



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 buscador Orión

Autora:

Wendy Díaz Hernández

Tutores:

Ing. José Gabriel Espinosa Ramírez

Lic. Regla. C Jiménez Hernández

La Habana, junio del 2017

“Año 59 de la Revolución”



Bueno es ir a la lucha con determinación, abrazar la vida y vivir con pasión, perder con clase y vencer con osadía, porque el mundo pertenece a quien se atreve y la vida es mucho para ser insignificante.

Charles Chaplin

Declaración de autoría

Declaro por este medio que yo Wendy Díaz Hernández con carné de identidad 94022127850 soy el autor principal del trabajo titulado “Capa de servicios web para el buscador Orión” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

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

Wendy Díaz Hernández

Firma de la Autora

Ing. José G Espinosa Ramírez

Firma del Tutor

Lic. Regla C Jiménez Hernández

Firma del Tutor

Agradecimientos

A mi amigos de la UCI: Hector, Arlenis, Laura gracias por estar conmigo en todo momento y apoyarme en todo.

A todos los que entraron conmigo al aula y hoy se gradúan junto conmigo.

A todos nuestros profesores docentes.

A mis tutores: Muchas gracias por toda la dedicación brindada a lo largo del desarrollo de este trabajo de diploma.

A mi pareja por oír mis quejas todo el tiempo y apoyarme en todo momento.

A mi mama por brindarme todo su apoyo, por entregarme su vida, te amo mamuti.

Dedicatoria

A mis abuelas, que me entregaron todo el amor del mundo mientras la vida se los permitió, por enseñarme a compartir lo que tengo, a luchar por mis sueños, nunca pensé celebrar este momento sin ellas, pero la presencia de su ausencia me da la valentía para ser más capaz porque quiero que estén donde estén, se sientan orgullosa de mí, las amo.

A mi madre porque es mucho lo que tengo que agradecerle por ser la madre más valiente y dedicada del mundo, por haber sido como fuiste conmigo, por haber estado las 24 horas disponibles para mí, por darme todo sin esperar nada a cambio, por llorar cuando yo sufría y por reír conmigo cuando triunfaba, por ser más que mi mejor amiga desde que tengo uso de razón, te amo mamuti.

A mi sobrino que es el regalo más lindo que me ha dado mi hermano.

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

Resumen

Desde el año 2011, en la Universidad de las Ciencias Informáticas, se desarrolla el motor de búsqueda Orión que permite localizar los contenidos publicados en la Web cubana. Este buscador posibilita 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 general y búsqueda avanzada. 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 de contenido

Resumen.....	5
Introducción	10
CAPÍTULO 1: Fundamentación teórica para el desarrollo de la capa de servicios web para el buscador Orión.....	15
1.1 Introducción	15
1.2 El surgimiento de la Web	15
1.3 Servicios web	15
1.3.1 Servicio de transporte.....	18
1.3.2 Servicio de mensajería.....	18
1.4 Interconexión de los motores de búsqueda con sistemas de terceros.....	22
1.4.1 Google	23
1.4.2 Yahoo!	24
1.5 Lenguajes, herramientas y tecnologías.....	25
1.5.1 Lenguajes de Programación	25
1.5.2 Marco de trabajo (<i>Frameworks</i>)	26
1.5.3 Metodologías de desarrollo	28
1.5.4 Entorno de desarrollo integrado	31
1.5.5 Servidores web	31
1.5.6 Sistemas de gestión de base de datos	32
1.5.7 Mecanismos existentes para la indexación de información	33
1.5.7 Herramientas CASE	36
1.6 Conclusiones parciales.....	36
CAPÍTULO 2: Análisis y diseño de la capa de servicios web para el buscador Orión.....	38
2.1 Introducción	38
2.2 Propuesta de la solución del sistema.....	38
2.3 Modelo de dominio	40
2.3.2 Descripción de las clases de dominio	40
2.4 Relación de los requerimientos del sistema	41
2.4.1 Requisitos funcionales.....	41
2.4.2 Requisitos no funcionales.....	41
2.5 Modelo de casos de uso del sistema.....	42
2.5.1 Definición de los actores del sistema.....	42

2.5.2 Definición de los Casos de Uso del sistema	43
2.5.3 Descripción de los Casos de Uso del sistema	44
2.6 Modelo de diseño	48
2.6.1 Estilo arquitectónico	48
2.6.2 Patrones de diseño empleados durante el desarrollo del sistema	49
2.7 Diagrama de clases del diseño	51
2.8 Diseño de la Base de Datos	52
2.9 Modelo de Despliegue	53
2.10 Conclusiones parciales	54
CAPITULO 3: Implementación y prueba del sistema	55
3.1 Introducción	55
3.2 Modelo de implementación del sistema	55
3.2.1 Diagrama de componentes	55
3.3 Estándares de código	56
3.4–Aspectos de Seguridad en el sistema	56
3.5 Tratamiento de errores	57
3.6 Validación del sistema	57
3.6.1 Pruebas	58
3.6.2 Descripción de los tipos de pruebas	58
3.8 Validación de la hipótesis de la investigación	66
3.9 Conclusiones parciales	69
Conclusiones	70
Recomendaciones	71
Referencias	72

ÍNDICE DE TABLAS

Tabla 1 Operacionalización de las variables.....	12
Tabla 2 Comparación JSON – XML.....	17
Tabla 3 Frameworks PHP.....	29
Tabla 4 Definición de autores del sistema.....	45
Tabla 5 Definición CU Autenticar Usuario.....	46
Tabla 6 Definición CU Devolver resultados de búsqueda.....	46
Tabla 7 Definición CU Gestionar credenciales de OAuth.....	46
Tabla 8 Descripción CU Autenticar usuario.....	47
Tabla 9 Descripción CU Devolver resultados de búsqueda.....	49
Tabla 10 Descripción CU Gestionar credenciales OAuth.....	50
Tabla 11 Respuesta del sistema ante errores.....	61
Tabla 12 Caso de Prueba CU Realizar búsqueda.....	62
Tabla 13 Descripción de variables de prueba CU Realizar búsqueda.....	63
Tabla 14 Matriz de Datos CU Realizar búsqueda.....	64
Tabla 15 Cantidad de no conformidades por cada iteración de las pruebas funcionales.....	64
Tabla 16 Cantidad de no conformidades por cada iteración de las pruebas de integración.....	66
Tabla 17 Resultados de las pruebas de rendimiento.....	68
Tabla 18 Vulnerabilidades del entorno.....	70
Tabla 19 Distribución de frecuencia.....	72

ÍNDICE DE FIGURAS

Figura 1 Funcionamiento SOAP.....	21
Figura 2 Uso de protocolos de servicios web.....	21
Figura 3 Diagrama de Clase de modelo de dominio.....	42
Figura 4 Diagrama de CU del sistema.....	45
Figura 5 Diagrama de Arquitectura del sistema.....	52
Figura 6 Diagrama de Clase de Diseño CU Realizar búsqueda.....	55
Figura 7 Diagrama de Base de Datos	55
Figura 8 Diagrama de Despliegue.....	56
Figura 9 Diagrama de Componentes de CU Realizar búsqueda.....	59
Figura 10 Comportamiento de las no conformidades por cada iteración de las pruebas funcionales.....	65

Introducción

El constante desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) es un factor determinante en el progreso social de la humanidad. Sus contribuciones en las últimas dos décadas, revolucionaron la manera de hacer negocios, la comunicación entre las personas y la interacción entre las empresas lo que favorece que cada país de una forma u otra se desarrolle y avance continuamente. A medida que se desarrollan las tecnologías, la sociedad ha tenido la necesidad de informatizar los procesos esenciales para lograr una mayor eficiencia en el trabajo cotidiano, interactuando con una gran cantidad de información digital, a través de los sistemas informáticos. En este sentido, se puede decir que el desarrollo tecnológico y la digitalización de la información se han convertido en algunos de los procesos más importantes del mundo moderno (Rossini, 2010).

Los contenidos disponibles en la red constantemente se encuentran en crecimiento a la vez que las necesidades de información de los usuarios evolucionan hacia requerimientos cada vez más complejos que deben ser satisfechos en un menor tiempo. En los albores de la red de redes el proceso para encontrar un recurso se realizaba de forma manual como un ejercicio complejo que demandaba de grandes cantidades de tiempo y paciencia. En el año 1993 surge la primera aplicación con características de un motor de búsqueda de la mano de Matthew Gray (Ledford, 2008).

Como parte de los programas de informatización de la sociedad cubana, surge en el año 2002 la Universidad de las Ciencias Informáticas UCI. Este centro de altos estudios cuenta con el Centro de Ideoinformática, el cual se encuentra enmarcado en la Facultad 1. Este centro de desarrollo de software se especializa en el estudio y la ejecución de proyectos para internet, de investigación + desarrollo + innovación (I+D+I). Como parte de los esfuerzos desplegados por este está el proyecto Orión que tiene como objetivo el desarrollo de un motor de búsqueda que permite localizar los contenidos publicados en la Web cubana. El motor de búsqueda Orión permite la recuperación de contenidos desde la red en base a las necesidades de información aportadas por los usuarios. Este motor de búsqueda ajusta su alcance al entorno en el cual sea desplegado, a la vez que sirve como base para el desarrollo de otros sistemas para facilitar el acceso a los recursos en la red. Orión cuenta con las funcionalidades de búsqueda general en la Web, búsqueda general avanzada en la Web, búsqueda de imágenes, búsqueda de documentos, búsqueda general avanzada de documentos, gestión de usuarios, alertas, y publicidad.

En la actualidad, se impone la necesidad de que los sistemas para Internet cuenten con la capacidad de comunicarse no solo con los usuarios finales, sino también con otros sistemas en la red con el fin de intercambiar contenidos e información; así como exponer las características propias como servicios para ser consumidos por terceros, como mecanismo de integración.

En estos momentos la integración con el motor de búsqueda Orión solo puede ser realizada mediante la integración punto a punto, lo cual no garantiza un control de las conexiones y expone elementos de la configuración que atentan contra la seguridad y estabilidad del sistema. La exposición de los detalles de configuración y la posibilidad de acceso directo a los componentes críticos de la arquitectura del sistema, afectan la seguridad y estabilidad del motor de búsqueda Orión y lo hace vulnerable ante diferentes tipos de ataques informáticos. En la medida que aumente el número de aplicaciones de terceros que deseen interactuar mediante una conexión punto a punto se encarece y se complejiza el proceso de mantenimiento del sistema (McKendrick, 2007). Las conexiones punto a punto se caracterizan por estar fuertemente enlazadas a los protocolos y aplicaciones involucrados por lo que son inflexibles ante los cambios que puedan tener lugar en la plataforma tecnológica sobre la que se soporta el sistema.

Al no existir el mecanismo para que las aplicaciones informáticas interactúen con el motor de búsqueda Orión, ve limitada sus posibilidades de explotación en un contexto en cual se impone la necesidad del intercambio de información y servicios con sistemas de terceros. Esto trae consigo que las funcionalidades que aportan valor agregado al motor de búsqueda Orión como producto informático, no puedan ser aprovechadas por las aplicaciones de los usuarios del mismo que se encuentran disponibles en la Web.

Considerando la situación problemática anteriormente descrita, se plantea como **problema de la investigación**: Los insuficientes mecanismos para que las aplicaciones informáticas interactúen con el motor de búsqueda Orión, limitan sus posibilidades de interoperabilidad con otros sistemas.

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** el 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 para el buscador Orión, que permita la integración con aplicaciones de terceros de forma segura y controlada.

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

1. Elaborar el marco teórico de la investigación que permita la comprensión del escenario.
2. Seleccionar los métodos y herramientas para la integración con aplicaciones de terceros mediante una capa de servicios web.
3. Implementar una capa de servicios web para el motor de búsqueda Orión.
4. Validar los resultados del sistema mediante pruebas de funcionalidad, pruebas de integración y pruebas de seguridad.

Luego de haber tratado los elementos fundamentales del área de la ciencia a incidir y los objetivos primordiales, se formula la siguiente hipótesis de investigación: el desarrollo de una capa de servicios web para el motor de búsqueda Orión incrementará, la seguridad y control de la integración del mismo con las aplicaciones de terceros. Teniendo en cuenta la hipótesis planteada se define como **variable independiente**: el desarrollo de una capa de servicios web para el motor de búsqueda Orión. Como **variables dependientes** se especifican: la seguridad y el control de la integración con aplicaciones de terceros.

A continuación, se detalla la operacionalización de las variables dependientes e independientes.

Tabla 1. Operacionalización de las variables.

Variable independiente	Dimensión	Definición conceptual	Indicadores	Unidades de medida
El desarrollo de una capa de servicios web para el motor de búsqueda Orión.	Capa de servicios web.	Involucra el proceso de que las aplicaciones informáticas interactúen con el motor de búsqueda Orión, con el fin de intercambiar contenidos e información; así como exponer las características propias como servicios para ser consumidos por terceros, como mecanismo de integración.	Satisfacción.	Alto (Más del 80 %) Medio (Del 40 % a 80 %) Bajo (Menos del 40 %)
Variable independiente	Dimensión	Definición conceptual	Indicadores	Unidades de medida
La seguridad y el control de la integración con aplicaciones de terceros.	La seguridad y el control.	Se refiere a la integración del motor de búsqueda con aplicaciones de terceros de forma segura y controlada.	Exhaustividad.	Alto (Más del 80 %) Medio (Del 40 % a 80 %) Bajo (Menos del 40 %)
			Precisión.	Alto (Más del 80 %) Medio (Del 40 % a 80 %) Bajo (Menos del 40 %)

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

1. Establecimiento de los fundamentos teóricos y metodológicos que sustentan el desarrollo de servicios web en motores de búsqueda para la interoperabilidad de aplicaciones informáticas.
2. Identificación de los servicios web que debe ofrecer el motor de búsqueda Orión.
3. Definición de la metodología a utilizar en el desarrollo de una capa de servicios web. Para así lograr la identificación de las necesidades de información, requisitos funcionales y no funcionales y la documentación de los artefactos generados durante los procesos de análisis, diseño e implementación de los servicios web.
4. Implementación de la capa de servicios web mediante la cual se expongan los servicios del motor de búsqueda Orión para ser consumidos por aplicaciones de terceros, garantizando la interoperabilidad de las mismas con el buscador.

