

Universidad de las Ciencias Informáticas

Facultad 1



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

Capa de servicios web para el motor de búsqueda Orión

Autor(s):

Manuel Antonio Novoa Proenza.

Yasmani Joaquin Alvarez Gómez

Tutor:

Ing. Eduardo Macías Sotolongo

La Habana, Junio de 2012

“Año 54 de la Revolución”

Declaración de autoría

Declaramos que somos los únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de este, con carácter exclusivo.

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

Manuel Antonio Novoa Proenza

Yasmani Joaquin Alvarez Gómez

Firma del autor

Firma del autor

Ing. Eduardo Manuel Macías Sotolongo

Firma del tutor

Agradecimientos

A nuestras familias.

A nuestros amigos y amigas.

A todos nuestros profesores docentes.

A nuestros profesores del proyecto: Mady, Paul, Eyeris y Jorgito.

A nuestro profesor consejero: Yusniel Hidalgo.

A nuestro tutor, Eduardo Manuel Macías Sotolongo. Muchas gracias por toda la dedicación brindada a lo largo del desarrollo de este trabajo de diploma.

A todo el tribunal de los talleres de tesis: Daileny, Liudmila, Aylín y Dovier.

A todos los que nos ayudaron y nos extendieron su mano amiga: Las profesoras Delly, Damaris y Graciela, y los profesores Yonny y Serguey.

Muchas gracias a Juan Carlos Lobaina por ser un buen gurú. -“Juanca, Java es lo mejor”.

A todos nuestros amigos de la Dirección de Televisión, por la constante preocupación sobre nuestro desempeño en el presente trabajo.

Gracias, mil gracias a Rislaidy Pérez y Yaisi Roque-escobar, por la gran ayuda que nos dieron.

A nuestros padres, por apoyarnos en cada momento.

Los Autores.

Dedicatoria

A todos los que un 9 de septiembre de 2007 entraron conmigo al aula 308 del docente 3.

A mi pareja, por soportarme hablar casi todo el tiempo de la tesis.

A mi prima María Haydee por demostrarse que “Sí se puede”.

A mi primo Luis Manuel y a mi hermana Nancy, por ser ejemplos a seguir para mí.

A mis abuelos Manuel y Tatá, yo sé que estén donde estén, estarán orgullosos de mí.

A mis abuelas Teresa y Haydee.

A mi madre querida por traerme a la vida. Viejíta gracias por estar ahí siempre y cuidar tu salud por encima de todo. Gracias por hacerme feliz.

A mi padre querido. Viejo, siempre agradeceré cada uno de tus consejos. Gracias por ayudarme a ser como soy. Gracias también por hacerme feliz.

Manuel.

A toda mi familia por apoyarme en mis metas.

A mi Mamá por ser la persona más importante del mundo para mí.

A mi abuela Pucha: Abuela gracias por tu amor y guiarme en mi vida.

A mi abuela Mercedes y mi tía Lazara, por quererme tanto y ser mis consejeras.

A mi padre Joaquín.

A mi padrastro Emigdio: Gordó gracias por tu apoyo en todas mis metas y tu cariño.

A mi abuelo Amado, mi tío Jorge, Machí y mi hermano Yunier por apoyarme en todo.

A Jorgito y Roxana; Maribel, Angela, todos mis primos de Ciego y de la Habana.

A mis amigos de Ciego, Helmer, Rosana, Asmaní y Juan Manuel.

A toda mi gente del Barrio: Lurdes, Orlanda, Marizol, Esther y todos los demás.

A mis hermanos de la UCI: Jorge, Ruben, Ernesto, Carlos Cesar, Ricardo y Novoa.

Yasmaní.

Resumen

Desde el año 2011, en la Universidad de las Ciencias Informáticas, se desarrolla el motor de búsqueda para la Web cubana. Este buscador posibilitará una mejor visibilidad de los contenidos de los sitios cubanos a los usuarios que busquen información sobre Cuba. Con la indexación de los contenidos existentes en las páginas web, el motor de búsqueda, almacenará en su índice un conjunto de datos útiles, que pueden permitirle a los webmaster mejorar el posicionamiento de sus sitios. El propósito del presente trabajo consiste en la implementación de una capa de servicios web que permita establecer la comunicación de aplicaciones informáticas con el motor de búsqueda. Esta comunicación posibilitará a cualquier sistema poder utilizar dentro de sus componentes, los servicios web brindados por el buscador. Dentro de los servicios disponibles se encuentran: los de búsqueda, los de estadísticas y los de configuración. En el marco de esta investigación, se sistematizó en el estudio teórico de los servicios web, su alcance dentro de los buscadores y se realizó la selección de los servicios que podría brindar el buscador Orión. Una vez realizada la investigación se desarrolló el análisis y diseño de las características del sistema, y con una base tecnológica bien definida se comenzó con la implementación de la aplicación. Todo el proceso finalizó con una etapa de pruebas para validar el correcto funcionamiento de la capa de servicios web. De forma general el sistema establece la interoperabilidad de las aplicaciones cubanas con el motor de búsqueda Orión.

Palabras Clave: servicios web, buscador, motor de búsqueda, Orión, Rest, API.

Índice

Declaración de autoría.....	I
Agradecimientos	II
Dedicatoria.....	III
Resumen	IV
Índice	5
Índice de Ilustraciones	7
Índice de Tablas	8
Introducción	10
Capítulo 1 Fundamentación Teórica.....	14
1.1 - Introducción.....	14
1.2 - La Web.....	14
1.3 - Los servicios web.....	15
1.3.1 - Servicio de Transporte	17
1.3.2 - Servicio de Mensajería	17
1.3.3 - Descripción del Servicio.....	21
1.3.4 - Descubrimiento de Servicios.....	22
1.4 - Arquitectura Orientada a Servicios	23
1.5 - Servicios web en la Universidad de las Ciencias Informáticas.....	24
1.6 - Servicios web en motores de búsqueda en el mundo.....	24
1.6.1 - Yahoo!	25
1.6.2 - Google.....	26
1.7 - Motores de Búsqueda en Cuba.....	28
1.8 - Metodologías, herramientas y tecnologías	29
1.8.1 - Frameworks para servicios web.....	29
1.8.2 - Lenguajes de Programación	30
1.8.3 - Metodologías de desarrollo.....	32
1.8.4 - IDEs de Desarrollo.....	34
1.8.5 - Sistemas de Gestión de Base de Datos.....	35
1.8.6 - Servidores Web	37

1.8.7 - Herramientas CASE.....	39
1.9– Conclusiones parciales.....	40
Capítulo 2: Características del Sistema	41
2.1 - Introducción.....	41
2.2 - Propuesta del sistema	41
2.3 – Modelo de dominio	44
2.4 – Relación de los requerimientos del sistema.....	45
2.4.1 – Requerimientos funcionales	46
2.4.2 – Requerimientos no funcionales	47
2.5 - Modelo de Casos de Uso del Sistema.....	48
2.5.1 - Definición de los actores del sistema	48
2.5.2 – Definición de los Casos de Uso del sistema.....	48
2.5.3 – Descripción de los Casos de Uso del sistema.....	51
2.6 – Conclusiones parciales.....	53
Capítulo 3: Análisis y Diseño del Sistema	54
3.1 – Introducción.....	54
3.2 Modelo de Diseño.....	54
3.3 Arquitectura del sistema	54
3.3.1 Patrones de diseño empleados durante el desarrollo del sistema.....	55
3.4 Diagrama de clases del diseño.....	56
3.4.1 Diagramas de interacción del Diseño.....	58
3.5 Diseño de la Base de Datos	60
3.6 Tratamiento de Errores	62
3.7 Aspectos de Seguridad en el sistema.....	62
3.8 Parámetros de consulta y acceso.....	63
3.9 Modelo de Despliegue.....	64
3.10 Conclusiones parciales.....	64
Capítulo 4: Implementación y prueba del sistema	65
4.1 – Introducción.....	65
4.2 – Modelo de implementación del sistema	65

4.2.1 –Diagrama de componentes	65
4.2.2 – Componentes Complementarios	66
4.2.3 – Estándares de código	67
4.3 – Modelo de prueba.....	67
4.3.1 –Plan de prueba.....	67
4.3.2 –Pruebas	69
4.4 –Resultado de las pruebas	71
4.5 – Conclusiones parciales.....	72
Conclusiones	73
Recomendaciones	74
Referencias Bibliográficas.....	75
Bibliografía.....	79
Anexos.....	80
Glosario de términos.....	92
Índice de Ilustraciones	
ILUSTRACIÓN 1 FUNCIONAMIENTO SOAP.	19
ILUSTRACIÓN 2 USO DE PROTOCOLOS DE SERVICIOS WEB.	19
ILUSTRACIÓN 3 BASE TECNOLÓGICA PROPUESTA.	40
ILUSTRACIÓN 4 PROPUESTA DEL SISTEMA	42
ILUSTRACIÓN 5 MODELO DE DOMINIO	44
ILUSTRACIÓN 6 DIAGRAMA DE CU DEL SISTEMA.	49
ILUSTRACIÓN 7 DIAGRAMA DE CLASES DEL DISEÑO CU AUTENTICAR USUARIO.....	57
ILUSTRACIÓN 8 DIAGRAMA DE CLASES DEL DISEÑO CU REALIZAR BÚSQUEDAS.....	57
ILUSTRACIÓN 9 DIAGRAMA DE SECUENCIA CU REALIZAR BÚSQUEDAS	59
ILUSTRACIÓN 10 DIAGRAMA DE SECUENCIA CU AUTENTICAR USUARIO.....	60
ILUSTRACIÓN 11 MODELO FÍSICO DE DATOS PARA MONGODB	61
ILUSTRACIÓN 12 DIAGRAMA DE DESPLIEGUE	64
ILUSTRACIÓN 13 DIAGRAMA DE COMPONENTES DE LA CAPA DE SERVICIOS WEB.	66
ILUSTRACIÓN 14 RESULTADOS DE PRUEBAS CON JMETER	72
ILUSTRACIÓN 15 CICLO DE VIDA DE RUP FUENTE: IBM	81

ILUSTRACIÓN 16 CLASES DE LA BIBLIOTECA MONGO PARA JAVA	81
ILUSTRACIÓN 17 CLASES DE LA BIBLIOTECA SORLJ	82
ILUSTRACIÓN 18 EJEMPLO DE USO DEL SERVICIO WEB ROBOTS.TXT	83
ILUSTRACIÓN 19 APLICACIÓN DE PRUEBA DE LOS SERVICIOS WEB.	84
Índice de Tablas	
TABLA 1 COMPARACIÓN JSON – XML.....	16
TABLA 2 PARÁMETROS DE SOLICITUD ANALIZADOR DE CONTENIDOS DE YAHOO!.	26
TABLA 3 PARÁMETROS DE SOLICITUD EXTRACTOR DE TÉRMINOS DE YAHOO!.	26
TABLA 4 PARÁMETROS FUNDAMENTALES SERVICIO WEB DE BÚSQUEDA DE GOOGLE.	28
TABLA 5 DEFINICIÓN DEL ACTOR DEL SISTEMA	48
TABLA 6 DEFINICIÓN CU AUTENTICAR USUARIO.....	49
TABLA 7 DEFINICIÓN CU CAMBIAR PROPIEDADES CONEXIÓN.....	49
TABLA 8 DEFINICIÓN CU OBTENER RESULTADO BÚSQUEDA.....	50
TABLA 9 DEFINICIÓN CU OBTENER CARACTERÍSTICAS PORTAL.....	50
TABLA 10 DEFINICIÓN CU GENERAR SITEMAP.XML.....	50
TABLA 11 DEFINICIÓN CU GENERAR ROBOTS.TXT.....	50
TABLA 12 DEFINICIÓN CU OBTENER TRÁFICO PORTAL	50
TABLA 13 DEFINICIÓN CU REGISTRAR TRÁFICO DESDE EL BUSCADOR	51
TABLA 14 DEFINICIÓN CU REALIZAR BÚSQUEDA VÍA E-MAIL.....	51
TABLA 15 DESCRIPCIÓN CU AUTENTICAR USUARIO	52
TABLA 16 DESCRIPCIÓN CU OBTENER RESULTADO DE BÚSQUEDA	53
TABLA 17 RESPUESTAS DEL SISTEMA ANTE ERRORES	62
TABLA 18 PARÁMETROS DE CONSULTA PARA LA BÚSQUEDA.	63
TABLA 19 CASO DE PRUEBA CU OBTENER RESULTADO BÚSQUEDA	70
TABLA 20 DESCRIPCIÓN DE VARIABLES DE PRUEBA CU OBTENER RESULTADO BÚSQUEDA	70
TABLA 21 MATRIZ DE DATOS CU OBTENER RESULTADO BÚSQUEDA.....	71
TABLA 22 CAMPOS DE RESPUESTA ANALIZADOR DE CONTENIDOS DE YAHOO! FUENTE: YAHOO!.....	80
TABLA 23 CAMPOS DE RESPUESTA EXTRACTOR DE TÉRMINOS DE YAHOO! FUENTE: YAHOO!.....	80
TABLA 24 CAMPOS DE RESPUESTA SERVICIO WEB DE BÚSQUEDA DE GOOGLE FUENTE: GOOGLE	80
TABLA 25 RELACIÓN CLASES DEL SISTEMA- BIBLIOTECAS-FRAMEWORKS.....	83

TABLA 26 RUTA DE SERVICIOS WEB	84
TABLA 27 DESCRIPCIÓN CU CAMBIAR PROPIEDADES CONEXIÓN	85
TABLA 28 DESCRIPCIÓN CU OBTENER CARACTERÍSTICAS PORTAL	86
TABLA 29 DESCRIPCIÓN CU GENERAR SITEMAP.XML	87
TABLA 30 DESCRIPCIÓN CU GENERAR ROBOT.TXT	88
TABLA 31 DESCRIPCIÓN CU OBTENER TRÁFICO PORTAL.....	89
TABLA 32 DESCRIPCIÓN CU REGISTRAR TRÁFICO DESDE EL BUSCADOR.....	90
TABLA 33 DESCRIPCIÓN CU REALIZAR BÚSQUEDA VÍA CORREO ELECTRÓNICO.....	91

Introducción

La creación de Internet supuso una revolución sin precedentes en el mundo de la informática y de las comunicaciones. El impacto causado por las tecnologías de la información en las últimas dos décadas del siglo pasado, revolucionó la manera de hacer negocios, la comunicación entre las personas y la interacción entre las empresas.

El surgimiento de la web y la proliferación de aplicaciones para esta, sustentaron el nacimiento de tecnologías que dieron origen a nuevas tendencias en el desarrollo de las comunicaciones [1].

La necesidad de lograr la interoperabilidad¹ y permitir compartir información entre distintas plataformas de software y hardware, trajo consigo la utilización de un lenguaje común de intercambio de información que aprovechara los estándares existentes en el mercado y bajo este contexto nacen los servicios web basados en XML [2] [3].

Miles de usuarios se conectaban a Internet para satisfacer necesidades de información sobre variados temas. Con el crecimiento de la WWW², localizar los contenidos expuestos en ella se convirtió en un proceso complicado. Para facilitar las búsquedas surgieron los buscadores web que permiten localizar la información mediante la indexación de los sitios, mostrándoles a los usuarios los resultados de la indagación de una forma rápida y efectiva [1].

Los servicios de los buscadores evolucionaron y pasaron a ser algo más que una herramienta de localización de información en la web. Estos sistemas comenzaron a brindar servicios web que permitieron a los desarrolladores, crear las más disímiles aplicaciones haciendo uso de los contenidos disponibles en ellos.

Actualmente la web cubana está en constante crecimiento. Existen hoy 2 293 subdominios registrados bajo el dominio .cu. Por ello la inclusión de los sitios cubanos en los sistemas de búsquedas desempeña un papel fundamental en la divulgación de la realidad de Cuba en el mundo [4].

¹ Interoperabilidad: habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.[47]

² World Wide Web

La indexación de los medios informativos cubanos en buscadores de Internet como Yahoo! y Google, mejora la localización de la información existente en la web cubana, para los usuarios que la solicitan. La inclusión de las páginas cubanas en los motores de búsqueda, ha posibilitado un aumento en la cantidad de visitas que estas reciben. Sin embargo, las aplicaciones cubanas se ven imposibilitada de acceder a determinados servicios que brindan estos sistemas de recuperación de información, debido a restricciones impuestas por el bloqueo norteamericano hacia Cuba.

En la actualidad los desarrolladores de aplicaciones para la web en cubana necesitan implementar funcionalidades de búsqueda para sus sistemas. Estas funcionalidades podrían de forma sencilla ser utilizadas desde algunos de los servicios web brindados por los buscadores, acortando el tiempo de creación de las aplicaciones web y evitando el uso innecesario de recursos humanos y tecnológicos.

La optimización de los sitios cubanos para lograr buenos lugares en de los resultados ofrecidos por los motores de búsquedas, implica la realización de estudios webmétricos para detectar las principales causas que afectan el posicionamiento de estos. Realizar ese proceso constituye una problemática, pues dichos estudios implican un gasto adicional, ya sea en recursos humanos, de software o hardware.

Como alternativa cubana para la búsqueda en la web surge el buscador Orión. El despliegue de dicho sistema tiene como objetivo, localizar la información publicada en la web nacional.

El buscador Orión contiene además, información valiosa para los webmaster acerca del comportamiento de sus sitios. Estos datos sirven de guía para trazar nuevas estrategias que permitan mejorar el desarrollo de las aplicaciones web cubanas.

Sin embargo, estos recursos no están disponibles, al no existir el mecanismo para que las aplicaciones informáticas interactúen con el motor de búsqueda Orión. Hoy los datos de búsqueda, webmetría y posicionamiento que procesa el buscador son completamente inaccesibles para cualquier sistema automatizado que intente acceder a ellos.

Por lo antes planteado se identifica el siguiente **problema a resolver**: ¿Cómo lograr la interoperabilidad de las aplicaciones informáticas con el motor de búsqueda Orión?

Lo antes expuesto lleva a analizar la comunicación entre sistemas informáticos planteando como **objeto de estudio** la interoperabilidad entre aplicaciones informáticas, enfocándose en el **campo de acción** del proceso de desarrollo de servicios web en motores de búsqueda.

Para darle solución al problema se plantea como **objetivo general** de este trabajo de diploma: Desarrollar una capa de servicios web haciendo uso de las tecnologías existentes, para lograr la interoperabilidad de aplicaciones informáticas con el motor de búsqueda Orión.

Desglosando de este los siguientes **objetivos específicos**:

1. Analizar el marco teórico conceptual respecto a las tecnologías actuales para el desarrollo de servicios web.
2. Diseñar una capa de servicios web para el motor de búsqueda Orión.
3. Implementar una capa de servicios web para el motor de búsqueda Orión.
4. Validar el correcto funcionamiento de los servicios web implementados.

Para el cumplimiento de estos objetivos se han trazado las siguientes **tareas de la investigación**:

1. Levantamiento bibliográfico y webgráfico para estudiar los principales elementos teóricos y conceptos que permitan analizar el estado actual del desarrollo de tecnologías para el desarrollo de servicios web y elaborar un marco teórico-conceptual.
2. Análisis de los servicios web que ofrecen los motores de búsqueda en Internet.
3. Identificación de los servicios web que debe ofrecer el motor de búsqueda Orión.
4. Definición de la metodología a utilizar en el desarrollo de una capa de servicios web.
5. Identificación de necesidades de información, requisitos funcionales y no funcionales.
6. Documentación de los artefactos generados durante los procesos de análisis, diseño e implementación de los servicios web según la metodología seleccionada.
7. Implementación de los servicios web que debe ofrecer el motor de búsqueda Orión.
8. Desarrollo de una aplicación cliente que consuma los servicios web desarrollados.

Durante el proceso de desarrollo de la capa de servicios web se realizarán investigaciones para profundizar en el objeto de estudio definido, dichas investigaciones serán guiadas por métodos teóricos de investigación. El Analítico y Sintético, que permitirá investigar acerca de los aspectos relacionados con los servicios web y su implementación, mediante el análisis de la documentación existente. Además, este método facilitará la extracción de los elementos más importantes relacionados con el campo de acción. El Histórico Lógico, que permitirá constatar teóricamente cómo ha evolucionado la utilización de servicios web hasta la actualidad. La Modelación, mediante el lenguaje de modelado UML se utilizará para reflejar la estructura, relaciones internas y características de la solución a través de diagramas.

El presente trabajo consta de cuatro capítulos, estructurados de la siguiente forma:

En el Capítulo 1 se abordarán los conceptos fundamentales que permitirán entender el mundo de los servicios web. Así como la fundamentación de las herramientas y tecnologías que se emplearán en el desarrollo del trabajo de diploma.

En el Capítulo 2 serán abordados los temas relacionados con el dominio y la caracterización del sistema a desarrollar, incluyendo los requerimientos planteados por el cliente y formulándose una propuesta de la solución a implementar.

