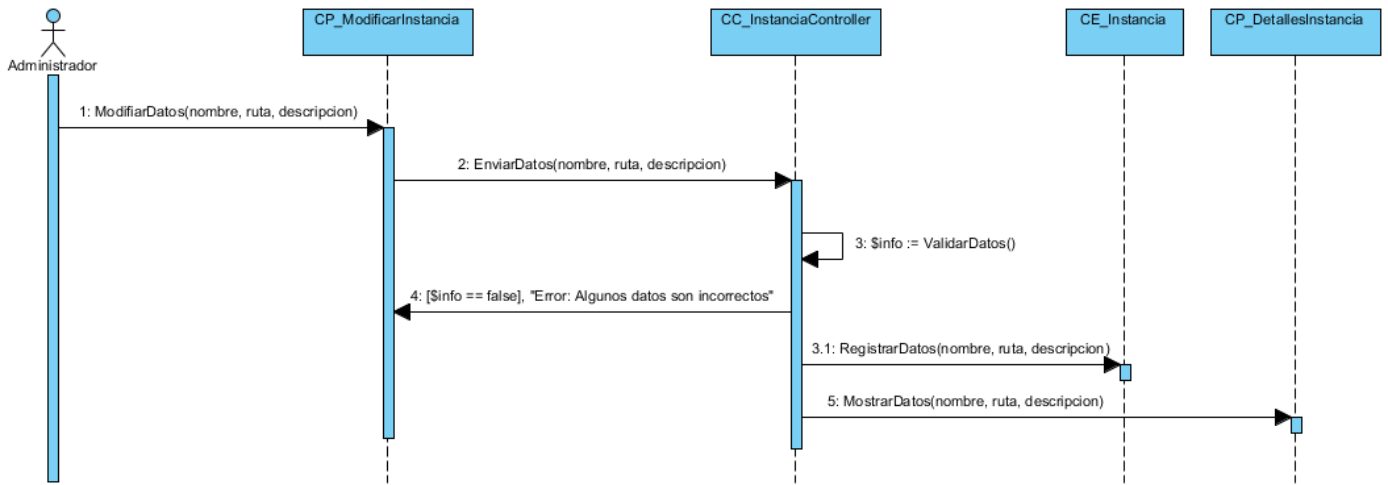


Figura 36: Diagrama de secuencia del escenario "Modificar instancia" del CU "Gestionar instancias".

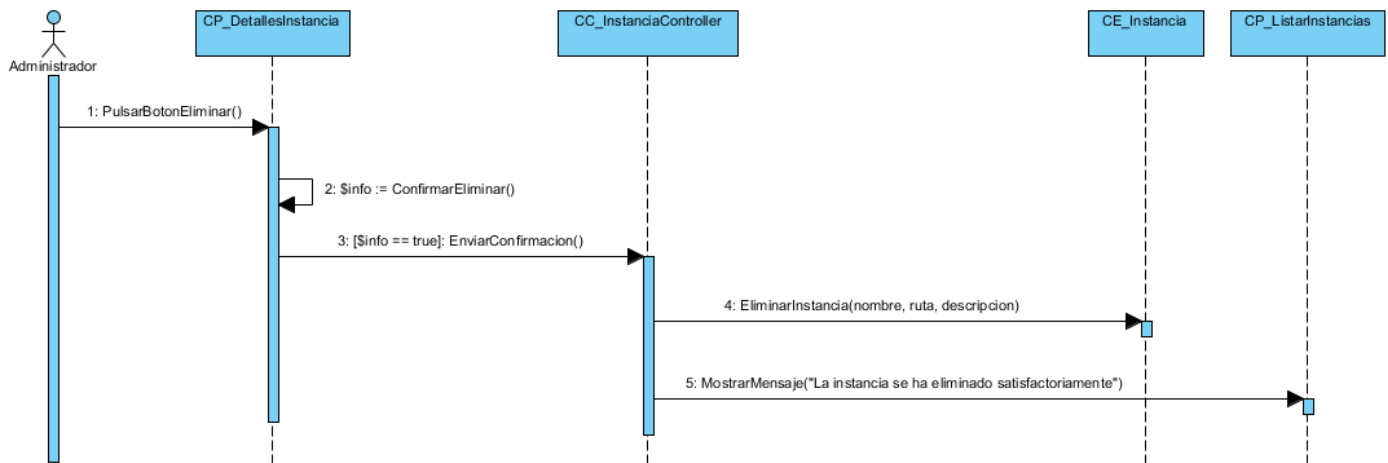


Figura 37: Diagrama de secuencia del escenario "Eliminar instancia" del CU "Gestionar instancias".