En el desarrollo de la investigación se utilizaron los siguientes métodos de investigación:

Métodos teóricos

Histórico – Lógico: con el objetivo de constatar teóricamente cómo ha evolucionado la utilización de servicios web hasta la actualidad y de igual forma las herramientas y tecnologías utilizadas en el desarrollo de aplicaciones web.

Analítico – Sintético: empleado para el análisis de los elementos esenciales referentes a las literaturas relacionadas 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.

Modelación: utilizado en la representación, se utilizará para reflejar la estructura, relaciones entre objetos, las actividades que intervienen en el proceso, relaciones internas y características de la solución a través de diagramas.

Métodos empíricos:

Entrevista: se empleó para obtener información de los miembros del Proyecto Orión perteneciente al centro de desarrollo de software CIDI.

Estructura del documento

Capítulo 1

Se expondrán un conjunto de conceptos y criterios fundamentales que permitirán entender el mundo de los

servicios web objeto de estudio de la investigación. Además, se estudiarán los principales motores de búsquedas existentes, con la finalidad de brindar una solución a la problemática planteada. Finalmente, se expondrán las distintas tecnologías a utilizar en el desarrollo de la herramienta, así como la metodología de desarrollo de software a utilizar.

Capítulo 2

Serán abordados los temas relacionados con el dominio y se exponen las características del sistema, patrones de diseño, requerimientos funcionales y no funcionales, y los artefactos que requiere la metodología de desarrollo utilizada.

Capítulo 3

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 para el desarrollo de la capa de servicios web para el buscador Orión.

1.1 Introducción

Con el objetivo de favorecer una mejor comprensión de la presente investigación en este capítulo se realiza una revisión bibliográfica acerca de los elementos y áreas del conocimiento que engloban al objeto de estudio y al campo de acción. Además, se analizan y evalúan los principales motores de búsqueda y se exponen las características esenciales de las herramientas, las tecnologías y la metodología de desarrollo de software seleccionadas para la implementación de la solución.

1.2 El surgimiento de la Web

El surgimiento y desarrollo de la Web como servicio imprescindible para compartir información, creó un excelente espacio para la interacción del hombre con la información hipertextual, a la vez que sentó las bases para el desarrollo de una herramienta integradora de los servicios existentes en Internet (Ronda Leon, 2008). Además, estas sustentaron el nacimiento de tecnologías que dieron origen a nuevas tendencias en el desarrollo de las comunicaciones.

La Web se convirtió en un espacio no sólo para obtener datos, sino para enviarlos, modificarlos y hacer transacciones económicas con ellos. Se convirtió en una plataforma donde la gente intercambia ideas, mensajes o productos de acuerdo a sus necesidades. Uno de los objetivos principales que se plantea los desarrolladores de la Web es ir más allá de los estándares técnicos y buscar una verdadera capacidad de compartir datos y conocimientos (Hernández, 2016).

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.

1.3 Servicios web

Los servicios web cobran relevancia al estandarizar la comunicación entre distintas aplicaciones de software. Normalmente se describe como servicio web una colección de procedimientos (métodos) a los que podemos llamar desde cualquier lugar de Internet o de nuestra intranet, siendo este mecanismo de invocación totalmente independiente de la plataforma que utilicemos y del lenguaje de programación en el que se haya implementado internamente el servicio (Dpto. de ciencias de la computación e inteligencia artificial, 2014).

Cuando conectamos con un servidor web desde el navegador, el servidor devuelve la página web solicitada, que es un documento que se mostrará en el navegador para que lo visualice el usuario, pero es difícilmente entendible por una máquina. En contraposición, los servicios web ofrecen información con un formato estándar que puede ser entendido fácilmente por una aplicación (Dpto.de ciencias de la computación e inteligencia artificial, 2014).

La necesidad de que las aplicaciones se comunicaran con otros sistemas dio surgimiento a los servicios web. El consorcio W3C¹ define los servicios web como: “sistemas de software diseñados para soportar una interacción interoperable máquina a máquina sobre una red”. Dicho de otro modo, los servicios web proporcionan una forma estándar de interoperar entre aplicaciones software que se ejecutan en diferentes plataformas. Por lo tanto, se caracterizan por una gran interoperabilidad y extensibilidad, así como por proporcionar información fácilmente procesable. 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” (Morales Machuca, 2008).

Los servicios web presentan un grupo de beneficios de interés para los usuarios, como son (Morales Machuca, 2008):

- ✓ **Promueven la interoperabilidad:** La interacción entre un proveedor y un solicitante de servicio está diseñada para que sea completamente independiente de la plataforma y el lenguaje.
- ✓ **Permiten la integración:** El proceso de descubrimiento se ejecuta dinámicamente, a medida que los solicitantes de servicio utilizan a los agentes para encontrar proveedores de servicio.
- ✓ **Reducen la complejidad por medio del encapsulamiento:** Los solicitantes y los proveedores del servicio se preocupan por las interfaces necesarias para interactuar.
- ✓ **Abren la puerta a nuevas oportunidades de negocio:** Los servicios web facilitan la interacción con socios de negocios, al poder compartir servicios internos con un alto grado de integración.

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.

¹ World Wide Web Consortium

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 (Lamarca, 2013).

Los servicios web basados en XML facilitan definir etiquetas y relaciones estructurales entre ellas, ofrece mecanismos más versátiles para mostrar datos y dentro de sus principales usos están los siguientes:

- ✓ **XML aplicado a los sitios web:** permite separar contenido y presentación, y que los mismos datos se puedan mostrar de varias formas distintas sin demasiado esfuerzo.
- ✓ **XML para la comunicación entre aplicaciones:** representación de los datos muy simple, fácil de transmitir por la red, estándar. En los últimos tiempos este uso se está haciendo muy popular con el surgimiento de los servicios web (Lamarca, 2013).

Otro formato o notación utilizada para el intercambio de datos entre aplicaciones es JSON². 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 (Ecma, 2013).

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

Tabla 2. Comparación JSON – XML.

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.

Con el profundo estudio realizado a las alternativas para el intercambio de datos de los servicios web se concluye: 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.

² JavaScript Object Notation

Para el desarrollo de los servicios web es importante conocer la estructura de los mismos. Este conjunto está conformado por cuatro subconjuntos entre los que se encuentran:

- ✓ Servicio de transporte.
- ✓ Servicio de mensajería.
- ✓ Descripción del servicio.
- ✓ Descubrimiento de servicios.

1.3.1 Servicio de transporte

Este servicio tiene como función lograr una comunicación confiable entre sistemas finales (extremo a extremo), asegurando que los datos lleguen en el mismo orden en que han sido enviados, y sin errores. Los servicios de transporte aíslan a la capa superior de los cambios del hardware y del sistema operativo y puede multiplexar varias conexiones sobre una conexión de red.

Se encarga del transporte entre host, de la confiabilidad del transporte de los datos, determinación y control de fallas y control del flujo de la información (Tolosa, 2014). Para su funcionamiento este incluye varios protocolos del nivel de aplicación del modelo OSI³, 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 utilizado para la transferencia de las transacciones XML a través de la red utilizando los mismos principios del HTML (W3C, 2011).
- ✓ **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.
- ✓ **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 (SMTP, 2006).

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.

³ Open System Interconnection. (Sistema de Intercomunicación Abierta)

REST (Representational State Transfer): Estilo de arquitectura de software para sistemas hipermedias distribuidos, tales como la World Wide Web. Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP.

REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RP (Marqués, 2013).

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 (Valero Menacho, 2015).

Los principios de REST son:

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

Existen tres niveles de calidad a la hora de aplicar REST en el desarrollo de una aplicación web. Estos niveles son (Marqués, 2013):

1. **Uso correcto de URI:** los objetos en REST siempre se manipulan a partir de la URI. La URI nos facilita acceder a la información para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros.
2. **Uso correcto de HTTP:** cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla.
3. **Implementar Hypermedia:** Ese concepto llevado al desarrollo de páginas web es lo que permite que el usuario puede navegar por el conjunto de objetos a través de enlaces HTML. En el caso de una API REST, el concepto de hipermedia explica la capacidad de una interfaz de desarrollo de aplicaciones de proporcionar al cliente y al usuario los enlaces adecuados para ejecutar acciones concretas sobre los datos.

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 (Dpto. de ciencias de la computación e inteligencia artificial, 2014).

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 (Dpto. de ciencias de la computación e inteligencia artificial, 2014):

- ✓ 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.

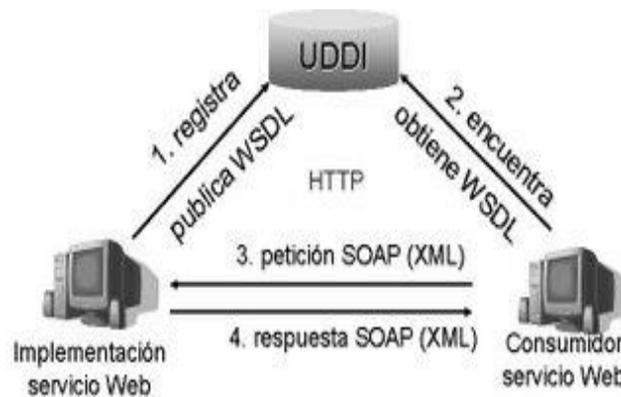


Figura 1 Funcionamiento SOAP.

Protocolo de comunicación que se debe usar

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 (Mejia, 2015).

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.

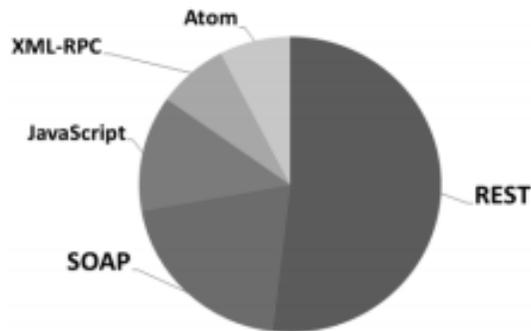


Figura 2 Uso de protocolos de servicios web.

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.

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.

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 (Dpto. de ciencias de la computación e inteligencia artificial, 2014).

Dentro de sus ventajas podemos encontrar:

- ✓ Seguridad, incrementa la privacidad, y es fuertemente tipado.
- ✓ Utiliza el Web Services Description Language (WSDL) que contiene el conjunto de normas comunes para definir los mensajes, los enlaces, las operaciones y la ubicación del servicio web.
- ✓ Es más seguro debido a que su implementación en la mayoría de las veces se hace del lado del servidor.
- ✓ Soporta varios protocolos y tecnologías, incluyendo WSDL, XSD, SOAP y WS-Addressing.
- ✓ Cuando se trabaja con componentes y se utiliza .NET o Java es muy sencillo de consumir.

El resultado consiste en un documento XML que contiene una definición específica del tipo de dato, lo que hace del protocolo algo muy estricto (Mejia, 2015).

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 (Singh, Tarandeep, 2011).

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 Fielding⁴ argumenta que la Web funciona mejor cuando se utiliza con el estilo que lo hace REST (Webber, 2010).

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 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 institucional como entorno de despliegue, por lo que habrá un bajo uso de recursos, así como el ancho de banda de transmisión. 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.

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

1.4 Interconexión entre los motores de búsqueda y sistemas de terceros

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 API⁵. Aplicaciones web de relevancia, como son las redes sociales que día a día demandan mayor disponibilidad

⁴ Roy T. Fielding – Uno de los principales autores de las especificaciones de HTTP.

⁵ Application Programming Interface-Interfaz de Programación de Aplicaciones

en el ciberespacio, utilizan fehacientemente este tipo de tecnologías, Facebook⁶ y Twitter⁷, por solo mencionar algunas.

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 más usado para localizar información en la Web a nivel mundial con los servicios web.

1.4.1 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 (StatCounter, 2011).

Hasta el 31 de agosto de 2011 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 (Google Custom Search , 2013).

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 (Google Custom Search , 2013).

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 (Google Custom Search , 2013).

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 en JSON.

Otro ejemplo de solicitud a este servicio web sería:

⁶ <http://www.facebook.com>

⁷ <http://www.twitter.com>

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

En el anexo número uno se resumen los principales parámetros a tener en cuenta para trabajar con el servicio web de búsqueda de Google 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 JSON.

1.4.2 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 (Tarun's Blog, 2012).

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! (Tarun's Blog, 2012).

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 (CollegiateLink Help Center, 2016).

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 YQL⁸ 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 HTTP⁹ utilizando el método POST.

Una petición de un servicio de análisis de contenidos de Yahoo! se demuestra con los parámetros relacionados en la tabla del anexo 2.

1.5 Lenguajes, herramientas y tecnologías

Para desarrollar la propuesta de solución, necesidad surgida del estudio realizado de herramientas homólogas, se hace necesario estudiar varias tecnologías, lenguajes y herramientas disponibles para llevar a cabo el objetivo general de la investigación.

1.5.1 Lenguajes de Programación

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:

✓ **Java**

Java es un lenguaje de programación multiplataforma que se introdujo a finales de 1995. Al programar en Java no se parte de cero. Cualquier aplicación que se desarrolle se apoya en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario, otras pueden ser comerciales, pero siempre hay un número muy importante de clases que forman parte del propio lenguaje (el API o Application Programming Interface de Java). Java incorpora en el propio lenguaje muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores. Java es un lenguaje muy completo. La compañía Sun Microsystems creadora del mismo describe el lenguaje Java como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico” (Jalón Garcia Javier, 2000).

✓ **PHP (*Hypertext Preprocesor*)**

Es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML (González, 2010). PHP es un lenguaje de programación de uso general,

⁸ Yahoo! Query Language (Lenguaje de consultas de Yahoo!)

⁹ Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto)

originalmente diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante.

Lo que distingue a PHP de los lenguajes de ejecución del lado del cliente como JavaScript, es que el código es ejecutado en el servidor, generando la 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 (González, 2010).