El Capítulo 3 tendrá como núcleo fundamental el diseño de la arquitectura con de la capa de servicios web. Además se exponen las ideas relacionadas con la validación de los datos, el tratamiento de errores y la seguridad.

En el Capítulo 4 se abordará la implementación y prueba de la solución propuesta y se generarán los artefactos correspondientes a la fase de desarrollo según la metodología seleccionada.

Capítulo 1 Fundamentación Teórica

1.1 - Introducción

La sistematización del estudio de los principales conceptos sobre los servicios web, permite entender cómo han evolucionado estas tecnologías y conocer cuáles son las últimas tendencias en este campo. Un estudio del desarrollo de los servicios web en el mundo, Cuba y en la Universidad de las Ciencias Informáticas sirve de referencia para apreciar hasta qué punto han evolucionado. Para un buen entendimiento de los servicios web, es necesario dominar las herramientas y tecnologías en las que se basan, así como hacer un análisis crítico y comparativo sobre las tecnologías estudiadas. Este estudio indicará cuáles de estas son las más recomendables para la implementación de la capa de servicios web, objetivo del presente trabajo de diploma.

1.2 - La Web

Para un correcto entendimiento de los servicios web es importante conocer el medio donde estos se despliegan: la Web. El desarrollo de esta no se debe a una sola persona, pero si se busca un padre para ella, sería Tim Bernes-Lee, pues a él se le deben los tres elementos fundamentales que fueron claves en su nacimiento.

HTML³ para crear los contenidos de la Web.

HTTP⁴ como protocolo de comunicación entre los ordenadores de la Web, encargado de la transferencia de las páginas web y demás recursos.

URL⁵ como medio de localización (direccionamiento) de los distintos recursos de la Web.

Con el paso del tiempo y unido al incremento exponencial del uso de la web, otros elementos se fueron sumando y estos a su vez fueron mutando la forma en la que desarrolladores y usuarios finales compartían sus datos en la red.

³ HyperText Markup Language (en español lenguaje de marcado de hipertexto)

⁴ Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto)

⁵ Uniform resource locator (en español localizador de recursos uniforme)

Los servicios web llegaron para quedarse y constituyen hoy día una de las vías más usadas para la interoperabilidad entre aplicaciones informáticas. Por ende, el estudio de los servicios web constituye uno de los objetivos primordiales en el presente trabajo de diploma.

1.3 - Los servicios web

Los servicios web cobran relevancia al estandarizar la comunicación entre distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma.

La necesidad de que las aplicaciones se comunicaran con otros sistemas dio surgimiento a los servicios web. El consorcio W3C⁶ define los servicios web como: *“sistemas software diseñados para soportar una interacción interoperable máquina a máquina sobre una red”*. Por otro lado, investigaciones como la realizada por el MSc. Carlos Andrés Morales Machuca de la Universidad de Colombia, plantea que los servicios web son: *“sistemas de software que permiten el intercambio de datos y funcionalidades entre aplicaciones sobre una red, soportando diferentes estándares que garantizan la interoperabilidad de estos sistemas”* [5].

Existen en el mundo organizaciones responsables de la arquitectura y reglamentación de los servicios web, ellas son OASIS⁷ y W3C [2].

La principal razón para usar servicios web es que se pueden utilizar con HTTP sobre TCP⁸ en el puerto 80. Dado que las organizaciones protegen sus redes mediante firewalls que filtran y bloquean gran parte del tráfico de Internet, cierran la mayoría de los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios web utilizan este puerto para la conexión ya que no resultan bloqueados. Es importante señalar que los servicios web se pueden utilizar sobre cualquier protocolo, sin embargo, TCP es el de preferencia [2].

Para la comunicación entre los servicios web es necesario un lenguaje que permita el intercambio de la información. Entre estos lenguajes se pueden mencionar XML y JSON como los utilizados preferentemente por los desarrolladores de servicios web.

⁶ World Wide Web Consortium

⁷ Organization for the Advancement of Structured Information Standards

⁸ *Transmission Control Protocol*

El lenguaje extensible de marcado (XML) fue desarrollado por la W3C para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que las aplicaciones puedan buscar y compartir información en la red de una forma segura y fácil [9]. XML tiene otras aplicaciones entre las que destaca su uso como estándar para el intercambio de datos entre diversas aplicaciones o software con lenguajes privados.

Otro formato o notación utilizada para el intercambio de datos entre aplicaciones es JSON⁹. Dada su simplicidad se ha generalizado su uso y constituye una alternativa al XML en AJAX, aunque aún existe preferencia al uso del XML. Esta notación ofrece varias herramientas de desarrollo tanto en el lado del cliente como del servidor [49].

En la siguiente tabla se muestra un resumen de las principales características de ambas alternativas para el intercambio de datos.

JSON	XML
Estructura de Datos	Estructura de Datos
No presenta sistema de validación	Para validar se usa: XSD
El análisis se realiza de forma directa en la evaluación.	El análisis requiere de bibliotecas Ej.: XPath.
En JavaScript se puede trabajar con objetos analizando el tipo en tiempo de ejecución.	En JavaScript se puede trabajar con cadenas, pero puede requerir análisis adicionales.

Tabla 1 Comparación JSON – XML

Por todo lo anteriormente planteado se decide dar soporte de comunicación utilizando ambos lenguajes para el intercambio de datos en la solución propuesta. Se recomienda el uso de JSON, dada su ligereza, en función de optimizar el uso del canal de comunicación.

Para el desarrollo de los servicios web es importante conocer la estructura de los mismos. Al conjunto de servicios y protocolos que usan los servicios web para su funcionamiento se les conoce como "*Web Services Protocol Stack*" (pila de protocolos de servicios web) y son utilizados para definir, localizar, implementar y hacer que un servicio web interactúe con otro. Este conjunto está conformado por cuatro subconjuntos entre los que se encuentran:

- Servicio de transporte;

⁹ JavaScript Object Notation

- Servicio de mensajería;
- Descripción del servicio;
- Descubrimiento de servicios;

1.3.1 - Servicio de Transporte

Este servicio se encarga del transporte de mensajes entre aplicaciones sobre una red. Para su funcionamiento este incluye varios protocolos del nivel de Aplicación del modelo OSI¹⁰, los más usados son:

•**HTTP (HyperText Transfer Protocol)**: Protocolo del nivel de Aplicación más utilizado en Internet.

Define la sintaxis y la semántica utilizada para la arquitectura web. En el contexto de los servicios web es utilizado para la transferencia de las transacciones XML a través de la red utilizando los mismos principios del HTML [6].

•**FTP (File Transfer Protocol)**: Es un protocolo de la capa de aplicación encargado de los servicios de transmisión de archivos a través de redes soportadas sobre TCP [5].

•**SMTP (Simple Mail Transfer Protocol)**: Es un estándar de la capa de aplicación ampliamente utilizado para el envío de mensajes de correo electrónico a través de Internet. Es un estándar basado en texto, que requiere como cliente software de tipo POP3 o IMAP [7].

1.3.2 - Servicio de Mensajería

Es el servicio encargado de la codificación de los mensajes a un lenguaje estándar y pueda así ser interpretado en cualquiera de los nodos de la red. Los componentes utilizados en este conjunto son los siguientes:

•**RPC (Remote Procedure Calls)**: Es una tecnología de software que permite ejecutar una rutina en un equipo o segmento de red de manera remota. Es un paradigma popular para la implementación de sistemas distribuidos bajo arquitecturas cliente servidor [5].

¹⁰ Open System Interconnection. (Sistema de Intercomunicación Abierta)

• **XML-RPC:** Es un protocolo de llamada remota que utiliza XML como lenguaje de codificación y HTTP como mecanismo de transporte. Es un protocolo sencillo ya que solo define algunos tipos de datos y comandos [5].

• **REST (Representational State Transfer):** estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la *World Wide Web* [8]. Existen otras definiciones válidas de REST como la que plantea que es un conjunto de principios para el diseño de redes, utilizado comúnmente para definir una interfaz de transmisión sobre HTTP.

REST para su funcionamiento, utiliza un conjunto de estándares tales como HTML, URL, XML, GIF, JPG y tipos MIME [9].

Los principios de REST son:

- Escalabilidad de la interoperabilidad con componentes;
- Generalidad de interfaces;
- Puesta en funcionamiento independiente;
- Compatibilidad con componentes intermedios;

• **SOAP (Simple Object Access Protocol):** Es un protocolo de la capa de aplicación para el intercambio de mensajes basados en XML sobre redes de computadoras. Básicamente es una vía de transmisión entre un SOAP transmisor y un SOAP receptor, pero los mensajes SOAP deben interactuar con un conjunto de aplicaciones para que se pueda generar un "diálogo" a través de mensajes SOAP. Un mensaje SOAP es la unidad fundamental de una comunicación entre nodos SOAP [10].

SOAP es básicamente un paradigma de una sola vía pero con la ayuda de las aplicaciones se puede llegar a crear patrones complejos. SOAP básicamente está constituido por:

- Un marco que describe el contenido del mensaje e instrucciones de proceso;
- Un conjunto de reglas para representar los tipos de datos definidos;
- Convenciones para representar llamadas a procedimientos remotos y respuestas;

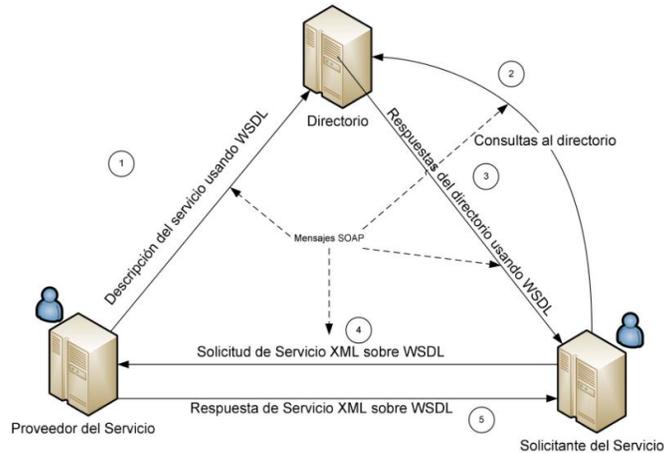


Ilustración 1 Funcionamiento SOAP.

REST o SOAP

Para la implementación de servicios web, la selección del estándar de comunicación a utilizar, resulta un factor importante en el éxito del despliegue de dichos servicios. Factores como la seguridad, la fiabilidad, el modelo extensible así como la descripción de estos, entre otros, tienen que ser tomados en cuenta poniendo en contraste las ventajas y desventajas de cada uno, para elegir entonces el más adecuado al entorno de implantación [48].

La siguiente ilustración tomada del directorio de servicios web (<http://programmableweb.com>) muestra el nivel de popularidad de los principales protocolos de mensajería para el desarrollo de servicios web.

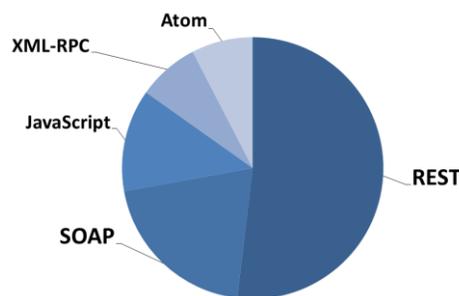


Ilustración 2 Uso de protocolos de servicios web.

Claramente, la mayor competencia se centra entre REST y SOAP, al ser considerados los estilos arquitectónicos de comunicación más usados. Un gran número de diseñadores de servicios web plantean que SOAP es complejo, por lo tanto, están comenzando a diseñar servicios web basados en REST para servir grandes volúmenes de datos.

Empresas como Amazon y Google, implementan sus servicios usando REST, y en el caso particular de la última, el uso de este estilo devino durante un proceso de mejora de sus servicios, tras anunciar la discontinuación de su API SOAP a partir del 31 de agosto de 2011. Los especialistas afirmaron, según un comunicado oficial en el Blog de Desarrolladores de Google (*The official Google Code blog*) que: “...dicha API fue disminuyendo su uso considerablemente durante los dos últimos años...”.

SOAP constituye la solución adecuada para dominios aislados, donde el entorno de implantación tiene un dominio bien definido. Por lo tanto cuando se necesita implantar un servicio web en un entorno empresarial el uso de SOAP es muy recomendado [48].

Cuando el número de usuarios es muy grande es necesario emplear una estrategia diferente. Se recomienda proponer un mecanismo explícito para la interoperabilidad de los sistemas que no poseen la misma API [48].

Por esta razón, se toma como modelo a la Web. A primera vista se puede pensar que SOAP lo hace, ya que utiliza HTTP como medio de transporte. Pero Fielding¹¹ argumenta que la Web funciona mejor cuando se utiliza con el estilo que lo hace REST [12].

Utilizar HTTP como medio de transporte para protocolos de aplicación a través de firewalls, según Fielding, es una idea equivocada. Esto reduce la efectividad de tener un firewall. Lo cual aumenta las posibilidades de nuevos agujeros de seguridad.

Los partidarios de SOAP argumentan que gracias a la tecnología existente se les permite a los diseñadores encapsular la lógica del sistema, dando lugar a interfaces que permiten facilitar el diseño.

Se puede concluir que la discusión entre ventajas y desventajas de estos estilos arquitectónicos de comunicación para los servicios web, no van incluidas de forma nativa en cada uno de ellos y la

¹¹Roy T.Fielding – Uno de los principales autores de las especificaciones de HTTP.

justificación de una selección correcta debe centrarse en el entorno de implantación del servicio demandado.

La capa de servicios web del presente trabajo de diploma tendrá a la web cubana como entorno de despliegue, por lo que un bajo uso de recursos, como el ancho de banda de transmisión, así como la escalabilidad y rendimiento para la distribución de contenidos hipermedia en una red que presenta aún un desarrollo incipiente, tienen que ser factores tomados en cuenta para la selección.

HTTP debe ser seleccionado con el objetivo de servir como protocolo de aplicación. Para agregar seguridad a la comunicación en los servicios que así lo requieran, se tendrá disponible entonces el canal que HTTP crea utilizando un cifrado basado SSL/TLS (HTTPS).

Todo lo abordado en este análisis justifica la selección de REST como estilo arquitectónico de comunicación de la capa de servicios web del buscador cubano.

1.3.3 - Descripción del Servicio

La descripción del servicio es un documento XML que define el formato del mensaje, comprensible para el servicio web. Sirve como acuerdo que establece el comportamiento de un servicio web XML e indica a los clientes potenciales cómo interactuar con él.

• **WSDL (*Web Services Description Language*)**: es un tipo de documento XML que describe lo que hace un servicio web, donde se encuentra y la forma de ser invocado [13]. Este provee información muy importante para los desarrolladores, describiendo el formato de los mensajes que utiliza y a cuáles puede responder. Siempre un documento XML WSDL presenta los siguientes elementos:

- *Tipos*: Tipos de datos usados por los mensajes.
- *Mensaje*: Que datos son enviados desde un nodo a otro.
- *Tipo de puerto*: Define las operaciones que pueden ser llamadas:
- *Límite*: Es la descripción del protocolo que se está utilizando para transportar el mensaje que puede ser HTTP POST, HTTP GET, SOAP y MIME.
- *Servicio*: Define una colección de puertos (nodos); el puerto especifica una dirección para el límite definiendo así la comunicación para un nodo específico.

- **WADL (Web Application Description Language):** es un formato basado en XML que proporciona una descripción de aplicaciones web basadas en HTTP. Estas aplicaciones suelen ser servicios web basados en REST. WADL es una tecnología de descripción presentada por la W3C, sin embargo, este consorcio no planea tenerle en cuenta para el desarrollo de sus trabajos. El éxito de este formato se ha visto frenado por WSDL 2.0, ya que este último se puede utilizar para describir servicios web basados en REST [12].

1.3.4 - Descubrimiento de Servicios

El descubrimiento de servicios es el proceso que consiste en localizar, o descubrir, uno o varios documentos relacionados que describen un servicio web determinado mediante un WSDL. A través del proceso de descubrimiento los clientes conocen la existencia de un servicio web y dónde encontrar el documento de descripción.

Al igual que con cualquier otro recurso de Internet, no se puede encontrar un servicio web XML en particular sin ayuda de algún medio que permita buscarlo. Los directorios de servicios web proporcionan ubicaciones centralizadas en las que los proveedores de estos servicios pueden publicar información acerca de los servicios de que disponen. Estos directorios pueden ser incluso servicios web XML [14].

Es necesario utilizar un directorio de servicios para localizar una organización que proporcione un servicio web con una función determinada o para especificar qué ofrece una organización concreta.

- **UDDI¹²** es un marco independiente de la plataforma para describir servicios, negocios e integrar servicios de negocios. La estructura de UDDI está basada sobre los servicios estándares de la web, lo que quiere decir que UDDI es accesible como un servicio web [2].

Su objetivo es ser accedido por los usuarios y clientes, y dar paso a documentos WSDL en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios web del catálogo de registros.

¹² Universal Description, Discovery, and Integration

1.4 - Arquitectura Orientada a Servicios

En la actualidad, el desarrollo de servicios web debe ser orientado por una arquitectura que brinde una forma bien definida de exposición e invocación de servicios, permitiendo la creación de sistemas de información altamente escalables que reflejen el negocio de la organización.

SOA¹³ es una arquitectura orientada a servicios de clientes que brinda un concepto de cómo dar soporte a los requisitos del negocio. No es una tecnología ni un producto que se pueda comprar e instalar. SOA es un conjunto de patrones, principios y prácticas para construir piezas de software que puedan interoperar independientemente de la tecnología utilizada en su implementación, definiendo las siguientes capas [15]:

- ⤴ **Aplicaciones básicas** - Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad;
- ⤴ **De exposición de funcionalidades** - Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (generalmente como servicios web);
- ⤴ **De integración de servicios** - Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración;
- ⤴ **De composición de procesos** - Que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio;
- ⤴ **De entrega** - donde los servicios son desplegados a los usuarios finales.

La clave de la arquitectura SOA está en la exposición de interfaces abstractas que aíslen de la implementación particular de cada pieza de software. Para conseguir este objetivo resulta especialmente útil el uso de servicios web [15].

13 Service Oriented Architecture

1.5 - Servicios web en la Universidad de las Ciencias Informáticas

La UCI¹⁴ desde la perspectiva de ciudad digital ha dado algunos pasos para lograr una integración de los servicios, por los beneficios que su uso reportaría en los diferentes procesos que se llevan a cabo dentro de la Universidad, pero aún este desarrollo es insuficiente.

Actualmente ya se cuenta con un UDDI¹⁵ en el que se pueden encontrar los servicios web para la gestión de trabajadores que son plantilla de la UCI (ASSETS- Sistema de Capital Humano), para la gestión de las personas con beca en la Universidad (Registro de Residentes) entre otros, como la autenticación y la obtención de la identidad del personal de esta casa de altos estudios. Este sistema de servicios web está desarrollado principalmente sobre el concepto de los servicios web basados en SOAP [16].

1.6 - Servicios web en motores de búsqueda en el mundo

Es notable el aumento de la cantidad de servicios web disponibles en Internet. La tendencia al uso de este tipo de servicios actualmente se realiza mediante la disponibilidad de los recursos a través de las API¹⁶. Aplicaciones web de relevancia, como son las redes sociales que día a día demandan mayor disponibilidad en el ciberespacio, utilizan fehacientemente este tipo de tecnologías. Facebook¹⁷ y Twitter¹⁸, por solo mencionar algunas, aumentan su campo de acción y su presencia gracias a los servicios web en los diversos *mashup*¹⁹ publicados en la red de redes [18].

Los buscadores, como herramientas de obligatoria consulta en Internet, extienden sus servicios más allá de su propia interfaz usando estas tecnologías que evidencian la evolución de la web. Es oportuno por lo tanto, realizar un estudio sobre algunos ejemplos actuales en los que se integren los sistemas de recuperación de información con los servicios web.

¹⁴ UCI (Universidad de Ciencias Informáticas).

¹⁵ <http://uddi.uci.cu>

¹⁶ Application Programming Interface-Interfaz de Programación de Aplicaciones.

¹⁷ <http://www.facebook.com>

¹⁸ <http://www.twitter.com>

¹⁹ Aplicación que usa y combina datos y funcionalidades procedentes de una o más fuentes para crear nuevos servicios.

1.6.1 - Yahoo!

Yahoo! comenzó como un hobby de estudiantes y se convirtió en un buscador que permite encontrar y acceder a la información en la web. Los dos fundadores de Yahoo!, David Filo y Jerry Yang, candidatos en Ingeniería Eléctrica en la Universidad de Stanford, comenzaron la guía Yahoo! en un tráiler del campus en febrero de 1994 como una manera de no perder de vista sus intereses personales en Internet. Rápidamente comenzaron a pasar más tiempo en su lista de enlaces favoritos que en sus tesis doctorales [19].

Con el tiempo, las listas de Jerry y David se hicieron demasiado largas y difíciles de manejar, y estas debieron agruparse en categorías. Cuando las categorías se llenaron demasiado, desarrollaron subcategorías y así nació el concepto central de Yahoo! [19].

