

Universidad de las Ciencias Informáticas



Facultad 1

Directorio Temático para el buscador Orión

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor: Rubén Roberto Peña Domínguez

Tutores: Ing. Miguel Ángel Chávez Alfonso
Ing. Geidis Sánchez Michel
Ing. Eyeris Rodríguez Rueda

La Habana, junio, 2016

Declaratoria de autoría

Yo, Rubén Roberto Peña Domínguez, declaro ser el autor del presente trabajo de diploma y reconozco a la Facultad 1 de la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Autorizo a dicha institución universitaria para que haga el uso que estime pertinente de este trabajo.

Para que así conste firman los presentes a los ____ días del mes de _____ del año _____.

Rubén Roberto Peña Domínguez

Autor

Ing. Eyeris Rodríguez Rueda

Tutor

Ing. Miguel Ángel Chávez Alfonso

Tutor

Ing. Geidis Sánchez Michel

Tutor



"Hacer es la mejor forma de decir"

José Martí

Agradecimientos:

A mis tutores, especialmente a Miguel Ángel Chávez Alfonso y Geidis Sánchez Michel por todo su empeño, dedicación, preocupación y exigencia, sin los cuales nada hubiese sido posible. A toda mi familia, en especial a mis padres y a mi abuelita Isabel, por siempre confiar en mí, en el logro de este paso tan importante en mi vida y apoyarme incondicionalmente en todos los sentidos. A mi pareja por su comprensión, dedicación y por acompañarme en cada momento de esta etapa tan difícil. A mis segundos padres que siempre me han ayudado y acogido como su hijo. A mis suegros por abrirme la puerta de su casa desinteresadamente y aceptarme con mucho cariño. A mis amigos de la vida que no menciono sus nombres porque ellos saben quiénes son y que fueron más que amigos, hermanos. A mi religión que es el árbol que alimenta mi alma y guía mis caminos por el destino correcto. A la Universidad de las Ciencias Informáticas por ser mi segundo hogar y lugar de importantes acontecimientos en mi vida y al secretariado de la FFU de la facultad 1 donde conocí a magníficas personas. A todos los que estuvieron a mi lado de una forma u otra durante estos cinco años, A TODOS GRACIAS.

Dedicatoria:

Dedico mi carrera a la memoria de mis abuelos del alma Consuelo y Roberto, que a pesar de no estar junto a mi físicamente, siempre estarán en mi corazón y en cada paso y suspiro que dé en la vida, que estará siempre consagrada a ellos, que fueron mis abuelos y mis padres. A Annia, mi madre, mi amiga, mi hermana, mi compañera, mi confidente, mi vida misma, que es el reflejo de la dedicación de la suya, te adoro y te amo. A Rubén Osvaldo, el mejor padre del mundo, que a pesar de no haber podido vivir todo el tiempo conmigo, es testigo y ha estado a mi lado en cada acontecimiento importante de mi vida y le ha dedicado mucho esfuerzo y sacrificio a formar la persona que soy, siempre con una entrega incalculable de amor, muchas veces sin saber cuánto lo amo. A mis hermanos Melissa y Osvaldo de los que espero ser guía y ejemplo en la vida y siempre podrán contar conmigo en cualquier circunstancia. A mi mujer Maibel, que no solo me da amor, cariño y comprensión, sino que me dio al bebé más lindo del mundo, te amo nena, espero compartir contigo cada segundo de mi vida y que seamos felices por siempre. Por último, pero no menos importante, a mi hijo Robert Anthony, que llegó a mi vida en el momento indicado para darme fuerzas, luz, esperanza y

poder lograr mis metas en un momento en que la vida me dio un duro golpe. A ti mi niño siempre estará destinado cada esfuerzo que haga, cada hora de desvelo, cada entrega de amor incondicional.

Resumen

El presente trabajo tiene como objetivo proponer un módulo denominado Directorio Temático para el buscador Orión, que permita al mismo mostrar a sus usuarios una alternativa adicional cuando se recupera información en la red, restringiéndose a ámbitos temáticos. El módulo permite apoyar el proceso de informatización de la sociedad cubana colocando a su disposición una herramienta de gran utilidad para la búsqueda en la web. Durante la investigación se analizaron los sistemas homólogos existentes a nivel internacional y nacional definiendo las principales tendencias en el uso de directorios temáticos, sus aspectos positivos y deficiencias; se comprobó que el buscador cubano Orión no posee una búsqueda restringida por categorías. Para darle cumplimiento al objetivo planteado, en la investigación se realizó el estudio del estado del arte sobre los sistemas de búsqueda y recuperación de información restringida a ámbitos temáticos. El proceso de desarrollo estuvo guiado por la metodología de desarrollo de software AUP, se seleccionaron como principales tecnologías: el marco de trabajo Symfony 2.7.9, para la programación el lenguaje PHP 5.6, el sistema de gestión de base de datos no relacional MongoDB 2.4.9 y Visual Paradigm 8.0 como herramienta para el modelado. A partir de la aplicación de las pruebas de software de tipo funcional, estructural, de integración y de carga y stress se verificó un correcto funcionamiento del módulo implementado.

Palabras clave: ámbitos temáticos, buscador Orión, directorio temático, recuperación de información.

Índice

Introducción	1
Capítulo1: Fundamentación teórica de los Sistemas de Recuperación de Información.....	7
1.1 Introducción.....	7
1.2 Desarrollo de la web	7
1.3 Recuperación de información.....	8
1.4 Sistemas de recuperación de información.....	9
1.5 Motores de búsqueda.....	12
1.6 Metabuscadore	13
1.7 Directorios temáticos	13
1.8 Estudio de los sistemas homólogos.....	15
1.9 Ambiente de desarrollo.....	17
1.10 Herramienta CASE	23
1.11 Metodologías de desarrollo.....	24
Conclusiones del capítulo	27
Capítulo 2: Análisis y diseño de la propuesta de solución Directorio temático de Orión.....	29
2.1 Introducción.....	29
2.2 Propuesta del módulo	29
2.3 Modelo de dominio.....	29
2.4 Requerimientos del sistema.....	31
2.5 Modelo de casos de uso del sistema.....	34
2.6 Arquitectura de la propuesta de solución	40
2.7 Patrones de diseño	42
2.8 Modelo de diseño	44
2.9 Diseño de la base de datos.....	45

2.10 Modelo de despliegue	47
Conclusiones del capítulo	48
Capítulo 3: Implementación y validación del Directorio temático de Orión.....	49
3.1 Introducción.....	49
3.2 Diagrama de componentes	49
3.3 Estándares de codificación	51
3.4 Validación del sistema	53
Conclusiones del capítulo	61
Conclusiones generales.....	62
Recomendaciones	63
Bibliografía.....	64

Introducción

El avance vertiginoso de las Tecnologías de la Información y las Comunicaciones, en lo adelante TIC, que se inició en la segunda mitad del pasado siglo, matiza el mundo actual y a una sociedad que ha dado en llamarse sociedad de la información. En ella, una parte importante del esfuerzo del hombre se ha concentrado en la producción, manejo y uso de la información (Gómez, 2002). El surgimiento, desarrollo y expansión de la informática y las telecomunicaciones ha supuesto una revolución sin precedentes en el mundo (Zayas, 2002).

En la actualidad, el crecimiento de la información presente en Internet es constante y acelerado, esta ha evolucionado para ser mucho más que una biblioteca virtual, abarca todo tipo de negocio e información del saber humano. Generalmente esta información se encuentra dispersa, poco estructurada e invisible lo que hace muy engorroso el proceso de obtención de información e imposibilita el rápido acceso a la información útil de la red (Núñez, 2002).

Cabe destacar que al realizar una búsqueda se encuentran una gran cantidad de resultados, que la mayoría de las veces resulta difícil clasificar y procesar sin que esto constituya una pérdida considerable de tiempo. Del gran cúmulo de elementos encontrados, la mayoría, presenta poca información o no es una información de calidad que satisfaga las necesidades del usuario. Esto se debe a la falta de estructuración y control sobre los contenidos publicados en la web, lo cual afecta fundamentalmente a científicos, investigadores, profesionales, universitarios y estudiantes, ávidos de información verídica y científica en cualquier área del conocimiento (Núñez, 2002).