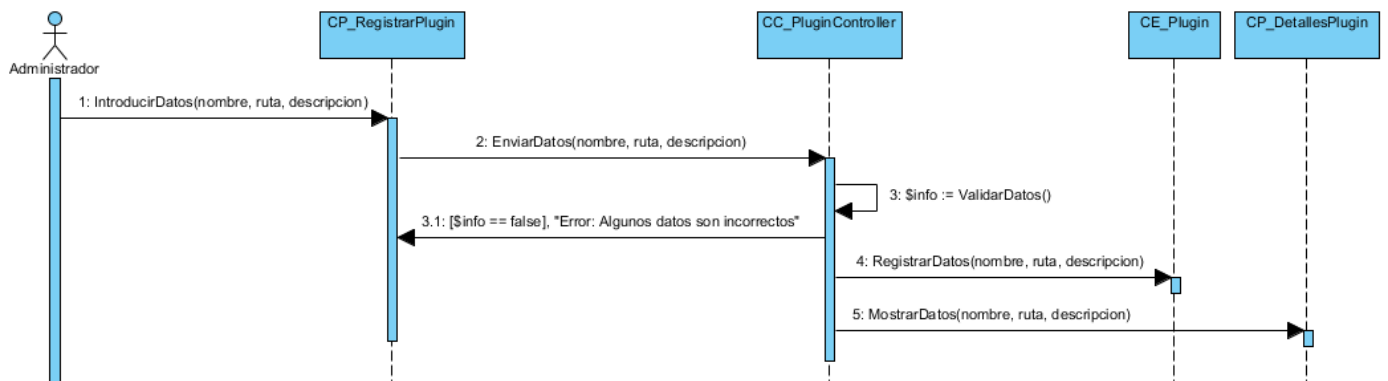


Figura 38: Diagrama de secuencia del escenario "Registrar plugin" del CU "Gestionar plugins".

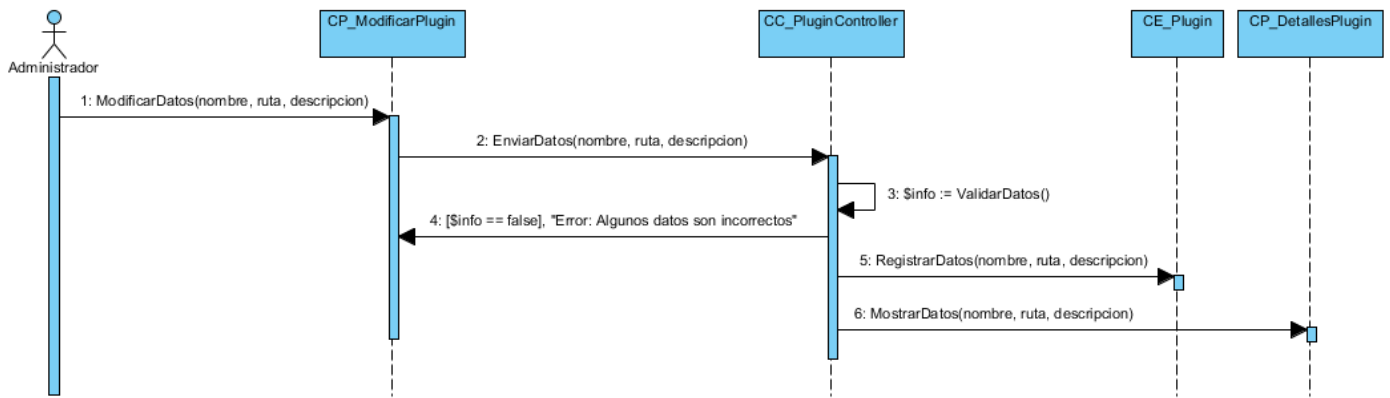


Figura 39: Diagrama de secuencia del escenario “Modificar *plugin*” del CU “Gestionar *plugins*”.