Teniendo en cuenta que PHP es multiplataforma, ya que el código es procesado completamente en el lado del servidor y se entrega como contenido dinámico al espectador. Permite operar en varios sistemas operativos, es un lenguaje completamente expandible, totalmente libre, que presenta una gran variedad de módulos y sus entornos de desarrollo son de rápida y fácil configuración y presenta fácil acceso a bases de datos. Además, de que es útil para realizar operaciones a gran escala que necesiten funcionar en todas las arquitecturas de sistemas. Tomando como base las características expuestas anteriormente en la descripción del lenguaje, se decide seleccionar a PHP como el lenguaje a utilizar.

1.5.2 Marco de trabajo (*Frameworks*)

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.

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 describe el *framework* que es utilizado en la implementación:

Symfony 2.8

Symfony es un marco de trabajo PHP de alto rendimiento de código abierto para el desarrollo de aplicaciones web, construido sobre una serie de componente web reutilizables. Posee la característica de que es fácil de instalar y puede ser usado tanto en Linux como en Windows. Symfony tiene una corta curva de aprendizaje y una comunidad de desarrollo comprometida para llevar PHP al próximo nivel. El proyecto tiene más de 2000 contribuidores y más de 300000 desarrolladores (Javier, 2011).

Ventajas del uso de Symfony como marco de trabajo:

- ✓ Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en las bases de datos y una mejor experiencia para el usuario.
- ✓ El manejo de cache reduce el uso de banda ancha y la carga del servidor.
- ✓ La presentación usa *templates* y *layouts* que pueden ser contruidos por diseñadores de HTML que no posean conocimientos del *framework*.
- ✓ Los *plugins* proveen un alto nivel de extensibilidad.
- ✓ Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.
- ✓ El *framework* de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas.
- ✓ La barra de depuración web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.
- ✓ La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.
- ✓ Es posible realizar cambios "en caliente" de la configuración (sin necesidad de reiniciar el servidor).
- ✓ El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

Yii

Yii es un marco de trabajo PHP de alto rendimiento para el desarrollo de aplicaciones web. Presenta una corta curva de aprendizaje apoyada en una amplia gama de tutoriales, ejemplos y guías de usuario. La seguridad es un *standard* con Yii, incluye validación de entradas, filtrado de salida y prevención frente a inyección SQL y Cross Site Scripting. Yii está explícitamente diseñado para trabajar eficientemente con AJAX (Ramirez, 2014).

Zend

Zend es un marco de trabajo PHP para el desarrollo de aplicaciones web. Zend tiene una arquitectura abierta basada en el Modelo-Vista-Controlador (MVC). Está desarrollado con buenas prácticas, almacena las contraseñas usando bcrypt y encripta utilizando AES-256 como algoritmo de encriptación. Posee una amplia colección de paquetes para el desarrollo profesional con PHP accesibles desde la comunidad (O'Phinney, 2016).

Se presentan los marcos de trabajo estudiados en la siguiente tabla comparativa.

Tabla 3. Frameworks PHP.

Característica	Framework		
	Symfony	Yii	Zend
Código Abierto	Si	Si	Si
MVC	Si	Si	Si
Almacenamiento en Cache	Si	Si	Si
Múltiples Bases de Datos	Si	Si	Si
Documentación Abundante	Si	Si	Si

Atendiendo a las características expuestas anteriormente, se decide seleccionar Symfony en su versión 2.8. Además, de ser sencillo de usar y lo suficientemente flexible como para adaptarse a los casos más complejos, puede integrarse al buscador cubano Orión que contiene una interfaz web desarrollada con este *framework*.

1.5.3 Metodologías de desarrollo

Con el objetivo de lograr una mayor organización en el proceso de desarrollo de software se seleccionan las metodologías de desarrollo, las cuales constituyen una filosofía de trabajo que proporciona una base de procesos para llevar a cabo con éxito cualquier proyecto informático. Las metodologías de desarrollo brindan soporte a la toma de decisiones en un equipo de trabajo; es decir, permiten conocer qué persona hace una determinada actividad, cuándo y cómo la debe hacer.

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 (Peñalver Romero, 2008).

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 el éxito del proyecto (Peñalver Romero, 2008).

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 (Peñalver Romero, 2008).

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 (Peñalver Romero, 2008).

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 (Peñalver Romero, 2008).

Algunas ventajas de SXP son:

- ✓ La metodología establece el uso de sistemas automatizados para la generación de algunos artefactos. Y además los recomienda de manera explícita.
- ✓ La descripción de una historia de usuario suele ser más sencilla que la de un caso de uso.
- ✓ SXP ha sido excelente para equipos pequeños, siempre ha sido recomendado para menos de 20 personas (CADAVID, 2013).

AUP

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- ✓ Desarrollo dirigido por pruebas (test driven development - TDD en inglés).
- ✓ Modelado ágil.
- ✓ Gestión de cambios ágil.
- ✓ Refactorización de base de datos para mejorar la productividad (Rodríguez Sánchez, 2013).

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva.

Fases AUP

1. Inicio: El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.

2. Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.

3. Construcción: Durante la fase de construcción el sistema es desarrollado y probado en el ambiente de desarrollo.

4. Transición: El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación para finalmente ser desplegado en los sistemas de producción.

AUP define 7 disciplinas (4 ingenieriles y 3 de gestión de proyectos), las disciplinas son (Rodríguez Sánchez, 2013):

Disciplinas de AUP

1. Modelo: El objetivo de esta disciplina es entender el negocio de la organización, el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio. Agrupa los flujos de trabajos de Modelado de negocio, Requisitos y Análisis y Diseño.

2. Implementación: El objetivo de esta disciplina es transformar su modelo en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas.

3. Prueba: El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificando que se cumplan los requisitos.

4. Despliegue: El objetivo de esta disciplina es la prestación y ejecución del sistema y que el mismo este a disposición de los usuarios finales.

5. Gestión de configuración: El objetivo de esta disciplina es la gestión de acceso a herramientas de su proyecto. Esto incluye no sólo el seguimiento de las versiones con el tiempo, sino también el control y gestión del cambio para ellos.

6. Gestión de proyectos: El objetivo de esta disciplina es dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos), coordinación con el personal y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.

7. Entorno: El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso sea el adecuado, la orientación (normas y directrices), y herramientas (hardware, software) estén disponibles para el equipo según sea necesario.

Variación de AUP para la UCI

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos) exigiéndose así que el proceso sea configurable. Se decide hacer uso de una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

Para guiar el proceso de desarrollo de la propuesta de solución se decidió utilizar la metodología AUP-UCI ya que esta metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas, para ello nos apoyaremos en el Modelo CMMI-DEV v1.3. El cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad (Rodríguez Sánchez, 2013).

1.5.4 Entorno de desarrollo integrado

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.

NetBeans V 8.1

Como entorno de desarrollo integrado se utilizará NetBeans. Liberado bajo el licenciamiento dual de CDDL y GPL (versión 2), NetBeans es un potente IDE para programadores que proporciona una plataforma ideal para escribir, compilar, depurar y ejecutar programas informáticos. Aunque inicialmente fue ideado para Java, puede ser empleado para la codificación de aplicaciones en múltiples lenguajes de programación. Este, además de ser gratuito y sin restricciones de uso, posee versiones para los distintos sistemas operativos del mercado, convirtiéndolo en una alternativa con grandes ventajas para los desarrolladores. La estructura modular de NetBeans le proporciona estabilidad y grandes posibilidades de ser extendido gradualmente por desarrollos comunitarios, permitiendo agregar continuamente nuevas funcionalidades. Su versatilidad lo ha convertido en el IDE por excelencia entre miles de programadores alrededor del mundo (Oracle, 2014).

1.5.5 Servidores web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales 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 (Kabir, 2004).

Apache

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 (Kabir, 2004). 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 (Kabir, 2004).

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 (Kabir, 2004).

JMeter

JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web. JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, servicios web, JMS, HTTP y conexiones TCP genéricas (Apache JMeter, 2016).

1.5.6 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 (López Gamino, 2001).

MongoDB

Un estudio enfocado en las bases de datos no relacionales arroja que MongoDB constituye una solución escalable y de alto rendimiento de almacenes de datos. MongoDB le permite a los desarrolladores implementar aplicaciones tan grandes como sea necesario. 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 (MongoDB, Inc., 2017).

La estructura se compone fundamentalmente por las colecciones de documentos, conformándose estos últimos por campos que almacenan los datos de interés. Esta estructura posee cierta analogía con los esquemas de bases de datos SQL, con la diferencia de que no se definen estrictamente los campos de cada documento. 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 (Chodorow, 2014).

MongoDB, aun siendo un sistema de gestión de bases de datos no relacionales, mantiene las características más valiosas de las bases de datos relacionales: consistencia fuerte, lenguaje de consulta expresivo e índices secundarios. Como resultado, los desarrolladores pueden crear aplicaciones altamente funcionales más rápido que las bases de datos del modelo clásico de sistema de gestión de bases de datos relacionales (Meyers, 2015).

Dada las características planteadas se selecciona MongoDB para el almacenamiento de los datos relacionados con la capa de servicios web a desarrollar. 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, proporciona la flexibilidad del modelo de datos y el alto rendimiento; los desarrolladores pueden mejorar continuamente las aplicaciones y entregarlas a una escala casi ilimitada en hardware.

MongoDB es considerada luego de un análisis en profundidad el almacén de documentos de la estructura JSON y el soporte de consulta de estructura arbitraria. MongoDB supera a PostgreSQL por órdenes de magnitud para la línea de tiempo y la consulta de comparación de *host*. Se eligió MongoDB para la base de datos del sistema ya que además, de las características planteadas, tiene otras características deseables como son, que cuenta con la capacidad para balancear la carga y recopilación de datos utilizando múltiples servidores para el almacenamiento de archivo (Meyers, 2015).

1.5.7 Mecanismos existentes para la indexación de información

Otro de los subsistemas presentes en la arquitectura de un buscador es el mecanismo de indexación. Entre los más utilizados se encuentran:

Elasticsearch

Es un servidor de búsqueda basado en Lucene. Provee un motor de búsqueda de texto completo, distribuido y con una interfaz web RESTful y con documentos JSON. Elasticsearch está desarrollado en Java y está publicado como código abierto bajo las condiciones de la licencia Apache.

Amazon Elasticsearch Service ofrece una API REST sencilla, un desempeño rápido, capacidades de búsqueda potentes y una escalabilidad perfecta que le permitirán crear aplicaciones de alto desempeño con capacidad para almacenar y recuperar miles de millones de documentos, con replicación integrada entre distintas zonas de disponibilidad situadas en una misma región.

Elasticsearch indexa los datos y permite que se analicen casi en tiempo real. Con el análisis geoespacial de Elasticsearch, puede identificar la región geográfica en la que se ha producido el problema. De este modo, los equipos de resolución de problemas pueden buscar el índice y realizar agregaciones estadísticas para identificar la causa raíz y resolver el problema (Amazon Web Services, Inc, 2017).

Entre sus características se destacan:

- ✓ Expone un API HTTP de tipo RESTful, y usa JSON tanto para peticiones como para respuestas.
- ✓ También se puede operar usando la API nativa de Java.
- ✓ Está libre de esquemas de datos, en el sentido de que no necesita disponer de una definición explícita del esquema.
- ✓ Búsquedas Facetadas - Muestra contador para cada categoría en los resultados de búsqueda.
- ✓ Búsqueda Geo-espacial - Búsqueda por localización y distancia.
- ✓ Los documentos (datos) no necesariamente tienen que ser planos, permite elementos anidados.
- ✓ Replicación - El índice podría ser replicado y proporciona soporte para conmutación por error.
- ✓ Posee búsqueda distribuida, es decir, la búsqueda puede realizarse en varios fragmentos/índices y al concluir la misma los resultados serán agregados.
- ✓ Presenta indexación distribuida lo que significa que los documentos van a ser almacenados en distintos nodos.
- ✓ Arquitectura diseñada pensando siempre en la distribución para permitir escalar una solución de un nodo a cientos, ofreciendo alta disponibilidad, soportando grandes cantidades de datos y cortos tiempos de respuesta.

Actualmente Elasticsearch presenta una comunidad pequeña de colaboradores, y consecuentemente una base de usuarios pequeña.

Solr

Solr es una plataforma de búsquedas basada en Apache Lucene, que funciona como un "servidor de búsquedas". Sus principales características incluyen búsquedas de texto completo, resaltado de resultados y manejo de documentos (como Word y PDF). Solr es escalable, permitiendo realizar búsquedas distribuidas y replicación de índices (Apache Solr, 2010). Actualmente es frecuentemente usado por importantes sitios en Internet tales como el sitio web de la Casa Blanca, que utiliza Solr vía Drupal para la búsqueda con resaltado y facetado de sitios. Está escrito en Java, pero puede ser usado en cualquier lenguaje, simplemente usando las peticiones GET para realizar búsquedas en el índice, y POST para agregar y actualizar documentos. Es fácil de configurar y de utilizar. Una de las características principales de Solr es su API con estilo REST.

Otras de sus características son:

- ✓ Ofrece un API REST y un API de Java.
- ✓ No se requiere un esquema y tipo de documento.
- ✓ Búsquedas facetadas - Muestra contador para cada categoría en los resultados de búsqueda.
- ✓ Los documentos (datos) tienen que ser planos, debido a que no permite elementos anidados.
- ✓ Permite importar datos desde una base de datos.
- ✓ Replicación: El índice podría ser replicado y proporciona soporte para conmutación por error.
- ✓ Búsqueda distribuida: La búsqueda puede realizarse en varios fragmentos/índices y al concluir la misma los resultados serán agregados.
- ✓ Resaltado de los resultados de búsqueda.
- ✓ Se ha desarrollado con la capacidad de extraer el contenido del archivo desde el sistema de archivos y añadirlo como parte de índice.

Solr, al igual que Elasticsearch, facilita a los programadores desarrollar aplicaciones sofisticadas y de alto rendimiento que incluyen facetado (resultado de búsqueda ordenado en columnas con cuentas numéricas). Solr se fundamenta en otra tecnología originaria de búsqueda-Lucene, una biblioteca de Java que provee la indexación y la tecnología de búsqueda. Ambos Solr y Lucene son manejados por la *Apache Software Foundation* (Yigal, 2016).