Yahoo! para la publicación de sus servicios web utiliza un API basada en una interfaz RESTful o REST haciendo uso de los principales métodos http para su funcionamiento y obteniendo sus resultados en un documento XML [20].

Los principales parámetros de las consultas hacia Yahoo! están dados por la llave de acceso a la API, el método de búsqueda y la consulta. A continuación se muestra un ejemplo de una consulta utilizando el método GET, donde se escribe la dirección en que se encuentra situado el recurso, seguido de un conjunto de parámetros que permiten la ejecución de la acción solicitada.

http://upcoming.yahooapis.com/services/rest/?api_key=<APIKey>&method=event.search&search_text=kill&me_id=1

Los servicios web de este motor de búsqueda hoy día van mucho más allá del sencillo servicio de consulta al índice de Yahoo! Otro servicio que sobresale es el dedicado al análisis de contenidos, este detecta entidades, conceptos, categorías y relaciones dentro de contenidos no estructurados.

El servicio de análisis de contenidos de Yahoo! se encuentra disponible como una tabla de YQL²⁰ y al existir la posibilidad de que la extensión de la cadena a analizar sea muy grande, las solicitudes al servicio se deberán realizar mediante peticiones HTTP POST.

Una petición a este servicio estará estructurada por los parámetros que a continuación relaciona la siguiente tabla.

²⁰ Yahoo! Query Language (Lenguaje de consultas de Yahoo!)

Parámetro	Tipo	Descripción
text	string	Contenido para analizar (UTF-8).
url	string	La URL de la página web a analizar.
related_entities	boolean: <i>true</i> (por defecto), <i>false</i>	Inclusión de las entidades relacionadas.
show_metadata	boolean: <i>true</i> (por defecto), <i>false</i>	Inclusión de metadatos de entidades.
enableCategorizer	boolean: <i>true</i> (por defecto), <i>false</i>	Inclusión de la categoría del documento.
unique	boolean: <i>true</i> , <i>false</i> (por defecto)	Detección de entidad de ocurrencia única.
max	integer: <i>100</i> (por defecto)	Número máximo de entidades a detectar.

Tabla 2 Parámetros de solicitud analizador de contenidos de Yahoo! [20].

Otro de los servicios web expuesto por Yahoo! es el extractor de términos, que basa su funcionamiento en la devolución de un conjunto de términos relevantes presentes en un extenso documento que se envíe para su análisis. Para acceder a este servicio la petición se dirigirá hacia la URL (Ver Anexo1).

http://search.yahooapis.com/ContentAnalysisService/V1/termExtraction

Las peticiones a este servicio deberán contar con los siguientes parámetros:

Parámetro	Tipo	Descripción
appid	string (requerido)	Identificación de la aplicación en la capa.
context	string (requerido)	Contenido a analizar (Codificado UTF-8).
query	string	Consulta opcional para facilitar el proceso de filtrado.
output	string: <i>xml</i> (default), <i>json</i> , <i>php</i>	Formato de salida.

Tabla 3 Parámetros de solicitud extractor de términos de Yahoo! [20].

1.6.2 - Google

Google, motor de búsqueda propiedad de la empresa Google Inc. y fundado en septiembre de 1998 por Larry Page y Sergey Brin; estudiantes de la Universidad de Stanford, es actualmente el buscador más usado para localizar información en la web a nivel mundial [21].

Hasta el 1 de noviembre de 2010 Google utilizaba para proveer sus servicios web la llamada **SOAP Search API**, pero a partir esta fecha dicha API fue obsoleta y se puso en uso la nueva API de

servicios web **JSON/Atom Custom Search API** la cual basa su funcionamiento en el uso del principio arquitectónico REST [22].

Google posibilita 100 consultas por días de forma gratuita para todos los usuarios que utilicen esta API. Luego de haber consumido dicha cuota, será necesario pagar 5 dólares por cada 1000 consultas adicionales [23].

El uso de la API REST de Google se realiza mediante la llamada a métodos HTTP como son GET y POST para acceder a las estructuras de datos JSON o Atom [22].

La siguiente URI corresponde a una petición al servicio web de búsqueda de Google:

<https://www.googleapis.com/customsearch/v1?q=universidad+cuba&output=xml>

La solicitud pide los primeros 10 resultados que coincidan con la consulta de los términos de búsqueda "universidad" y "cuba", esta a su vez especifica que el formato del documento respuesta será XML.

Otro ejemplo de solicitud a este servicio web sería:

https://www.googleapis.com/customsearch/v1?q=universidad+cuba&output=xml&lr=lang_es&start=10&num=5

Aquí se puede encontrar como novedad la inclusión de parámetros como ***lr***, ***start*** y ***num***. Estos datos restringen la búsqueda hacia páginas de un determinado lenguaje, posición inicial de los resultados a obtener así como la cantidad de resultados a mostrar respectivamente. En la solicitud anterior los resultados que se obtendrán serán los que coincidan con la consulta de los términos de búsqueda "universidad" y "cuba", las páginas tendrán que estar en el idioma español, se mostrará los resultados del 11 al 15 y el formato del documento respuesta será XML.

A continuación se resumen los principales parámetros a tener en cuenta para trabajar con el servicio web de búsqueda de Google [22].

Parámetros	Descripción
as_epq	Añade una determinada frase a la consulta de búsqueda del parámetro q.
as_filetype	Especifica los tipos de archivos a incluir o excluir en los resultados de búsqueda.
as_lq	Centra los resultados de búsquedas en páginas de una determinada URL.
getfields	Indica los nombres de las etiquetas a retornar por cada resultado de búsqueda.

ie	Establece la codificación de caracteres para interpretar la cadena de consulta.
lr	Restringe la búsqueda hacia páginas de determinado lenguaje.
num	Máximo número de resultados a retornar.
oe	Establece la codificación de caracteres para codificar los resultados.
output	Indica el formato del documento de salida. Este parámetro es requerido.
q	Consulta de búsqueda. Este parámetro es requerido.
requiredfields	Restringe la búsqueda hacia los documentos que posean exactamente las etiquetas especificadas.
start	Especifica el número de índice de la primera entrada del documento a retornar.

Tabla 4 Parámetros fundamentales servicio web de búsqueda de Google [22].

Existen otros campos de respuesta de este servicio web (Ver Anexo 2).

1.7 - Motores de Búsqueda en Cuba

El primer buscador cubano en Internet para los sitios con dominio .cu fue presentado el 15 de febrero de 2007 por Roberto del Puerto, director de la Oficina para la Informatización, en la XII Convención y Expo Internacional Informática 2007. El sistema, conocido como 2x3 (versión Beta), es resultado de una revisión de más de 100 000 direcciones de sitios cubanos en Internet [24].

La característica esencial de 2x3, y que constituye su principal novedad, es que su acción de búsqueda se desarrolla en el dominio .cu. Con ello se facilita considerablemente el hallazgo de información generada en el país, cuya exploración hasta el momento dependía de los buscadores de cada sitio y ramales de dominio [24].

Con el paso de los años el buscador 2x3 quedó totalmente sin soporte y prácticamente inutilizado, debido a ello el Ministerio de la Informática y las Comunicaciones (MIC) le solicita a la Universidad de las Ciencias Informáticas (UCI) la implementación de un sistema de búsqueda que permitiera sustituir 2x3 y al cual se le pudiera brindar soporte por parte de la Universidad.

El motor de búsqueda Orión es una sistema de búsqueda web desarrollado por el centro CIDI²¹ de la Universidad de las Ciencias Informáticas. Entre sus principales servicios se encuentran los servicios

²¹ Centro de Ideoinformática

de búsqueda, posicionamiento y webmetría. La arquitectura de este buscador está compuesta por la integración de varias herramientas libres como: Solr en su versión 3.4 y Nutch versión 1.4.

1.8 - Metodologías, herramientas y tecnologías

El desarrollo de un software²² es un proceso complejo en el cual se tienen muchos elementos que permitirán su construcción con calidad.

Para la implementación de una aplicación, es necesario tener en cuenta las tecnologías sobre las cuales se va a desarrollar el sistema, pues de estas dependerá totalmente su funcionamiento y uso.

1.8.1 - Frameworks para servicios web

En el mundo de desarrollo de aplicaciones informáticas el uso de los frameworks se vuelve inminente, ya que estas estructuras conceptuales y tecnológicas de soporte definido, constituyen la base con la cual otro proyecto de software puede ser fácilmente organizado y desarrollado [10].

Para la creación de servicios web, estas tecnologías también ganan su espacio debido que permiten a los desarrolladores dedicar mayor cantidad de tiempo a diseñar los servicios para brindar, y no en profundizar en detalles de bajo nivel de implementación para ponerles operativos.

Actualmente, son muchos los frameworks que existen y que implementan en diversos lenguajes los protocolos que se usan para la creación de los servicios web. El uso y popularidad de estos está relacionado con el lenguaje sobre el que están escritos y la cantidad de protocolos que implementan. A continuación se relacionan algunos de estos frameworks:

- **Apache Axis2:** Se trata de un completo rediseño y reescritura de Apache Axis. Este framework tiene como principal capacidad añadir interfaces de servicios web a las aplicaciones, pero también puede funcionar como una aplicación de servidor independiente.

Axis2 no solo es compatible con SOAP 1.1 y SOAP 1.2, pues también tiene soporte integrado para REST, lo que lo vuelve una de las elecciones certeras para los que desarrollan servicios web en Java o C [25].

Se encuentra bajo la licencia Apache License 2.0, descrita por la Apache Software Foundation [25].

²² Aplicación Informática

- **Zend Framework (ZF) 1.11:** es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de este framework es algo única; cada componente está construido con una baja dependencia de otros componentes [26].

Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "*use-at-will*" (uso a voluntad). Este framework tiene la ventaja de implementar en PHP los protocolos usados para crear servicios web: SOAP, JSON, JSON-RPC, REST, XML-RPC [26].

- **Jersey (JAX-RS: Java API for Restful Web Services)** (en español: Interfaz de Abstracta de Programación en Java para servicios web Restful): es una implementación referencia y de código abierto de la Sun Microsystem para la implementación de servicios web en Rest. Contiene una API que provee soporte nativo desde el lenguaje Java para la creación de los servicios. Jersey forma parte oficial de la edición empresarial de la plataforma Java en su versión 6 [27].

Internamente esta API incluye una serie de anotaciones necesarias para un completo trabajo sobre el protocolo HTTP, haciendo el proceso de creación de recursos para exponer, mucho más sencillo para los desarrolladores.

Este frameworks resulta significativo para la implementación de servicios utilizando la tecnología REST sobre el lenguaje Java, característica que mejora el intercambio de datos con el núcleo del buscador. Por todo lo planteado se selecciona Jersey para el desarrollo de la aplicación, objetivo del presente trabajo.

1.8.2 - Lenguajes de Programación

Los lenguajes de programación constituyen el elemento esencial en la construcción de un software. Para la implementación de la capa de servicios web se decide estudiar una serie de lenguajes de desarrollo, afines a las tecnologías sobre las cuales se implementan los servicios web:

- **PHP**

PHP (*Hypertext Preprocessor*): es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML [28].

En lugar de usar muchos comandos para mostrar HTML (como en C o Perl), páginas PHP contienen HTML con código incluido en ellas. El código PHP está entre etiquetas de comienzo y final especiales `<?php` y `?>` que nos permitirán entrar y salir del "modo PHP". [28].

Lo que distingue a PHP de lenguajes de ejecución del lado del cliente como Javascript, es que el código es ejecutado en el servidor, generando el fuente HTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el script, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los archivos HTML con PHP [28].

• Ruby

Ruby es un "lenguaje de guiones (scripts) para una programación orientada a objetos rápida y sencilla", fue creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU [51].

Está diseñado para la productividad y la diversión del desarrollador, siguiendo los principios de una buena interfaz de usuario. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de Software Libre [29].

• Java

Java es un lenguaje de programación orientado a objeto desarrollado por Sun Microsystem a principio de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria [30].

Se utiliza en los principales sectores de la industria de todo el mundo y está presente en un gran número de dispositivos, equipos y redes [30].

La versatilidad y eficiencia de la tecnología Java, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en la tecnología ideal para su aplicación a redes. De portátiles a centros de datos, de consolas de juegos a súper-equipos científicos, de teléfonos móviles a Internet, Java está

en todas partes. Java ha sido probado, mejorado, ampliado y probado por una comunidad especializada de más de 6,5 millones de desarrolladores, la mayor y más activa del mundo [30].

La selección del framework de desarrollo de servicios, la robustez y la capacidad de integración con el núcleo del motor de búsqueda justifican la utilización de Java como lenguaje de desarrollo.

1.8.3 - Metodologías de desarrollo

Las metodologías de desarrollo de software constituyen una guía para estructurar, planificar y controlar el proceso de desarrollo de sistemas informáticos.

Cada metodología de desarrollo de software tiene su propio enfoque para el desarrollo de aplicaciones. Además, tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado, que permitan desarrollar aplicaciones de calidad, incluyendo heurísticas de construcción y criterios de comparación de modelos de sistemas [31].

Para tal fin se describen, fundamentalmente, herramientas de Análisis y Diseño Orientado a Objetos, sus diagramas, especificación, y criterios de aplicación de estas [31].

SXP

SXP es una metodología de desarrollo de software compuesta por las metodologías SCRUM y XP, que ofrece una estrategia tecnológica a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva, fomentando el desarrollo de la creatividad y aumenta el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control de este [32].

SXP es una metodología encaminada para el desarrollo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto [32].

Consta de cuatro fases principales: Planificación-Definición, donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto. Desarrollo, es donde se realiza la implementación del sistema hasta que esté listo para ser entregado. Entrega, puesta en marcha. Mantenimiento, donde se realiza el soporte para el cliente [32].

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras. En estas fases se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, lo que permite mejorar el diseño cada vez que se le añade una nueva funcionalidad [32].

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro. Permite además, seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo [32].

RUP

Proceso Unificado de Rational (*Rational Unified Process*) es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización [33].

El ciclo de vida RUP es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semiordenadas y con un ciclo de vida que organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades [33]. (Ver Anexo3)

Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos que son los productos tangibles del proceso de desarrollo como por ejemplo, el modelo de casos de uso, el código fuente y otros.

También define los roles que constituyen el papel que desempeña una persona en un determinado momento. La metodología RUP es apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas [33].

Con el objetivo de lograr una uniformidad de documentación en todos los componentes del motor de búsqueda Orión, además de estar considerada una metodología muy completa para el desarrollo de software. Se selecciona RUP para guiar el desarrollo de la capa de servicios web.

1.8.4 - IDEs de Desarrollo

Para agilizar la construcción de aplicaciones, los desarrolladores se auxilian de un entorno de desarrollo integrado IDE (*Integrated Development Environment*). Estos programas contienen un conjunto de herramientas de programación que permiten el desarrollo de otras aplicaciones en determinado lenguaje. Estos pueden realizar las funciones de un editor de código, un compilador, depurador y hasta un constructor de interfaz gráfica.

Zend Studio V 7.1.0

Zend Studio es un ambiente de desarrollo que abarca todos los componentes necesarios para desarrollar una aplicación PHP. Es multiplataforma y su interfaz de usuario proporciona rasgos que ayudan en el proceso de desarrollo.

Contiene una ayuda contextual con todas las bibliotecas de funciones del lenguaje que asiste en todo momento, ofreciendo nombres de las funciones y parámetros que deben recibir. Zend Studio dispone de una herramienta muy útil que permite ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno: las cookies [34].

NetBeans V 7.0-m1

NetBeans IDE es un entorno de desarrollo creado por Sun Microsystems en junio 2000, es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. Es de código abierto y gratuito para uso tanto comercial como no comercial.

El código fuente está disponible para su reutilización de acuerdo con la *Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2.23* [35].

²³ <http://netbeans.org/cddl-gplv2.html>

Su selección para el desarrollo de la aplicación objetivo de esta tesis, está sustentada por ser gratuito, libre y potente. Proporcionar todas las herramientas y funciones necesarias para realizar el presente trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar.

La integración con el lenguaje y framework seleccionados lo convierten en la selección más certera para este tipo de herramienta.

1.8.5 - Sistemas de Gestión de Base de Datos

Los sistemas de gestión de bases de datos son un tipo de software dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de estos sistemas es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización [36].

Por su parte, los almacenes de datos pueden ser relacionales y no relacionales, a estos últimos también se les conoce comúnmente como bases de datos No SQL [36].

Las bases de datos relacionales cumplen con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer conexiones entre los datos que están guardados en tablas y a través de dichas conexiones relacionar los datos de ambas tablas; de ahí proviene su nombre: "Modelo Relacional" [36].

Por otro lado, los almacenes de datos no relacionales no proporcionan garantías ACID²⁴. Normalmente no tienen esquemas fijos de tablas ni sentencias "join"²⁵. Estos sistemas responden a las necesidades de escalabilidad horizontal que tienen cada vez más empresas.

Dentro de los sistemas de gestión de bases de datos relacionales usados a nivel internacional se encuentran: PostgreSQL, MySQL, ORACLE y SQL-Server.

Actualmente los almacenes de datos No SQL que más sobresalen son: MongoDB, Cassandra y BigTable, todos usados en aplicaciones que manejan grandes volúmenes de datos como son: Twitter, Facebook y Google respectivamente.

²⁴ Conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

²⁵ Sentencia para relacionar tablas.

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales [36].

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando [36].

MySQL

MySQL es el Sistema de Gestión de Base de Datos de código abierto más popular del mundo, con más de 100 millones de copias a lo largo de su historia. Con su velocidad, fiabilidad y facilidad de uso, se ha convertido en la elección predilecta por los desarrolladores web.

MySQL es parte importante de LAMP (Linux, Apache, MySQL y PHP), toda una suite o compendio de aplicaciones de código abierto con amplia aceptación por las empresas [37].

MongoDB

Un estudio enfocado en las base de datos no relacionales arroja que MongoDB constituye una solución escalable y de alto rendimiento de almacenes de datos No SQL.

De código abierto y escrito en C++, este sistema tiene orientado el almacenamiento de datos en documentos al estilo JSON con esquemas dinámicos, que ofrecen potencia y simplicidad. Se destaca por conservar los índices de todos los atributos y hacer mucho más flexible la agregación y procesamiento de datos [38].

Esto tiene ventajas y desventajas, sin embargo, existe un sesgo en el esquema, que puede hacer ciertas consultas un poco menos elegante que en el esquema relacional.

Las colecciones en MongoDB son análogas a las tablas en una base de datos relacional. Cada colección contiene varios documentos. Como se mencionó anteriormente estos documentos pueden ser grandes.

No existe ninguna declaración explícita de los campos de los documentos de una colección. Sin embargo, hay un diseño conceptual del esquema de lo que esos campos representarán y cómo los documentos de la colección estarán estructurados [38].

MongoDB no requiere que en una colección todos los documentos tengan la misma estructura. Sin embargo, en la práctica, la mayoría de las colecciones son altamente homogéneas [38].

Dada las características planteadas se selecciona MongoDB para el almacenamiento de los datos relacionados con la capa, así como para los parámetros fundamentales del tráfico que se genere, con fines estadísticos. La selección de este sistema de almacenamiento de datos se justifica por lograr un mejor manejo de la información frente a un volumen considerable de datos.

1.8.6 - Servidores Web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente. Estos generan una respuesta en cualquier lenguaje entendible por el cliente. Para la transmisión de todos estos datos suele utilizarse generalmente el protocolo HTTP, perteneciente a la capa de aplicación del modelo OSI. A continuación se describirán algunas de estas aplicaciones servidoras [39].

Apache 2

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual [39].

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la *Apache Software Foundation*. Presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración [39]. Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado.

Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM WebSphere Application Server. Mac OS X integra apache como

parte de su propio servidor web y como soporte de su servidor de aplicaciones WebObjects. Apache es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux. Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable [39].

Algunos de los grandes sitios web del mundo están ejecutándose sobre Apache. La capa frontal (front end) del motor de búsqueda Google está basada en una versión modificada de Apache, denominada Google Web Server (GWS). La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original [39].

Apache Tomcat

Es un servidor que funciona como un contenedor de servlets²⁶ desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software Licence [40].

Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache. Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era solo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java [40].

Por las características anteriormente expuestas se selecciona Apache Tomcat como servidor web para la capa de servicios, pues como contenedor de servlets brindará estabilidad y robustez al sistema las cuales son dos garantías muy necesarias y que Tomcat brinda.

²⁶ Programa que se ejecuta en un servidor para generar páginas web de forma dinámica.

1.8.7 - Herramientas CASE

Las herramientas de Ingeniería de Software Asistida por Computadora (CASE) (Computer Aided Software Engineering) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero [41].