Núñez afirma: “Muchos individuos publican y utilizan información en Internet sin considerar su exactitud, validez o sesgos, cada día se sitúan en la Web grandes cantidades de información en todo el mundo, pero sólo una pequeña cantidad de ella es de calidad”. Los Sistemas de Recuperación de la Información (SRI), constituyen el mecanismo ideal para resolver este tipo de problemas, aunque no lo resuelven del todo, se siguen encontrado en las primeras listas de los buscadores importantes información poco confiable o de baja calidad científica. Estos permiten encontrar, procesar y consumir la información de forma rápida y en forma automática. Además, son sistemas capaces de localizar casi cualquier contenido existente en la web, desde textos e imágenes hasta archivos de sonido y videos, entre otros. Los principales SRI existentes actualmente son: los directorios temáticos, los motores de búsqueda o buscadores y los metabuscadores.

Los directorios o índices temáticos crean y mantienen colecciones de enlaces a otros sitios web, clasificándolos y presentándolos según una lista jerárquica de temáticas. Un directorio web temático no

es un motor de búsqueda y, por tanto, no ofrece listas de páginas web basadas en palabras clave. En su lugar, muestra las listas organizadas por categorías y sub-categorías. Dentro de cada categoría se especifica el conjunto de dominios o sitios web asociados al ámbito temático identificado por el nombre y descripción de la categoría. La categorización se basa usualmente en la totalidad del contenido de un sitio web en lugar de una página individual y, normalmente, cada sitio no se encuentra en más de una o dos categorías (Olivas, 2011).

Los directorios emplean una tecnología más sencilla y barata que los buscadores automáticos pues utilizan menos recursos informáticos, pero a cambio requieren más soporte humano y mantenimiento, ya que deben contar con editores que visitan, analizan y clasifican los distintos sitios web (se dice que son *human-edited directories* -directorios editados por personas-). Los algoritmos que utilizan son mucho más sencillos. No recorren las webs almacenando sus contenidos. Sólo mantienen algunos de los datos de cada sitio, como el título y la descripción que se introducen a la hora de ser dados de alta. Así, los resultados de la búsqueda están determinados por la información que se haya suministrado al directorio cuando se registra cada sitio web.

Los directorios suelen permitir que los *webmaster* soliciten la inclusión de sus sitios web en el directorio. Los editores del directorio revisan y filtran la información brindada en cada solicitud según la política particular del directorio de que se trate (Wallis, y otros, 2014).

Los buscadores y los metabuscadores descubren el nuevo recurso en la web principalmente de forma automática, por medio de robots denominados spiders que guardan la información de forma indexada. Representan el contenido a través de palabras clave, operadores y las consultas de cada búsqueda. Sus resultados son muy exhaustivos y poco precisos. En cambio, los directorios web o temáticos, descubren el nuevo recurso en la red a través de la inscripción de las personas con un análisis y aprobación previa por parte de un grupo de expertos. La forma de representar el contenido es mediante una clasificación manual implícita navegando por categorías de dominios donde se va de lo general a lo particular. Las consultas se representan por páginas creadas antes de la búsqueda y sus resultados son poco exhaustivos pero muy precisos (Wallis, y otros, 2014).

Las principales ventajas que poseen los buscadores y los metabuscadores son la búsqueda exhaustiva mediante un proceso automático y la constante actualización a través de sus *spiders*. Tienen dentro de sus desventajas que los resultados mostrados generalmente no han pasado por un proceso de selección de calidad, por lo que toda la información no es confiable y se dificulta distinguir una de otra. Los directorios web o temáticos son fáciles de usar por cualquier tipo de usuario, permiten tener una idea global del contenido, los resúmenes temáticos suelen estar elaborados intelectualmente por lo que

describen muy bien su contenido. Cubren una pequeña fracción de los recursos disponibles en la web y su actualización no permite seguir los cambios de contenido en la red de forma constante y veloz ya que este proceso no es automático e implica el análisis humano, estas son sus principales desventajas.

En Cuba existen empresas o entidades estatales oficiales que poseen presencia en la web a través de sitios o portales, por lo que sus publicaciones y la información que proporcionan son de gran importancia y utilidad. Esta se considera información altamente verídica ya que proviene de fuentes oficiales y que son la imagen en internet de la vida del país. Como parte de los programas de formación e informatización de la sociedad y en plena batalla de ideas que libra el pueblo cubano, surge en el año 2002 la Universidad de las Ciencias Informática (UCI). En el marco de este proceso se ha implementado un motor de búsqueda llamado Orión, cuyo objetivo es dotar al país de una herramienta que permita realizar búsquedas en todos los sitios cubanos y adaptarse a las características de la red en donde sea implantada. Este proyecto fue desarrollado por el Centro de Ideoinformática (CIDI) de la UCI.

Actualmente el buscador Orión a través de su función de búsqueda utiliza la misma función de relevancia para todas las páginas encontradas por el componente de rastreo. La relevancia de un sitio viene dada por la relación que se guarda entre el contenido del mismo, la adecuada estructura del diseño html de la página y el criterio utilizado en la búsqueda, lo que permite mostrar páginas relacionadas en un orden calculado automáticamente. En ocasiones, la búsqueda en Orión de determinados términos muestra los resultados en un orden que no coincide con los mostrados por los buscadores más utilizados internacionalmente a partir del uso del mismo criterio, lo que provoca que en algunos casos los dominios que provienen de fuentes oficiales no aparezcan como resultados en las primeras páginas de búsqueda. Esta problemática plantea al equipo de desarrollo la necesidad de mejorar la función de relevancia o de establecer algún tipo de asociaciones de páginas relevantes en la red frente a ámbitos temáticos específicos.

Orión posee un mecanismo de rastreo que indexa los dominios que logra encontrar a partir de un conjunto inicial de url. A partir del conjunto de dominios iniciales no es posible llegar a todas las url que existen en la web. Este rastreo no posee un mecanismo para que un administrador de una o varias páginas web pueda solicitar la inclusión de sus dominios en los resultados de búsqueda de Orión.

A partir de una búsqueda en ocasiones se muestran varias url de un mismo dominio, provocando que un usuario no pueda encontrar cuales son los dominios más importantes asociados a un ámbito temático. Orión no permite una búsqueda categorizada mediante la cual el usuario pueda explorar sitios nuevos a partir de temáticas generales, donde vaya de lo general a lo particular, acotando su búsqueda por las disímiles líneas temáticas.

Teniendo en cuenta la situación problemática descrita anteriormente se enuncia el siguiente **problema de investigación** a resolver: **¿Cómo restringir a ámbitos temáticos la búsqueda y recuperación de la información existente en la red para el buscador Orión?**

Para llevar adelante la investigación se plantea el siguiente **objeto de estudio** los procesos de búsqueda y recuperación de la información en la red y el **campo de acción** se define como: búsqueda y recuperación de información restringida a ámbitos temáticos.

Para darle solución al problema descrito, se ha planteado el siguiente **objetivo general**:

- Desarrollar un directorio temático que permita la búsqueda restringida por ámbitos en los procesos de búsqueda y recuperación de información del buscador Orión.

Objetivos específicos:

- Construir el marco teórico conceptual y el estado del arte respecto a las tecnologías y funcionalidades de los directorios temáticos.
- Diseñar un directorio temático para la agrupación de dominios en ámbitos temáticos en Orión.
- Implementar un módulo que permita la gestión de dominios en ámbitos temáticos en Orión.
- Validar el correcto funcionamiento del software desarrollado.

Una vez planteado los objetivos que se desean vencer en el presente trabajo y las áreas de la ciencia donde se pretende llevar a cabo dichos objetivos se plantea la siguiente **idea a defender**: el desarrollo de un directorio temático integrado al buscador Orión permitirá restringir a diferentes ámbitos temáticos los procesos de búsqueda y recuperación de información.

Para lograr los objetivos descritos anteriormente se plantean las siguientes **tareas de investigación**:

- Revisión de la bibliografía existente sobre los conceptos asociados al marco teórico de la investigación.
- Búsqueda de las tendencias más actuales sobre directorios temáticos y su uso en la recuperación de la información.
- Diseño del módulo a desarrollar, generando los artefactos de Ingeniería de Software correspondientes de acuerdo a la metodología de desarrollo, patrones y arquitectura a utilizar.
- Desarrollo de las funcionalidades asociadas a la gestión de categorías y dominios.
- Desarrollo de las funcionalidades asociadas al envío de una solicitud por parte de un usuario y la gestión de la misma, lo cual incluye notificación del cambio de estado por correo electrónico.