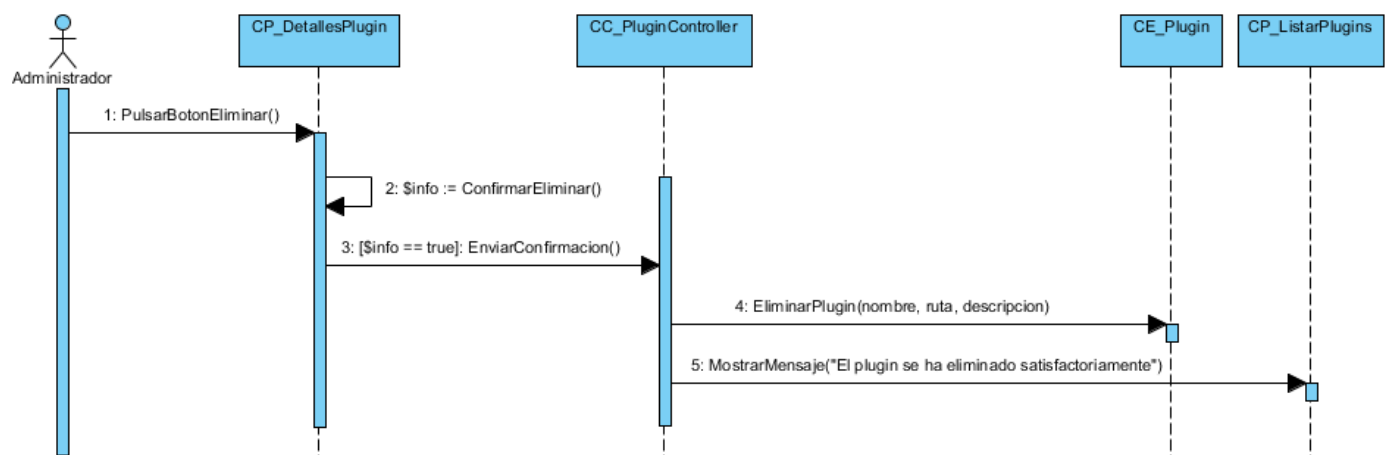


Figura 40: Diagrama de secuencia del escenario “Eliminar *plugin*” del CU “Gestionar *plugins*”.

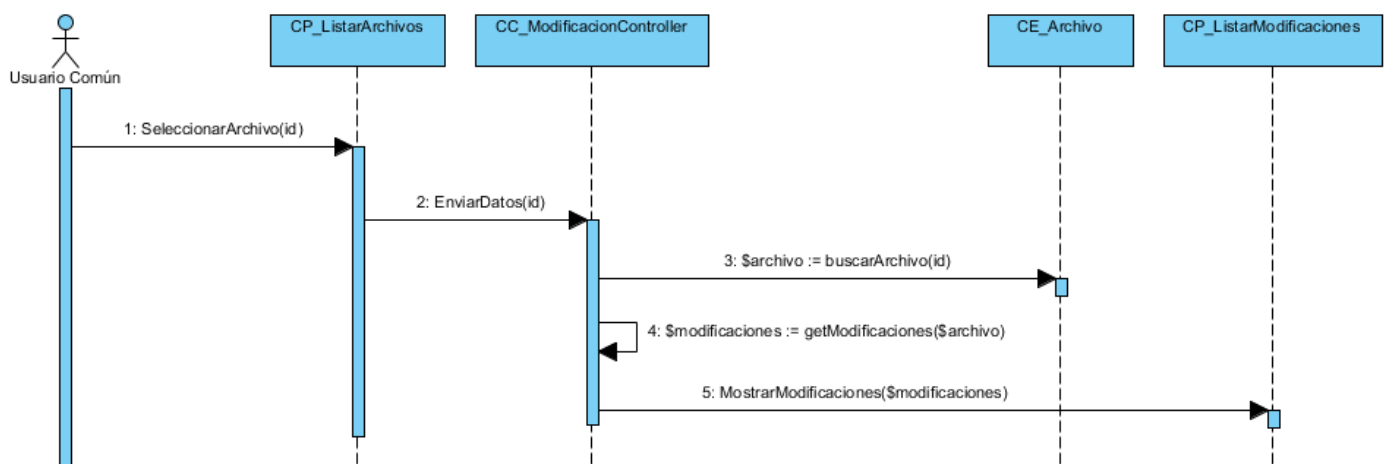


Figura 41: Diagrama de secuencia del escenario “Listar modificaciones” del CU “Gestionar modificaciones”.

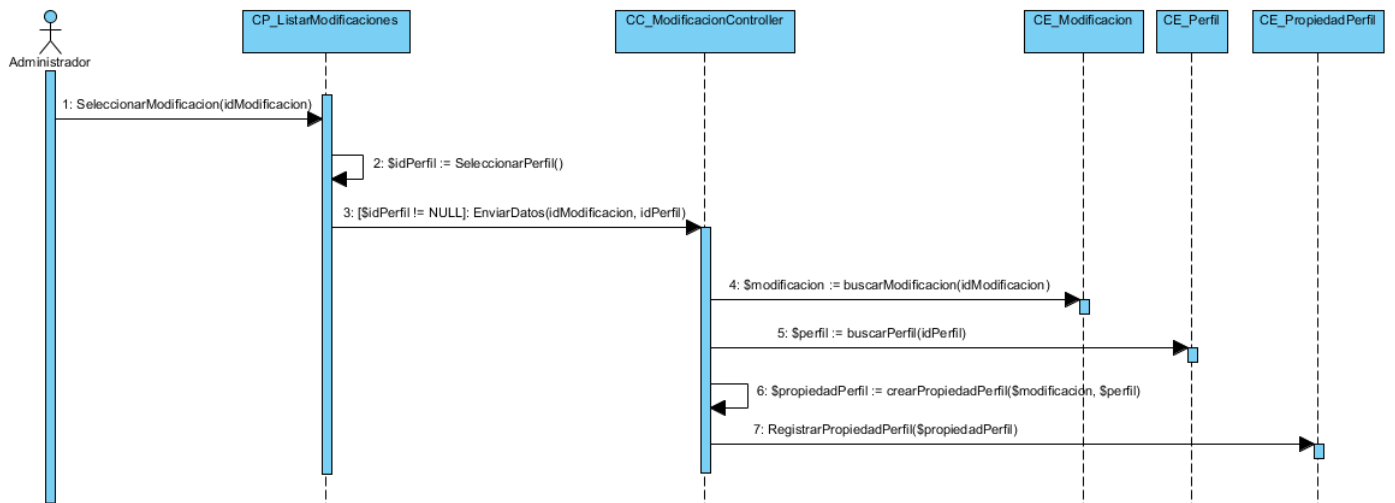


Figura 42: Diagrama de secuencia del escenario “Crear propiedad de perfil a partir de modificación” del CU “Gestionar modificaciones”.