Solr es un mecanismo de indexación que utiliza un modelo de recuperación de información híbrido ya que se basa en el modelo booleano y en el modelo vectorial para establecer el subconjunto de documentos relevantes.

Por otra parte, aunque tanto Solr como Elasticsearch gozan de una gran popularidad, ambos son escritos en Java. Solr presenta soporte para la manipulación de datos concurrentemente, además de que garantiza la integridad de los datos mientras que Elasticsearch no.

Por tanto, teniendo en cuenta la anterior comparación, añadiendo que Solr hoy es el mecanismo de indexación utilizado por el buscador cubano Orión se ha determinado que como mecanismo de indexación se utilizará Solr. Es importante señalar que Solr implementa una variante del modelo.

1.5.7 Herramientas CASE

Visual Paradigm

Visual Paradigm es una herramienta CASE multiplataforma, que soporta el ciclo completo de desarrollo de software: análisis, diseño, implementación y pruebas. Facilita la construcción de aplicaciones informáticas con un menor coste que destacan por su alta calidad y contribuye a mejorar la experiencia de usuario mediante el diseño de un gran número de artefactos de ingeniería de software. Permite la generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería directa e inversa, la gestión de requisitos de software y la modelación de procesos del negocio (Patricia, 2015).

Visual Paradigm (VP) 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 (Visual Paradigm, 2010).

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 facilitan licencias especiales para fines académicos (Visual Paradigm, 2010).

Visual Paradigm for UML por su estabilidad de ejecución en diferentes sistemas operativos y la facilidad de abrir y trabajar con un modelo UML utilizando el mismo programa independientemente del sistema operativo y sin afectar en absoluto el trabajo hecho es seleccionada para realizar el proceso de modelado. Además, se debe destacar que esta herramienta guarda todo el modelo en un solo fichero (Quatrani, 2002).

1.6 Conclusiones parciales

En este capítulo se trataron los elementos teóricos que dan sustento a la propuesta de solución del problema planteado, arribando a las siguientes conclusiones:

- ✓ El estudio de los conceptos asociados a los servicios web permitió lograr un mejor entendimiento de la investigación que se realiza.

- ✓ El análisis de las funciones de interconexión con sistemas de terceros de algunos de los buscadores más utilizados del mundo permitió identificar requerimientos necesarios para nuestra investigación.
- ✓ La selección de la metodología, herramientas y tecnologías con soporte multiplataforma y basadas en software libre, permitió obtener una base tecnológica enfocada en los componentes que se utilizaran para la integración con aplicaciones de terceros mediante una capa de servicios web en el buscador Orión.

CAPÍTULO 2: Análisis y diseño de la capa de servicios web para el buscador Orión

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.

En el presente capítulo se identifican 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 de la solución del sistema

La capa está enfocada en dotar al buscador de la capacidad para comunicarse no solo con los usuarios finales, sino también con otros sistemas en la red con el fin de intercambiar contenidos e información, así como exponer las características propias como servicios para ser consumidos por terceros, realizar peticiones y por cada petición generar un mensaje de respuesta en lenguaje JSON que contenga la información solicitada. Para que estas funcionalidades puedan ser consumidas desde otros entornos, se implementó una capa de servicios web utilizando la base tecnológica propuesta en el capítulo anterior, que se conecte con la capa de servicios del buscador Orión. Esta proporciona una serie de funciones que simplifican la interacción de otras aplicaciones con los servicios del motor de búsqueda Orión.

El sistema consumidor del servicio representa una aplicación informática que necesita consumir los servicios del motor de búsqueda. Esta aplicación interactúa con 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 de servicios web del motor de búsqueda Orión se encuentran:

-Autenticar Usuario por OAuth: Permitirá la autorización por parte del proveedor de los servicios web (Orión), a usuarios o a aplicaciones clientes para poder consumir su información o parte de ella de manera segura y controlada.

OAuth (Open Authorization): es un protocolo abierto, que permite la autorización por parte de las aplicaciones web o proveedores de servicios web a usuarios (a través de la red, aplicaciones de escritorio

o móviles), para poder compartir su información o parte de ella de manera segura. El protocolo pretende ayudar y proporcionar tanto a desarrolladores de aplicaciones como a proveedores de servicios una completa funcionalidad de intercambio de información protegida de forma segura, simple y controlada, sin la necesidad de que el usuario tenga que entregar su usuario y contraseña para cada una de las aplicaciones web que tenga que utilizar (Oscar, 2012).

OAuth es ideal para desarrolladores porque:

- ✓ Permite la construcción de aplicaciones web, de escritorio, *widgets* o bibliotecas que realicen transferencia o compartan información protegida o sensible.

OAuth es ideal para proveedores porque:

- ✓ Permite soportar API de lado del servidor.
- ✓ Desplegar servicios que sirven información sensible.

-Obtener resultados de búsqueda: Representa una serie de servicios que serán consumidos por una aplicación tercera mediante la acción de realizar búsquedas haciendo uso de la capa de servicios web, esta acción brinda respuesta a las peticiones realizadas por una solicitud con cierto criterio de búsqueda que incluye los resultados según la consulta y los parámetros de filtrado incluidos, el mensaje de respuesta será en un documento con lenguaje JSON.

-Gestionar credenciales de OAuth: Las credenciales almacenan información de autenticación de sistemas externos, por lo que si se desea realizar una integración con sistemas que exigen que los usuarios se registren, casi con seguridad será necesario hacer uso de las credenciales. Las credenciales admiten la autenticación de contraseña por OAuth de modo que los usuarios pueden gestionar sus propias credenciales lo que permite que estos puedan configurar y gestionar sus propios parámetros de autenticación para acceder al sistema externo.

OAuth utiliza 3 tipos de credenciales para realizar el protocolo de autenticación y permiso (Oscar, 2012):

- ✓ **Credenciales del cliente:** Son utilizadas para autenticar el cliente. Permite al servidor recoger información sobre el cliente que utiliza sus servicios.
- ✓ **Credenciales temporales:** Son utilizadas para identificar peticiones de autorización.
- ✓ **Credenciales token (*access token*):** Son utilizadas en lugar del nombre de usuario y contraseña del dueño del recurso, esto es para evitar que el usuario tenga que compartir estos datos con el cliente. Con estas credenciales el cliente accede a la información protegida sin conocer la contraseña del dueño del recurso.

2.3 Modelo de dominio

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases de software, u objetos de software con responsabilidades, sino que modela clases conceptuales significativas en un determinado problema (Craig, 2012).

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.

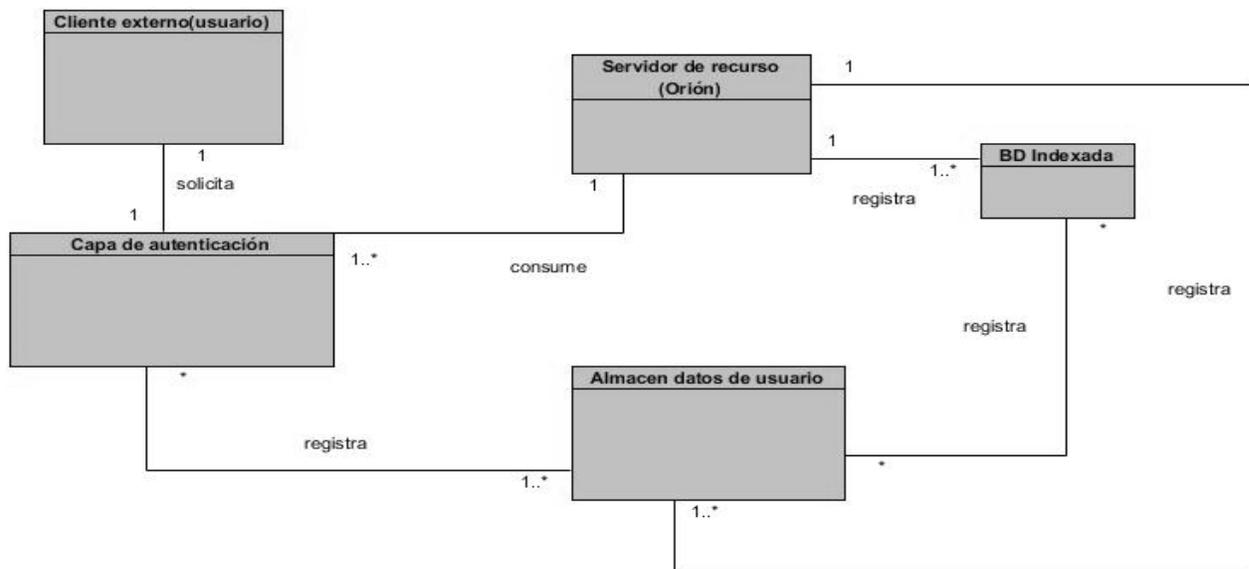


Figura 3 Diagrama de Clase de Modelo de dominio.

2.3.2 Descripción de las clases de dominio

- ✓ **Servidor de recurso (Orión):** Proveedor que brinda sus servicios para ser consumidos por el cliente externo.
- ✓ **Cliente externo:** Aplicación informática que va a consumir los servicios web que se brindan en el servidor de recurso.
- ✓ **BD Indexada:** Aplicación informática que maneja la información referente a las búsquedas así como al registro de usuarios que consumen los servicios del motor, de forma organizada y persistente.
- ✓ **Almacén de datos de usuario:** Información relacionada con los consumos de los servicios web y en la que se retendrán los datos de cada una de las demandas por parte de los clientes (dirección IP, fecha, criterio de búsquedas).

- ✓ **Capa de autenticación:** En un modelo tradicional de autenticación, el cliente interactúa con el servidor e intercambia, el cliente solicita el acceso a los recursos controlados por el dueño del recurso (Orión) y que están almacenados en el servidor. Cuando un cliente desea acceder a un recurso, primero deberá obtener permiso del dueño del mismo, este permiso está definido en forma de *token* y su correspondiente clave compartida.

2.4 Relación de los requerimientos del sistema

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 (Institute of Electrical and Electronics Engineers, 1990).

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 Requisitos funcionales

RF1. Autenticar usuario con OAuth.

RF2. Realizar búsqueda general.

RF2.1. Realizar búsquedas en la web.

RF2.2. Realizar búsquedas de imágenes.

RF2.3. Realizar búsquedas de documentos.

RF3. Realizar búsqueda avanzada.

RF3.1. Realizar búsquedas avanzada en la web.

RF3.1. Realizar búsquedas avanzada de documentos.

RF4. Gestionar credenciales de OAuth.

2.4.2 Requisitos no funcionales

✓ **Requerimientos de software**

RNF 1 Se requiere del sistema operativo Ubuntu 16.04.

RNF 2 Se requiere la instalación del servidor web y de servlets Tomcat 7 para el correcto funcionamiento del servidor de Solr.

RNF 3 Se requiere la instalación del servidor web Apache en su versión 2.4 y PHP 5.5 o superior para poder visualizar la interfaz web.

✓ **Requerimientos de hardware**

RNF 4 CPU con 2 núcleos de procesamiento a velocidad de 2.0 GHz.

RNF 5 2 GB de RAM.

RNF 6 160 GB de espacio libre en disco.

✓ **Requerimientos de diseño e implementación**

RNF 7 Como lenguaje de programación para la interfaz web se deberá utilizar PHP en su versión 5.5.

RNF 8 Para el desarrollo de la aplicación web se deberá utilizar Symfony 2.8.

RNF 9 La aplicación se desarrollará sobre el sistema operativo Ubuntu 16.04 LTS.

✓ **Requerimientos de usabilidad**

RNF 10 Debe contar con la portabilidad necesaria para poder ser transferido de un ambiente a otro o reemplazado por nuevas versiones.

✓ **Requerimientos de eficiencia**

RNF 11 La respuesta del sistema debe encontrarse entre los 3 y 5 segundos aproximadamente.

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.

Tabla 4. Definición de autores del sistema.

Actores del sistema	Cliente externo (aplicación tercera).
Justificación	Cliente externo (aplicación tercera): Es la aplicación o sistema que se conecta para consumir los servicios web que brinda el buscador a través de la capa de servicios web.

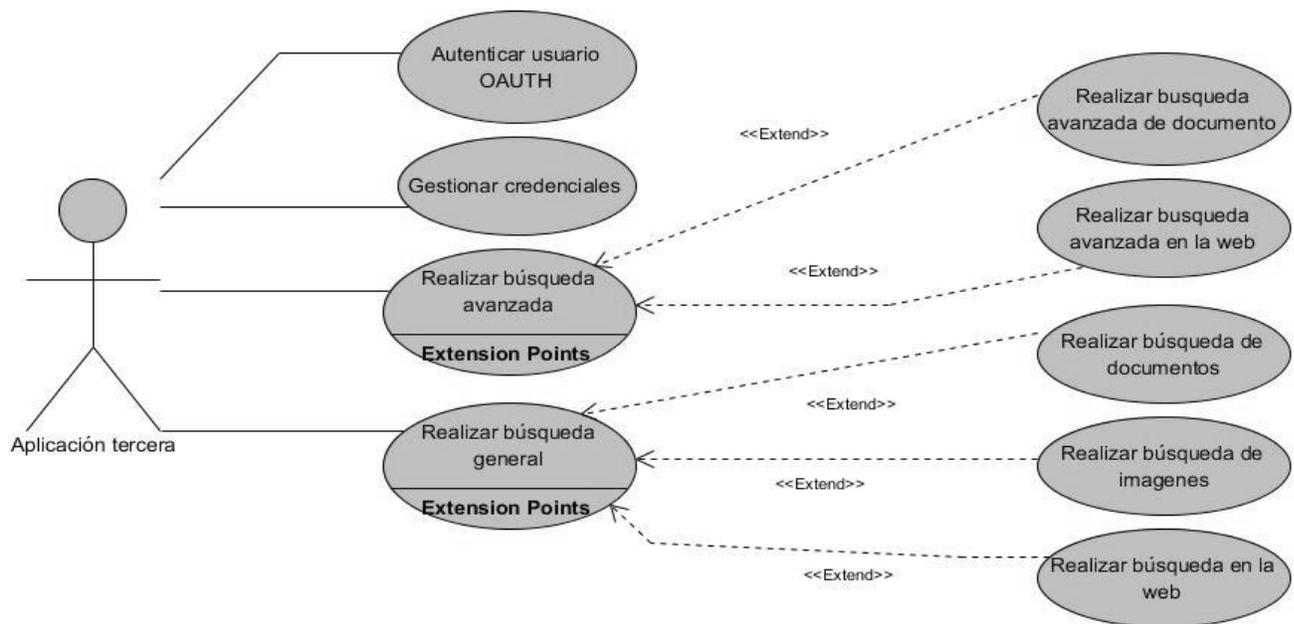


Figura 4 Diagrama de CU del sistema.