- Desarrollo de las funcionalidades necesarias para la visualización del directorio temático en la interfaz principal de Orión.
- Determinación de la estructura de base de datos a utilizar para el almacenamiento de la información.
- Diseño y ejecución de las pruebas de software que serán aplicadas a la solución para la correcta validación de la misma.

Para dar solución a las tareas de investigación se utilizan los siguientes métodos de investigación científica:

Histórico-lógico: Utilizado cuando se estudia el desarrollo progresivo de la información hasta su masificación actual a través de las TIC, el motor de búsqueda Orión desde su surgimiento, así como cada una de las etapas por las que ha transcurrido el mismo para lograr una mayor comprensión de su funcionamiento, el surgimiento y desarrollo de los SRI en especial los buscadores y sus directorios web.

Analítico-Sintético: Utilizado cuando se analiza la teoría existente sobre los SRI en especial los buscadores y los directorios web.

Inductivo-Deductivo: Utilizado para analizar la teoría existente, realizar comparaciones necesarias y arribar a conclusiones que garanticen el desarrollo de la investigación.

Modelado: Utilizado para realizar una reproducción simplificada de la realidad, mostrando las relaciones internas y características del sistema a través de diagramas.

El presente trabajo de diploma consta de la siguiente estructura: Introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos.

El Capítulo 1 titulado “Fundamentación teórica de los Sistemas de recuperación de información” ofrece los conceptos básicos asociados al dominio del problema. Además, brinda el estado del arte referido al tema de investigación, así como las técnicas, tecnologías, metodologías y herramientas usadas en la actualidad en las que se apoya la solución del problema.

El Capítulo 2 titulado “Análisis y diseño de la propuesta de solución Directorio temático de Orión” contiene una visualización del entorno real del proyecto, sus objetos y la forma en que interactúa, a partir del modelo de dominio. Describe las funciones que el módulo debe cumplir, las propiedades o cualidades que el software debe tener, los casos de uso del sistema. Propone el uso de la arquitectura MVC, así como los patrones de diseño a emplear.

En el Capítulo 3 titulado “Implementación y validación del Directorio temático de Orión” se muestra el diagrama de componentes y su descripción, con el fin de comprender los elementos de software del subsistema y sus relaciones. También se expone la validación del diseño y el módulo implementado mediante las pruebas de software de tipo funcional, estructural, de integración y de carga y estrés.

Capítulo1: Fundamentación teórica de los Sistemas de Recuperación de Información

1.1 Introducción

En el presente capítulo se brinda una visión general de los aspectos relacionados con los SRI, en especial los directorios web. Se mencionan las características, tendencias, tecnologías de cada herramienta, así como la descripción de las principales definiciones asociadas al dominio del problema y que son necesarias para entender el negocio y la propuesta de solución. Se definen y valoran los conceptos más importantes a emplear en la investigación y se hace una revisión de aplicaciones homólogas existentes en el mundo y en Cuba y la comparación entre estas. Se detalla una breve descripción de las metodologías de análisis y diseño, se especifica las ventajas de la escogida y de las herramientas de desarrollo usadas para la confección de la aplicación.

1.2 Desarrollo de la web

Dadas las problemáticas presentadas durante varios años para la recuperación de la información se han buscado soluciones para resolverlas y lograr una mayor utilidad e influencia de la red de redes en la vida cotidiana del ser humano, una de ellas es la creación de la web semántica la cual busca estructurar y organizar la información eficientemente con diferentes tipos de tecnologías. Otra de ellas, es el desarrollo de la web 2.0 que apunta a un enfoque social donde la participación es el eje principal para compartir información.

1.2.1 Web 2.0

La Web 2.0 no es otra cosa que la actual fase de un imparable y expansivo proceso evolutivo de Internet, que tiene como característica esencial un fuerte acento en lo social (Cobo, y otros, 2008), donde el rol de las comunidades en red genera mayor impacto sobre los recursos compartidos, con intereses comunes en cuanto a requerimientos de información se refiere es decir, “una plataforma abierta, construida sobre una arquitectura basada en la participación de los usuarios” (Arroyo, y otros, 2007). Arroyo destaca los dos principios fundamentales de la Web 2.0: inteligencia colectiva y arquitectura de participación. La primera es la suma del saber de cada uno de estos individuos, que al ser compartida puede dar lugar a una obra colectiva (Ej. Wikipedia), y la segunda implica una nueva forma de construir los sitios web para permitir la participación de la gran masa de usuarios (Ej. Facebook).

1.2.2 Web semántica

La Web semántica representa la visión de una futura web, que busca un proceso inteligente el cual permitirá la interoperabilidad entre máquinas y la comunicación con las personas (Méndez, 2002).

“la Web semántica aspira a mejorar sustancialmente la interacción entre los sistemas informáticos y los seres humanos, dotando a los primeros de una mayor inteligencia y autonomía, y a los segundos de un nuevo infouniverso conceptual ergonomizado intelectivamente que satisfaga la necesidad de convertir información en conocimiento” (Bonilla, 2011).

Las tecnologías fundamentales que sustentaran esta web, según Arias son: lenguaje de marca extensible (XML), Marco de Descripción de Recursos (RDF) y las ontologías (por ejemplo, el lenguaje de ontológico web (OWL)). Cada uno de estos componentes permiten: dotar a la red de significado, brindar interoperabilidad con otros sistemas, coordinación de trabajo entre humanos y ordenadores, intercambio de información y la transformación de una web invisible a una web visible (Arias, 2006).

1.3 Recuperación de información

La recuperación de información, en lo adelante (RI), en inglés *Information Retrieval*, es un término muy usado ya que abarca el saber de diferentes ciencias, desde las ciencias informática, la documentación y la ciencia de la información, se ha emitido un gran número de criterios y consideraciones, por su complejidad, de diferentes autores que van desde Baeza-Yates (1999), Gómez (2002) hasta Tolosa y Bordignon (2008). “Los datos se pueden estructurar en tablas, arboles, para recuperar exactamente lo que se quiere, el texto no posee una estructura clara y no resulta fácil crearla.... dada una necesidad de información (consulta) y un conjunto de documentos, ordenar los documentos de más a menos relevantes para esa necesidad y presentar un conjunto de aquellos con mayor relevancia” (Baeza-Yates, 1999).

Por RI se entiende el proceso por el cual, una vez analizado un documento es identificada la necesidad de información, se produce una comparación entre ambos para obtener resultados satisfactorios para el usuario (Gómez, 2002). Otro de los criterios plantea que la recuperación de información consiste en encontrar y *rankear* documentos relevantes que satisfagan la necesidad de información de un usuario, expresada en un determinado lenguaje de consulta (Tolosa, y otros, 2008).

Para el autor de esta investigación la RI se define como la condición de encontrar información variada, sobre cualquier ciencia del saber humano, en las distintas fuentes existentes, con la finalidad de procesarla y ordenarla según su importancia, para la satisfacción de sus necesidades de conocimiento.

1.4 Sistemas de recuperación de información

Los sistemas de recuperación de la información agrupan a varias aplicaciones informáticas que “permiten almacenar, recuperar y mantener información” (Shannon, 1981) de la web, que se encuentra disperso e imposibilita al usuario encontrar información útil que responda a sus exigencias sin la ayuda de los SRI, para ampliar el significado de este término se analizarán los conceptos de varios autores. De acuerdo con Baeza–Yates (2005) los SRI “deben de alguna manera interpretar el contenido de la información dentro de una colección de documentos y establecer con ellos, un orden de acuerdo al grado de relevancia que estos posean, los datos se pueden estructurar en tablas y árboles, para recuperar exactamente lo que se quiere para las consultas de los usuarios” (Baeza-Yates, y otros, 2005).