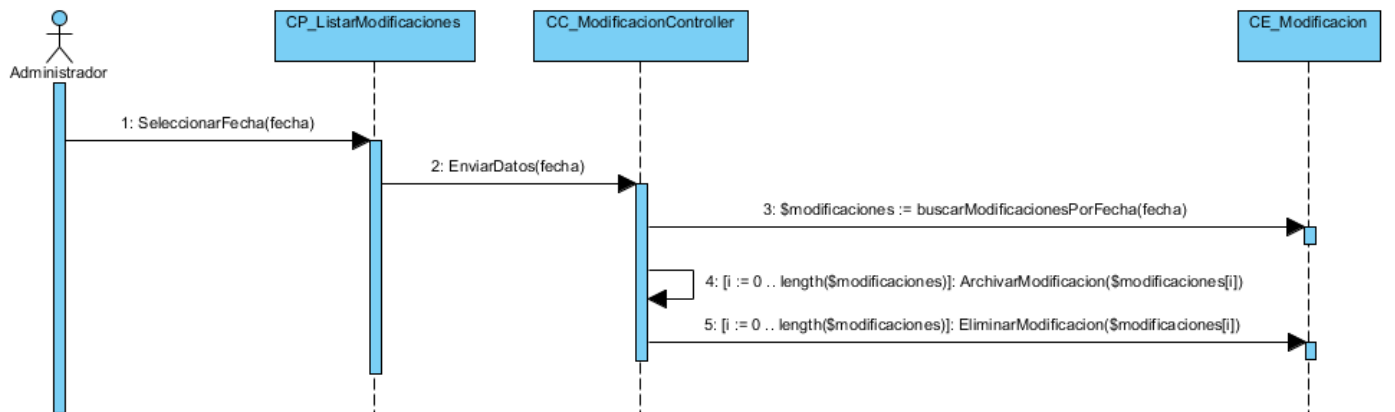


Figura 43: Diagrama de secuencia del escenario “Archivar modificaciones” del CU “Gestionar modificaciones”.

Anexo # 7: Principales imágenes del sistema.



Copyright © 2014 - Universidad de las Ciencias Informáticas
Sistema para la configuración del mecanismo de rastreo de Orión 2.0 | Motor de Búsqueda Cubano

Figura 44: Vista de autenticación del sistema.