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.

A continuación, se describen los casos de uso del sistema, la descripción del caso de uso realizar búsqueda avanzada se puede encontrar en el anexo 3.

Tabla 5. Definición CU Autenticar Usuario.

CU_1	Autenticar Usuario con OAuth
Autor	Cliente externo (aplicación tercera)
Descripción	<p>El cliente solicita petición de autorización a los recursos en el servidor utilizando credenciales temporales con la intención de obtener un “<i>access token</i>”.</p> <p>El servidor otorga esta autorización o “<i>access token</i>” previa validación del usuario en su plataforma. Este paso involucra 2 formas de realizarlo, la primera es que la petición se realice de forma automática en la cual el protocolo OAuth redirigirá al usuario a la página de autenticación del servidor y una vez autenticado le pregunte por si desea otorgar dicha autorización al</p>

	cliente, y la segunda forma es la que el dueño del recurso o usuario previamente ha otorgado el permiso o <i>token</i> y lo suministra al cliente en su desarrollo o al llamar a las clases que lo implementan. El cliente envía la petición con el <i>access token</i> para realizar la operación con la API, u obtener información protegida.
Referencia	RF1

Tabla 6. Definición CU Devolver resultados de búsqueda general.

CU_2	Realizar búsqueda general
Autor	Cliente externo (aplicación tercera)
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, RF2.1, RF2.2, RF2.3

Tabla 7. Definición CU Gestionar credenciales de OAuth.

CU_4	Gestionar credenciales de OAuth
Autor	Cliente externo (aplicación tercera)
Descripción	Es el caso de uso que da la posibilidad de utilizar la autenticación por usuario, de modo que los usuarios pueden gestionar sus propias credenciales lo que permite a los usuarios configurar y gestionar sus propios parámetros de autenticación para acceder al sistema externo. Las credenciales denominadas admiten la autenticación de contraseña por OAuth.
Referencia	RF7

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 que se desarrollara para el motor de búsqueda Orión.

Tabla 8. Descripción CU Autenticar Usuario.

Caso de uso	Autenticar Usuario con OAuth
Autor	Cliente externo (aplicación tercera)

Resumen	<p>El caso de uso se inicia cuando un sistema externo o cliente solicita la petición de autorización a los recursos en el servidor utilizando credenciales temporales con la intención de obtener un “<i>access token</i>”.</p> <p>El servidor otorga esta autorización o “<i>access token</i>”, previa validación del usuario en su plataforma. Este paso involucra 2 formas de realizarlo:</p> <p>La primera es que la petición se realice de forma automática en la cual el protocolo OAuth redirigirá al usuario a la página de autenticación del servidor y una vez autenticado le pregunte por si desea otorgar dicha autorización al cliente.</p> <p>La segunda forma es la que el dueño del recurso o usuario previamente ha otorgado el permiso o <i>token</i> y lo suministra al cliente en su desarrollo o al llamar a las clases que lo implementan.</p> <p>El cliente envía la petición con el <i>access token</i> para realizar la operación con la API, u obtener información protegida.</p>
Referencias	RF1
Prioridad	Crítico
Flujo de eventos	
Acción de autor	Respuesta del sistema
1. El sistema externo realiza la petición del servicio para la autenticación, para ello introduce los datos necesarios (usuario, contraseña).	2. Verifica la validez de los datos. 3. Crea sesión y devuelve la clave de esta.
Flujo alterno	
Acción del autor	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 9. Descripción CU Devolver resultados de búsqueda.

Caso de uso	Realizar búsqueda general
Autor	Cliente externo (aplicación tercera)
Resumen	El caso de uso se inicia cuando un sistema externo o (usuario) desea consumir algunos de los servicios (haciendo uso del cliente externo) para obtener información del motor de búsqueda haciendo uso de sus servicios 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 en un mensaje JSON.
Referencias	RF2, RF2.1, RF2.2, RF2.3
Prioridad	Crítico
Precondiciones	El sistema (cliente externo) debe haber obtenido una clave de sesión o sea el usuario tiene que estar autenticado.
Flujo de eventos	
Acción de autor	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"> ✓ Realizar búsqueda general 	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 autor	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 general: <ul style="list-style-type: none"> ✓ Realiza búsquedas en la web. ✓ Realizar búsquedas de imágenes. 	2. Si los datos introducidos no son correctos se envía mensaje JSON con la información solicitada. 3. Si los datos introducidos no son correctos se envía mensaje JSON con los criterios de búsqueda en blancos

✓ Realizar búsquedas de documentos.	
Postcondiciones	Se obtiene respuesta del sistema.

Tabla 10. Descripción CU Gestionar credenciales OAuth.

Caso de uso	Gestionar credenciales OAuth
Autor	Cliente externo (aplicación tercera)
Resumen	<p>Las credenciales almacenan información de autenticación de sistemas externos. En el caso de uso se pueden administrar las credenciales guardadas que le dan acceso directo a los datos publicados. Estas credenciales representan <i>tokens</i> de acceso OAuth.</p> <p>Puede eliminar credenciales siempre y cuando no se estén utilizando en alguno de los componentes. Si alguna de las credenciales está en uso, aparecerá un mensaje de error. No se pueden eliminar el grupo de credenciales si alguna de las credenciales seleccionadas está en uso.</p>
Referencias	RF7
Prioridad	Crítico
Precondiciones	El sistema (cliente externo) debe haber obtenido una clave de sesión o sea el usuario tiene que estar autenticado.
Flujo de eventos	
Acción de autor	Respuesta del sistema
<p>1. El usuario gestionara las credenciales OAuth así como las actualiza y las elimina.</p> <p>✓ Gestionar credenciales de OAuth.</p>	<p>2. Verifica que el usuario este autenticado.</p> <p>3. Gestiona las credenciales</p>
Flujo alterno	
Acción del autor	Respuesta del sistema
	Conseguir el <i>token</i> de acceso necesario para poder acceder al servicio sin tener que en ningún momento

	introducir las credenciales de acceso en la aplicación cliente que desea acceder a los datos.
Postcondiciones	El usuario se autentica en nuestra aplicación, bien con un par usuario/contraseña. A partir de entonces, cada petición HTTP que haga el usuario va acompañada de un <i>token</i> en la cabecera. Este <i>token</i> no es más que una firma cifrada que permite a nuestro API identificar al usuario.

2.6 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 (Estrada Melgarejo, 2009). El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

2.6.1 Estilo arquitectónico

Un estilo arquitectónico es la organización fundamental de un sistema, las relaciones de los componentes con cada uno de los otros y con el entorno, y los principios que orientan su diseño y evolución.

Arquitectura del sistema

Symfony basa su funcionamiento interno en la arquitectura Modelo - Vista – Controlador (MVC), utilizada por la mayoría de frameworks web. No obstante, según su creador Fabien Potencier: "Symfony no es un framework MVC. Symfony sólo proporciona herramientas para la parte del Controlador y de la Vista. La parte del Modelo es responsabilidad del usuario (Fabien, 2011).

A través del controlador son recibidas y atendidas todas las peticiones al sistema. Cuando el controlador recibe una petición del usuario, consulta los datos almacenados en el mecanismo de indexación a través de un servicio desarrollado en Symfony. Los datos son almacenados en el mecanismo de indexación luego del recorrido realizado por el mecanismo de rastreo. Esta arquitectura está basada en el patrón arquitectónico MVC, donde la función del modelo la tienen los servicios de Symfony los cuales son los encargados de implementar la lógica del negocio gestionando todos los accesos a la información almacenada.

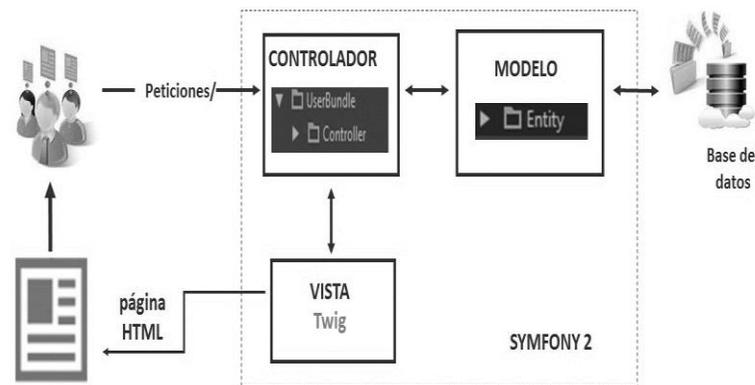


Figura 5. Arquitectura del sistema.

El controlador es el centro del módulo donde se implementa la lógica de negocio; responde a las peticiones que hace el usuario y se encarga de realizar las solicitudes de datos al modelo, en el sistema estas clases se encuentran en la carpeta Controller de cada uno de los bundles muestra de esto son los bundles APICommonBundle con la carpeta AbstractSearchController.php, y el bundle APIImageBundle con la carpeta ImageSearchController.php. En el modelo se encuentran las clases que permiten la representación de los datos del módulo, evitando mezclar los accesos a bases de datos con el código que implementa la lógica de negocio actuando como capa de abstracción entre el usuario y los datos, un ejemplo de esto se evidencia en la clase Client.php y AccessToken.php. La vista es la capa de presentación de los datos de la propuesta de solución, se limita a representar de manera gráfica los datos y todos los elementos que el usuario visualiza dentro la aplicación mediante los archivos con extensión .twig implementadas dentro de UserBundle/Resources/view/Credential.

2.6.2 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.

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 GRASP¹⁰ los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Larman, 2004).

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. En Symfony2 en la clase Controller.php se definen y ejecutan las acciones. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Controller.php es creador de dichas entidades.

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 (Larman, 2004).

Este patrón se evidencia en cada una de las clases, pues cada una de ellas tiene una responsabilidad bien definida. Se ve en concordancia con el bajo acoplamiento porque si las clases modelan toda la información asociada a un objeto, significa que hay poca dependencia de otras clases externas para su correcto funcionamiento. En el sistema se utiliza en la implementación de las clases entidades se evidencia en las clases Client.php y AccessToken.php.

Bajo acoplamiento:

Asigna una responsabilidad a una clase para mantener bajo acoplamiento, o sea, disminuir la dependencia entre clases, evitando que una modificación en alguna de ellas repercuta en gran medida en el resto, posibilitando además una mayor reutilización. En la solución la clase AuthorizeController.php hereda de Controller.php, que es una clase estable en cuanto a su implementación por lo que se obtiene un grado bajo de acoplamiento entre las clases. Además, al no asociar las clases del modelo con las de la vista o el controlador, la dependencia entre las clases, en este caso, se mantiene baja (Larman, 2004).

¹⁰ General Responsibility Assignment Software Patterns: Patrones de Software de Asignación de Responsabilidades

Polimorfismo:

Este patrón se puede decir que es el más importante patrón estratégico en el diseño orientado a objeto. Es un principio fundamental en que se fundan las estrategias globales o planes de ataque al diseñar como organizar un sistema para que se encargue del trabajo un diseño basado en la asignación de responsabilidades mediante el polimorfismo puede ser extendido para que realice nuevas variantes. Por ejemplo, agregar una nueva clase, un ejemplo de este patrón se evidencia en las clases `AbstractSearchController.php` y `AbstractAdvancedSearchController.php`. Uno de los beneficios que brinda este patrón estratégico es que permite agregar las futuras extensiones que requieren las variaciones imprevistas (Larman, 2004).

Experto:

Este patrón es el principio básico de asignación de responsabilidades. Pretende asignar una responsabilidad al experto en información, o sea, la clase que cuenta con información necesaria para cumplir la responsabilidad, ya sea crear un objeto o implementar un método. La utilización de este patrón dentro del sistema permite un diseño con mayor cohesión y se mantiene el encapsulamiento un ejemplo se puede ver en la clase `AbstractSearchController.php`, ayudando a entender y mantener el código fácilmente. El patrón experto fue utilizado en la capa de abstracción del modelo. Las clases generadas poseen un grupo de funcionalidades que facilitan el acceso y la manipulación de los datos de las entidades persistentes en la base de datos (Larman, 2004).

2.7 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.