Los sistemas de recuperación de información constan de una misma arquitectura que los relaciona entre si y los distingue de otros sistemas informáticos (López, 2006):

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

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

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

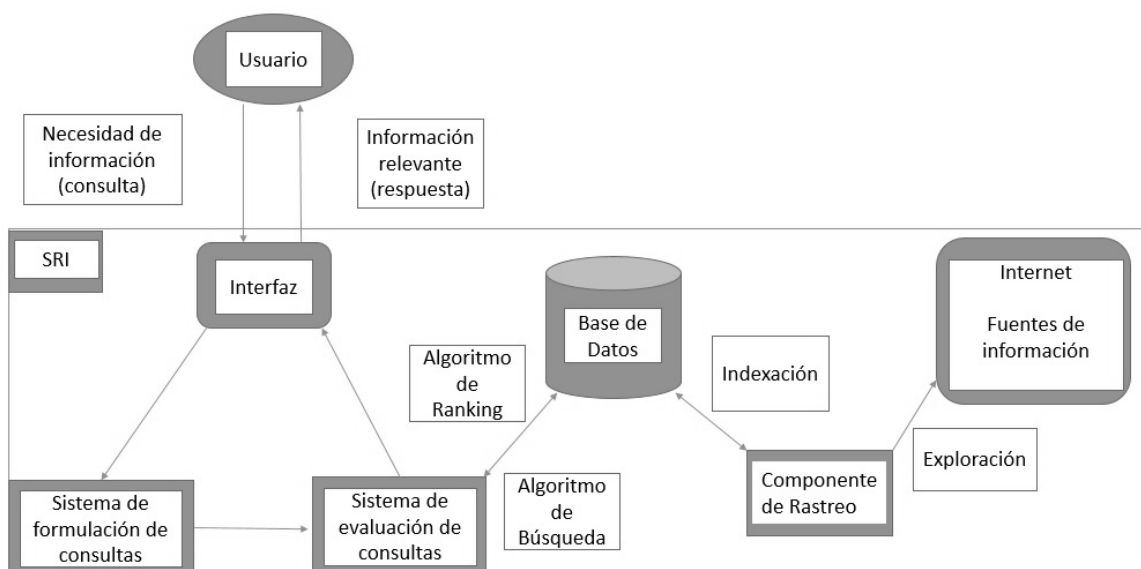


Figura 1: Esquema de Arquitectura de los SRI.

Se concluye que los SRI constituyen un mecanismo de encontrar información variada sobre cualquier ciencia del saber humano en las distintas fuentes existentes, con la finalidad de procesarla y ordenarla según su importancia para la satisfacción de las necesidades de conocimiento del usuario.

1.4.1 Clasificación de los SRI

Estos se clasifican a partir de las diferencias según la forma en que indexan el contenido de la web, como representan el contenido, como realizan la consulta y la forma que muestra el resultado al usuario. Las clasificaciones más conocidas y usadas son: los Directorios temáticos o web, los buscadores o motores de búsqueda y los metabuscadores o buscador en buscadores (Olivas, 2011).

En las siguientes tablas se presenta una comparativa entre los diferentes tipos de SRI, destacando las ventajas y desventajas que posee cada uno.

Tabla 1: Comparativa por características de los SRI

	Descubrimiento de recursos	Representación del contenido	Representación de la consulta	Presentación de los resultados
Directorios	Lo realizan las personas.	Clasificación manual implícita (navegación por categorías).	Páginas creadas antes de la consulta.	Poco exhaustivos, muy precisos.
Motores de búsqueda	Principalmente de forma automática por medio de robots. Indización automática.	Explícita (palabra clave, operadores).	Páginas creadas dinámicamente en cada consulta.	Muy exhaustivos, poco precisos.
Metabuscadores	Envían su consulta a varios motores de búsqueda y crean	Explícita (palabra clave, operadores,).	Páginas creadas dinámicamente en cada consulta dependiendo de	Muy exhaustivos, poco precisos, tienen menos control de la

	una lista con los principales resultados encontrados en cada uno de ellos.		los resultados obtenidos de los motores.	búsqueda al interrogar varias bases de datos con interfaces diferentes.
--	--	--	--	---

Cuadro comparativo de fortalezas y debilidades de motores de búsqueda y directorios (Villota, 2004).

Tabla 1.1: Fortalezas y Debilidades de los SRI.

Herramientas de Búsqueda	Fortalezas	Debilidades
Motores de Búsquedas	<ul style="list-style-type: none"> ✓ Permiten búsqueda exhaustiva, ya que el proceso es automático. ✓ Utilizan mecanismos automáticos para seguir los cambios de contenido, direcciones <i>Web</i>. ✓ Existen buscadores especializados en todos los campos del conocimiento. 	<ul style="list-style-type: none"> ✓ Los resultados que se muestran generalmente no han pasado por ningún proceso de selección de calidad, por lo cual toda la información no es confiable. ✓ Requieren mayor esfuerzo del usuario al configurar la herramienta "búsqueda avanzada".
Directorios temáticos	<ul style="list-style-type: none"> ✓ Fáciles de utilizar, especialmente para usuarios no experimentados. ✓ Permiten tener una idea global del contenido. ✓ Los resúmenes temáticos suelen estar elaborados intelectualmente, por lo que realmente describen su 	<ul style="list-style-type: none"> ✓ Cubren una pequeña fracción de los recursos de la <i>Web</i>. ✓ La indexación documental sin recursos automáticos no permite el seguimiento a los cambios de contenido, aparición o desaparición de documentos.

	contenido.	
Metabuscadores	<ul style="list-style-type: none"> ✓ Permite al usuario explorar varios buscadores a través de una sola interfaz, planteando una sola estrategia de búsqueda y de forma automática. ✓ Se recomiendan para temas “oscuros” o difíciles de encontrar. ✓ Se obtienen los resultados integrados, a partir de varios buscadores. 	<ul style="list-style-type: none"> ✓ No almacenan información porque no poseen base de datos propia. ✓ Son difíciles de usar para búsquedas muy precisas, porque tienen menos control de la búsqueda al interrogar varias bases de datos con interfaces diferentes.

1.5 Motores de búsqueda

Los motores de Búsqueda son herramientas que permiten a los cibernautas localizar información disponible en Internet de manera interactiva. Sus bases de datos son una especie de inventario de archivos existentes en la red. Constan de un programa que registra, generalmente, estos archivos sin utilizar criterios específicos de selección y, con ellos, forma un índice de búsqueda que otro programa consulta cuando el usuario realiza su solicitud. Los buscadores proporcionan formularios y diversas interfaces mediante los cuales los interesados pueden realizar su exploración; como resultado, el solicitante recibe una lista de referencias que responden a su pedido.

Componentes básicos de los motores de búsqueda (Cruz, 2004):

- Spider: recorre la Web buscando las páginas para indexar.
- Indexador: gestiona el índice con la información de las páginas localizadas.
- Motor de Búsqueda: realiza la búsqueda sobre el índice de documentos
- Interfaz: interactúa con el usuario.

Los buscadores se han convertido en la puerta de acceso a la red para la gran mayoría de los usuarios que buscan información, y generan la mayor parte del tráfico en la Web; pero su funcionamiento dista

mucho de ser el más adecuado, esencialmente por las enormes diferencias de calidad que existen entre los recursos atesorados en sus colecciones, así como por las insuficiencias propias de los algoritmos de recuperación de la información (Olivas, 2011). Desde un principio, los contenidos de la Web despertaron recelos en sectores académicos y profesionales a causa de la presencia de abundante información sin valor científico alguno (Olivas, 2011). Sobre estos aspectos se ha trabajado intensamente durante los últimos años con vista a mejorar tanto el rigor de los recursos coleccionados como la precisión de los resultados de sus búsquedas. Incluso, desde hace algún tiempo, muchos motores solo recolectan información de sitios autorizados.

1.6 Metabuscadores

Se conoce como metabuscadores, a aquellos que permiten interrogar varias bases de datos simultáneamente desde una única interfaz; aunque ellos no ofrecen el mismo nivel de control sobre la lógica y la interfaz de búsqueda que los motores y directorios, la mayoría son bastante rápidos. Apunta, además que la tendencia es mejorar sus capacidades con la incorporación de opciones de búsqueda al uso de los operadores convencionales, entre ellas, se destaca la posibilidad de poder escoger los motores de búsqueda donde debe buscarse, la búsqueda por campos (título, URL) entre otras. Otras cualidades se agregan a la definición anterior, se plantea que, al no tener sus propias bases de datos, a diferencia de los motores de búsqueda y los directorios temáticos, los metabuscadores no almacenan páginas web, no agregan direcciones, ni clasifican y reseñan sitios web, sino que envían una solicitud de búsqueda a varios buscadores a la vez (Stanley, 2001).

1.7 Directorios temáticos

Los índices o directorios temáticos (web), se pueden definir como un sistema de búsqueda por temas o categorías escalonadas (aunque también suelen incluir sistemas de búsqueda por palabras clave), es un sistema muy útil cuando el cibernauta quiere encontrar sitios web que contengan información referida a temas relevantes, como lo haría usando el índice alfabético de las enciclopedias o de las guías telefónicas. Los directorios web se compilan por personas (expertos), quienes deciden los sitios de Internet que se registrarán en sus bases de datos. Ellos ordenan sus recursos de forma cronológica, por temas, categorías o por una combinación de estos, generalmente en forma de árbol jerárquico, lo que facilita la búsqueda de la información existente en un área determinada del conocimiento, los resultados son recorridos en profundidad, lo que garantiza que al final de la jerarquía, exista una alta probabilidad de encontrar lo que realmente se necesita (Kiva, 2011).