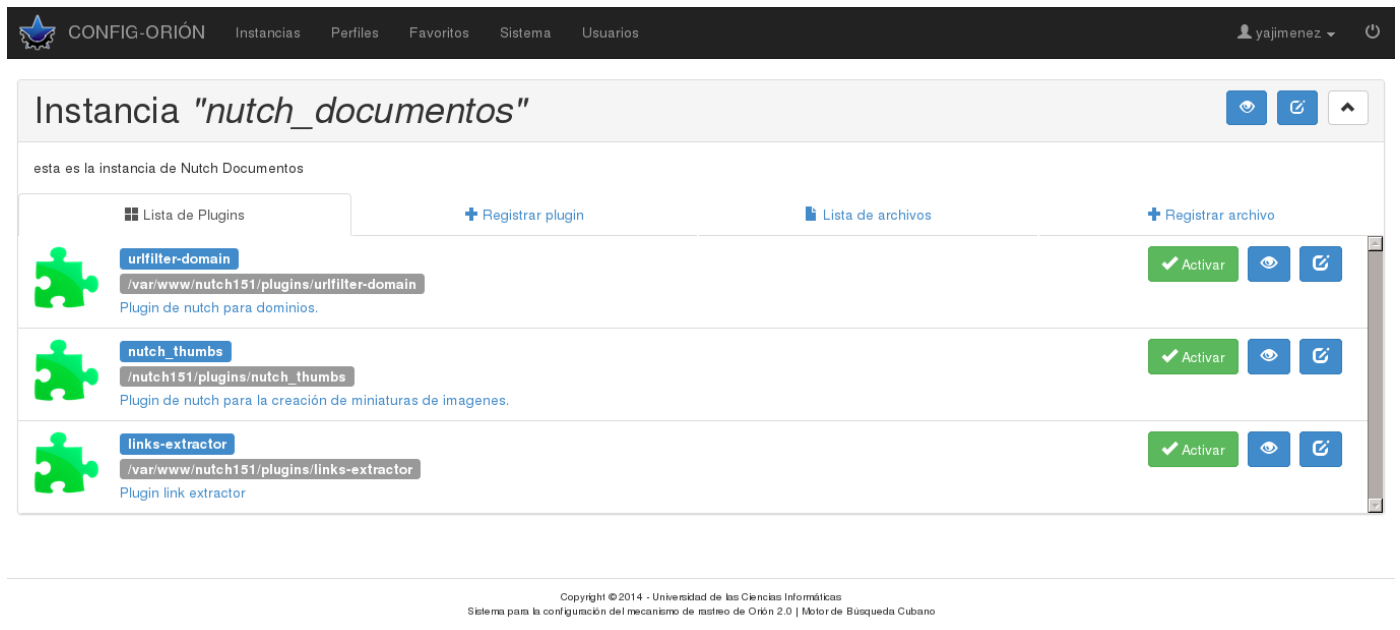


Figura 45: Vista de gestión de *plugins*.

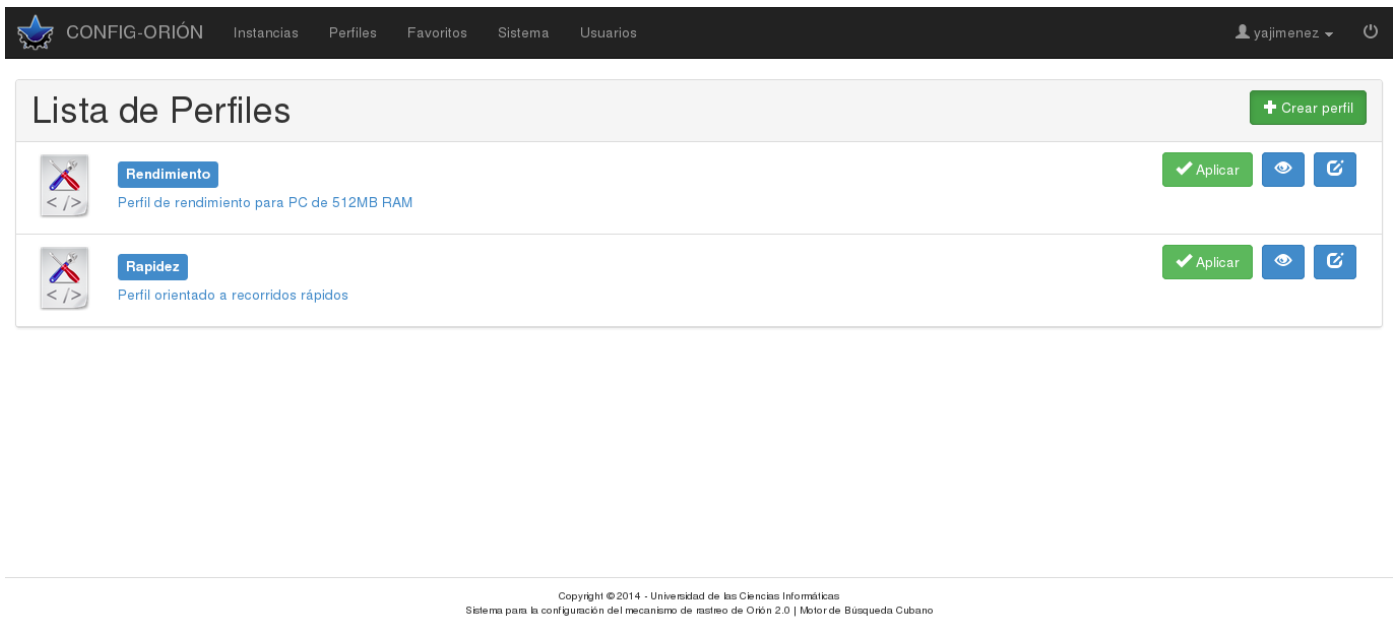


Figura 46: Vista de gestión de perfiles de configuración.