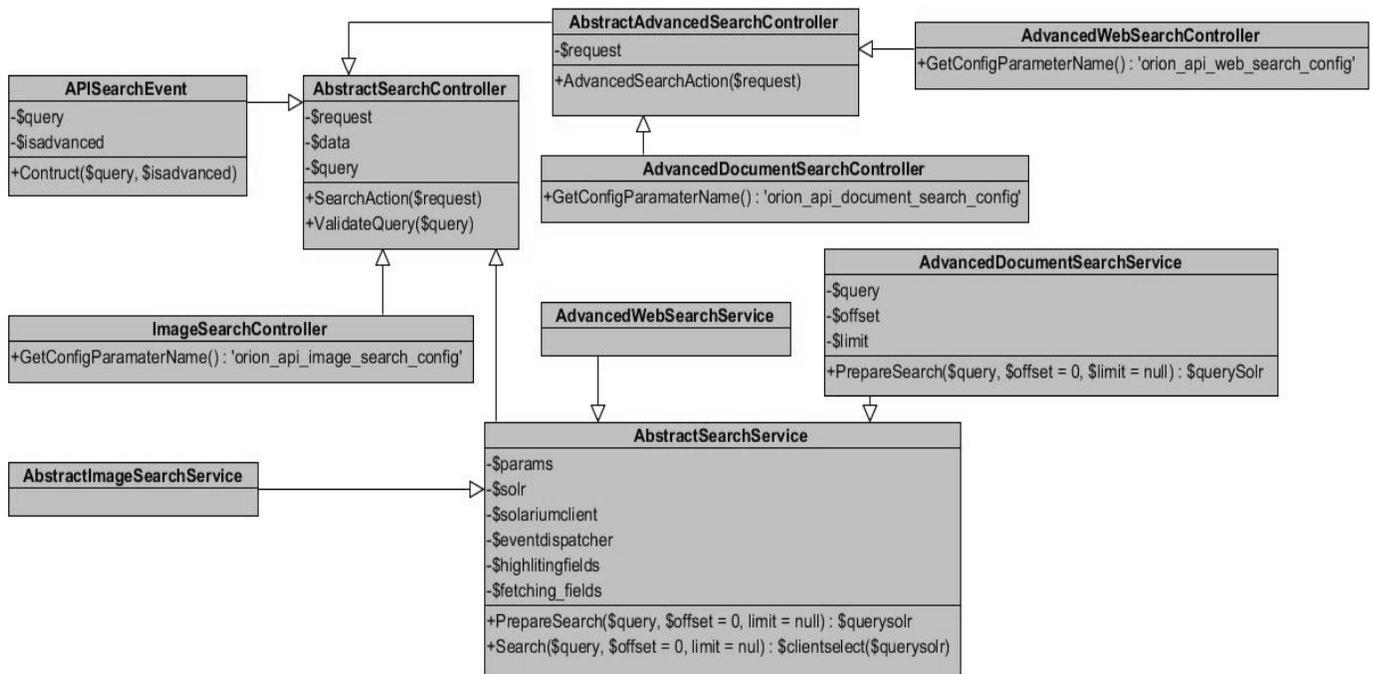


Figura 6 Diagrama de Clase de Diseño CU Realizar búsqueda.

2.8 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 servicios web en cuanto a estadísticas se refiere (consultas, direcciones IP, resultados más relevantes), así como los usuarios que tendrán acceso a los servicios web expuestos en la capa.

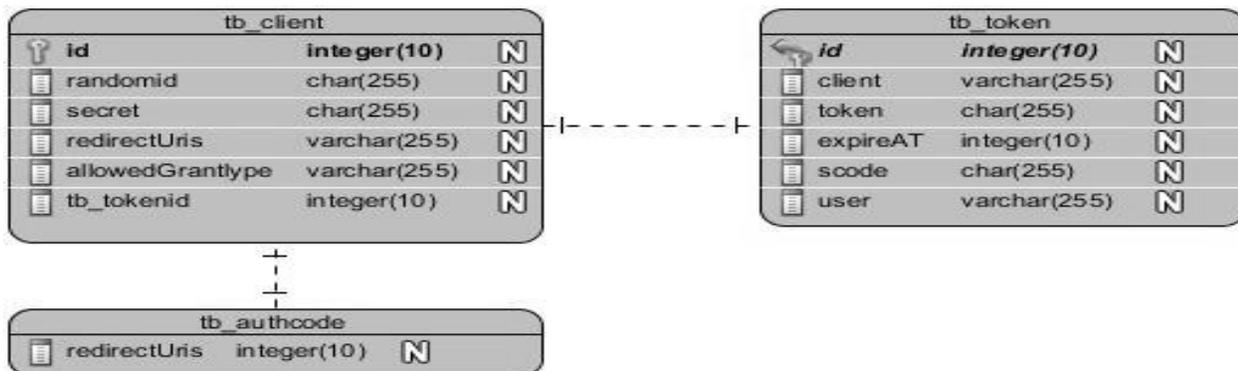


Figura 7 Diagrama de base de datos.

2.9 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.

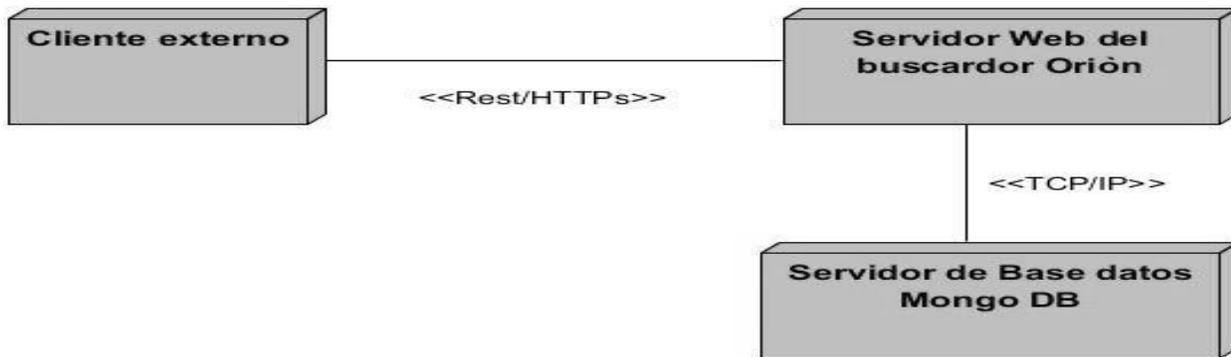


Figura 8 Diagrama de Despliegue.

Descripción de elementos e interfaces de comunicación:

- ✓ **Cliente externo:** Se necesita un navegador web para conectarse al módulo hospedado en el servidor de aplicaciones utilizando el protocolo de comunicación Rest/HTTPS.
- ✓ **Servidor web del buscador Orión:** Es la estación de trabajo que hospeda el código fuente de la aplicación y que le brinda al usuario las interfaces para realizar los procesos del sistema. Esta estación se comunica con el servidor de base de datos donde se almacenan los datos de la aplicación realizando la comunicación mediante el protocolo TCP/IP.
- ✓ **Servidor de BD:** Este servidor es el encargado del almacenamiento de los datos del sistema. Se comunica con el servidor de aplicaciones del sistema, posibilitando el acceso mediante el usuario con privilegios para las operaciones determinadas a realizarse en el mismo.
- ✓ **Rest/HTTPS:** HTTPS Protocolo para establecer a través del puerto 443 la conexión segura entre el dispositivo de acceso cliente y el servidor de aplicaciones. Cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST). En la actualidad se usa en el sentido más amplio para describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes, como por ejemplo SOAP.

- ✓ **TCP/IP:** Estos protocolos establecen la conexión entre el servidor de aplicaciones y el servidor de base de datos. Para el servidor de base de datos de MongoDB se define el puerto 27017. La conexión entre el servidor web y el servidor de base de datos permite dar órdenes y obtener información de esta.

2.10 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 requisitos funcionales y no funcionales que debe cumplir la capa de servicios web a desarrollar 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.

CAPITULO 3: Implementación y prueba del sistema

3.1 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.

3.2 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.

3.2.1 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.

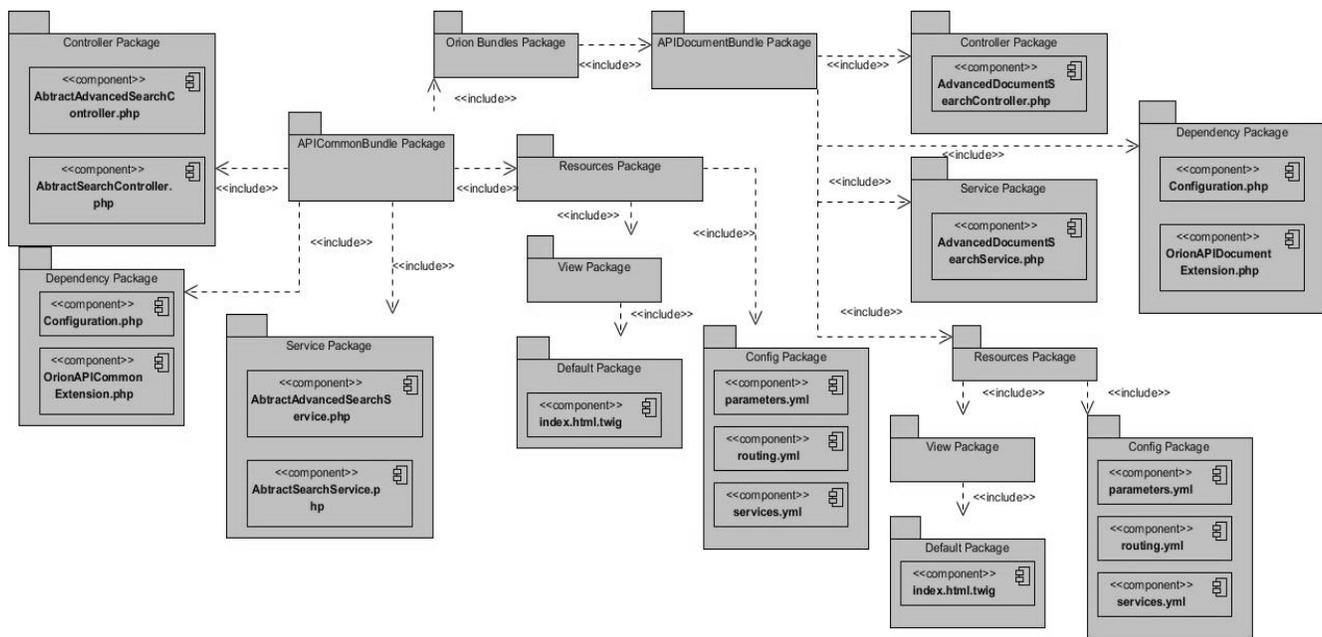


Figura 9 Diagrama de Componentes de CU Realizar búsqueda de documentos.

3.3 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.

Para la implementación de la capa de servicios web, como se definió anteriormente, se seleccionó el lenguaje de programación PHP. La propuesta de solución cumplirá con los estándares de código seguidos por Symfony2 PSR-0, PSR-1, PSR-2 y PSR-4.

3.4 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 el usuario se autentica en nuestra aplicación, bien con un par usuario/contraseña, a partir de entonces, cada petición HTTP que haga el usuario va acompañada de un *token* en la cabecera. Este *token* no es más que

una firma cifrada que permite a nuestro API identificar al usuario. Esta clave va a ser única .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.

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.

3.5 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.

Tabla 11. Respuesta del sistema ante errores.

Error	Causa	Respuesta del Sistema
Identidad Incorrecta	Los datos (usuario y contraseña) que fueron introducidos para obtener un <i>token</i> de sesión no coinciden con lo 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.
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.

3.6 Validación del sistema

La validación del sistema incluye un conjunto de actividades para asegurar que el software desarrollado se corresponde con los requisitos del cliente. Dentro de estas actividades se encuentran las pruebas de

validación las cuales tienen como objetivo evaluar la calidad y de manera más pragmática, descubrir errores en el sistema.

Para esto se traza un modelo de pruebas que describe cómo se prueban los componentes ejecutables en el modelo de implementación con pruebas de integración, pruebas funcionales, pruebas de carga y estrés, pruebas de rendimiento y pruebas de seguridad. Es una colección de casos de prueba, procedimientos de prueba y componentes de prueba (Pressman, 2006).

3.6.1 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 (Perez Rivas Nadiesda, Otros, 2007).

3.6.2 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**

Las pruebas funcionales son aquellas que se aplican a un software determinado, con el objetivo de validar que las funcionalidades implementadas funcionen de acuerdo a las especificaciones de los requisitos definidos con anterioridad. Para la ejecución de este tipo de pruebas, suelen emplearse dos métodos fundamentales: el método de Caja Blanca y el método de Caja Negra. El primero se centra en las pruebas al código de las aplicaciones; mientras que el segundo permite a los probadores enfocar su atención en el funcionamiento de la interfaz, a través del análisis de los datos de entrada y los de salida. En este epígrafe se exponen los aspectos concernientes a las pruebas funcionales realizadas utilizando el método de Caja Negra. Se selecciona entre sus distintas técnicas la partición equivalente (Pressman, 2006).

Diseño de los casos de prueba basados en casos de uso

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (Pressman, 2006).

Técnica a emplear

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?

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 Realizar Búsqueda.

Tabla 12. Caso de Prueba CU Realizar búsqueda.

Caso de Uso: CU_2 Realizar Búsqueda
Requisitos: RF2, RF3, RF4, RF5, RF6.
Condición: Debe tener el <i>token</i> de seguridad asignado al existir el usuario en la base de datos para poder ejecutarse la búsqueda.

Descripción de las variables

Tabla 13. Descripción de variables de prueba CU Realizar Búsqueda.

Parámetro	Valor Nulo	Descripción
q	Si	Consulta de la búsqueda
type	No	Indica el formato de documentos a buscar(doc,pdf)
Limit	No	Indica el número de resultados a mostrar.
host	No	Indica el sitio en el cual se realizará la búsqueda.
offset	No	Numero de página a consultar

Tabla 14. Matriz de Datos CU Realizar búsqueda.

Escena rio	Descripción	q	Limit	type	host	offset	Respuesta del Sistema	Flujo central
EC 1 Búsqueda Simple	Realizar búsqueda de un término determinado.	cuba	nulo	nulo	nulo	o	Documento JSON con la estructura definida previamente que muestra los resultados para el	Enviar la URL por post con el parámetro q=cuba

							criterio de búsqueda "cuba" a partir de la página 0.	
EC 1.1	Realizar búsqueda sin criterio.	nulo	nulo	nulo	nulo	nulo	Muestra un error devolviendo un documento JSON con los parámetros de búsqueda en blanco.	Enviar la URL por post con el parámetro q vacío.
EC 2	Búsqueda personalizada	cuba	10	pdf	dragones.uci.cu	10	Documento JSON con la estructura definida previamente que muestra los resultados para el criterio de búsqueda q="cuba" type=pdf, host=dragones.uci.cu a partir de la página 10.	Enviar la URL por post con el parámetro q=cuba, type=pdf, host=dragones.uci.cu, offset=10.

Resultados de las pruebas funcionales

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.

En la Tabla 17 se muestran los resultados obtenidos en cada iteración de prueba a la capa de servicios web para el motor de búsqueda Orión, así como la corrección de cada uno de los errores.

Tabla 15. Cantidad de no conformidades por cada iteración de las pruebas funcionales.

No Conformidades	1era iteración	2da iteración	3era iteración	4ta iteración
Detectadas	10	8	3	0
Resueltas	10	8	3	0

Durante su desarrollo se detectaron un total de 21 no conformidades que fueron corregidas satisfactoriamente. Los principales errores detectados fueron de tipo ortográfico, de tipo diseño donde se muestran errores de estructuración del contenido mostrado en la vista de las credenciales.