Las categorías de los directorios temáticos suelen organizarse según el orden alfabético, esta es la forma predefinida en que estarán ordenadas las listas de sitios. Los sitios web en la categoría aparecen según el orden alfabético de sus títulos, también pueden estar ordenados basados en la popularidad, los sitios que son más populares entre los usuarios, o los más relevantes de la categoría, aparecerán al principio de la lista (este orden se basa en la tecnología que utilizaba el buscador de Yahoo!). Para obtener el listado ordenado de acuerdo a la popularidad y relevancia. Los sitios nuevos (recién incluidos), siempre aparecerán al principio del listado, sin importar cómo esté configurado el orden de la lista de sitios. Además, los sitios "oficiales" sobre un tema o persona, siempre aparecerán al principio del orden alfabético (Veron, 2004).

El volumen de crecimiento de la red supera la capacidad de los recursos humanos, por ello suelen ser selectivos y suelen recoger el 5% de la información en la red (lo cual no es poco si se tiene en cuenta que la información realmente interesante se mueve en un 10% de la red). Además, su actualización es lenta (Universidad Carlos III de Madrid, 2012).

Según la sección de internet para ingenieros de la universidad Carlos III de Madrid, "algunos índices han automatizado la recopilación con "robots", que, por frecuencia de palabras van asignando las páginas a las distintas categorías. Pero siempre hay una revisión posterior humana y nunca se pretenderá crear un índice de frecuencia de palabras de todas las páginas." (Universidad Carlos III de Madrid, 2012)

Muchos han incluido sistemas de búsqueda por palabras, que puede actuar de dos formas distintas:

- Sobre la clasificación o una sección de ella: resultan un instrumento útil cuando no se sabe en qué parte del directorio puede estar la "clase" que interesa.
- Sobre las páginas: pero se limitan sólo a lo recopilado por el Índice (fundamentalmente sitios, no páginas).

Por esto es muy fácil confundirlos con un buscador, pero los distingue la propia naturaleza de la recopilación, su volumen y calidad. Algunos autores utilizan el término "buscador temático". No puede hablarse propiamente de un "buscador" porque su finalidad no es crear un índice de palabras sino una clasificación temática (Universidad Carlos III de Madrid, 2012).

Otras definiciones de índices o directorio temáticos son (Universidad Carlos III de Madrid, 2012):

Los **índices especializados**, cuando la información compilada se limita a un área temática, ejemplo: Índices para Ingeniería, Directorio de revistas, entre otros.

Los **índices analíticos** especializados, o en ocasiones generales, que proporcionan una descripción de las páginas y sedes, muy superior a la que proporcionan los directorios convencionales, condensando su contenido, explicando su utilidad y evaluando su calidad y creando índices complejos de conceptos para orientar en la búsqueda de información. Son instrumentos muy dinámicos (constante actualización) y muy especializados en un área temática, aunque algunos pueden tocar varias áreas (acuerdos de cooperación), algunos ejemplos, *JISC Resource Guide for Engineering, Mathematics and Computer Science, Engineering - Science & Mathematics - Computers & Information Technology, Social Sciences Information Gateway*.

Los **índices orientativos**, sin llegar a ser índices analíticos, intentan recoger aquellos recursos de mayor interés para sus usuarios. No es un instrumento "oficial" pero muchas organizaciones (Bibliotecas y centros de información) se han dedicado a elaborar este tipo de guías, según el modelo de los directorios, que ofrecen la ventaja de que han sido seleccionados por un "experto", dando al usuario un valor añadido sobre otros instrumentos de información. No son exhaustivos, sólo recogen lo más interesante y lo más útil. Alguien las ha llamado "Webbibliografías", Acceso a INGENIE -Guía temática de Ingeniería-, es un ejemplo.

1.8 Estudio de los sistemas homólogos

En el ámbito internacional

Google Directorio.com

Es un directorio que permite realizar búsquedas por diferentes líneas temáticas, además brinda la posibilidad de utilizar un buscador general que se encuentra en la página inicial, algunas funcionalidades están visibles pero no funcionan en estos momentos, como algunos enlaces de direcciones de sitios que están listados en las categorías, dado que todavía está en desarrollo y al ser sin fines de lucro no cuenta con un financiamiento alto ni una estabilidad en su funcionamiento, al adentrarse en una temática, no permite continuar filtrando la búsqueda por otras más específicas, en forma de árbol jerárquico donde se va de lo general a lo particular , además de que posee errores ortográficos en su interfaz principal (GoogleDirectorio, 2014).

CubawebDirectory

Es un directorio creado fuera del país, especializado en la búsqueda de sitios web sobre Cuba. Posee categorías con una amplia variedad de temas que permiten al usuario más exigente satisfacer sus

necesidades de información en las más diversas materias, tiene la opción de restringir la búsqueda y dirigirla a una provincia específica y permite filtrar palabras dentro de una categoría devolviendo los resultados dentro de estos límites. Te permite adicionar nuevos enlaces, de esta forma un cibernauta añade su sitio una vez verificado por el administrador, otra de sus ofertas es la selección de los más populares y sugerir nuevas categorías, así como realizar recomendaciones de su experiencia en el directorio. Como deficiencias es que es un sistema propietario donde para añadir un sitio o utilizar algunos de sus servicios complementarios a la búsqueda hay que pagar por ellos, está alojado en un servidor extranjero dificultando su uso desde el territorio nacional a los usuarios que solo tienen acceso a la intranet. Además de que su interfaz no tiene un diseño agradable a los ojos de usuario (Cubawebdirectory, 2013).

DondeBuscar.net

Es un directorio que tiene búsqueda general, posee temáticas de búsqueda, pero no muy diversificadas, tiene un matiz más recreativo y comercial, no posee una interfaz amigable y con colores que ayuden a los débiles visuales, no permite búsquedas de interés académico o profesional. El sistema está alojado en un servidor internacional lo que proporciona una gran desventaja en su uso (Dondebuscar, 2013).

En el ámbito nacional

Directorio Prensa de Cubadebate

“El presente directorio de la Prensa Cubana incluye sitios de periódicos, medios digitales, emisoras de radio y televisión de Cuba” (Cubadebate, 2015).

Solo se enmarca en el ámbito de la prensa y no permite más opciones al usuario que no sea la búsqueda por categorías según el tipo de medio, nacional o provincial, y se accede a las páginas principales de estos. No existe una jerarquía por temáticas que vaya de lo general a lo específico acotando según diferentes categorías como política, cultura o deportes, solo muestra un listado con el nombre del medio, su dirección y si es periódico, emisora de radio o televisión. El alcance que posee lo restringe la categoría donde se encuentre. Para acceder al mismo es a través del Portal web de Cubadebate. Destacar que es el único medio que tiene hoy el internauta cubano de acceder a un listado de todos los sitios de prensa del país. Otras ventajas, está alojado en un servidor de acceso nacional y tiene una interfaz amigable para el usuario (Cubadebate, 2015).

Directorio de Cubarte

Es un directorio especializado en el ámbito cultural cubano, posee una búsqueda general y una búsqueda por categorías, todas relacionadas con las diferentes manifestaciones del arte. Es de fácil acceso desde la red nacional. Posee un directorio institucional dentro del ámbito de la especialidad, permite la búsqueda clasificada de los sitios de cultura por provincias. Como deficiencia es que al ser especializado no permite al usuario buscar otras temáticas. Posee un espacio de comentarios, donde los clientes pueden expresar sus opiniones e inquietudes al respecto, pero no permite sugerir sitios nuevos al administrador (Cubarte, 2014).

1.8.1 Resultados del estudio de sistemas homólogos

Aspectos a tener en cuenta:

- ✓ La navegación por categorías que vayan de lo general a lo particular.
- ✓ Tener una amplia variedad de temáticas.
- ✓ Uso de tecnología de software libre.
- ✓ Diseño de interfaz mediante el empleo de componentes, librerías y estándares para optimizar el uso de HTML 5, CCS 3 y Java Script y su visualización en varios tipos de dispositivos.
- ✓ Solicitud de inclusión de nuevos dominios en el directorio temático.
- ✓ Filtrado de categorías y dominios a partir de parámetros de búsqueda general.