Rational Rose Enterprise

Rational Rose Enterprise es compatible con el lenguaje UML (*Unified Modeling Language*) y es uno de los productos más completos de la familia Rational Rose. Es capaz de analizar la calidad del código y de generarlo gracias a las capacidades de sincronización configurable entre el modelo y el código. Incluye un complemento de modelado web, que proporciona la capacidad de visualización y el modelado, además de herramientas para desarrollar aplicaciones web. Permite el modelado UML para diseñar bases de datos, con la posibilidad de representar la integración de los requisitos de datos y aplicaciones mediante diseños lógicos y físicos. Es una herramienta CASE propietaria y es admitida solo por el sistema operativo Windows [42].

Visual Paradigm

Visual Paradigm (VP) en su versión 5.3 constituye una herramienta robusta. Además de que utiliza UML como lenguaje de modelado es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto [43].

Permite control de versiones y realizar ingeniería tanto directa como inversa en diferentes lenguajes. Se puede representar todos los tipos de diagramas UML para las distintas fases que propone RUP, como la captura de requisitos, análisis, diseño e implementación. Esta herramienta facilita la comunicación, ya que utiliza un lenguaje estándar común a todo el equipo de desarrollo.

Presenta la posibilidad de la interoperabilidad con otras aplicaciones como es el Rational Rose. Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community (que es gratuita). También se facilita licencias especiales para fines académicos [43].

Por todo lo anterior se selecciona Visual Paradigm 5.3 para realizar el proceso de modelado.

1.9– Conclusiones parciales

En este capítulo se abordaron los principales basamentos teóricos necesarios para el presente trabajo de diploma y se realizó un estudio del estado del arte en el mundo de los servicios web.

Populares sistemas de recuperación de información como son Google y Yahoo! publican sus capas de servicios web para la realización de consultas a sus índices y estas convergen en poseer una serie de parámetros que según el valor especificados en la petición influyen en el filtrado del contenido que se desea recuperar en cada consulta.

Se puede concluir que:

Se obtuvieron los elementos, como tecnologías y principales parámetros recomendados para la disponibilidad de los recursos de los servicios web.

La sistematización del estudio del estado del arte de los servicios web permitió seleccionar la siguiente base tecnológica:

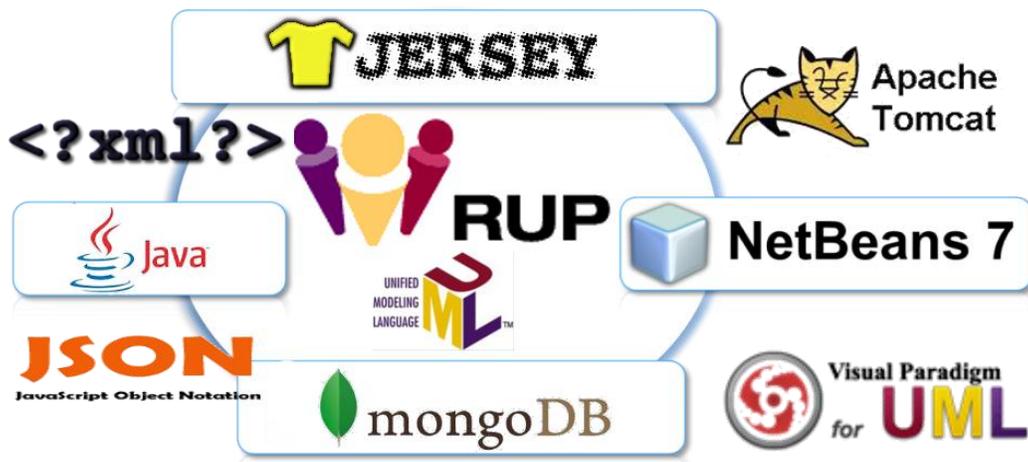


Ilustración 3 Base tecnológica propuesta.

Capítulo 2: Características del Sistema

2.1 - Introducción

Dadas las necesidades planteadas en la situación problemática del presente trabajo de diploma, se determina desarrollar una capa de servicios web. Esta aplicación debe contar con las funcionalidades necesarias para que sistemas informáticos se comuniquen con el motor de búsqueda Orión. Para el desarrollo de los servicios se necesitan los resultados para determinado criterio de búsqueda y los datos estadísticos existentes en el motor de indexación.

En el presente capítulo se identifica las clases del dominio y la relación entre ellas, los requerimientos funcionales y no funcionales que debe cumplir la capa de servicios web. Se realiza el modelado de casos de uso del sistema y las descripciones textuales de estos. Este capítulo tiene el objetivo de dejar abiertas las puertas al diseño de la solución a implementar y que esta contribuya a dar la solución deseada a las necesidades que dieron su origen.

2.2 - Propuesta del sistema

La capa estará enfocada en permitirle a los sistemas que interactúen con ella, realizar peticiones que incluirán consultas parametrizadas que contribuyan con el filtrado de los resultados encontrados y por cada petición generar un mensaje de respuesta que contenga la información solicitada.

Para que estas funcionalidades puedan ser consumidas desde otros entornos, se implementará una interfaz de programación de aplicaciones o API (*Application Programming Interface*) utilizando la base tecnológica propuesta en el capítulo anterior, que se conecte con la capa de servicios del buscador. Esta proporcionará una serie de funciones que darán mayor abstracción a los desarrolladores facilitando el uso de la capa de servicios web del motor de búsqueda Orión.

La siguiente ilustración muestra una idea gráfica de lo que se propone como sistema general:

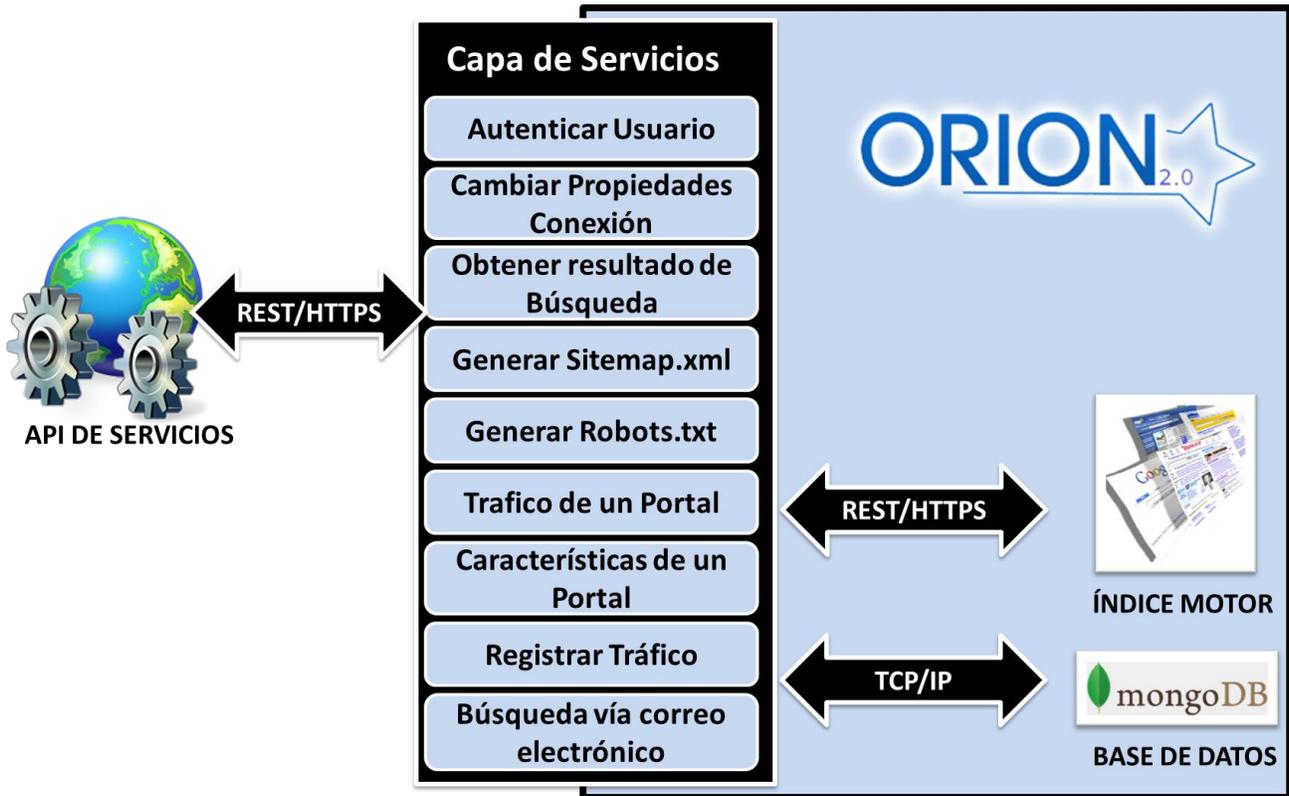


Ilustración 4 Propuesta del Sistema

El sistema externo representa una aplicación informática que necesita consumir los servicios del motor de búsqueda. Esta aplicación agrega la API de servicios web que se encargará de brindar las funciones necesarias que garanticen la correcta interacción entre ambos sistemas.

A modo general dentro de los servicios que expondrá la capa del motor de búsqueda Orión se encuentran:

- **Autenticar Usuario:** Permitirá la autenticación del usuario con sus parámetros de conexión: usuario y contraseña, creándose una sesión correspondiente al mismo, devolviendo la clave que lo identifica, para que posteriormente pueda consumir los demás servicios que se brindan pasando dicha clave en cada petición. Con esto se evita tener que enviar este tipo de datos sensibles con frecuencia por el canal de comunicación.

Esta clave va a ser única y contará de 28 caracteres. Se generará automáticamente y tendrá una vigencia igual al tiempo que el usuario esté usando los servicios web. Las sesiones van a contar con un tiempo de expiración, marcado por la inactividad del usuario.

El sistema periódicamente chequea las sesiones y cierra aquellas cuyo tiempo inactivo sea igual al de expiración.

- **Cambiar Propiedades Conexión:** Permitirá cambiar las propiedades de conexión de una sesión creada previamente, dígame contraseña, país y nombre del cliente.

- **Obtener resultados de búsqueda:** Representa una serie de funcionalidades que se encargan de devolver al sistema que realizó la petición un mensaje que incluye los resultados de búsqueda según la consulta y los parámetros de filtrado incluidos.

Este filtrado se podrá realizar mediante la selección de URL, MIME, contenidos, etcétera.

- **Obtener Características de un portal:** Representa una serie de funcionalidades que se encargan de devolver al sistema que realizó la petición un mensaje que incluye algunas de las características de determinado portal. Estas funcionalidades solo podrán ser consumidas por usuarios registrados como administradores del portal en consulta.

- **Generar Sitemap.xml:** Se refiere a la funcionalidad de generar un mapa de determinado sitio web, donde se listan las páginas presentes en él, generando el contenido del documento sitemap.xml que puede ser utilizado como herramienta de planificación que lista las páginas y que mejora el posicionamiento en los buscadores. Este mapa se generará con los resultados obtenidos por la última recolección realizada al portal señalado.

- **Generar robots.txt:** Funcionalidad que genera el contenido del fichero que sigue el estándar de exclusión de robots, también conocido como protocolo de robots.txt y cuyo objetivo es evitar que ciertos bots que analizan los sitios web u otros robots que investigan todo o una parte del acceso de un sitio web, público o privado, agreguen información innecesaria a los resultados de búsqueda. Este fichero será generado con el fin de realizarles recomendaciones de optimización a los administradores del sitio para el cual se solicite el servicio.

- **Obtener tráfico de un portal en el buscador:** Representa una serie de funcionalidades que se encargan de devolver al sistema que realizó la petición un mensaje que incluye la información

referente a la presencia de un portal en el buscador. La presencia de un sitio en el buscador estará basada en la cantidad de impresiones que dicho portal haya tenido para ciertos criterios de búsqueda, cantidad de click, fecha de las consultas, ip desde donde se generen estas consultas, entre otros datos que servirán de información útil a los webmaster.

- **Registrar tráfico:** Permitirá registrar todos los datos relacionados con los sitios que se visiten tras la muestra de resultados creada previamente, dígame usuario y contraseña.

- **Búsqueda vía correo electrónico:** Este servicio permitirá obtener los resultados de la búsqueda para un criterio determinado mediante el envío de un correo electrónico.

2.3 – Modelo de dominio

Debido a la falta de un proceso de negocio bien definido se decidió desarrollar un modelo de dominio (MD) que representa una parte visual de las clases y sus interacciones con el entorno real del proyecto. El diagrama que se elabora en el MD representa los objetos relacionados con los principales conceptos que se trabajarán en el desarrollo de este trabajo.

Este diagrama permitirá a los usuarios, desarrolladores e interesados lograr una mayor comprensión de los conceptos que se manejan, permitiéndoles utilizar un vocabulario común para comprender el contexto en el que se encuentra el sistema. Contribuirá, además, a identificar personas, eventos, transacciones y objetos involucrados en el sistema. [33].

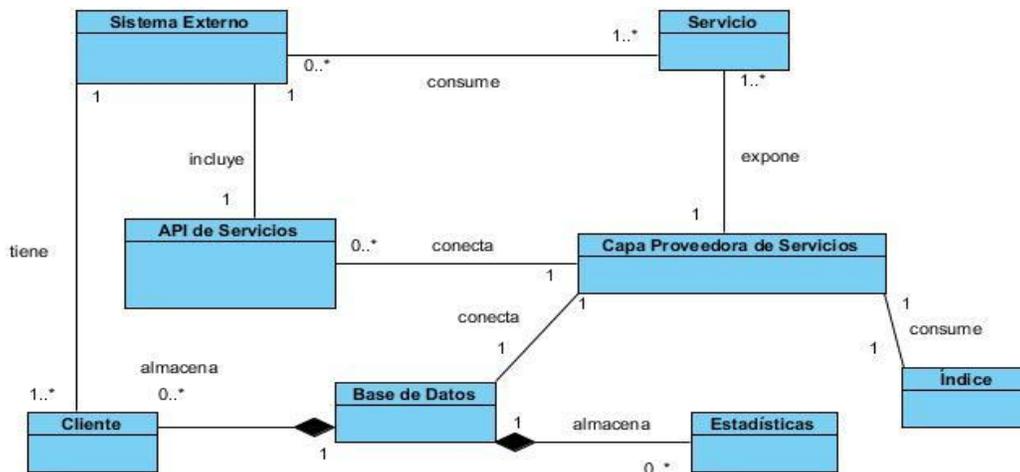


Ilustración 5 Modelo de Dominio

A continuación, se explica en qué consiste cada una de las clases del Modelo de Dominio.

- **Servicios:** Funcionalidad que es ofrecida para ser utilizada por los sistemas externos.
- **Sistema Externo:** Aplicación informática que va a consumir los servicios web que se brindan.
- **Base de Datos:** Aplicación informática que maneja la información referente a las estadísticas de búsquedas así como al registro de usuarios consumidores de los servicios del motor, de forma organizada y persistente.
- **Capa proveedora de Servicios:** Sistema que expone los servicios web del motor de búsqueda, atiende las solicitudes que recibe generando respuestas a estas. Consume directamente del índice invertido del servidor de indexación del buscador e interactúa con la base de datos en el registro de cada petición que se realice.
- **Índice:** Se refiere al servidor de indexación Solr, sistema en el que se ubican los resultados de búsqueda para determinados criterios y parámetros.
- **API de servicios:** Interfaz o librería que contiene las funciones necesarias para realizar conexiones exitosas con la capa proveedora de servicios del motor. Punto de enlace entre los sistemas externos y el buscador. Dicha interfaz estará disponible en varios lenguajes.
- **Estadísticas:** Información relacionada con los consumos de servicios web y la que se almacenarán los datos de cada una de las peticiones (dirección IP, fecha, criterio de consulta, parámetros).
- **Cliente:** Usuarios autorizados a consumir los servicios web del buscador. Administrador del sistema externo que interactuará con la capa.

2.4 – Relación de los requerimientos del sistema

La IEEE Standard Glossary of Software Engineering Terminology (en español Glosario Estándar de Terminologías de Ingeniería de Software) define requerimiento como condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

También se define como requerimiento a la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente [44].

A continuación se presentan los requerimientos que debe cumplir el sistema que se propone. Estos se han clasificado en funcionales y no funcionales.

2.4.1 – Requerimientos funcionales

RF1. Autenticar usuario.

RF2. Devolver N resultado de búsquedas.

RF3. Devolver posición de un sitio en los resultados para un criterio de búsqueda.

RF3.1 Devolver posición de una página para los resultados de una búsqueda dentro de un sitio.

RF4. Devolver N resultado de páginas mejor posicionadas de determinado sitio para un criterio de búsqueda.

RF5. Devolver la cantidad de enlaces entrantes que tiene un sitio.

RF5.1 Devolver los sitios que apuntan al sitio cliente.

RF6. Devolver la cantidad de páginas de un portal indexadas en el buscador.

RF7. Devolver las páginas de un portal indexadas en el buscador.

RF8. Devolver el Page Rank en el buscador de las páginas.

RF9. Registrar tráfico desde el buscador.

RF10. Generar sitemap.xml

RF11. Generar robots.txt

RF12. Devolver los criterios de búsqueda por los que más se visita una página desde el buscador.

RF13. Devolver el tráfico a un sitio desde el buscador.

RF13.1. Devolver cantidad de clic.

RF13.2. Devolver direcciones IP.

RF13.3. Devolver navegador.

RF13.4. Devolver por rango de fecha.

RF13.5. Devolver sistema operativo.

RF14. Devolver la cantidad de resultados encontrados para un criterio de búsqueda.

RF15. Devolver por tipo MIME los resultados de búsqueda.

RF16. Devolver fecha de creación de los resultados para criterios de búsquedas.

RF17. Devolver tiempo de respuesta de la consulta para un criterio de búsqueda.

RF18. Cambiar propiedades de la conexión.

RF19. Devolver los resultados de búsqueda mediante una consulta vía correo electrónico.

2.4.2 – Requerimientos no funcionales

Seguridad:

Para mantener una mayor confiabilidad se necesita conocer la identidad del usuario (clave de sesión) que consume los servicios, cada vez que realice una petición.

Por las características del sistema, se ha definido que el tiempo límite de demora que un consumidor tiene para solicitar un servicio antes de que expire su sesión es de 15 minutos, pasado ese tiempo tendría que volver a obtener una nueva sesión.

Se le debe incorporar la opción de poder usar certificados digitales, permitiendo enviar los datos vía HTTPS. Además, se debe garantizar la disponibilidad de los servicios en cualquier momento para los desarrolladores.

Validar la entrada de datos contra los ataques más frecuentes.

No permitir conexiones remotas al servidor de base de datos, Solr, Nutch, MongoDB.

Hardware:

Pentium 4/Core2Duo/DualCore/ u otro superior.

Mínimo de Memoria RAM: se recomienda 1 GB.

Software:

Tener Debian 6 como sistema operativo del servidor.

Máquina Virtual de Java.

Portabilidad: Independientemente de la plataforma de desarrollo en que fueron implementados los servicios, las aplicaciones que lo consuman pueden hacerlo sin tener en cuenta esta, desde cualquier

plataforma o lenguaje de programación.

Soporte: Se requiere una documentación apropiada que describa todas las funcionalidades del sistema desarrollado así como una guía para su uso.

Flexibilidad: El sistema permitirá que se defina mediante parámetros las diferentes formas en las que se realizarán las peticiones.

2.5 - Modelo de Casos de Uso del Sistema

En esta sección se identifican los actores y casos de uso del sistema, quedando determinado el Diagrama de Casos de Uso del Sistema. Se describen además cada uno de estos casos de uso.

2.5.1 - Definición de los actores del sistema

Los actores del sistema intercambian información con él, aunque no forman parte de este. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

A continuación se muestra el actor y la justificación que tiene en el sistema.

Actores del Sistema	Sistema Externo
Justificación	Es la aplicación o sistema que se conecta para consumir los servicios web.