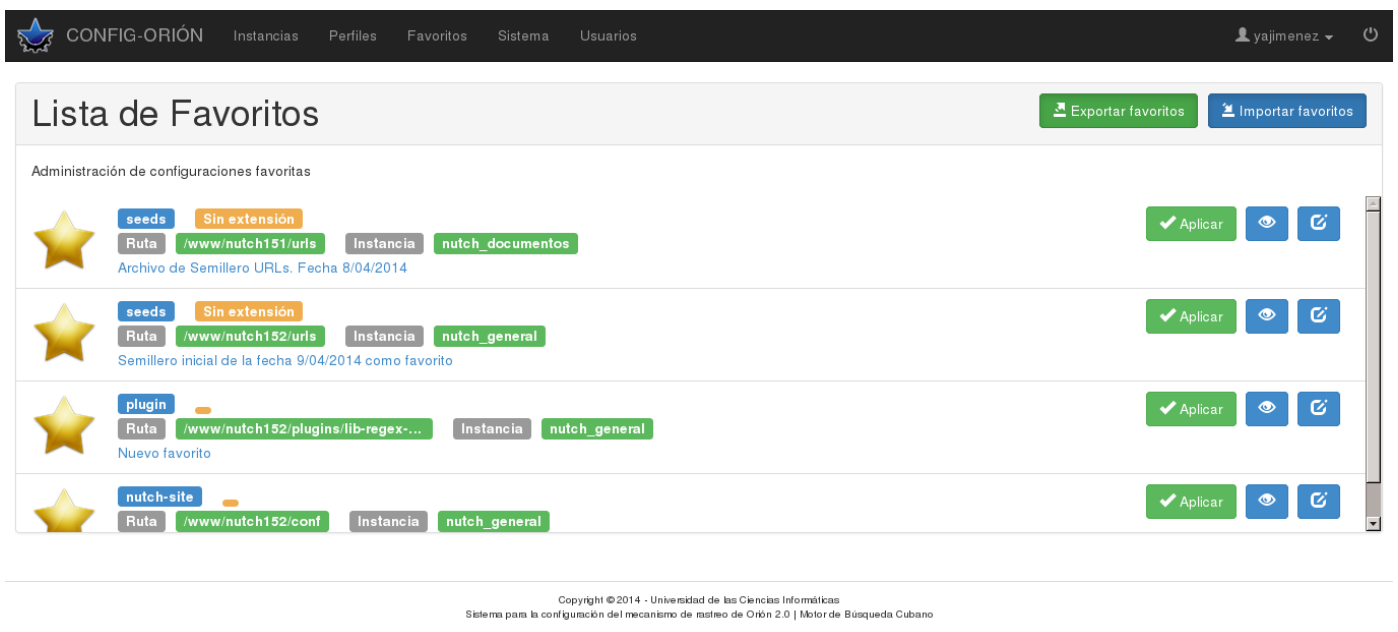


Figura 47: Vista de gestión de archivos de configuración favoritos.

Gráficos estadísticos de modificaciones

Gráfico Spline

Últimas modificaciones

Instancia | Plugin | Archivo

Rango de fecha

[Cantidad]

[Seleccionar]

[Fecha inicio] - [Fecha fin]

Filtrar

Restablecer

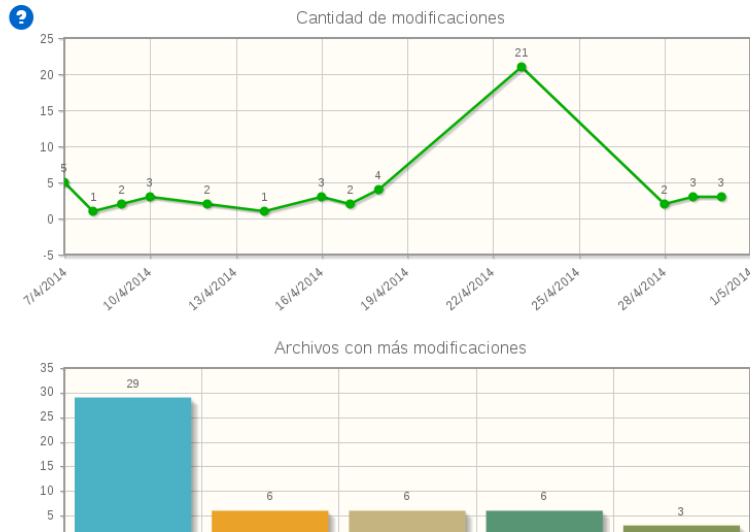


Figura 48: Vista de gestión de gráficos estadísticos respecto a las modificaciones.

Spline de historial de modificaciones

Gráficos generales

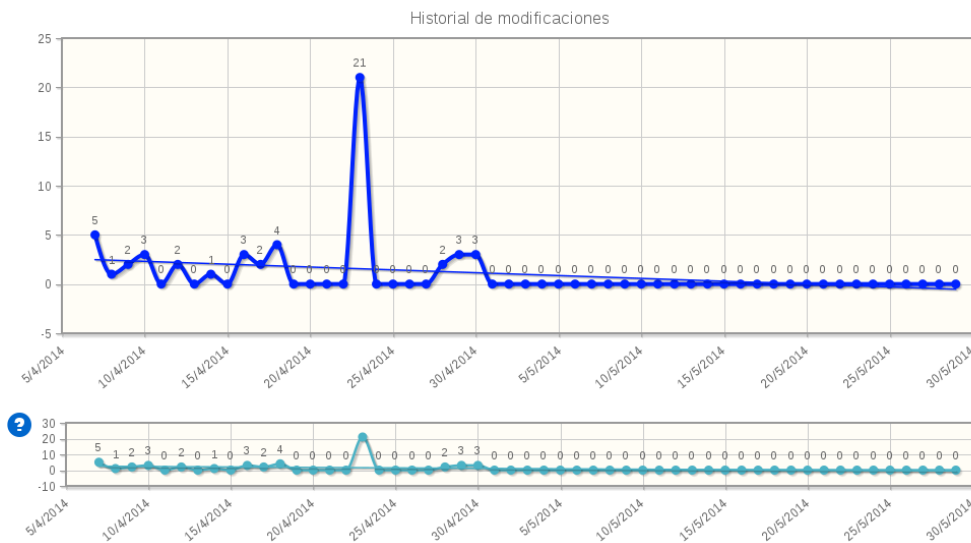


Figura 49: Vista de gráfico Spline de historial de modificaciones.