1.9 Ambiente de desarrollo

1.9.1 Lenguajes de programación del lado del cliente

HTML

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

adecuadas para la creación de documentos relativamente simples que permiten simplificar su estructura (Consortium, 2014).

CSS

Con el crecimiento de Internet y la aparición del lenguaje HTML para la creación de páginas web, el W3C demostró la necesidad de un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos creados, de manera que pudieran visualizarse de igual forma en cualquier navegador web. A partir de entonces surgieron entre varias propuestas los lenguajes CHSS (*Cascading HTML Style Sheets*) y SSP (*Stream-based Style Sheet Proposal*), el primero realizado por Hakon Wium Lie y el segundo por Bert Bos, que a finales de 1994 y 1995 se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta denominado CSS (*Cascading Style Sheets*, por sus siglas en inglés), (Consortium, 2014).

Java Script

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

Uno de los aportes más significativos de JavaScript como lenguaje, son todas las librerías que han sido codificadas y que permiten agregarle mayor dinamismo, estética, funcionalidad y usabilidad a los sitios web; entre ellas se destacan: ExtJS, JQuery, Node.JS, entre otras (Java, 2012).

1.9.2 Lenguajes de programación del lado del servidor

Lenguaje PHP

PHP es un lenguaje de programación interpretado de propósito general. Es ampliamente utilizado en el desarrollo web y puede ser incluido en páginas HTML. La versión estable de PHP que se pretende usar es la 5.6 o superior, este último requisito propuesto por el proyecto Orión para poder integrar la nueva característica al sistema existente, por ser esta la utilizada. La mayor parte de su sintaxis ha sido

tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas (PHP, 2015).

Es el acrónimo de *Hipertext Preprocesor*. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones web actuales.

Características principales del lenguaje

- Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas (PHP, 2015).
- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF, analizar código XML, entre otras.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que se puede realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.
- Tiene manejo de excepciones (desde PHP 5).

En los últimos años, PHP ha crecido hasta ser el más generalizado en Internet, funcionando en más de un tercio de los servidores web a través del mundo. PHP ha tenido un crecimiento no solo cuantitativo sino también cualitativo (PHP, 2015).

PHP ha mantenido el liderazgo en la web. PHP no solo revoluciona el modelo orientado a objetos sino que también contiene muchas nuevas características que lo convierten en un lenguaje mucho más maduro (López, 2011).

La llegada de PHP5 vino emparejada de una reestructuración del *Core* (núcleo) de PHP, lo que los creadores de PHP llaman Zend Engine 3. Así, como el lejano PHP3 incluye su Zend Engine 0.5, y PHP4 el Zend Engine 1.0, se tiene Zend Engine 2.0 en PHP5. El cambio de versión no fue trivial, incluye la reescritura casi total del modelo de objetos, entre sus cambios más sustanciales (López, 2011).

Lenguaje Java

Java es un lenguaje de programación orientado a objetos, toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Los programas escritos en el lenguaje Java pueden ejecutarse en cualquier tipo de hardware. La recolección de basura de Java es un proceso prácticamente invisible al desarrollador, es decir, el programador no tiene conciencia de cuándo la recolección de basura tendrá lugar, ya que ésta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente. Desarrollado por Sun Microsystem, a principio de los años 90 (Java, 2012).

1.9.3 Framework de desarrollo

En el mundo el desarrollo de aplicaciones informáticas con el uso de los marcos de trabajos 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 (SOAP, 2015).

Debido a la necesidad de estandarizar las prácticas comunes, así como los criterios y conceptos a la hora de desarrollar aplicaciones de software, surgen los marcos de trabajos (en inglés, *frameworks*) de desarrollo o marcos de trabajo, los cuales definen una estructura conceptual y tecnológica de soporte definido, a partir de artefactos o módulos de software previamente desarrollados, que sirven de base para la organización y construcción de un software (Riehle, 2012).

Symfony

Symfony es un completo *framework* para el desarrollo ágil de complejas aplicaciones web. Está completamente desarrollado en PHP 5. Symfony está basado en la experiencia, no reinventa la rueda,

sino que usa las mejores prácticas del desarrollo web e integra una gran variedad de librerías. Posee una activa comunidad de usuarios y desarrolladores, los cuales desarrollan *plugins* que extienden las funcionalidades del *framework*. Existe abundante documentación en diferentes idiomas, fundamentalmente en inglés, español y francés (Potencier, 2014).

Características de Symfony:

- Fácil de instalar y configurar en la mayoría de las plataformas y con la garantía de que funciona correctamente en los sistemas Windows y Unix.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor15 y que permite un mantenimiento muy sencillo.

Por todas estas características anteriormente expuestas el proyecto Orión utiliza este *framework* para el desarrollo de sus componentes. Con el objetivo de lograr una mejor integración del directorio temático al buscador Orión se decide utilizar el *framework* Symfony 2 para el desarrollo de la propuesta de solución.

1.9.4 Entorno de desarrollo integrado

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

Netbeans

NetBeans es un entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto de NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y

móviles utilizando la plataforma Java, así como PHP, JavaScript y Ajax, Ruby y Ruby, Groovy y C/C++. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de formación exhaustivos, así como una selección diversa de complementos de tercero (Netbeans, 2014).

1.9.5 Sistema de gestión de base de datos

Este tipo de sistema tiene la función principal de servir de interfaz de comunicación entre el usuario final, la base de datos y las aplicaciones que la utilizan. De esta manera, se garantizan requisitos indispensables en toda aplicación que requiera la persistencia de grandes volúmenes de datos, por ejemplo: la independencia de la lógica de programación respecto a la forma de almacenamiento, la seguridad de la información, la abstracción de las operaciones, la seguridad y la rapidez de las respuestas a consultas. Si bien existen varios tipos de bases de datos, atendiendo a su estructura y a la forma en que almacenan los datos, todas contribuyen al aumento de la calidad de los sistemas informáticos (Castrillon , y otros, 2008).

MongoDB

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

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 no SQL. Con esquemas dinámicos ofrece 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, 2011).

Solr

Es una popular herramienta de indexación de datos rastreados por un motor de búsqueda. Su desarrollo está patrocinado por la Fundación Apache y Lucid Imagination. Sus características principales incluyen una búsqueda de texto completo, resaltado de búsqueda de facetas, la integración de bases de datos y documentos (por ejemplo, Word, PDF). Solr es altamente escalable, proporcionando búsqueda distribuida y replicación de índices, es desarrollado en Java y se ejecuta como un servidor de búsqueda independiente de texto completo dentro de un contenedor de *servlets*.

Solr utiliza la biblioteca de Java Lucene para la búsqueda e indexación de texto completo y tiene APIS que hacen que sea fácil de utilizar desde cualquier lenguaje de programación. Presenta una configuración externa de Solr, permitiendo adaptarse a casi cualquier tipo de aplicación sin necesidad de codificación en Java.

1.9.6 Servidores Web

Apache

Apache es un servidor web flexible, rápido, eficiente y continuamente actualizado. Entre sus características se destacan: (Bannister, 2012)

- Es multiplataforma.
- Es un servidor web conforme al protocolo HTTP.
- Es modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y con la API de programación de módulos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Es extensible.
- Es popular (fácil de conseguir ayuda y soporte).

1.10 Herramienta CASE

Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Además es una herramienta de diseño multiplataforma. Soporta los principales estándares de la industria, tales como; UML, SysML, BPMN, XMI. Ofrece un conjunto completo de herramientas que los equipos de desarrollo necesitan para la captura de requisitos, planificación de software, planificación de pruebas, modelamiento de clases, modelamiento de datos y otras actividades (Paradigm, 2012).

1.11 Metodologías de desarrollo

Las metodologías de desarrollo se pueden clasificar dentro de 2 grandes categorías: Las metodologías tradicionales y las metodologías ágiles, teniendo sus ventajas y sus desventajas cada una de ellas.

Extreme Programming (XP)

Metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Metodologías Águilas en el Desarrollo de Software, 2000).

Open Unified Process (OpenUP)

Es una metodología ágil basada en *Rational Unified Process* (RUP), conservando las características esenciales de esta metodología, que incluye el desarrollo iterativo e incremental, casos de uso y escenarios de conducción de desarrollo, gestión de riesgos, y el enfoque centrado en la arquitectura. El resultado ha sido una metodología mucho más simple y ligera, definiéndose en el campo de proyectos pequeños que abarquen de 3 a 6 personas (Proceso de entrega: OpenUP, 2014).