Tabla 5 Definición del Actor del Sistema

2.5.2 – Definición de los Casos de Uso del sistema

Los casos de uso son fragmentos de funcionalidades que agrupan los requisitos que debe cumplir el sistema. El modelo de casos de uso describe las funciones que realizan en el sistema y la interacción de estos con el actor. En el sistema que se modela se han definido los siguientes casos de uso:

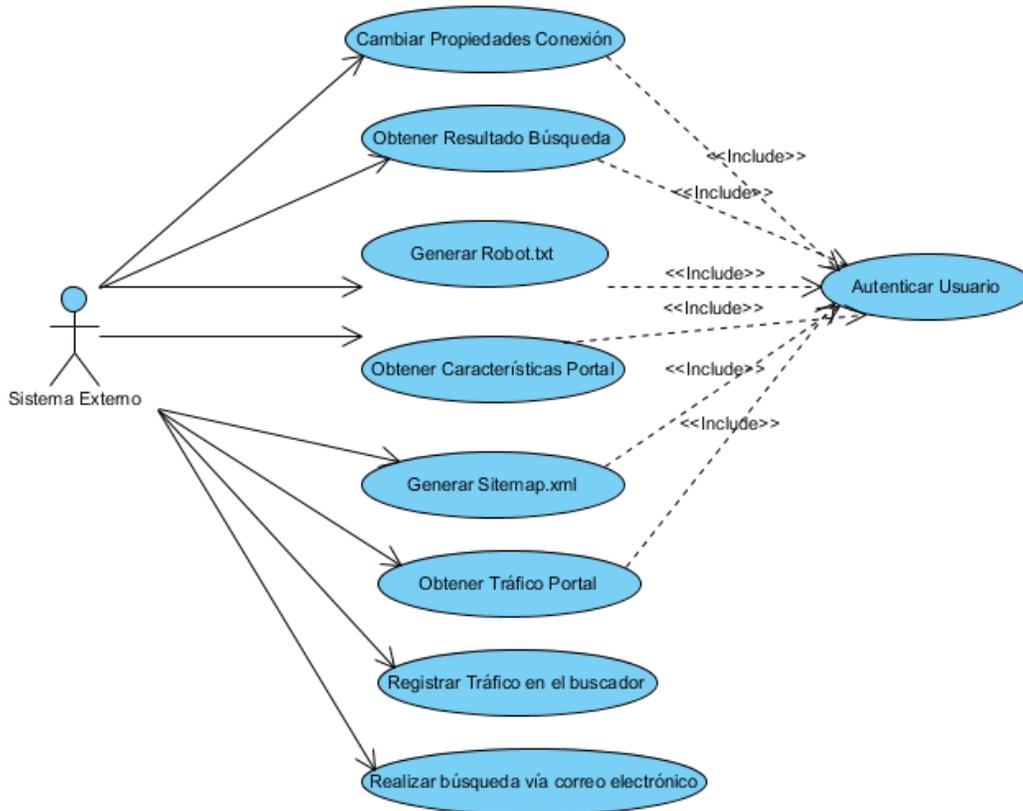


Ilustración 6 Diagrama de CU del sistema.

CU_1	Autenticar Usuario
Actor	Sistema externo
Descripción	Es el caso de uso donde se obtiene la clave de sesión, la cual es requerida para consumir las demás funcionalidades que brinda la capa de servicio web.
Referencia	RF1

Tabla 6 Definición CU Autenticar Usuario

CU_2	Cambiar Propiedades Conexión
Actor	Sistema externo
Descripción	Es el caso de uso que da la posibilidad de poder modificar una vez que haya obtenido la clave de sesión, los valores de las propiedades de la conexión, dígame: correo del usuario y contraseña.
Referencia	RF18

Tabla 7 Definición CU Cambiar Propiedades Conexión

CU_3	Obtener Resultado Búsqueda
Actor	Sistema externo
Descripción	Es el caso de uso que agrupa los servicios que brindan respuesta con los resultados de una solicitud con cierto criterio de búsqueda, entiéndase como la acción de realizar búsquedas con el motor usando la capa de servicios web.
Referencia	RF2, RF3,RF3.1, RF4, RF14, RF15, RF16, RF17, RF9,RF19

Tabla 8 Definición CU Obtener Resultado Búsqueda

CU_4	Obtener Características Portal
Actor	Sistema externo
Descripción	Es el caso de uso que agrupa los servicios que brindan respuesta con las características de los sitios web indexados en el buscador, entiéndase como características a factores como (enlaces entrantes, cantidad de páginas, Page Rank, palabras claves, etc.).
Referencia	RF5, RF6, RF7, RF8

Tabla 9 Definición CU Obtener Características Portal

CU_5	Generar Sitemap.xml
Actor	Sistema externo
Descripción	Es el caso de uso que da la posibilidad de generar el mapa de un determinado sitio con los datos recolectados por el motor.
Referencia	RF10

Tabla 10 Definición CU Generar Sitemap.xml

CU_6	Generar Robots.txt
Actor	Sistema externo
Descripción	Es el caso de uso que da la posibilidad de generar una propuesta de estándar de exclusión de robots.
Referencia	RF11

Tabla 11 Definición CU Generar Robots.txt

CU_7	Obtener Tráfico Portal
Actor	Sistema externo
Descripción	Es el caso de uso que da la posibilidad de devolver la información referente a la presencia de determinado sitio en el buscador, teniéndose en cuenta las consultas realizadas y en las que los contenidos de ese sitio fueron impresos y clickeados.
Referencia	RF12, RF13, RF13.1, RF13.2, RF13.3, RF13.4, RF13.5,

Tabla 12 Definición CU Obtener Tráfico Portal

CU_8	Registrar tráfico desde el buscador.
Actor	Sistema externo
Descripción	Es el caso de uso que da la posibilidad de registrar el tráfico desde el buscador de los recursos expuestos en los resultados de cada consulta.
Referencia	RF9

Tabla 13 Definición CU Registrar tráfico desde el buscador

CU_9	Realizar búsqueda vía correo electrónico.
Actor	Sistema externo
Descripción	Es el caso de uso que da la posibilidad de obtener los resultados de búsqueda vía correo electrónico.
Referencia	RF19

Tabla 14 Definición CU Realizar búsqueda vía e-mail

2.5.3 – Descripción de los Casos de Uso del sistema

A continuación se describirán los casos de usos del sistema identificados para la capa de servicios web del motor de búsqueda Orión.

Caso de Uso	Autenticar Usuario	
Actor	Sistema Externo	
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir el servicio para obtener la clave de sesión y poder ejecutar los demás servicios que se brindan. Para ello se introducen los datos necesarios para el procesamiento del servicio a través de la aplicación, el servicio solicitado es ejecutado y se brinda la respuesta correspondiente.	
Precondiciones		
Referencias	RF1	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El sistema externo realiza la petición del servicio para la autenticación, para ello introduce los datos necesarios (correo, contraseña).	2. Verifica la validez de los datos. 3. Crea sesión y devuelve la clave de esta.	
Flujo Alterno		
Acción del Actor	Respuesta del Sistema	

	2. Si los datos no son correctos se envía mensaje de error.
Poscondiciones	El usuario es autenticado y se crea una sesión.

Tabla 15 Descripción CU Autenticar Usuario

Caso de Uso	Obtener Resultado de Búsqueda
Actor	Sistema Externo
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir algunos de los servicios para obtener información del motor de búsqueda para determinado criterio. Para ello se introducen los datos necesarios para el procesamiento del servicio a través de la aplicación, como puede ser el término o frase a consultar así como la clave de sesión activa que le permita consumir dicho servicio. El servicio solicitado es ejecutado y se brinda la respuesta correspondiente.
Precondiciones	El sistema externo debe haber obtenido una clave de sesión.
Referencias	RF2, RF3, RF3.1, RF4, RF14, RF15, RF16, RF17
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El sistema externo realiza la petición de uno de los siguientes servicios, pasando la clave de sesión y el criterio de búsqueda: <ul style="list-style-type: none"> a. Devolver N resultado de búsquedas. b. Devolver valor de la posición de determinada página en los resultados. c. Devolver N resultado de páginas mejor posicionadas de determinado sitio. d. Devolver la cantidad de resultados encontrados para un criterio de búsqueda. e. Devolver tipo MIME de los resultados de búsqueda. f. Devolver fecha de modificación de los resultados para criterios de búsquedas. g. Devolver tiempo de respuesta de la consulta para un criterio de búsqueda. 	2. Verifica la validez de los datos. 3. Ejecuta la petición y envía la respuesta. 4. Registra los datos de la petición en la Base de Datos.

Flujo Alterno

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

	2. Si los datos introducidos no son correctos se envía mensaje de error: "Parámetros incorrectos"
Poscondiciones	Se obtiene respuesta del sistema.

Tabla 16 Descripción CU Obtener Resultado de Búsqueda

Para ver la descripción de los casos de uso restantes ver anexo 8.

2.6 – Conclusiones parciales

En el presente capítulo se presentó la propuesta del sistema a desarrollar. Se estructuró el modelo de dominio, describiendo los conceptos fundamentales que en él se exponen, pues no existen procesos de negocio bien definidos. Se listan los requerimientos funcionales y no funcionales que debe cumplir el software para satisfacer las necesidades del cliente. Además, se definieron los actores y casos de uso del sistema, así como la descripción textual de cada uno de ellos, lo que constituye un paso de gran importancia para las próximas etapas de desarrollo.

La descripción de los casos de uso propuesta servirá de base para comenzar con el siguiente flujo de trabajo, el análisis y diseño, teniendo presente las definiciones del sistema planteadas en este capítulo.

Capítulo 3: Análisis y Diseño del Sistema

3.1 – Introducción

El análisis y diseño es una de las etapas más importantes del desarrollo de un software, en ese momento se está en condición de traducir los casos de uso a una especificación que describa cómo implementar el sistema. Precisamente, lograr esto mediante la modelación del análisis y el diseño del sistema es el objetivo del presente capítulo. Además de ofrecer una descripción de los patrones de diseño empleados, algunos aspectos referentes a la seguridad del sistema y el manejo de los posibles errores que puedan ocurrir.

3.2 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que describe la realización de los casos de uso, y sirve como una abstracción del modelo de implementación. Está compuesto por clases del diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos. El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en resumen: cómo cumple el sistema sus objetivos [16]. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

3.3 Arquitectura del sistema

La arquitectura de software es el diseño de más alto nivel de la estructura de un sistema, es también denominada arquitectura lógica, que consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco.

Una arquitectura de software se selecciona y diseña con base en objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como el mantenimiento, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información.

Para el desarrollo de la capa de servicios web se decidió emplear una arquitectura de software de tres capas, separando la lógica de negocios de la lógica de diseño. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo se corrige al nivel requerido sin tener que revisar todo el código.

Además se enfoca el sistema a una arquitectura orientada a servicios donde la aplicación está formada por servicios débilmente acoplados y altamente interoperables. Para comunicarse entre sí, estos servicios se basan en una definición formal independiente de la plataforma subyacente y del lenguaje de programación. La definición de la interfaz encapsula las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo. Con esta arquitectura, se pretende que los componentes de software desarrollados sean muy reutilizables, ya que la interfaz se define siguiendo un estándar; así, un servicio en Java podría ser usado por una aplicación PHP.

3.3.1 Patrones de diseño empleados durante el desarrollo del sistema

La habilidad más importante en el análisis y el diseño orientado a objetos es asignar eficientemente las responsabilidades a los componentes del software, por ser una actividad que repercute en la solidez, capacidad de mantenimiento y en la reutilización de los componentes de software.

Los diseñadores expertos en orientación a objetos y otros diseñadores de software, van formando un amplio repertorio de principios generales y de expresiones que los guían al crear software. A algunos se les puede asignar el nombre de patrones, si se codifican en un formato estructurado que describe el problema y su solución y se les asigna un nombre.

En palabras más simples un patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas.

Durante el desarrollo del sistema se usaron los patrones **GRASP**²⁷ los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones [45].

A continuación se hace una breve descripción de los mismos.

Creador: Con este patrón se resuelve el problema de saber quién debería ser responsable de crear una nueva instancia de alguna clase, pues la solución sería asignarle la responsabilidad a una clase de crear instancias de otra, ya sea agregando, utilizando, conteniendo, registrando o creando objetos de la otra clase. Su uso permite un bajo acoplamiento y mejores oportunidades de reutilización.

²⁷ General Responsibility Assignment Software Patterns: Patrones de Software de Asignación de Responsabilidades

Este patrón se evidencia en la clase AdminSession que crea convenientemente la clase Session para el manejo de las sesiones de los usuarios.

Alta Cohesión: Con el uso de este patrón, se resuelve el problema de saber cómo mantener la complejidad dentro de límites manejables, pues es el encargado de asignar una responsabilidad de modo que la cohesión siga siendo alta. Además de que aumenta la capacidad de reutilización.

Este patrón se evidencia en cada una de las clases, pues cada una de ellas tiene una responsabilidad bien definida.

Se hace uso además del patrón **Singleton** que garantiza que una clase solo tenga una instancia y proporciona un punto de acceso global a tal instancia.

3.4 Diagrama de clases del diseño

Un diagrama de clases del diseño es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Las clases de diseño se especifican utilizando la sintaxis del lenguaje de programación elegido. A continuación se muestran los diagramas de clases del diseño para cada uno de los casos de uso.

El primer diagrama muestra la relación existente entre las clases que componen el Caso de Uso Autenticar Usuario. En él se muestra la relación entre las clases WS_AuthSession, la cual tiene como controladora la clase AuthSession, esta última por su parte depende de las clases AdminClient y AdminSession y se apoya en la clase ClientDao para la conexión a la base de datos.

La segunda ilustración muestra la relación entre clases pertenecientes al Caso de Uso Realizar Búsquedas. En el diagrama se aprecia la relación de la clase WS_Search con su clase controladora Search la cual se apoya en las clases sesión para validar el correcto acceso a la aplicación y en la clase SearchDAO para el acceso a la base de datos.

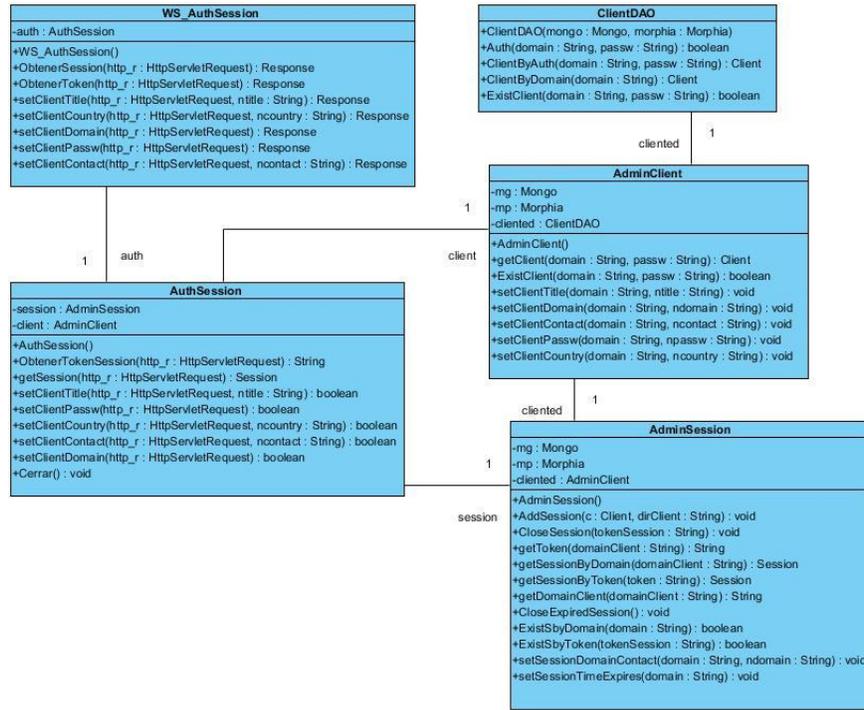


Ilustración 7 Diagrama de Clases del Diseño CU Autenticar Usuario

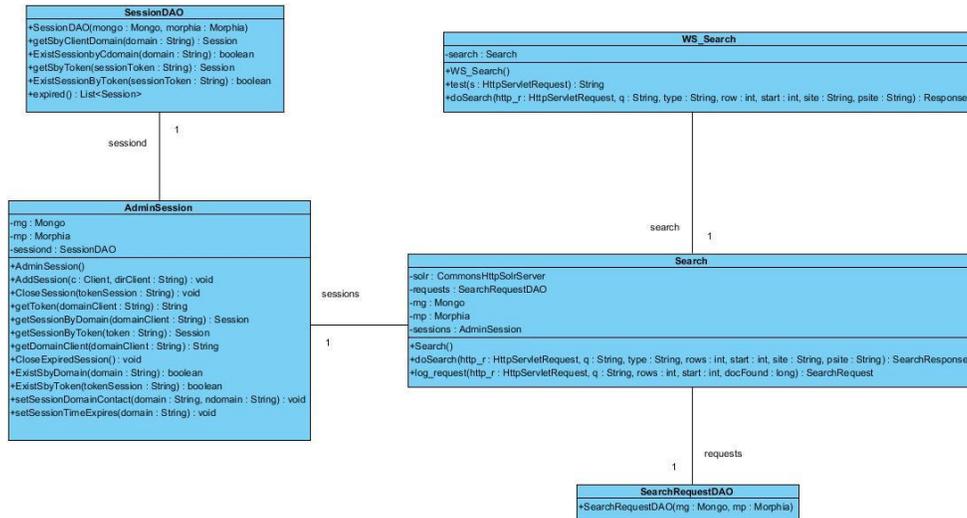


Ilustración 8 Diagrama de Clases del Diseño CU Realizar Búsquedas

Para una mejor interrelación entre clases y sistemas se hace uso de la biblioteca **solrj-3.4.0** y del driver **mongo** para Java, de ellos se utilizan sus funcionalidades aportándole al sistema un mejor funcionamiento e integración con el núcleo del motor de búsqueda Orión.

Solrj-3.4.0 SolrJ es la interfaz cliente Java de Solr que le aísla de los detalles de analizar y enviar mensajes de ida y vuelta entre una aplicación y Solr, por defecto se comunica usando un formato binario rápido en lugar de XML. Permite trabajar con los objetos como SolrQuery, QueryResponse y SolrDocument de Solr. SolrJ es una parte fundamental del proyecto Solr, y por lo general, se actualizará tan pronto como las nuevas características se añaden a Solr en el lado del servidor.

Para una mejor comprensión de la interacción de las clases del sistema con la biblioteca se puede consultar la tabla del Anexo 4.

3.4.1 Diagramas de interacción del Diseño

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Están formados por un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

A continuación se muestran los diagramas de secuencia correspondientes a algunos de los escenarios reflejados en los caso de uso. Estos a diferencia de los de colaboración, destacan el orden temporal de los mensajes.

El primer diagrama corresponde al caso de uso Realizar Búsquedas donde se aprecia las secuencias que ocurren en el momento de solicitarse una petición para localizar información.

Al realizarse una búsqueda la clase interfaz WS_Search parsea la url enviada, apoyándose en las clases del frameworks `jax-rs`, enviando posteriormente los datos a la clase controladora para su procesamiento. La clase controladora Search verifica mediante la clase AdminSession la existencia de una sesión para un usuario y permite o deniega el acceso al servicio web. Si es denegado el acceso muestra un mensaje, en caso de ser autorizado se consulta el índice del buscador, se registra en la base de datos la búsqueda y devuelve posteriormente los resultados, parseándolos nuevamente por las clases del frameworks `jax-rs` a un formato de comunicación, json o xml.

El segundo diagrama corresponde al caso de uso Autenticar Usuario donde se muestra la secuencia de pasos que ocurren al momento de acceder a los servicios web.

El sistema externo debe enviar un usuario y contraseña. Al realizarse una solicitud la clase WS_AuthSession, apoyándose en las clases del frameworks Jersey, procesará los datos y llamará los métodos de la clase AuthSession. Las funciones de esta clase ejecutarán las consultas a la base de datos mediante las diferentes clases que implementa, y verificará la existencia del usuario. En caso de no existir el cliente el sistema enviará un error, en otro caso devolverá el token de sesión el cual será manejado por la clase AuthSession para crear seguidamente la sesión en la base de datos, permitiendo al usuario el acceso a los servicios web de la capa.