A continuación se muestra un gráfico donde se evidencian los resultados obtenidos durante las cuatro iteraciones de pruebas funcionales realizadas al sistema desarrollado.

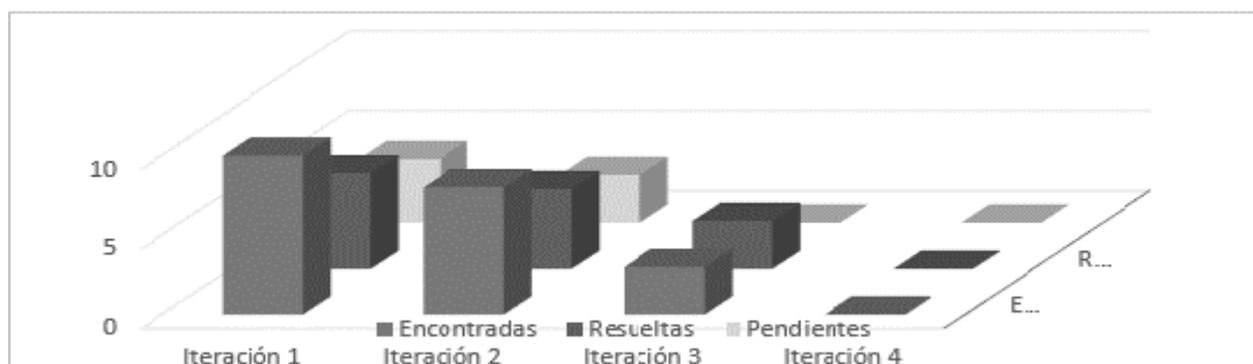


Figura 10 Comportamiento de las no conformidades por cada iteración de las pruebas funcionales.

✓ **Pruebas de integración**

Una vez realizadas las pruebas funcionales a cada componente interno de manera independiente, y verificado que las funcionalidades implementadas se corresponden de acuerdo a los requisitos funcionales y no funcionales establecidos; se pudo comprobar el correcto funcionamiento de los componentes mediante el estudio del flujo de datos entre ellos. Posterior a estas pruebas, se hace necesaria la realización de pruebas de integración (Sommerville, 2007).

Estas pruebas son una forma de comprobar la correcta interrelación de los distintos componentes del sistema. En el caso de la solución desarrollada, es la verificación de una correcta interoperabilidad entre el sistema desarrollado y el buscador Orión, validando la compatibilidad y el funcionamiento de las diferentes partes que componen la solución informática.

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

- ✓ Integración entre la capa de servicios web con el motor de búsqueda Orión.
- ✓ Verificación de la conexión entre el sistema y solr.

Se muestra, en la Tabla 18, los resultados de la ejecución de las tres iteraciones de las pruebas de integración realizadas al sistema:

Tabla 16. Cantidad de no conformidades por cada iteración de las pruebas de integración.

No Conformidades	1era iteración	2da iteración	3era iteración
Detectadas	5	3	0
Resueltas	5	3	0

La ejecución de las pruebas de integración permitió verificar el trabajo conjunto de los componentes de la herramienta en cuestión. Se hizo énfasis en el mecanismo de indexación Solr, para detectar incoherencias en el funcionamiento de la aplicación, encontrándose un total de ocho no conformidades. Los principales errores detectados fueron al integrar la capa de servicios web con el mecanismo de indexación Solr ya que al mostrar el resultado de las búsquedas realizadas se detectaron incoherencias y errores de contenido en los datos mostrados. Las no conformidades fueron corregidas en su totalidad llegándose a la conclusión que existe una correcta integración entre los componentes internos de la herramienta.

Pruebas de rendimiento

Las pruebas de carga consisten en probar el funcionamiento del software bajo condiciones extremas. Mediante las pruebas de rendimiento es posible hallar tendencias y comportamientos para los elementos de una aplicación. Este tipo de pruebas permite identificar cuellos de botella, capacidad de concurrencia de usuarios, tiempos de respuesta de operaciones del negocio a nivel de sistema, establecer un marco de referencia para pruebas futuras, determinar el cumplimiento de los objetivos de rendimiento y requerimientos no funcionales, entre otros (Quality, 2013).

Dependiendo del objetivo perseguido se puede diseñar distintos tipos de pruebas, entre ellas se puede encontrar las pruebas de carga y las pruebas de estrés. Una **prueba de carga** se realiza generalmente para determinar y validar la respuesta de la aplicación cuando es sometida a una carga de usuarios o transacciones que se espera en el ambiente de producción (Rincon, 2015).

Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga.

Las **pruebas de estrés** se utilizan para encontrar el volumen de datos o de tiempo en que la aplicación comienza a fallar o es incapaz de responder a las peticiones (Rincon, 2015). Una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del sistema en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y procesos, que se verá reflejada en una óptima operación de negocio.

Resultados de las pruebas de rendimiento

Para la realización de las mismas se utiliza la herramienta *JMeter* v2.8.4. La prueba realizada consistió en definir las pruebas con 150 hilos de concurrencia, las cuales simulan 150 accesos de usuarios respectivamente.

Hardware:

- ✓ Tipo de procesador: corei3 3110M a 2.40 GHz.
- ✓ Memoria: 2GB RAM.
- ✓ Tipo de Red: Ethernet 10/100Mbps.

Software:

- ✓ Tipo de Servidor Web: Apache v2.4.
- ✓ Máximo de hilos concurrentes (simulación de usuarios): 150.

- ✓ Plataforma: sistema operativo Ubuntu 16.04 de 64 bits.
- ✓ Servidor de BD: MongoDB 2.6.6.

Para un mejor entendimiento de las componentes “Reporte de Rendimiento”, se explica cada parámetro que la compone a continuación:

- ✓ **Usuarios:** total de usuarios.
- ✓ **# Muestras:** El número de peticiones.
- ✓ **Media:** El tiempo medio transcurrido para un conjunto de resultados.
- ✓ **Mín:** El mínimo tiempo transcurrido en milisegundos para las muestras de la URL dada.
- ✓ **Máx:** El máximo tiempo transcurrido en un milisegundo para las muestras de la URL dada.
- ✓ **% Error:** Porcentaje de las peticiones con errores.
- ✓ **Rendimiento:** Rendimiento medido en base a peticiones por segundo/minuto/hora.

Tabla 17. Resultados de pruebas de carga y estrés.

Usuarios	# Muestras	Media	Mín	Máx	% Error	Rendimiento. (peticiones/segundos)
150	2010	2.050	2	6.025	0	10.5

El sistema desarrollado, para un total de 150 usuarios conectados de forma concurrente respondió 2010 peticiones al servidor en un promedio de 2.050 segundos, lo que equivale a 10.5 peticiones por segundo. Atendiendo a la cantidad de peticiones por cada segundo que se enviaron, las prestaciones del hardware donde se realizaron las pruebas, se considera que, de manera general, es un resultado satisfactorio para el autor de la presente investigación.

✓ Prueba de seguridad y control de acceso

Se considera un producto (de software) seguro aquel que protege la confidencialidad¹¹, integridad¹² y disponibilidad¹³ de los recursos y el servicio, bajo el control del propietario o administrador del sistema (LEBLANC, 2006).

El objetivo principal de la prueba y análisis de la seguridad de software es la comprobación de que el software exhibe las propiedades y conductas deseables siguiente (Vega, 2012) :

¹¹ Sólo los "usuarios" debidamente autorizados deben tener acceso a los datos y/o los recursos.

¹² Sólo los usuarios debidamente autorizados pueden modificar los datos y/o los recursos de modo apropiado y controlado.

¹³ Los recursos deben poder ser accedidos cuando son necesarios y funcionar a un nivel aceptable.

- ✓ Su comportamiento es predecible y seguro.
- ✓ No expone ninguna vulnerabilidad o debilidades.
- ✓ Las rutinas que se ocupan de los errores y excepciones le permite que mantenga un estado seguro frente a modelos de ataque o fallas intencionales.
- ✓ Satisface todos los requisitos no funcionales de seguridad especificados e implícitos.
- ✓ No viola ninguna restricción de seguridad especificada.
- ✓ El código ha sido disimulado para evitar la ingeniería inversa.

En sentido general las pruebas de seguridad y control de Acceso se enfocarán 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.

Al sistema desarrollado se le realizaron una serie de pruebas de seguridad mediante el software Acunetix, las cuales se presentan a continuación:

- ✓ Ataques de inyección.
- ✓ Falsificación de petición (CSRF).
- ✓ Peticiones vacías.

Se encontraron 5 alertas en total, de ellas ninguna de clasificación alta, 1 de clasificación media, 4 de clasificación baja y. De manera general los resultados obtenidos se presentan en la tabla 20.

Tabla 18. Vulnerabilidades del entorno.

Tipo	Cantidad	Descripción	Recomendaciones
Software	1	La versión de PHP instalada contiene vulnerabilidades que han sido solucionadas en versiones más recientes.	Actualizar la versión por una superior.
Configuración	4	La configuración del servidor permite la	Esta vulnerabilidad aunque no permite al atacante tomar el control acerca de la información a obtener puede

		captación de información sensible por el atacante.	proporcionar al atacante información sensible que puede utilizar para realizar ataques más específicos, pueden solucionarse limitando el acceso a los archivos del servidor y los elementos de la configuración
--	--	--	---

Resultados de las pruebas de seguridad

La realización de pruebas de seguridad contribuye a la detección temprana de vulnerabilidades y la toma de medidas para la disminución de amenazas de ataque. Todas las vulnerabilidades fueron resueltas en el servidor y se recomienda que se tenga en cuenta una nueva configuración en el servidor para un futuro despliegue del sistema, llegando a la conclusión que el sistema desarrollado es seguro y está en condiciones de ser usado por el cliente con el propósito de consumir los servicios que brinda el motor de búsqueda Orión.

3.8 Validación de la hipótesis de la investigación

Para la validación de la hipótesis de investigación se utiliza el método criterio de expertos en su variante Delphi empleando los siguientes pasos (Sanchez, 2015):

- ✓ Identificación de los posibles expertos.
- ✓ Selección de los expertos.
- ✓ Realización de la consulta a los expertos y procesamiento y valoración de la información obtenida.

Para identificar los posibles expertos se tuvieron en cuenta, la experiencia profesional en relación con el objeto de investigación, la participación en investigaciones relacionadas con esta temática, el dominio teórico de la temática, la preparación académica y científica, y la experiencia, de modo que estuvieran en capacidad de ofrecer valoraciones y hacer recomendaciones pertinentes, en relación con los aspectos que le serían consultados. Se identificaron 10 posibles expertos.

Selección de expertos

La selección de los expertos, se inició con la aplicación de una encuesta a los posibles expertos para determinar su coeficiente de competencia (K), el cual se obtiene de la expresión: $K = (kc + ka) / 2$.

- **kc**: coeficiente de conocimiento o información.
- **ka**: coeficiente de argumentación.

Esta tarea se ejecuta mediante la autovaloración de los especialistas seleccionados sobre los niveles de información y argumentación que tienen sobre el tema planteado. De los 10 especialistas valorados se seleccionaron 8 expertos cuyo coeficiente de competencia (K) resultó Alto y Medio. Los 8 expertos son graduados universitarios y han obtenido títulos de formación académica.

Realización de la consulta a expertos

Después de contar con los 8 expertos seleccionados se sometió a su consideración un instrumento para la validación de la capa de servicios web para el motor de búsqueda. El instrumento se compone de 5 categorías evaluativas que permitan conocer la opinión de los expertos. Las categorías evaluativas empleadas fueron muy adecuado (MA), bastante adecuado (BA), adecuado (A), poco adecuado (PA) e inadecuado (I). La variable independiente se evaluó mediante las sentencias de interfaz intuitiva, visualización de consultas realizadas y recopilación de datos del usuario durante la interacción con el servidor del buscador mediante la utilización de la capa de servicios web, mientras que, la evaluación de la variable dependiente midió el nivel de la seguridad y el control de la integración del motor de búsqueda con aplicaciones terceras.

Análisis de la concordancia de las valoraciones de los expertos

En un segundo momento se calcula el coeficiente de Kendall que permite analizar la concordancia en las valoraciones realizadas por los expertos (Sampieri, 2014). En este caso el coeficiente concordancia (W) será un índice de la divergencia del acuerdo efectivo entre los expertos.

El coeficiente de concordancia de Kendall se obtiene de la expresión:

$$W = 12S / K^2(N^3 - N).$$

$$W = 12S / K^2(N^3 - N)$$

$$W = 12 * 30.08 / 8^2(2^3 - 2)$$

$$W = 0,94$$

- ✓ **S:** cuadrado de las desviaciones medias.
- ✓ **K:** número de expertos.
- ✓ **N:** número total de aspectos a evaluar.

El valor de W oscila entre 0 y 1. El valor de 1 significa una concordancia de acuerdos total y el valor de 0 un desacuerdo total. Como **W = 0.94** existe concordancia de acuerdos entre los expertos.

Se aplica además la prueba de significación de hipótesis para comprobar el grado de significación de Kendall, planteándose la hipótesis nula y la alternativa (H0: no existe concordancia entre los expertos y H1: existe concordancia entre los expertos).

El cálculo del Chi cuadrado real: $x^2 = K(N - 1)W$

$$x^2 = 8(2 - 1)0,94$$

$$x^2 = 7,52$$

El Chi cuadrado calculado se compara con el tabulado en la tabla del percentil de la distribución Chi cuadrado. Para tener un 95% de confianza se utilizará $\alpha = 0.05$. Si se cumple que $x^2_{calculada} < x^2(\alpha, N - 1)$, entonces se infiere que existe concordancia de criterios entre los expertos al considerar válida la hipótesis alternativa H1. Como $7,52 < 33,924$ existe concordancia entre los expertos.

Procesamiento estadístico

Los criterios aportados por los expertos se someten a una prueba estadística no paramétrica que permite concluir qué valoración final tiene cada uno de los aspectos a evaluar. Para los datos anteriores se debe confeccionar una distribución de frecuencia a partir de los datos primarios para cada uno de los aspectos sometidos a consulta.