Durante el proceso de desarrollo de software solo el contenido fundamental y necesario es incluido. La mayoría de los elementos de OpenUP están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances (OpenUP, 2015).

Principios de OpenUP

- ✓ Colaborar para sincronizar intereses y compartir conocimientos.
- ✓ Equilibrar las prioridades para maximizar el beneficio obtenido por los intereses en el proyecto.
- ✓ Centrarse en la arquitectura para minimizar el riesgo y organizar el desarrollo.
- ✓ Desarrollo evolutivo para obtener retroalimentación y mejoramiento continuo.

OpenUP estructura el ciclo de vida de un proyecto en cuatro fases: concepción, elaboración, construcción y transición. El ciclo de vida del proyecto provee a los interesados un mecanismo de supervisión y dirección para controlar los fundamentos del proyecto, su ámbito, la exposición a los riesgos, el aumento de valor y otros aspectos.

Beneficios de usar esta metodología:

- ✓ Disminuye las probabilidades de fracaso en los proyectos pequeños, incrementando las probabilidades de éxito.
- ✓ Permite la detección de errores tempranos por medio de un ciclo iterativo.
- ✓ Evita la elaboración de documentación, diagramas e iteraciones redundantes.
- ✓ Cuenta con un enfoque orientado al cliente.

Agile Unified Process (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), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad. Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados (Flores, y otros, 2014):

- Concepción: 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.
- Elaboración: el objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- Construcción: durante la fase de construcción el sistema es desarrollado y probado en el ambiente de desarrollo.
- Transición: el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Las disciplinas se llevan a cabo de manera sistemática, a la definición de las actividades que realizan los miembros del equipo de desarrollo a fin de desarrollar, validar, y entregar el software

de trabajo que responda a las necesidades de sus interlocutores. Las disciplinas son: (Flores, y otros, 2014)

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.
2. Aplicación. El objetivo de esta disciplina es transformar su modelo (s) 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, etc.), 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, etc.) estén disponibles para el equipo según sea necesario.

La AUP es ágil, porque está basada en los siguientes principios (Flores, y otros, 2014):

1. El personal sabe lo que está haciendo. La gente no va a leer detallado el proceso de documentación, pero algunos quieren una orientación de alto nivel y / o formación de vez en cuando. La AUP producto proporciona enlaces a muchos de los detalles, si usted está interesado, pero no obliga a aquellos que no lo deseen.
2. Simplicidad. Todo se describe concisamente utilizando un puñado de páginas, no miles de ellos.
3. Agilidad. El ajuste a los valores y principios de la Alianza Ágil.
4. Centrarse en actividades de alto valor. La atención se centra en las actividades que se ve que son esenciales para el de desarrollo, no todas las actividades que suceden forman parte del proyecto.

5. Herramienta de la independencia. Usted puede usar cualquier conjunto de herramientas que usted desea con el ágil UP. Lo aconsejable es utilizar las herramientas que son las más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.
6. Adaptación de este producto para satisfacer sus propias necesidades.

1.11.1 Selección de la metodología de desarrollo de software a utilizar

- ✓ Se seleccionó la metodología de desarrollo de software AUP para lo que se tuvieron en cuenta los aspectos siguientes:
- ✓ Describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software.
- ✓ Es una metodología ágil por tanto se adapta al trabajo de equipos de pequeño formato.
- ✓ Permite la gestión de cambios ágiles sin la necesidad de generar mucha documentación.
- ✓ Es la metodología utilizada en el proyecto Orión, al cual se integrará el módulo final.
- ✓ Permite la descripción simplificada del proceso de desarrollo de software acortando los tiempos de desarrollo de una aplicación.

Conclusiones del capítulo

El análisis de los principales elementos teóricos sobre los SRI, así como la importancia de estos para la recuperación de información aportó los elementos necesarios para la comprensión del trabajo y el arribo a los resultados que se desean. Los directorios temáticos existentes a nivel nacional e internacional no satisfacen las necesidades requeridas para la recuperación de información restringida a ámbitos temáticos del buscador Orión, la mayoría necesitan ser accedidas mediante una conexión a Internet lo que dificulta el acceso de los usuarios de redes locales. La utilización de MongoDB garantiza una adecuada gestión de configuración y almacenamiento de la información. El uso de Visual Paradigm 8.0 como herramienta CASE y la metodología de desarrollo de software AUP permite diseñar y elaborar la documentación necesaria para guiar el desarrollo del sistema de forma ágil y metódica. El empleo del lenguaje de programación PHP y el uso de Symfony 2.7, permiten el desarrollo de las funcionalidades necesarias para la búsqueda de información de calidad, restringida a ámbitos temáticos, y con un nivel

jerárquico. La agrupación por categorías mediante el nombre y la descripción de la misma, permitirá acotar la búsqueda en ámbitos temáticos en el buscador Orión.

Capítulo 2: Análisis y diseño de la propuesta de solución Directorio temático de Orión

2.1 Introducción

En el presente capítulo se definen los requerimientos de software, las estructuras de datos y los artefactos necesarios para la solución de la propuesta del Directorio temático del buscador Orión. Se describen los principales requisitos funcionales y no funcionales para lograr un correcto funcionamiento del módulo. Se presenta además el modelo de dominio y de despliegue utilizado para la construcción del mismo, los elementos relacionados al modelado del negocio y se definen los flujos de proceso y una descripción de las principales funcionalidades. Se explicará la estructura de la base de datos desarrollada.

2.2 Propuesta del módulo

Para dar solución a los objetivos expuestos se propone el desarrollo de un directorio temático, utilizando el marco de trabajo Symfony 2.7, como gestor de base de datos MongoDB 2.4.9 y el servidor web Apache 2.4.7, esta aplicación será capaz de integrarse al motor de búsqueda Orión. El usuario podrá acceder desde la red a este directorio temático y realizar una búsqueda navegando por sus categorías donde encontrará información relevante y confiable, filtrada ya por las categorías preestablecidas y por un proceso de selección de los contenidos que se publiquen.

La aplicación contará con tres roles de usuarios principales, el usuario anónimo que solo tendrá acceso a la búsqueda por categorías; el usuario común que tendrá permiso para realizar la solicitud de inscripción de su dominio en una categoría, enviar reporte de algún problema al administrador del sistema y el usuario administrador, que podrá gestionar las categorías, los dominios y las solicitudes de inscripción. El usuario con rol de tipo administrador tendrá los máximos privilegios en el módulo.

2.3 Modelo de dominio

Es una representación visual estática del entorno real del proyecto, sus objetos y la forma en que estos interactúan. Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis, como paso previo al diseño de un sistema, ya sea de software o de otro tipo. El objetivo del modelado del dominio es comprender y describir las clases más importantes del sistema y ayudar a los usuarios, clientes, desarrolladores y otros interesados a utilizar un vocabulario común (Jacobson, y otros, 2000).

El modelo de dominio se representa con un diagrama de clases en el cual se muestran las clases conceptuales, las asociaciones entre ellas y sus atributos.

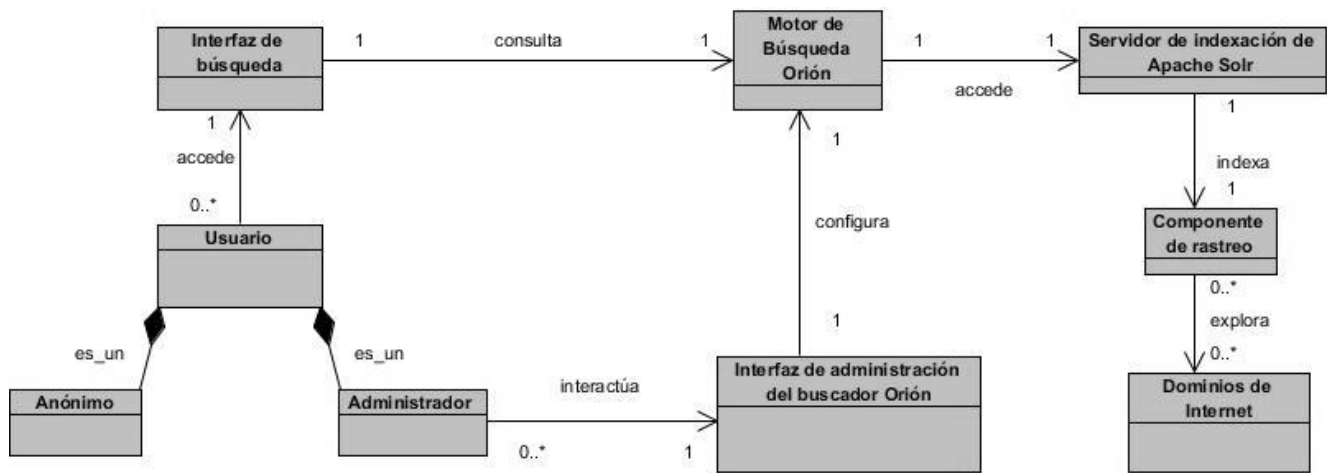


Figura 1: Modelo de dominio

El usuario accede a la interfaz principal del buscador Orión, este le muestra en su interfaz las opciones que proporciona y los resultados de la búsqueda realizada. El Motor de Búsqueda Orión consulta el servidor de indexación de Apache Solr que le devuelve el resultado de la búsqueda y que este a su vez indexa los contenidos del Componente de rastreo que explora la web aleatoriamente en busca de nuevos contenidos.