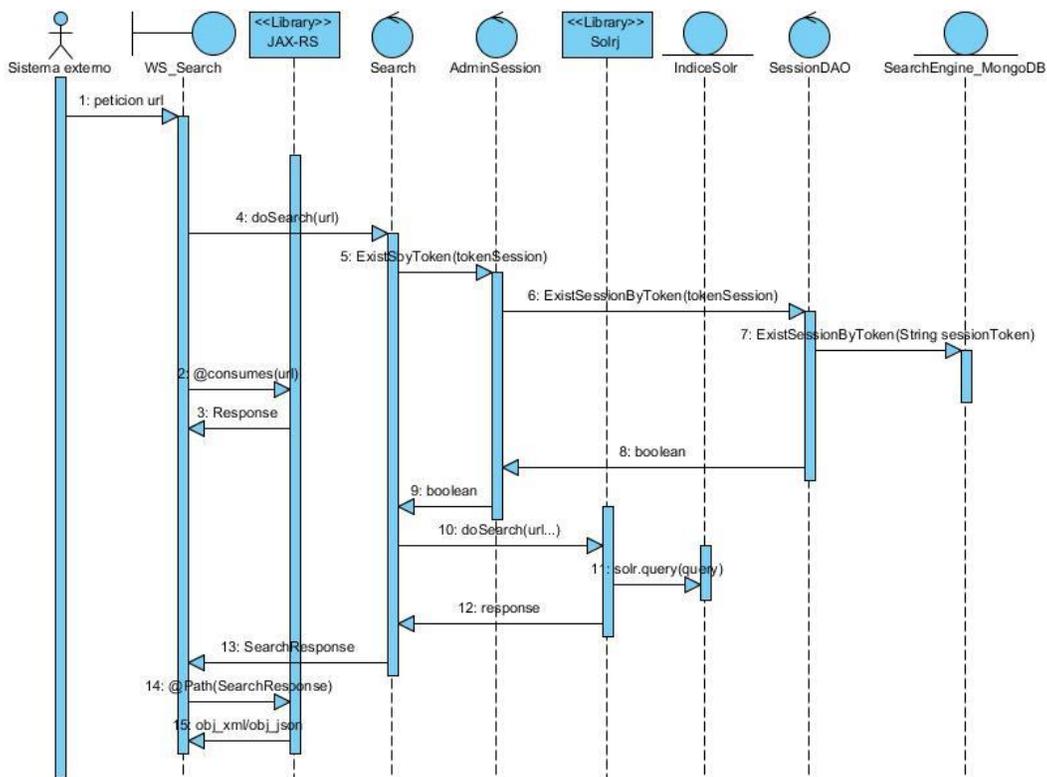


Ilustración 9 Diagrama de Secuencia CU Realizar Búsquedas

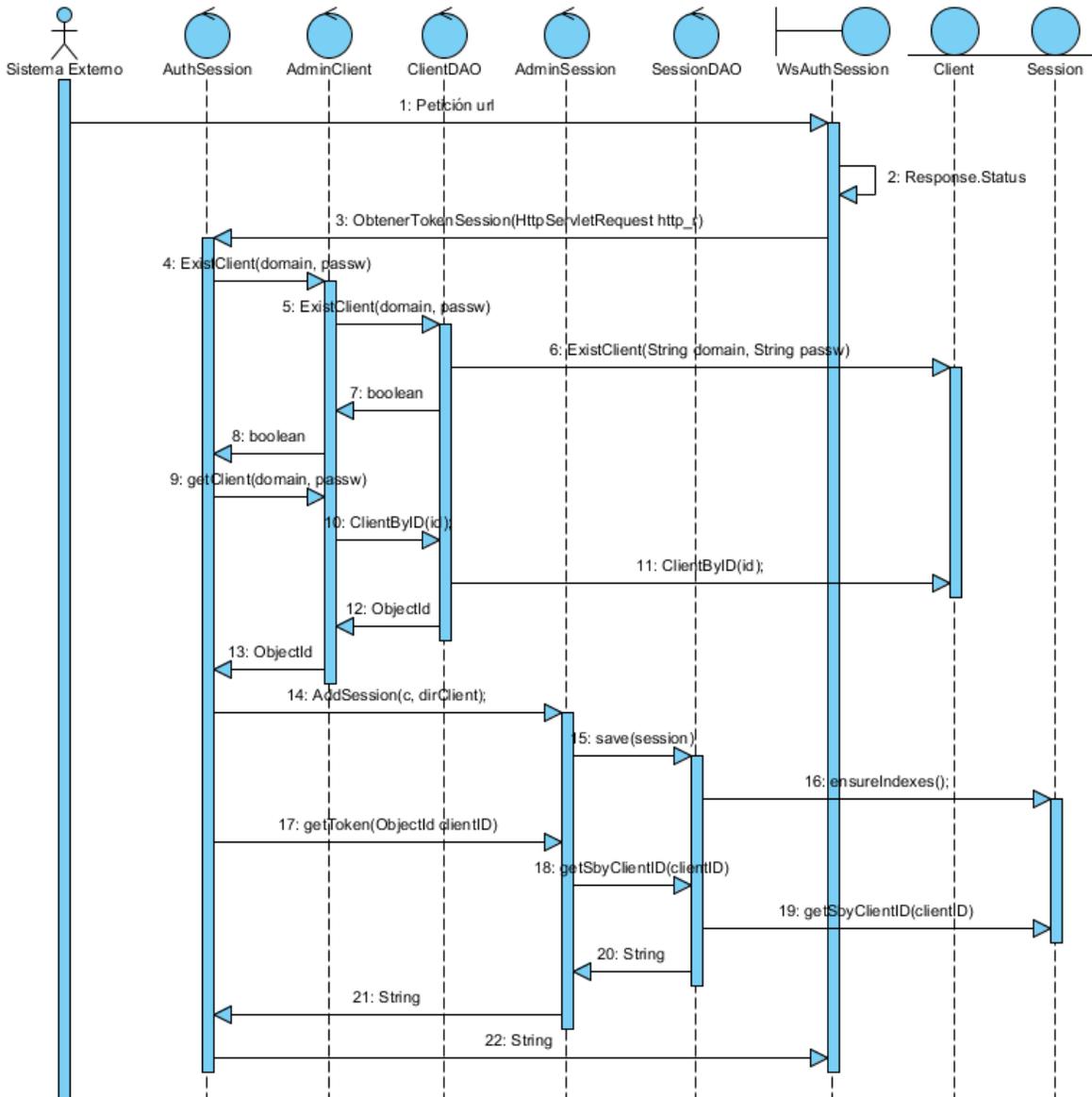


Ilustración 10 Diagrama de secuencia CU Autenticar Usuario.

3.5 Diseño de la Base de Datos

Las bases de datos desempeñan un papel crucial en casi todas las áreas de desarrollo de software. Su diseño es una de las tareas más importantes en la construcción de un sistema que requiera manejar la persistencia de la información. El sistema que se propone hace uso de una base de datos,

pues la información a persistir solo será la obtenida del tráfico entre las aplicaciones externas y la capa de servicio web en cuanto a estadísticas se refiere (consultas, direcciones ip, resultados más relevantes, etcétera.) así como los usuarios que tendrán acceso a los servicios web expuestos en la capa.

MongoDB es una base de datos que se basa en colecciones. Cuando se dice que se *basa en colecciones*, significa que los datos se agrupan en conjuntos llamados "colecciones". Cada colección tiene un nombre único en la base de datos y puede contener un número ilimitado de documentos. Una colección es análoga a una tabla, excepto que no tienen un esquema definido.

La colección de datos Session almacenará los datos correspondientes a la sesión activa en ese momento, datos como el token, cliente y el inicio y fin de la expiración de la sesión.

La colección Client almacenará los datos correspondientes a los sistemas clientes de la capa de servicios web, entre estos datos se pueden encontrar el nombre, correo, país, sitio y contraseña del cliente.

La colección SearchRequest almacenará los datos correspondientes a los registros generado por las búsquedas realizadas en Orión, entre estos datos se pueden encontrar, el ip de origen, fecha, criterio de búsqueda, sistema operativo y navegador web.

La colección TrafficLog almacenará los datos correspondientes a los registros de tráfico realizado sobre la capa de servicios web.

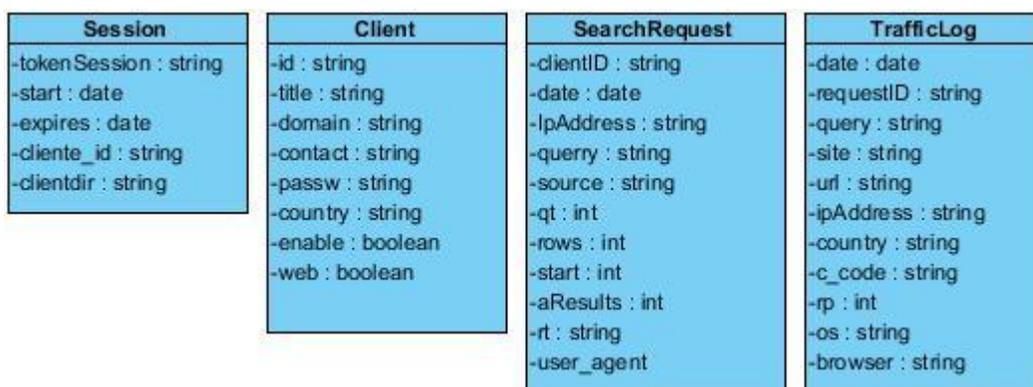


Ilustración 11 Modelo Físico de Datos para MongoDB

3.6 Tratamiento de Errores

Durante el desarrollo del sistema, se han definido una serie de acciones que pueden provocar un mal funcionamiento, para cada una de ellas se definió un tratamiento específico, para asegurar una respuesta adecuada que garantice cierto grado de estabilidad.

Los mensajes de error llegarán a los usuarios en un lenguaje común, con el objetivo de facilitar su comprensión, pero sin revelar información propia del sistema que pueda atentar contra su seguridad.

En la siguiente tabla se describen las posibles acciones, que pueden producir errores.

Error	Causa	Respuesta del sistema
Identidad incorrecta	Los datos (correo electrónico y contraseña) que fueron introducidos para obtener un token de sesión no coinciden con los de ningún usuario registrado.	Notifica al usuario que los datos incluidos son incorrectos.
Sesión no existente	El usuario intenta consumir un servicio web pasando una clave de sesión no válida.	Notifica al usuario que la clave de sesión pasada no existe.
No autorizado	El usuario solicita un servicio web de tráfico a determinado portal del cual no es su webmaster.	Notifica al usuario que no está autorizado a consumir dicho servicio para ese portal.
Conexiones duplicadas	El usuario intenta obtener una clave de sesión con propiedades de conexión que ya se encuentran en uso por otra sesión.	Notifica al usuario que esas propiedades de conexión ya se encuentran en uso por otra sesión.

Tabla 17 Respuestas del sistema ante errores

3.7 Aspectos de Seguridad en el sistema.

En aras de garantizar la seguridad del sistema, y pretendiendo lograr la confiabilidad, integridad y disponibilidad de los datos, se han tomado una serie de medidas. Primeramente se trabajó con las sesiones de usuarios.

Una sesión es creada cuando se desea consumir alguna de las funcionalidades del servicio web. Para ello, previamente el usuario debe obtener la clave de sesión a través de la funcionalidad GetToken(), donde se autentica el usuario con los parámetros de conexión entrados (correo y contraseña) y es creada la sesión. Esta clave va a ser única y contará de 28 caracteres combinando

números y letras. Ella se generará automáticamente y tendrá una vigencia igual al tiempo que el usuario esté usando el servicio web. Después de esto, solo se comprobará su identidad a través de su clave de sesión, evitando que datos sensibles como el usuario y la contraseña viajen por el canal de comunicación frecuentemente.

Las sesiones van a contar con un tiempo de expiración, marcado por la inactividad del usuario, con el objetivo de prevenir que una aplicación que se haya quedado abierta, no sea utilizada por personas no autorizadas. El sistema periódicamente chequea las sesiones y cierra aquellas cuyo tiempo inactivo sea igual al de expiración.

Por otra parte, se ha tenido en cuenta utilizar HTTPS, que es la versión segura del protocolo HTTP y viene dada por el protocolo de seguridad SSL. Se utilizó la opción de certificado autofirmado, permitiendo asegurar la transferencia de la información de forma segura, ya que el envío de datos en los servicios web normalmente se hace mediante REST/HTTP, donde los mensajes viajan en texto plano por la red.

3.8 Parámetros de consulta y acceso

Durante el diseño se definieron una serie de parámetros mediante los que se podrá consultar y configurar la capa de servicios web.

Un ejemplo de cómo estaría construida la petición sería de la siguiente forma.

<https://example.cu/ws-api/api/search?q=universidad>

Algunos de los principales campos presentes en el documento de respuesta de este servicio web se relacionan y describen en la siguiente tabla:

Parámetros	Descripción
q	Consulta de la búsqueda. Este parámetro es requerido.
alt	Indica el formato del documento de salida (json,xml).
type	Indica el formato de documentos a buscar.
rows	Indica el número de resultados a mostrar.
start	Indica el número de resultados por el cual comenzar a mostrar.
site	Indica el sitio en el cual se realizará la búsqueda.

Tabla 18 Parámetros de consulta para la búsqueda.

3.9 Modelo de Despliegue

El Modelo de Despliegue provee un modelado detallado de la forma en la que los componentes se desplegarán. Detalla las capacidades de red, las especificaciones del servidor y otra información relacionada al despliegue del sistema propuesto.

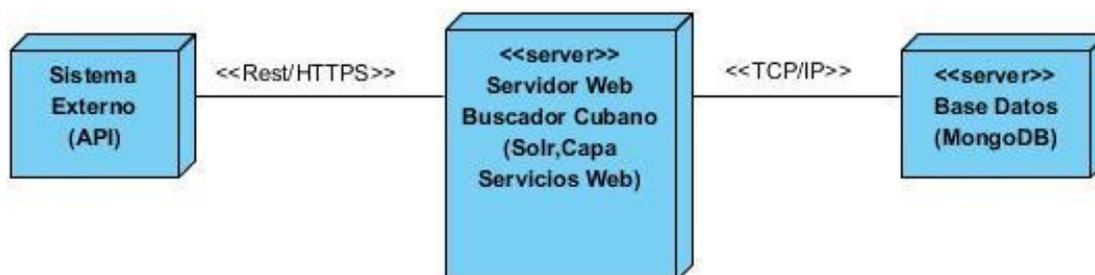


Ilustración 12 Diagrama de Despliegue

3.10 Conclusiones parciales

Este capítulo abordó la realización de los diagramas de clases de análisis y diseño del sistema, así como los diagramas de interacción, correspondientes a cada caso de uso, lo cual constituye una importante entrada para el próximo flujo de trabajo: la implementación. Se realizó además una descripción de los patrones de diseño usados. Se llevaron a cabo acciones para manejar los posibles errores que puedan ocurrir, así como la descripción de algunos aspectos que se tuvieron en cuenta para garantizar la seguridad del sistema.

Se puede concluir que a través de la obtención de los artefactos generados por este flujo de trabajo se obtuvo una definición de los elementos que se tomarán en cuenta en la implementación del sistema. Con la obtención del modelo de datos se define cuáles son los elementos que se almacenarán para su utilización posteriormente. Se definieron estrategias para el tratamiento de requisitos no funcionales del sistema. Se obtuvo una visión de los posibles parámetros de consulta y configuración que brindará el sistema para la prestación de los servicios web y su posible despliegue en un determinado ambiente.

Capítulo 4: Implementación y prueba del sistema

4.1 – Introducción

El periodo de implementación y pruebas es el momento en el que luego de haber realizado el diseño del sistema, se comienza la elaboración de los elementos de código que convertirán lo planificado en un producto real. Posteriormente este sistema debe ser validado como correcto en cuanto a los resultados que se esperan obtener de él. En este proceso se generan todos los artefactos de esta etapa de desarrollo y se valida el correcto funcionamiento de la aplicación.

4.2 – Modelo de implementación del sistema

En el modelo de implementación se describe cómo se transforman en artefactos los elementos descritos en el diseño. El modelo de implementación está constituido por el diagrama de componentes, por algunos elementos de código del sistema así como alguna de las principales pantallas de la aplicación.

4.2.1 –Diagrama de componentes

Un diagrama de componentes es la representación de los componentes del sistema, interfaces y relaciones. Un componente es la implementación física de un conjunto de otros elementos lógicos, como clases y colaboraciones.

La siguiente ilustración muestra el diagrama de componentes de la capa de servicios web. Se puede apreciar la estructura de los paquetes y la distribución de los componentes dentro de estos. De forma general cada componente del paquete Service tendrá a la par un componente dentro del paquete Control que se relacionará con su respectivo componente dentro del paquete Model, todos ellos a su vez utilizarán los componentes del paquete Library en dependencia de sus necesidades.

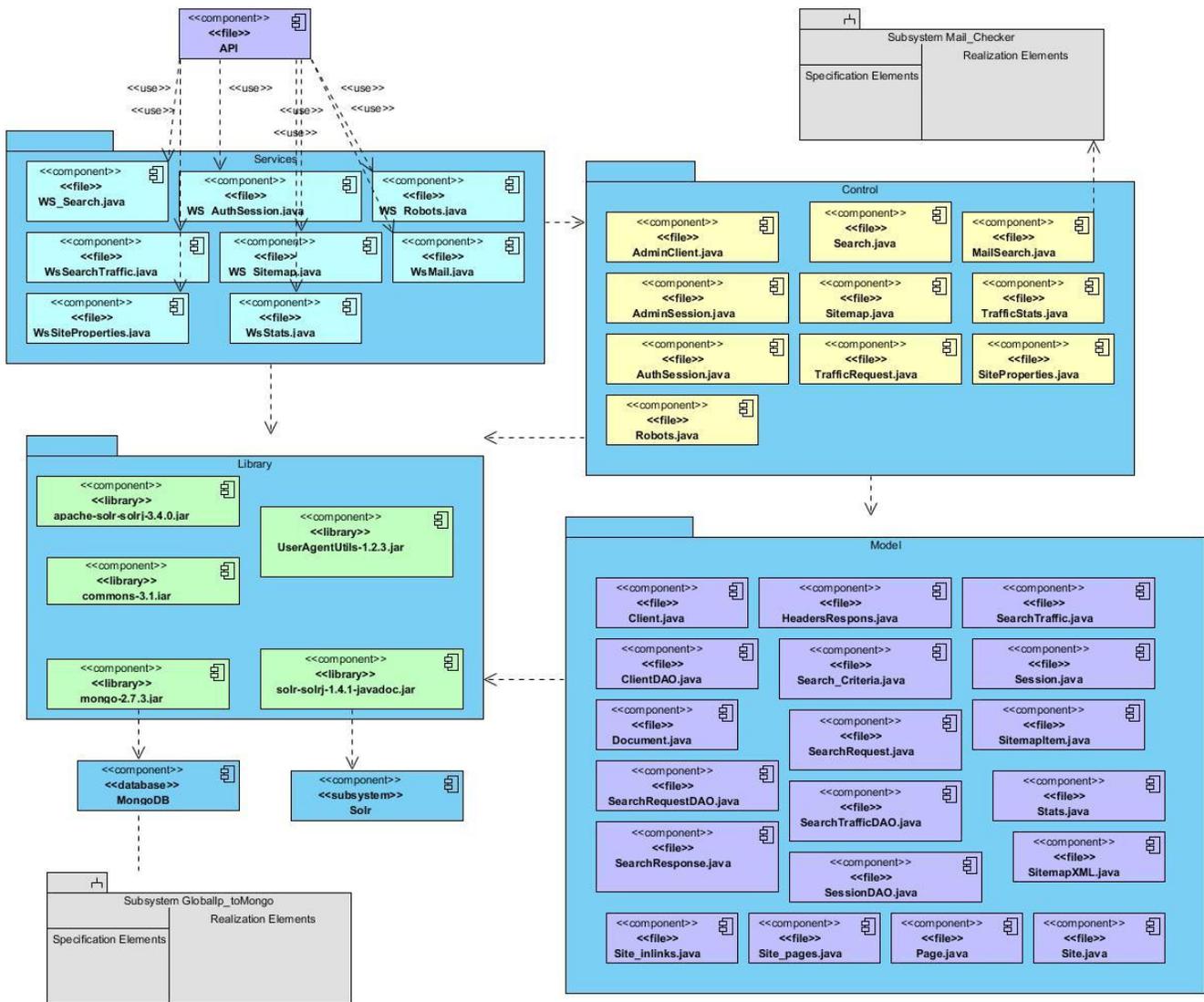


Ilustración 13 Diagrama de componentes de la capa de servicios web.

4.2.2 – Componentes Complementarios

Durante el proceso de implementación de la capa de servicios web se desarrollaron de manera general tres artefactos más de implementación. Estos fueron la aplicación web (ver anexo 7) que incluye el API para PHP de la capa de servicios, que valida las funcionalidades del sistema. Además se desarrolló un script, incluido en el diagrama anterior, que mantiene la vigilancia sobre la cuenta de

correo del buscador para atender las solicitudes de los usuarios que utilicen el servicio de búsqueda vía correo electrónico. Para el registro de las estadísticas fue necesaria la implementación de un sistema, incluido en el diagrama previo, que actualizará la base de datos constantemente con los nuevos dominios que se van creando en el mundo, estos datos se necesitan para el registro de las visitas por países.

4.2.3 – Estándares de código

Un estilo de programación, también llamado estándares de código o convención de código, es un término que describe convenciones para escribir código fuente en cierto lenguaje de programación. El estilo de programación es dependiente del lenguaje de programación que se haya elegido.

Para lograr la legibilidad de un programa es importante considerar aspectos tales como el nombre de los identificadores, escribir el código con cierta alineación y líneas en blanco en lugares apropiados así como realizar una buena documentación. [46]

Para la implementación de la capa de servicios web, como se definió anteriormente, se seleccionó el lenguaje de programación Java. La propuesta de solución cumplirá con el documento “Convenciones del Código Java” establecido por sus creadores *Sun Microsystems*. [46]

4.3 – Modelo de prueba

El modelo de pruebas describe cómo se prueban los componentes ejecutables en el modelo de implementación con pruebas de integración y de sistema. Es una colección de casos de prueba, procedimientos de prueba y componentes de prueba.

Para la validación del software se decidió la realización de las pruebas de caja negra, estas demuestran que las funciones de un software son completamente operativas y validan el correcto funcionamiento de los requisitos funcionales del software.

4.3.1 –Plan de prueba

El plan de pruebas señala el enfoque, los recursos y el escenario de prueba, así como los elementos a probar, las características, las actividades de prueba, el personal responsable y los riesgos.