Tabla 19. Distribución de frecuencia.

Categorías evaluativas	Frecuencia Absoluta	Frecuencia Relativa
Muy Adecuado	15	0.80
Bastante Adecuado	3	0.10
Adecuado	3	0.10
Poco adecuado	0	0
Inadecuado	0	0

De acuerdo con los datos el 80% de los aspectos analizados fueron valorados de muy adecuado, el 10% de bastante adecuado, el 10% de adecuado. El análisis de los resultados obtenidos de la consulta de expertos permitió identificar que existe una coincidencia en las valoraciones realizadas sobre el alto valor que presenta la capa de servicios web para el buscador Orión. Además, todos los indicadores fueron evaluados satisfactoriamente evidenciando la calidad de la propuesta presentada.

3.9 Conclusiones parciales

En este capítulo se abordaron una serie de aspectos correspondientes a la implementación y validación de la capa de servicios web llegándose a la siguiente conclusión:

- ✓ 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 elaboración de los artefactos propuestos por la metodología de desarrollo y el levantamiento de requisitos permitieron una mayor comprensión, identificación de los procesos y características del sistema desarrollado.
- ✓ El sistema implementado permitió solucionar los problemas existentes planteados en la problemática de la presente investigación relacionada con los servicios web.
- ✓ Las pruebas de software aplicadas a la capa de servicios web para el buscador Orión, demostraron que es una solución funcional, segura, con un rendimiento adecuado y que se integra sin dificultad al buscador Orión.

Recomendaciones

Una vez concluida la investigación y el desarrollo de la propuesta de solución, se recomienda:

- ✓ Integrar la capa de servicios web realizada en el entorno real de Orión y a la Plataforma C.U.B.A.

Referencias

- Google Custom Search . 2013. Using REST to Invoke the API . [En línea] 26 de Septiembre de 2013. <https://developers.google.com/custom-search/docs/overview>.
- Amazon Web Services, Inc. 2017. Amazon Web Services. [En línea] 2017. <https://aws.amazon.com/es/elasticsearch-service/>.
- Apache JMeter. 2016. Apache JMeter. [En línea] 2016. <http://jmeter.apache.org/usermanual/>.
2010. Apache Solr: una introducción. *Apache Solr: una introducción*. [En línea] 11 de Junio de 2010. <https://dosideas.com/noticias/java/913-apache-solr-una-introduccion>.
- Burke, Bill. 2010. *RESTful Java with JAX-RS. s.l. . O'Reilly, : s.n., 2010. 978-0-596-15804-0,179-185*.
- CADAVID, A. 2013. Revisión de metodologías ágiles para el desarrollo de software. [En línea] 2013. <http://ojs.uac.edu.co/index.php/prospectiva/article/viewFile/36/21>. vol. 11, no 2, p. 30-39..
- Chodorow, Kristina. 2014. *The Definitive Guide*. 2014. ISBN:978-1-449-34468-9.
- CollegiateLink Help Center. 2016. API (Version 2.0) - Introduction to the API. [En línea] 2016. <http://support.collegiatelink.net/hc/en-us/articles/204034074-API-Version-2-0-Introduction-to-the-API>.
- Craig, Larman. 2012. *UML y Patrones*. Montevideo-Uruguay : s.n., 2012.
- Dpto.de ciencias de la computación e inteligencia artificial. 2014. Introducción a los Servicios Web. [En línea] 26 de Junio de 2014. <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion02-apuntes.html>.
- . 2014. Introducción a los Servicios Web. [En línea] 26 de Junio de 2014. <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion02-apuntes.html>.
- . 2014. Invocación de servicios web SOAP. [En línea] 26 de 06 de 2014. <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.html#Servicios+Web+SOAP+y+Java+EE>.
- E-business corporativo: cómo implantar software libre, servicios web y el ...* García Valcárcel, Ignacio y Munilla Calvo, Eduardo. 2003. España : s.n., 2003. ISBN:84-95428-98-9.
- Ecma. 2013. *The JSON Data*. 2013.
- El web como sistema de información*. RODRIGUEZ PEROJO, Keilyn y RONDA LEON, Rodrigo. 2006. 1, 2006, ACIMED, Vol. 14. ISSN 1024-9435.
- Engineers, Institute of Electrical and Electronics. 1990. *IEEE Standard Computer Dictionary:A Compilation IEEE Standard Computer Glossaries*. New York : s.n., 1990.
- Estrada Melgarejo, Yadira y Jorge Martínez, Yasiel. 2009. *Servicios Web para la creación de aplicaciones SEO*. Cuba,Universidad de las Ciencias Informatica : s.n., 2009.
- Fabien, Potencier. 2011. *Arquitectura de software*. 2011.
- Fraga Quevedo, Jorge. 2015. *Subsistema de promoción de anuncios del motor de búsqueda ORION*. La Habana : s.n., 2015. 1.

- González, Enrique. ¿QUÉ ES PHP? ¿PARA QUÉ SIRVE PHP? ¿QUÉ ES PHP? ¿PARA QUÉ SIRVE PHP? [En línea] http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=501:comunicacion-cliente-servidor-sin-php-y-con-php-interprete-php-y-gestor-de-bases-de-datos-cu00804b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193.
- Hernández, Pedro. 2016. Tendencias de Web aplicadas a la educación en línea. *Tendencias de Web aplicadas a la educación en línea*. [En línea] 15 de Marzo de 2016. [Citado el: 4 de 12 de 2016.] <http://www.nosolousabilidad.com/articulos/web20.htm>.
- . 2016. Tendencias de Web aplicadas a la educación en línea. [En línea] 15 de Marzo de 2016. <http://www.nosolousabilidad.com/articulos/web20.htm>.
- Inc., Sun Microsystems. *JAVA CODE CONVENTIONS*.
- Jalón Garcia Javier, otros. 2000. *Aprenda java como si estuviera en primero*. 2000.
- Javier, Equiluz. 2011. *Desarrollo web agil con synfony 2*. 2011.
- Kabir, Mohammed J. 2004. *La biblia de Servidor Apache2*. 2004. 39-50.
- Lamarca, María. 2013. El nuevo concepto de documento en la cultura de la imagen. [En línea] Tesis doctoral. Universidad Complutense de Madrid., 08 de 12 de 2013. <http://www.hipertexto.info>.
- Larman, C. 2004. *UML y Patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda. s.l. : Prentice Hall, 2004. pág. 520.
- Larman, Craig. 2004. *UML y Patrones.Introduccion al analisis y diseño orientado a objeto*. La Habana : Felix Varela, 2004.
- . 1999. *UML y Patrones: Introducción al análisis y diseño orientado a objeto*. 1999.
- LEBLANC, y otros. 2006. *Writing Secure Code*. 2006. ISBN-13: 978-0735617223..
- Ledford, Jerri L. 2008. *Search Engine*. Indianápolis : Wiley Publishing : s.n., 2008. ISBN 978-0-470-45264-6..
- Ledford, Jerri L. 2008. *Search Engine*. Indianápolis : Wiley Publishing, 2008. ISBN 978-0-470-45264-6.
- López Gamino, Felipe. 2001. *Introducción a los sistemas de bases de datos*. Mexico : s.n., 2001.
- Marqués, Asier. 2013. Apuntes personales, desarrollo de software y negocios en Internet.Conceptos sobre APIs REST. [En línea] 2013. <http://asiermarques.com/2013/conceptos-sobre-apis-rest/>.
- MARTÍNEZ, MIGUEL VITALA. 2014. Artículos y tutoriales sobre php. *Artículos y tutoriales sobre php*. [En línea] 17 de Marzo de 2014. Disponible en: <http://www.miguelvilata.com/blog/tag/php>.
- McKendrick, Joe. 2007. Why Point-to-Point Integration Costs More than SOA. [En línea] 2007. http://www.ebizq.net/blogs/soainaction/2007/04/why_point-to-point_integration.php.
- Mejia, Dante. 2015. REST VS SOAP. [En línea] 26 de Octubre de 2015. <http://restysoap.blogspot.com/feeds/posts/default>.

- Meyers, Chris. 2015. How We Chose MongoDB for Ansible Tower System Tracking Feature. [En línea] 30 de Julio de 2015. <https://www.ansible.com/blog/how-we-chose-mongodb-for-ansible-tower-system-tracking-feature>.
- Momjian, Bruce. 2002. *PostgreSQL: Introduction and Concepts*. 2002.
- MongoDB, Inc. 2017. MongoDB. [En línea] 2017. <https://www.mongodb.com/cloud/atlas?jmp=homepage&jmp=homepage>.
- Morales Machuca, Carlos Andrés. 2008. *Estado Del Arte: Servicios Web*. Colombia : s.n., 2008.
- Morales Machuca, MSc Carlos Andrés. 2008. *Estado Del Arte: Servicios Web*. Universidad Nacional de Colombia : s.n., 2008.
- O'Phinney, Matthew Weier. 2016. Zend Framework 3 Update. [En línea] Marzo de 2016. <https://framework.zend.com/blog/2016-03-24-zf3-update.html>.
- Oracle. 2014. *NetBeans Developing Applications with NetBeans*. 2014.
- Oscar, Garcia. 2012. [OAuth] – Autorización segura de APIs con OAuth (Open Authorization). [En línea] 23 de abril de 2012. <http://www.elclubdelprogramador.com/2012/04/23/oauth-autorizacion-segura-de-apis-con-oauth-open-authorization/>.
- Patricia, Lopez. 2015. *Herramienta case visual paradigm*. 2015.
- Peñalver Romero, Gladys M. 2008. *Metodología ágil para proyectos de software libre*. . Habana : s.n., 2008. MAGMPR-UR2.
- Perez Rivas Nadiesda, Otros. 2007. Pruebas de Aceptación Parciales del cliente. [En línea] 2007. http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_0479_07..
- Pressman, R S. 2006. *Ingeniería del software. Un enfoque práctico*. 2006. Vols. ISBN 970-10-5473-3.
- QATécnico. 2012. QATécnico. [En línea] 3 de marzo de 2012. [Citado el: 16 de marzo de 2017.] <http://qatecnico.blogspot.com/2012/03/pruebas-de-rendimiento-tipos-y.html>.
- Quality, V&V. 2013. *Política de Privacidad*. 2013.
- Quatrani, Terry. 2002. *Visual Modeling with Rational Rose*. 2002.
- Ramirez, Antonio. 2014. Yiiframework. [En línea] Junio de 2014. <http://www.yiiframework.com/about/>.
2015. REST VS SOAP. REST VS SOAP. [En línea] 26 de octubre de 2015. <http://restysoap.blogspot.com/2015/10/normal-0-21-false-false-false-es-pe-x.html>.
- Rincon. 2015. *Pruebas de carga y estrés*. 2015.
- Rincon, Mafe. 2015. Prezi. [En línea] 30 de abril de 2015. [Citado el: 16 de marzo de 2017.] <https://prezi.com/hnxdes2szvdp/pruebas-de-carga-y-estres/>.
- Rodríguez Sánchez, Tamara. 2013. *Metodología de desarrollo para la*. La Habana.Cuba : Universidad de las Ciencias Informáticas, 2013.

- Ronda Leon, Rodrigo. 2008. *El web como sistema de información*. 2008. ISSN 1024-9435..
- Rossini, Daniel. 2010. *LOS ARCHIVOS Y LAS NUEVAS TECNOLOGIAS DE LA INFORMACION*. 2010.
- . 2010. *LOS ARCHIVOS Y LAS NUEVAS TECNOLOGIAS DE LA INFORMACION*. 2010.
- Sampieri, R. 2014. *Metodología de la investigación*. 2014.
- Sanchez. 2015. *Estrategia de soporte técnico para el proceso de migración a código abierto en los Organismos de la Administración Central del Estado*. La Habana, Universidad de las Ciencias Informáticas : s.n., 2015.
- Singh, Tarandeep. 2011. REST vs. SOAP – The Right Webservice. [En línea] 21 de noviembre de 2011. <http://geeknizer.com/rest-vs-soap-using-http-choosing-the-right-webserviceprotocol..>
- SMTP. 2006. *SMTP extension for email delivery failure*. 2006. US7093025 B1.
- Sommerville. 2007. *Software Engineering*. 2007.
- StatCounter. 2011. Top 5 Search Engines from . [En línea] 20 de Noviembre de 2011. http://gs.statcounter.com/#search_engine-wwmonthly-201010-201110-bar..
- Tarun's Blog. 2012. The History of Yahoo! - How It All Started. [En línea] 18 de Septiembre de 2012. <https://tarunrattan.wordpress.com/.../the-history-of-yahoo-how-it->
2012. The Apache Software Foundation. Apache Axis2/Java Version 1.6.2 Documentation Index. *The Apache Software Foundation. Apache Axis2/Java Version 1.6.2 Documentation Index*. [En línea] 2012. <http://axis.apache.org/axis2/java/core/docs/contents.html..>
- Tolosa, Gabriela H. 2014. *Protocolos y modelos OSI*. 2014.
- Valero Menacho, Sara. 2015. Desarrollo de una interfaz web para la composición de servicios REST. [En línea] 26 de Noviembre de 2015. <http://repositori.uji.es/xmlui/handle/10234/149706>.
- Vega. 2012. *Guía metodológica para implementar la seguridad durante el desarrollo de aplicaciones informáticas*. La Habana : s.n., 2012.
- Vega, Cutiño Ruth Yurina. 2012. *Guía metodológica para implementar la seguridad durante el desarrollo de aplicaciones informáticas*. La Habana : s.n., 2012.
- Visual Paradigm. 2010. Visual Paradigm. Visual Paradigm for UML. [En línea] 2010. [http://www.visual-paradigm.com/..](http://www.visual-paradigm.com/)
- W3C. 2011. Hypertext Transfer Protocol Overview. [En línea] 20 de noviembre de 2011. [http://www.w3.org/Protocols/..](http://www.w3.org/Protocols/)
- Webber, Jim. 2010. *Rest in Practice. Hypermedia and System Architecture*. 2010.
- Yigal, Asaf. 2016. Solr vs. Elasticsearch. [En línea] 19 de 27 de 2016. <https://logz.io/blog/solr-vs-elasticsearch/>.