Anexo # 8: Diseño de casos de prueba basados en casos de uso.

Caso de uso: Gestionar archivos.

Condiciones de ejecución:

- El usuario ha sido autenticado en el sistema.
- El usuario autenticado es un Administrador.
- El usuario ha accedido a la lista de archivos de configuración. **Ver CU Listar archivos**

Tabla 23: Sección "Modificar datos de archivo" del caso de prueba basado en el CU "Gestionar archivos".

Escenario	Descripción	Nombre	Ruta	Formato	Contenido	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Modificar datos de archivo de forma exitosa.	El sistema introduce los datos del archivo de forma correcta.	V	V	V	N/A	V	El sistema modifica los datos del archivo de forma correcta y muestra el mensaje "El archivo se ha modificado satisfactoriamente" .	1 - El usuario se autentica en el sistema con rol de Administrador. 2 - El sistema muestra la vista "Administrar instancias" . 3 - El usuario pulsa en el botón de una Instancia. 4 - El sistema muestra la vista "Administrar archivos" . 5 - El usuario pulsa el botón "Editar" de un ítem de la lista de archivos. 6 - El sistema muestra la vista "Modificar archivo" . 7 - El administrador llena los datos de los campos y pulsa el botón "Guardar" .
EC 1.2 Modificar datos de archivo de forma incorrecta.	El sistema introduce los datos del archivo de forma incorrecta.	I	V	V	NA	V	El sistema no modifica los datos del archivo y muestra el mensaje "Error: El archivo especificado no existe" .	
		V	I	V	NA	V	El sistema no modifica los datos del archivo y muestra el mensaje "Error: La ruta no puede contener dos caracteres '/' seguidos" .	
		V	I	V	NA	V	El sistema no modifica los datos del archivo y muestra el mensaje "Error: La ruta no puede terminar con el carácter '/' " .	
		V	V	I	NA	V	El sistema no modifica los datos del archivo y muestra el mensaje "Error: El archivo no es realmente de tipo 'TXT' o 'Texto plano' " .	
		V	V	I	NA	V	El sistema no modifica los datos del archivo y muestra el mensaje "Error: El archivo no es realmente de tipo 'XML' " .	

EC 1.3 Modificar datos de archivo duplicado	El sistema introduce los datos de un archivo que se encuentra registrado en el sistema.	V	V	V	N/A	V	El sistema no modifica los datos del archivo y muestra el mensaje "Atención! El archivo que intenta registrar ya existe" .
EC 1.4 Modificar datos de archivo con campos en blanco	El sistema introduce los datos del archivo con campos en blanco.	I	I	I	N/A	I	El sistema no modifica los datos del archivo y muestra el mensaje "Por favor, complete este campo" debajo del campo en blanco.

Tabla 24: Sección "Eliminar datos de archivo" del caso de prueba basado en el CU "Gestionar archivos".

Escenario	Descripción	Nombre	Ruta	Formato	Contenido	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar datos de archivo de forma exitosa.	El sistema elimina los datos de archivo de forma correcta.	N/A	N/A	N/A	N/A	N/A	El sistema elimina los datos del archivo de forma correcta y muestra el mensaje "El archivo se ha eliminado satisfactoriamente" .	1 - El usuario se autentica en el sistema con rol de Administrador. 2 - El sistema muestra la vista "Administrar instancias" . 3 - El usuario pulsa en el botón de una Instancia. 4 - El sistema muestra la vista "Administrar archivos" . 5 - El usuario pulsa el botón "Editar" o "Ver detalles" de un ítem de la lista de archivos. 6 - El sistema muestra la vista "Modificar archivo" o "Ver detalles de archivo" según corresponda. 7 - El administrador pulsa el botón "Eliminar" . 8 - El administrador pulsa el botón "Si" para confirmar la eliminación.
EC 1.2 Cancelar la acción eliminar datos de archivo.	El sistema cancela la acción eliminar los datos de archivo.	NA	NA	NA	NA	NA	El sistema no elimina los datos del archivo y permanece en la vista actual.	

Caso de uso: Gestionar instancias.

Condiciones de ejecución:

- El usuario ha sido autenticado en el sistema.
- El usuario autenticado es un Administrador.
- El usuario ha accedido a la lista de instancias de Nutch. **Ver CU Listar instancias**

Tabla 25: Sección “Registrar datos de instancia” del caso de prueba basado en el CU “Gestionar instancias”.