Definición de precondiciones

Para un comienzo exitoso de las pruebas se decidió establecer un grupo de precondiciones que aseguren la calidad de las pruebas. Entre las precondiciones se debe asegurar el espacio físico incluyendo dentro el equipamiento tecnológico necesario para la realización de las pruebas, así como una buena definición de los juegos de datos a utilizar.

Roles y responsabilidades

Para la realización de las pruebas de aceptación es necesaria la participación de ambas partes del negocio, el cliente y miembros del equipo de desarrollo, probador y analista.

El incumplimiento de este elemento puede traer malos entendimientos, dudas sin respuestas inmediatas o mala calidad en la realización de las pruebas.

Requisitos de hardware y software

Los requisitos mínimos de hardware y software son necesarios para la realización exitosa de cualquiera de las pruebas. Se realizó un análisis de los requerimientos del sistema y su disponibilidad para ser usado así como las herramientas de software necesarias para la realización de las pruebas.

Descripción de los tipos de pruebas

Para la realización de las pruebas al sistema, se definieron una serie de pruebas fundamentales para la validación de los servicios web.

➤ Pruebas funcionales

La prueba de funcionalidad se enfocará en los requerimientos para verificar que se corresponden directamente a casos de usos.

Técnica

Ejecutar cada proceso o función usando datos válidos y no válidos, para verificar lo siguiente:

- ¿Se obtienen los resultados esperados cuando se usan datos válidos?
- ¿Cuándo se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados?

Criterio de aceptación

Todas las pruebas planificadas se realizaron. Todos los defectos encontrados han sido debidamente identificados.

➤ **Prueba de seguridad y control de acceso**

La prueba de seguridad y control de Acceso se enfocará a la seguridad en el ámbito de la aplicación, asegurando que los usuarios solo accedan cuando estén registrados en el sistema.

Técnica

Crear pruebas para un token de seguridad dado y verificar que las funciones o datos adicionales están correctamente disponibles o sean denegados.

Criterio de aceptación

Para el token de seguridad, las funciones y datos apropiados de la aplicación están disponibles y todas las operaciones funcionan como se espera y se ejecutan las pruebas de funcionalidad de la aplicación.

➤ **Pruebas de rendimiento de software**

La prueba de rendimiento se enfoca en la capacidad de recibir peticiones mediante la utilización de alguna herramienta, verificando cuantas peticiones pueda sostener el sistema sin que este se vea afectado, así como la velocidad de respuesta de este.

Técnica

Las pruebas serán realizadas utilizando Jmeter que es una aplicación de escritorio y de código abierto, diseñada para realizar pruebas de carga y medir el comportamiento de las pruebas y medir el rendimiento de un sistema.

4.3.2 –Pruebas

Las pruebas tienen como fin, ratificar si el sistema cumple con los requisitos definidos por el cliente y permite al usuario determinar la aceptación del sistema. Por este motivo, las pruebas son realizadas por el cliente y el desarrollador participa de forma pasiva facilitando la comunicación del cliente con el

sistema. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. [47].

Las pruebas se le realizaron a cada uno de los casos de uso del sistema y en sentido general a cada uno de los requisitos de la capa de servicios web.

Las siguientes tablas muestran el caso de prueba para el caso de uso Obtener Resultado de Búsqueda.

Caso de Uso: CU_3 Obtener Resultado Búsqueda
Requisitos: RF2, RF3, RF4, RF14, RF15, RF16, RF17
Condición: Debe tener el token de seguridad asignado al existir el usuario en la base de datos para poder ejecutarse la búsqueda.

Tabla 19 Caso de Prueba CU Obtener Resultado Búsqueda

Descripción de variables.

Parámetro	Valor Nulo	Descripción
q	No	Consulta de la búsqueda
alt	Sí	Indica el formato del documento de salida (json,xml).
type	Sí	Indica el formato de documentos a buscar.
rows	Sí	Indica el número de resultados a mostrar.
start	Sí	Indica el número de resultados por el cual comenzar a mostrar.
site	Sí	Indica el sitio en el cual se realizará la búsqueda.

Tabla 20 Descripción de variables de prueba CU Obtener Resultado Búsqueda

Escenario	Descripción	q	alt	type	rows	start	site	Respuesta del Sistema	Flujo central
EC 1 Búsqueda Simple	Realizar búsqueda de un término.	cuba	nulo	nulo	nulo	nulo	nulo	Documento xml con la estructura definida previamente que muestra los resultados para el criterio de búsqueda "cuba".	Enviar la url por post con el parámetro q=cuba

EC 1.1 Búsqueda sin criterio.	Realizar búsqueda dejando en blanco el parámetro de búsqueda	nulo	nulo	nulo	nulo	nulo	nulo	Muestra un error denegando acceso.	Enviar la url por post con el parámetro q vacío.
EC 2 Búsqueda personalizada.	Búsqueda avanzada con varios parámetros	cuba	json	pdf	12	10	feu.uci .cu	Documento json con la estructura definida previamente que muestra los resultados para el criterio de búsqueda "cuba" y cumpliendo con los parámetros definidos para la misma.	Enviar la url por post con los parámetro q=cuba,alt=json,type=pdf,rows=12,start=10,site=feu.uci.cu

Tabla 21 Matriz de Datos CU Obtener Resultado Búsqueda

4.4 –Resultado de las pruebas

La implementación de las pruebas funcionales de los casos de uso muestra una visión de la calidad del software. En una primera iteración, se obtuvo como resultado del análisis de los escenarios de pruebas, un conjunto de conformidades y de no conformidades correspondientes a los servicios web implementados hasta el momento de la ejecución de las pruebas. Las no conformidades fueron corregidas posteriormente, dándole paso a la segunda iteración de pruebas, en las cuales se evaluaron nuevamente, obteniendo resultados satisfactorios.

El análisis de los nuevos casos de pruebas correspondientes a los nuevos servicios web implementados, también arrojaron conformidades y no conformidades.

En una tercera iteración de las pruebas se verificó la corrección de las no conformidades arrojadas en la segunda iteración y se verificó nuevamente los requisitos correspondientes a todos los casos de uso del sistema. Se validó en sentido general el correcto funcionamiento de la capa de servicios y el cumplimiento de los requisitos funcionales del software.

Para probar la escalabilidad del sistema es necesario conocer acerca de la carga que puede soportar. Para ello se implementaron las pruebas de rendimiento de software, las cuales se apoyaron en el uso

de la herramienta Jmeter, que ofrece datos importantes acerca del stress que puede soportar el sistema. Primeramente se probó el sistema con un total de **12951** peticiones enviadas en **120** hilos de ejecución concurrentes, arrojando un **0,4%** de error al atender dichas peticiones a los servicios. En una segunda prueba se probaron un total de **28 121** peticiones en **600** hilos de ejecución concurrentes, arrojando como error un **1.72%**.

El porcentaje de error estuvo originado por peticiones no atendidas, que no fueron responsabilidad de la capa de servicios, pues fueron rechazadas al detener la herramienta de pruebas.

# Muestras	% Error	Rendimiento	Kb/sec
28121	1,72%	44,4/sec	221,5
28121	1,72%	44,4/sec	221,5

Ilustración 14 Resultados de pruebas con Jmeter

Para una total verificación de las funcionalidades de la capa de servicios web, se implementó una aplicación que sirviera para probar cada una de ellas. Esta aplicación confirmó el correcto funcionamiento de la capa. También servirá de manual de usuario para los usuarios interesados en utilizar la capa de servicios web.

4.5 – Conclusiones parciales

La implementación de la capa de servicios web en el lenguaje Java, significó una ventaja para la comunicación de la capa de servicios web con el núcleo del motor de búsqueda. El intercambio de los datos se realiza a nivel de codificación binaria lo que le aporta eficiencia y agilidad en las consultas al índice.

- ✓ La capa de servicios puede soportar una gran concurrencia de usuarios utilizando y solicitando concurrentemente información al sistema.
- ✓ El sistema cumple con todos los requerimientos establecidos para su implantación y cumple con su objetivo de posibilitar la interoperabilidad de aplicaciones con el motor de búsqueda.

Conclusiones

- La sistematización del estudio del estado del arte de las tecnologías para el desarrollo de servicios web en buscadores permitió implementar una aplicación actualizada en tecnología, así como la selección adecuada de las herramientas para su desarrollo.
- La implementación de la capa de servicios web le aporta un valor agregado al motor de búsqueda Orión.
- La aplicación web de pruebas implementada que utiliza los servicios de la capa, permitió validar el correcto funcionamiento del sistema y permite demostrar la interoperabilidad de una aplicación informática con el buscador Orión.
- Se logró establecer la vía de comunicación para permitir la interoperabilidad de las aplicaciones informáticas con el buscador Orión.

Recomendaciones

Con el objetivo de mejorar y seguir perfeccionando la capa de servicios, los autores del presente trabajo de diploma recomiendan al equipo de desarrollo de Orión:

- ✓ La implementación de nuevas funcionalidades a la capa de servicios web.
- ✓ La implementación de nuevas APIs para los posibles lenguajes de programación con las que puedan ser implementadas las aplicaciones cubanas.
- ✓ Desplegar la capa de servicios web en el entorno real del buscador Orión.

Referencias Bibliográficas

1. **Leiner, Barry M y Otros.** Una breve historia de Internet. [En línea] Noviembre de 2001. <http://www.ati.es/DOCS/internet/histint/histint1.html>.
2. **W3C.** World Wide Web. [En línea] 09 de Enero de 2008. [Citado el: 10 de Octubre de 2011.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
3. **Fernández Calvo, Rafael.** Glosario básico inglés-español para usuarios de Internet. [En línea] Asociación de Técnicos de Informática, Julio de 2001. http://www.ati.es/novatica/glosario/glosario_internet.html#eXtensible%20Markup%20Language.
4. **Centro Cubano de Información de Red, CUBANIC.** Estadísticas. [En línea] 2011. <http://www.nic.cu/estadisticas.php>.
5. **Morales Machuca, MSc.Carlos Andrés.** *Estado Del Arte: Servicios Web*. Colombia : Universidad Nacional de Colombia, 2008,1- 8.
6. **W3C.** HTTP - Hypertext Transfer Protocol Overview. [En línea] 2011. [Citado el: 20 de Noviembre de 2011.] <http://www.w3.org/Protocols/>.
7. **Klensin, J.** Simple Mail Transfer Protocol. [En línea] The Internet Engineering Task Force. <http://www.ietf.org/rfc/rfc2821.txt>.
8. **Fielding, Roy T.** Architectural Styles and the Design of Network-based Software Architectures. [En línea] PhD Thesis.2000. <http://roy.gbiv.com/pubs/dissertation/top.htm>.
9. **SoftZone.** Software Zone. Diccionario Informático. [En línea] 2007. [Citado el: 18 de Octubre de 2011.] <http://software.adslzone.net/glosario/s-t-y-u/>.
10. **W3C.** SOAP Version 1.2 Pa 1: Messanging Framework. [En línea] W3C Consortium., Abril de 2007. <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/#encapsulation>.
11. **LeBlanc, Jonathan.** Programing Social Aplications.O´Reilly, 2011, 25-34.
12. **Webber, Jim.** Rest in Practice. *Hypermedia and System Architecture*. O´Reilly, 2010,12-15.
13. **Tidwell, Doug, Snell, James y Ku Ichenko.** *Programming Web Service with SOAP*. O'Really, 2002,110-114.
14. **Microsoft.** Directorios De Servicios Web XML. [En línea] 2011. [Citado el: 16 de Noviembre de 2011.] [http://msdn.microsoft.com/es-es/library/7e29kfs9\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/7e29kfs9(v=vs.80).aspx).

15. **OASIS**. Reference Model for Service Oriented Architecture 1.0. [En línea] 2006. [Citado el: 16 de Octubre de 2011.] <http://docs.oasis-open.org/soa-rm/v1.0/>.
16. **Estrada Melgarejo, Yadira y Jorge Martínez, Yasiel**. *Servicios Web para la creación de aplicaciones SEO*. s.l., Cuba : Universidad de las Ciencias Informáticas, 2009.
17. **Meng Li, Zhao, Junfeng, Wang, Lijie, Cai, Sibó y Xie, Bing**. *CoWS: An Internet-Enriched and Quality-Aware Web Services Search Engine*. Washington, DC : 2011 IEEE International Conference on Web Services, 2011.
18. **SEOMoz**. Social Media Marketing Guide. [En línea] Agosto de 2011. [Citado el: 14 de Noviembre de 2011.] <http://www.seomoz.org/article/social-media-marketing-tactics>.
19. **Yahoo! Inc**. The History of Yahoo! - How It All Started. [En línea] 2005. [Citado el: 20 de Noviembre de 2011.] <http://docs.yahoo.com/info/misc/history.html>.
20. **Yahoo! Inc**. Upcoming API Documentation-Version2.0. [En línea] 2009. [Citado el: 19 de Noviembre de 2011.] <http://upcoming.yahoo.com/services/api/>.
21. **StatCounter**. Top 5 Search Engines from Oct 2010 to Oct 2011. *StatCounter Global Stats*. [En línea] 2011. [Citado el: 20 de Noviembre de 2011.] http://gs.statcounter.com/#search_engine-ww-monthly-201010-201110-bar.
22. **Google Inc**. Developer's Guide.Google Code. [En línea] 2010. [Citado el: 15 de Noviembre de 2011.] <http://code.google.com/intl/es-ES/apis/websearch/docs/>.
23. **Google Inc**. JSON/Atom Custom Search API.Google Code. [En línea] 2011. [Citado el: 16 de Noviembre de 2011.] <http://code.google.com/intl/es-ES/apis/customsearch/v1/overview.html>.
24. *2x3 : el primer buscador cubano en Internet*. **Santovenia Díaz, Javier y Cañedo Andalia, Rubén**. 5, 2007, Vol. 15.
25. **The Apache Software Foundation**. Apache Axis2/Java Version 1.6.2 Documentation Index. [En línea] 2012. [Citado el: 25 de Noviembre de 2011.] <http://axis.apache.org/axis2/java/core/docs/contents.html>.
26. **Zend Technologies Ltd**. Zend Framework Manual. [En línea] 2011. [Citado el: 19 de Octubre de 2011.] <http://framework.zend.com/manual/en/introduction.overview.html>.
27. **Burke, Bill**. *RESTful Java with JAX-RS*. s.l. : O'Reilly, 2010. 978-0-596-15804-0,179-185.
28. **The PHP Group**. ¿Qué es PHP? PHP Manual. [En línea] 2011. [Citado el: 19 de Noviembre de 2011.] <http://www.php.net/manual/es/intro-what-is.php>.

29. **Carlson, Lucas y Richardson, Leonard.** *Ruby Cookbook*. s.l. : O'Reilly, 2006. 0-596-52369-6, xix-xx.
30. **Team, Java.** Conozca más sobre la tecnología Java. [En línea] 2009. [Citado el: 19 de Noviembre de 2011.] <http://www.java.com/es/about/>.
31. **Sommerville, Ian.** *Ingeniería de Software*. s.l. : Pearson Educación, 2005.
32. **Peñalver Romero, Gladys M.** *Metodología ágil para proyectos de software libre*. Habana : Universidad de las Ciencias Informáticas, 2008. MA-GMPR-UR2.
33. **IBM.** Rational Unified Process. [En línea] IBM, 2011. [Citado el: 13 de Noviembre de 2011.] <http://www-01.ibm.com/software/awdtools/rup/>.
34. **Zend Technologies Ltd.** Zend Studio - The Leading PHP IDE. [En línea] 2011. [Citado el: 5 de Diciembre de 2011.] <http://www.zend.com/products/studio/index?src=plobby>.
35. **Corporation, Oracle.** Bienvenido a NetBeans. [En línea] 2011. [Citado el: 19 de Noviembre de 2011.] http://netbeans.org/index_es.html.
36. **Martínez, Rafael.** Principal | www.postgresql-es.org. [En línea] [Citado el: 20 de Noviembre de 2011.] <http://www.postgresql-es.org/principal>.
37. **Oracle Corporation.** About MySQL. [En línea] 2010. [Citado el: 20 de Noviembre de 2011.] <http://www.mysql.com/about/>.
38. **10gen.** MongoDB. [En línea] 2011. [Citado el: 20 de Enero de 2012.] <http://www.mongodb.com>.
39. **Kabir, Mohammed J.** *La biblia de Servidor Apache2*. 2004,39-50.
40. **Foundation, The Apache Software.** Apache Tomcat Specifications. [En línea] <http://tomcat.apache.org/>.
41. **Davis, Williams.** *Herramientas CASE: Metodología estructurada para el desarrollo de los sistemas*. s.l. : Paraninfo, 1991.
42. **Quatrani, Terry.** *Visual Modeling with Rational Rose 2002 and UML*. s.l. : Addison-Wesley Professional, 2003.
43. **Visual Paradigm.** Visual Paradigm for UML. [En línea] 2010. [Citado el: 20 de Noviembre de 2011.] <http://www.visual-paradigm.com/>.
44. **Engineers, Institute of Electrical and Electronics.** *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York, NY : s.n., 1990.

45. **Olivares Rojas, Juan Carlos.** *Patrones de Diseño*. México : Instituto Tecnológico de México, 2009.
46. **Sun Microsystems Inc.** *JAVA CODE CONVENTIONS*. 1995-1999.
47. **Sanz Carmenates, Nadiesda Perez Rivas, Geiser Arcio.** Pruebas de Aceptación Parciales del Cliente. [En línea] 2007. http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_0479_07.
48. **Singh, Tarandeep.** REST vs. SOAP – The Right WebService. [En línea] 2011. [Citado el: 21 de Noviembre de 2011.] <http://geeknizer.com/rest-vs-soap-using-http-choosing-the-right-webservice-protocol>.
49. Introducing JSON. [En línea] 2011. [Citado el: 15 de Noviembre de 2011.] <http://www.json.org/>.

Bibliografía

Sandoval, Jose. *RESTful Java Web Services*. BIRMINGHAM - MUMBAI, Packt Publishing, Nov 2009. ISBN 978-1-847196-46-0.

Chodorow, Kristina. Dirolf, Michael. *MongoDB: The Definitive Guide*. O'Reilly Media, Sept 2010. ISBN: 978-1-449-38156-1.

Richardson, Leonard. Ruby, Sam. *RESTful Web Services*. O'Reilly Media, May 2007, ISBN-10: 0-596-52926-0.

Burke, William J. *RESTful Java with JAX-RS*. O'Reilly Media, Nov 2009, 978-0-596-15804-0.

Smiley, David. Pugh, Eric. *Apache Solr 3 Enterprise Search Server*. BIRMINGHAM - MUMBAI Packt Publishing, Aug 2009, ISBN 978-1-84951-606-8

Webber, Jim. *Rest in Practice. Hypermedia and System Architecture*. s.l. : O'Reilly, 2010.

LeBlanc, Jonathan. *Programing Social Aplications*. s.l. : O'Reilly, 2011.

Anexos

Anexo 1

Campo	Descripción
categories	Lista de categorías.
categories/yct_categories	Lista de categorías YCT.
categories/yct_categories/yct_category	Categorías YCT. Este elemento tiene un atributo de puntuación numérica. Las categorías se enumeran en orden decreciente según la puntuación.
entities	Lista de las entidades y conceptos detectados.
entities/entity/text	Texto de las entidades y conceptos detectados.
entities/entity/wiki_url	Dirección en Wikipedia del concepto o entidad. No se muestra si no existe dicha página en Wikipedia.
entities/entity/types	Listado de tipos de entidades y conceptos.
entities/entity/metadata_list	Lista de Metadatos.

Tabla 22 Campos de respuesta analizador de contenidos de Yahoo! **Fuente:** Yahoo!

Campo	Descripción
ResultSet	Lista con todos los términos extraídos.
Result	Texto de cada término extraído.

Tabla 23 Campos de respuesta extractor de términos de Yahoo! **Fuente:** Yahoo!

Anexo 2

Campos	Descripción
U	La URL de los resultados de búsqueda.
TM	Tiempo total de ejecución de la consulta.
S	Muestra los términos resaltados para cada resultado de búsqueda.
T	El título de los resultados de búsqueda.
M	Número estimado de resultados para un criterio de búsqueda.
q	Consulta de búsqueda. Este parámetro es requerido.