Escenario	Descripción	Nombre	Ruta	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar datos de instancia de forma exitosa.	El sistema introduce los datos de la instancia de forma correcta.	V	V	V	El sistema registra los datos de la instancia de forma correcta y muestra el mensaje "La instancia se ha creado correctamente" .	1 - El usuario se autentica en el sistema con rol de Administrador. 2 - El sistema muestra la vista "Administrar instancias" . 3 - El usuario pulsa el botón "Registrar Instancia" . 4 - El sistema muestra la vista "Registrar Instancia" . 5 - El administrador llena los datos de los campos y pulsa el botón "Guardar" .
EC 1.2 Registrar datos de instancia de forma incorrecta.	El sistema introduce los datos de la instancia de forma incorrecta.	V	I	V	El sistema no registra los datos de la instancia y muestra el mensaje "Error: La ruta no puede contener dos caracteres '/' seguidos" .	
		V	I	V	El sistema no registra los datos de la instancia y muestra el mensaje "Error: La ruta no puede terminar con el símbolo '/' " .	
EC 1.3 Registrar datos de instancia duplicada	El sistema introduce los datos de una instancia que se encuentra registrada en el sistema.	V	V	V	El sistema no registra los datos de la instancia y muestra el mensaje "Atención! La instancia que intenta registrar ya existe" .	
EC 1.4 Registrar datos de instancia con campos en blanco	El sistema introduce los datos de la instancia con campos en blanco.	I	I	I	El sistema no registra los datos de la instancia y muestra el mensaje "Por favor, complete este campo" debajo del campo en blanco.	

Tabla 26: Sección “Modificar datos de instancia” del caso de prueba basado en el CU “Gestionar instancias”.

Escenario	Descripción	Nombre	Ruta	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Modificar datos de instancia de forma exitosa.	El sistema introduce los datos de la instancia de forma correcta.	V	V	V	El sistema modifica los datos de la instancia de forma correcta y muestra el mensaje "La instancia se ha editado correctamente" .	1 - El usuario se autentica en el sistema con rol de Administrador. 2 - El sistema muestra la vista "Administrar instancias" . 3 - El usuario pulsa el botón "Editar" de un ítem de la lista de instancias. 4 - El sistema muestra la vista "Editar Instancia" . 5 - El administrador llena los datos de los campos y pulsa el botón "Guardar" .
EC 1.2 Modificar datos de instancia de forma incorrecta.	El sistema introduce los datos de la instancia de forma incorrecta.	V	I	V	El sistema no modifica los datos de la instancia y muestra el mensaje "Error: La ruta no puede contener dos caracteres '/' seguidos" .	
		V	I	V	El sistema no modifica los datos de la instancia archivo y muestra el mensaje "Error: La ruta no puede terminar con el símbolo '/' " .	
EC 1.3 Modificar datos de	El sistema introduce los datos de una instancia que	V	V	V	El sistema no modifica los datos de la instancia y muestra el mensaje "Atención! La instancia que intenta registrar ya existe" .	

instancia duplicada	se encuentra registrada en el sistema.				
EC 1.4 Modificar datos de instancia con campos en blanco	El sistema introduce los datos de la instancia con campos en blanco.				El sistema no modifica los datos de la instancia y muestra el mensaje "Por favor, complete este campo" debajo del campo en blanco.

Tabla 27: Sección "Eliminar datos de instancia" del caso de prueba basado en el CU "Gestionar instancias".

Escenario	Descripción	Nombre	Ruta	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar datos de instancia de forma exitosa.	El sistema elimina los datos de instancia de forma correcta.	N/A	N/A	N/A	El sistema elimina los datos de la instancia de forma correcta y muestra el mensaje "La instancia se ha eliminado correctamente" .	1 - El usuario se autentica en el sistema con rol de Administrador. 2 - El sistema muestra la vista "Administrar instancias" . 3 - El usuario pulsa el botón "Editar" o "Ver detalles" de un ítem de la lista de instancias. 4 - El sistema muestra la vista "Modificar instancia" o "Ver detalles de instancia" según corresponda. 5 - El administrador pulsa el botón "Eliminar" . 6 - El administrador pulsa el botón "Si" para confirmar la eliminación.
EC 1.2 Cancelar la acción eliminar datos de instancia.	El sistema cancela la acción eliminar los datos de instancia.	NA	NA	NA	El sistema no elimina los datos de la instancia y permanece en la vista actual.	

Tabla 28: Variables empleadas en el diseño del caso de prueba basado en el CU "Gestionar instancias".

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Debe ser un nombre sugerente con el tipo de instancia que se está registrando. (Obligatorio)
2	Ruta	Campo de texto	No	Debe ser una ruta de instancia existente. La instancia debe estar dentro del directorio especificado en las propiedades de sistema, para el despliegue de instancias. (Obligatorio)
3	Descripción	Área de texto	Si	