Tabla 24 Campos de respuesta servicio web de búsqueda de Google **Fuente:** Google

Anexo 3

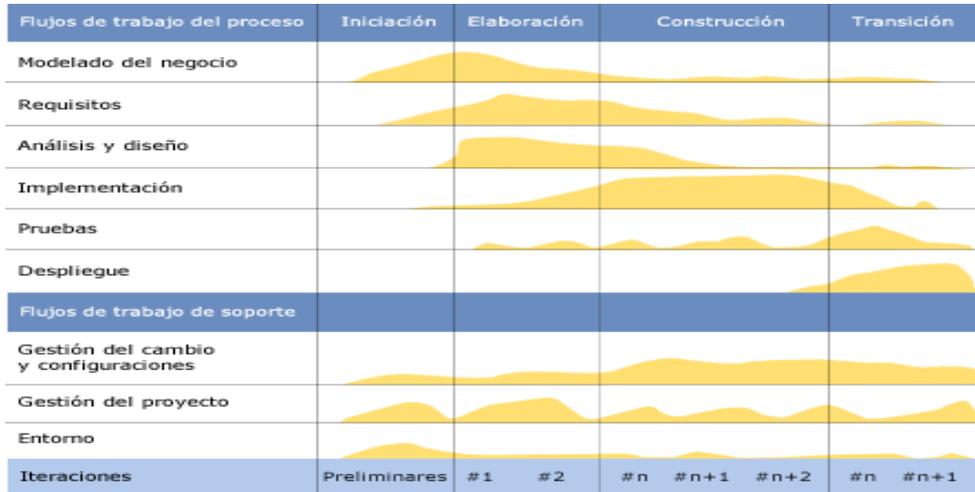


Ilustración 15 Ciclo de vida de RUP Fuente: IBM

Anexo 4



Ilustración 16 Clases de la biblioteca Mongo para JAVA

Anexo 4

SolrQuery	SolrRequest
<pre> -scorePattern : Pattern +SolrQuery() +SolrQuery(q : String) +setTerms(b : boolean) : SolrQuery +getTerms() : boolean +addTermsField(field : String) : SolrQuery +getTermsFields() : String [] +setTermsLower(lower : String) : SolrQuery +getTermsLower() : String +setTermsUpper(upper : String) : SolrQuery +getTermsUpper() : String +setTermsUpperInclusive(b : boolean) : SolrQuery +getTermsUpperInclusive() : boolean +setTermsLowerInclusive(b : boolean) : SolrQuery +getTermsLowerInclusive() : boolean +setTermsLimit(limit : int) : SolrQuery +getTermsLimit() : int +setTermsMinCount(cnt : int) : SolrQuery +getTermsMinCount() : int +setTermsMaxCount(cnt : int) : SolrQuery +getTermsMaxCount() : int +setTermsPrefix(prefix : String) : SolrQuery +getTermsPrefix() : String +setTermsRaw(b : boolean) : SolrQuery +getTermsRaw() : boolean +setTermsSortString(type : String) : SolrQuery +getTermsSortString() : String +setTermsRegex(regex : String) : SolrQuery +getTermsRegex() : String +setTermsRegexFlag(flag : String) : SolrQuery +getTermsRegexFlags() : String [] +addFacetField(fields : String []) : SolrQuery +addNumericRangeFacet(field : String, start : Number, end : Number, gap : Number) : SolrQuery +addDateRangeFacet(field : String, start : Date, end : Date, gap : String) : SolrQuery +getFacetFields() : String [] +removeFacetField(name : String) : boolean +setFacet(b : boolean) : SolrQuery +setFacetPrefix(prefix : String) : SolrQuery +setFacetPrefix(field : String, prefix : String) : SolrQuery +addFacetQuery(f : String) : SolrQuery +getFacetQuery() : String [] +removeFacetQuery(q : String) : boolean +setFacetLimit(lim : int) : SolrQuery +getFacetLimit() : int </pre>	<pre> -method : SolrRequest\$METHOD -path : String -responseParser : ResponseParser +SolrRequest(m : SolrRequest\$METHOD, path : String) +getMethod() : SolrRequest\$METHOD +setMethod(method : SolrRequest\$METHOD) : void +getPath() : String +setPath(path : String) : void +getResponseParser() : ResponseParser +setResponseParser(responseParser : ResponseParser) : void +getParams() : SolrParams +getContentStreams() : Collection +process(parameter : SolrServer) : SolrResponse </pre>
	<pre> SolrServer -binder : DocumentObjectBinder +SolrServer() +add(docs : Collection) : UpdateResponse +addBeans(beans : Collection) : UpdateResponse +add(doc : SolrInputDocument) : UpdateResponse +addBean(obj : Object) : UpdateResponse +commit() : UpdateResponse +optimize() : UpdateResponse +commit(waitFlush : boolean, waitSearcher : boolean) : UpdateResponse +optimize(waitFlush : boolean, waitSearcher : boolean) : UpdateResponse +optimize(waitFlush : boolean, waitSearcher : boolean, maxSegments : int) : UpdateResponse +rollback() : UpdateResponse +deleteById(id : String) : UpdateResponse +deleteById(ids : List) : UpdateResponse +deleteByQuery(query : String) : UpdateResponse +ping() : SolrPingResponse +query(params : SolrParams) : QueryResponse +query(params : SolrParams, method : SolrRequest\$METHOD) : QueryResponse +request(parameter : SolrRequest) : NamedList +getBinder() : DocumentObjectBinder </pre>
	<pre> SolrResponse +SolrResponse() +getElapsedTime() : long +setResponse(parameter : NamedList) : void +getResponse() : NamedList </pre>

Ilustración 17 Clases de la biblioteca Solrj

Anexo 5

Clases	Frameworks	Bibliotecas	
		Solrj	Mongo
WS_AuthSession, WS_Robots, WS_Sitemap, WS_Busqueda, WS_CaracteristicasPortal, WS_TraficoPortal, WS_Stats	JAX-RS +Consumes(...) +MediaType(...) +Path(...) +Response(...)		
AuthSession, Session, Cliente, AdminSession			+Mongo(...) +QueryBuilder(...) +Response(...) +WriteResult(...)
Robots, Sitemap		+SolrQuery(...) +SolrRequest(...) +SolrServer(...) +SolrResponse(...)	
Búsqueda, CaracteristicasPortal		+SolrQuery(...) +SolrRequest(...) +SolrServer(...) +SolrResponse(...)	+Mongo(...) +QueryBuilder(...) +Response(...) +WriteResult(...)
TraficoPortal, Stats, RegistroTrafico, RegistroConsulta			+Mongo(...) +QueryBuilder(...) +Response(...) +WriteResult(...)

Tabla 25 Relación Clases del Sistema- Bibliotecas-Frameworks

Anexo 6

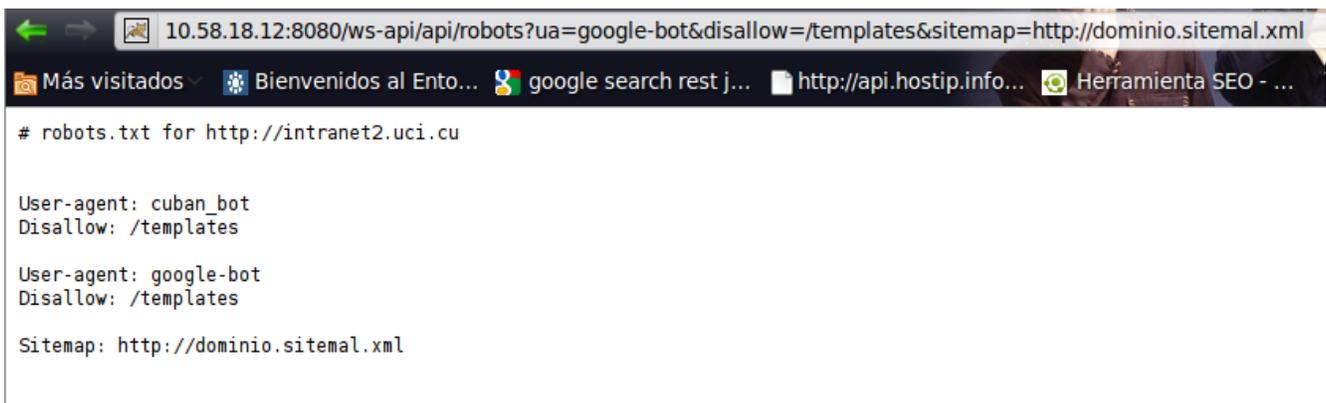


Ilustración 18 Ejemplo de uso del servicio web robots.txt

Anexo 7

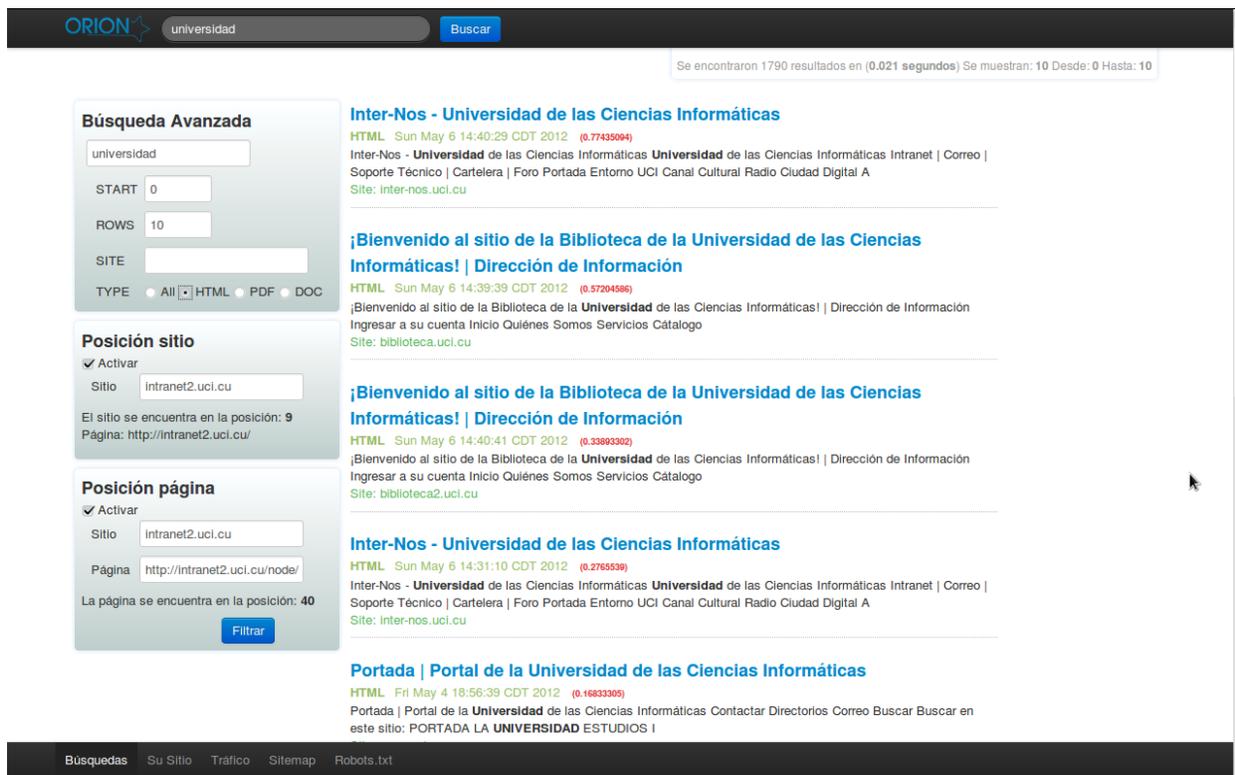


Ilustración 19 Aplicación de prueba de los servicios web.

Parámetros	Descripción
/ws-api/api/auth/	Ruta para los servicios web de autenticación.
/ws-api/api/search/?	Ruta para los servicios web de búsqueda general.
/ws-api/api/search/wsp?	Ruta para los servicios web de búsqueda de un sitio específico.
/ws-api/api/search/wpp?	Ruta para los servicios web de búsqueda de una página específica.
/ws-api/api/sitemap?	Ruta para el servicio de sitemap.xml
/ws-api/api/robots?	Ruta para el servicio de robots.txt

Tabla 26 Ruta de servicios web

Anexo 8

Caso de Uso	Cambiar Propiedades Conexión	
Actor	Sistema Externo	
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir el servicio destinado a cambiar las propiedades de su conexión, dígame: correo del usuario y contraseña. Para ello se introducen los parámetros que desea modificar y la clave de sesión actual que identifica los mismos. El servicio solicitado es ejecutado y por último, el sistema actualiza las propiedades de conexión correspondientes.	
Precondiciones	El sistema externo debe haber obtenido una clave de sesión.	
Referencias	RF18	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El sistema externo realiza la petición del servicio para cambiar las propiedades de conexión, para ello introduce los datos necesarios (correo del usuario y contraseña, clave de sesión).	2. Verifica validez de los datos. 3. Modifica las propiedades de conexión que se desean cambiar. 4. Devuelve mensaje de respuesta de estado exitoso.	
Flujo Alterno		
Acción del Actor	Respuesta del Sistema	
	2. Si los datos no son correctos se envía mensaje de error: "Error. Parámetros incorrectos."	
Poscondiciones	Las propiedades de conexión son modificadas	

Tabla 27 Descripción CU Cambiar Propiedades Conexión

Caso de Uso	Obtener Características Portal
Actor	Sistema Externo
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir algunos de los servicios para obtener las características de determinado portal. Para ello se introducen los datos necesarios para el procesamiento del servicio a través de la

	aplicación, como puede ser la url del portal a consultar así como la clave de sesión activa que le permita consumir dicho servicio. El servicio solicitado es ejecutado y se brinda la respuesta correspondiente.	
Precondiciones	El sistema externo debe haber obtenido una clave de sesión.	
Referencias	RF5, RF6, RF7, RF8	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El sistema externo realiza la petición de uno de los siguientes servicios, pasando la clave de sesión y la url del portal:</p> <ul style="list-style-type: none"> a. Devolver la cantidad de enlaces entrantes que tiene una página. b. Devolver la cantidad de páginas de un portal indexadas en el buscador. c. Devolver las páginas de un portal indexadas en el buscador. d. Devolver el Page Rank en el buscador de un sitio determinado. e. Devolver los criterios de búsqueda por los que más se visita una página desde el buscador. 	<p>2. Verifica la validez de los datos.</p> <p>3. Ejecuta la petición y envía la respuesta.</p>	
Flujo Alterno		
Acción del Actor	Respuesta del Sistema	
	2. Si los datos entrados no son correctos se envía mensaje de error: "Parámetros incorrectos"	
Poscondiciones	Se obtiene respuesta del sistema.	

Tabla 28 Descripción CU Obtener Características Portal

Caso de Uso	Generar Sitemap.xml	
Actor	Sistema Externo	
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir el servicio de generar un mapa a determinado portal. Para ello se introducen los datos necesarios para el procesamiento del servicio a través de la aplicación, como puede ser la url del portal al que se le desee generar el sitemap.xml así como la clave de sesión activa que le permita consumir dicho servicio. El servicio solicitado es ejecutado y se brinda la respuesta correspondiente.	
Precondiciones	El sistema externo debe haber obtenido una clave de sesión.	
Referencias	RF10	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El sistema externo realiza la petición del servicio para generar el sitemap.xml a un portal, para ello introduce los datos necesarios (url del portal, clave de sesión).	2. Verifica la validez de los datos. 3. Ejecuta la petición y envía la respuesta.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
	2. Si los datos entrados no son correctos se envía mensaje de error: "Parámetros incorrectos"	
Poscondiciones	Se genera el sitemap.xml al portal especificado.	

Tabla 29 Descripción CU Generar Sitemap.xml

Caso de Uso	Generar robots.txt	
Actor	Sistema Externo	
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir el servicio de generar un fichero de exclusión de robots. Para ello se introducen los datos necesarios para el procesamiento del servicio a través de la aplicación, como puede ser la url del portal al que se le desee generar el robots.txt así como la clave	

	de sesión activa que le permita consumir dicho servicio. El servicio solicitado es ejecutado y se brinda la respuesta correspondiente.	
Precondiciones	El sistema externo debe haber obtenido una clave de sesión.	
Referencias	RF11	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El sistema externo realiza la petición del servicio para generar un fichero de exclusión de robots a un portal, para ello introduce los datos necesarios (url del portal, clave de sesión).	2. Verifica la validez de los datos. 3. Ejecuta la petición y envía la respuesta.	
Flujo Alterno		
Acción del Actor	Respuesta del Sistema	
	2. Si los datos entrados no son correctos se envía mensaje de error: "Parámetros incorrectos"	
Poscondiciones	Se genera el fichero de exclusión de robots al portal especificado.	

Tabla 30 Descripción CU Generar Robot.txt

Caso de Uso	Obtener Tráfico Portal
Actor	Sistema Externo
Resumen	El caso de uso se inicia cuando un sistema externo desea consumir algunos de los servicios para obtener los datos referentes al tráfico de determinado portal en el buscador. Para ello se introducen los datos necesarios para el procesamiento del servicio a través de la aplicación, como puede ser la url del portal a consultar así como la clave de sesión activa que le permita consumir dicho servicio. El servicio solicitado es ejecutado y se brinda la respuesta correspondiente.
Precondiciones	El sistema externo debe haber obtenido una clave de sesión con los datos de un usuario registrado como webmaster del sitio a consultar.
Referencias	RF12, RF13, RF13.1, RF13.2, RF13.3, RF13.4, RF13.5,
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El sistema externo realiza la petición de uno de los siguientes servicios, pasando la clave de sesión:</p> <ul style="list-style-type: none"> a. Devolver los criterios de búsqueda por los que más se visita una página desde el buscador. b. Devolver cantidad de clic. c. Devolver direcciones IP. d. Devolver criterios de búsqueda utilizados. e. Devolver navegador. f. Devolver fecha y hora. g. Devolver sistema operativo. 	<p>2. Verifica la validez de los datos. 3. Ejecuta la petición y envía la respuesta.</p>
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	2. Si los datos entrados no son correctos se envía mensaje de error: "Parámetros incorrectos"
Poscondiciones	Se obtiene respuesta del sistema.

Tabla 31 Descripción CU Obtener Tráfico Portal

Caso de Uso	Registrar tráfico desde el buscador.
Actor	Sistema Externo
Resumen	El caso de uso se inicia cuando un sistema externo tras haber consumido un servicio de búsqueda visita alguno de los resultados mostrados. Para ello los links de estos resultados tendrán una estructura que accederá a la capa de servicios, donde esta registrará la visita y redireccionará posteriormente a la dirección original del resultado. El servicio solicitado es ejecutado y se produce un registro de tráfico desde el buscador.
Precondiciones	El sistema externo debe estar realizando una búsqueda.

Referencias	RF9
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El sistema externo realiza la petición de un recurso de los expuestos como resultado por el buscador.	2. Obtiene la petición parametrizada. 3. Inserta el registro de tráfico. 4. Redirección a la dirección original del recurso solicitado.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
Poscondiciones	Se inserta un registro de tráfico, se redirecciona a la dirección original del recurso.

Tabla 32 Descripción CU Registrar tráfico desde el buscador

Caso de Uso	Realizar búsqueda vía correo electrónico.
Actor	Sistema Externo
Resumen	El caso de uso se inicia cuando un sistema externo solicita a través de un correo electrónico un servicio de búsqueda. Para ello el sistema accederá a la capa de servicios, donde obtendrá los resultados de la misma, enviando posteriormente a la dirección del remitente el resultado de la búsqueda.
Precondiciones	
Referencias	RF19
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El sistema externo envía un correo electrónico, donde en el asunto irán los parámetros de la búsqueda.	2. Obtiene los parámetros del correo. 3. Ejecuta la petición y envía la respuesta al remitente. 4. Registra los datos de la petición en la Base de Datos.

Flujo Alterno	
Acción del Actor	Respuesta del Sistema
Poscondiciones	Se obtiene respuesta del sistema.

Tabla 33 Descripción CU Realizar búsqueda vía correo electrónico

Glosario de términos

API: interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Artefactos: Un artefacto es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software.

GPL: La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Java: Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Biblioteca: En ciencias de la computación, una biblioteca (o librería) es un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos.

Paquete: Un paquete de software es una serie de programas que se distribuyen conjuntamente. Algunas de las razones suelen ser que el funcionamiento de cada uno complementa a o requiere de otros, además de que sus objetivos están relacionados como estrategia de mercadotecnia.

WEB 2.0: El término, Web 2.0 fue acuñado por Tim O'Reilly en 2004 para referirse a una segunda generación en la historia de la Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs y los wikis.

WWW : World Wide Web o Red Global Mundial es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de internet.

XML: Siglas de Extensible Markup Language, lenguaje desarrollado por el W3 Consortium para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar y compartir información en la red.

Webmasters: Persona encargada de la gestión y mantenimiento de un servidor web, fundamentalmente desde el punto de vista técnico.

UML: Siglas de Unified Modeling Language, lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

URL: Siglas de Uniform Resource Locator, es la dirección global de cualquier documento o recurso en la Web, visto como localizador.

Página Web: Documento situado en el Web con información y diversos enlaces con otros documentos también situados en el Web.

Posicionamiento web: Consiste en la aplicación de técnicas tendentes a lograr que los buscadores de Internet encuadren nuestra página web en una posición y categoría deseada dentro de su página de resultados para determinados conceptos clave de búsqueda.

RF: Requisito funcional.

Robots: Programa que navega la web yendo desde cada página a todas las que ésta da enlace y enviando e incorporando el código o partes de él a una base de datos de referencia

Servicio Web: sistemas de software que permiten el intercambio de datos y funcionalidades entre aplicaciones sobre una red soportado en diferentes estándares que garantizan la interoperabilidad de estos.