Se entiende por:

La **Interfaz de búsqueda de Orión** a la interfaz web que permite a los usuarios realizar una búsqueda en la red y listar sus resultados.

El **servidor de indexación Apache Solr** permite almacenar de forma indexada los dominios escaneados en la red.

El **usuario** es la persona que se encuentra utilizando las facilidades que brinda el buscador Orión.

El usuario **anónimo** solo realiza búsquedas introduciendo una consulta en el buscador Orión.

El **administrador** es un tipo de usuario autenticado que tiene total dominio de las funcionalidades del sistema y permiso para gestionar cualquiera de sus funciones.

El **motor de Búsqueda Orión** es el sistema informático que se comunica con el servidor de indexación para dar respuesta a las solicitudes de los usuarios y devuelve una lista de resultados encontrados ordenados por un algoritmo de *ranking*.

El **componente de rastreo** es un programa automatizado que recorre la web, buscando dominios y las páginas que los componen, y en general todos los archivos que pueda encontrar, visitan los diferentes sitios web de una manera generalmente aleatoria.

Los **dominios de internet** son las distintas direcciones que componen la Web, es una serie de letras, números, y/o símbolos con los que se identifica el lugar donde se almacena la información.

La **interfaz de administración del buscador Orión** muestra al usuario administrador, que posee los máximos privilegios, las opciones necesarias para gestionar el sistema y consultar los cambios efectuados.

2.4 Requerimientos del sistema

Los requerimientos del sistema no son más que los requisitos funcionales confeccionados de los diferentes encuentros y contactos con el cliente que resumen lo que el software será capaz de realizar, estos describen que se debe hacer para satisfacer las necesidades de nuestros clientes. Estos se convierten en los algoritmos, códigos y la lógica del sistema.

2.4.1 Requisitos funcionales

Un requerimiento funcional es una 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 (Jacobson, y otros, 2000).

A continuación, se listan los requisitos funcionales de la propuesta de solución:

- RF1: Crear categoría
- RF2: Modificar categoría
- RF3: Eliminar categoría
- RF4: Listar categoría
- RF5: Buscar categoría
- RF6: Aprobar solicitud de un dominio propuesto por un usuario
- RF7: Enviar notificación a partir de la solicitud de un usuario a la inclusión de un dominio

- RF8: Desplazar prioridad de uno o varios dominios para ordenar visualización
- RF9: Establecer parámetros para ordenar la visualización de los dominios de una categoría en un orden previamente definido
- RF10: Crear dominio
- RF11: Modificar dominio
- RF12: Eliminar dominio
- RF13: Listar dominio
- RF14: Buscar dominios
- RF15: Crear solicitud
- RF16: Modificar solicitud
- RF17: Eliminar solicitud
- RF18: Listar solicitud
- RF19: Buscar solicitud
- RF20: Mostrar un directorio con categorías y sus dominios
- RF21: Permitir varios niveles de categorías

2.4.2 Requisitos no funcionales

Los requerimientos no funcionales son capacidades o cualidades que el producto debe tener, o sea, características que hagan al producto atractivo, rápido, usable o confiable. Están estrechamente vinculados a los requisitos funcionales, puesto que una vez que está definido lo que el sistema debe hacer, es necesario especificar cómo ha de hacerlo. Pueden llegar a marcar la diferencia entre un producto bien aceptado por los clientes y usuarios o uno de poca o ninguna calidad y aceptación (Jacobson, y otros, 2000).

Requisitos de Apariencia o Interfaz

- Diseño orientado a una interfaz simple, visible, con una tipografía clara, intuitiva e interactiva, amigable para cualquier tipo de usuario.
- El diseño gráfico estará guiado por los colores de interfaz que establezca el proyecto Orión.
- El módulo de Administración estará sujeto a las pautas de diseño de Xilema Base-Web.

Requisitos de Seguridad

- Garantizar que la información sea editada únicamente por los usuarios que tienen este permiso.

- Avalar que las funcionalidades del módulo se muestren según el nivel de acceso del usuario activo.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminaciones).
- Asegurar que los mecanismos utilizados para lograr la seguridad, no interrumpan el acceso de los usuarios al directorio web.

Requisitos de Hardware del servidor

- CPU con 2 *cores* de procesamiento a velocidad de 2.0 GHz.
- 2 Gb de RAM.
- 10 Gb de espacio libre en disco.

Requisitos de software del servidor

- Ubuntu Server 14.04 LTS.
- PHP 5.6.
- Apache 2.4.7.

Requisitos de Soporte

- Se permitirá realizar modificaciones posteriores para adaptar mejoras al sistema o en caso que cambien las necesidades de los clientes.
- El software estará bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse.

Requerimientos de Disponibilidad

- El sistema deberá estar disponible las 24 horas del día para trabajar en el momento deseado y contar con servidores de respaldo en caso de alguna contingencia.

Requerimientos de Confiabilidad

- Solamente los usuarios registrados con permiso de administración pueden modificar la configuración de la aplicación.
- Todas las salidas del sistema tienen que tener 100% de veracidad y precisión.

Restricciones de diseño

- La aplicación se desarrolla sobre el sistema operativo Ubuntu 14.04 LTS.
- Como herramienta para la gestión, procesamiento, almacenamiento y consulta de la base de conocimiento se debe utilizar MongoDB 2.4.9.
- El módulo se organiza utilizando el marco de trabajo Symfony 2.7.
- El IDE que se debe utilizar es Netbeans 8.0.

Requisitos legales, de Derecho de Autor y otros

- Las herramientas que se utilizan para el desarrollo del producto están basadas en licencias GNU/GPL.

2.5 Modelo de casos de uso del sistema

El modelo contiene actores como entidad externa que ejecutan e interactúan con los casos de uso del sistema. Los casos de uso, representan las funcionalidades de la aplicación y satisfacen los requisitos del cliente definidos por el equipo de desarrollo del proyecto Orión. El diagrama de casos de uso del sistema define las relaciones e interacciones entre casos de uso y actores del sistema.

2.5.1 Definición de los actores del sistema

A continuación, se definen los actores del sistema y la jerarquía que estos presentan en la aplicación por el rol que juegan en la misma.

Usuario administrador: Representa una persona (usuario) que una vez autenticado en la aplicación posee la habilidad de realizar cualquier cambio en el directorio temático, desde su configuración hasta el análisis de las solicitudes de inscripción y reportes enviados por los demás usuarios, por lo que posee la máxima jerarquía que otorga el módulo.

Usuario común: Es la persona (usuario) que una vez autenticado en la aplicación puede realizar una solicitud de inscripción de un dominio en una categoría, este tipo de usuario posee una jerarquía intermedia ya que puede realizar otras funciones que no sea la básica búsqueda de dominios por categorías, pero no llega a realizar cambios directos en la configuración del módulo.

Usuario anónimo: Se le asigna a cualquier persona (usuario) que accede a la aplicación a través de la web para realizar una búsqueda. Posee la jerarquía mínima en el módulo, solo puede realizar las búsquedas de dominios restringidas por categorías.

2.5.2 Definición de los casos de uso del sistema

El diagrama de casos de uso del sistema es el modelo donde se definen los actores del sistema, los casos de uso que tendrá cada uno y la relación entre ellos.

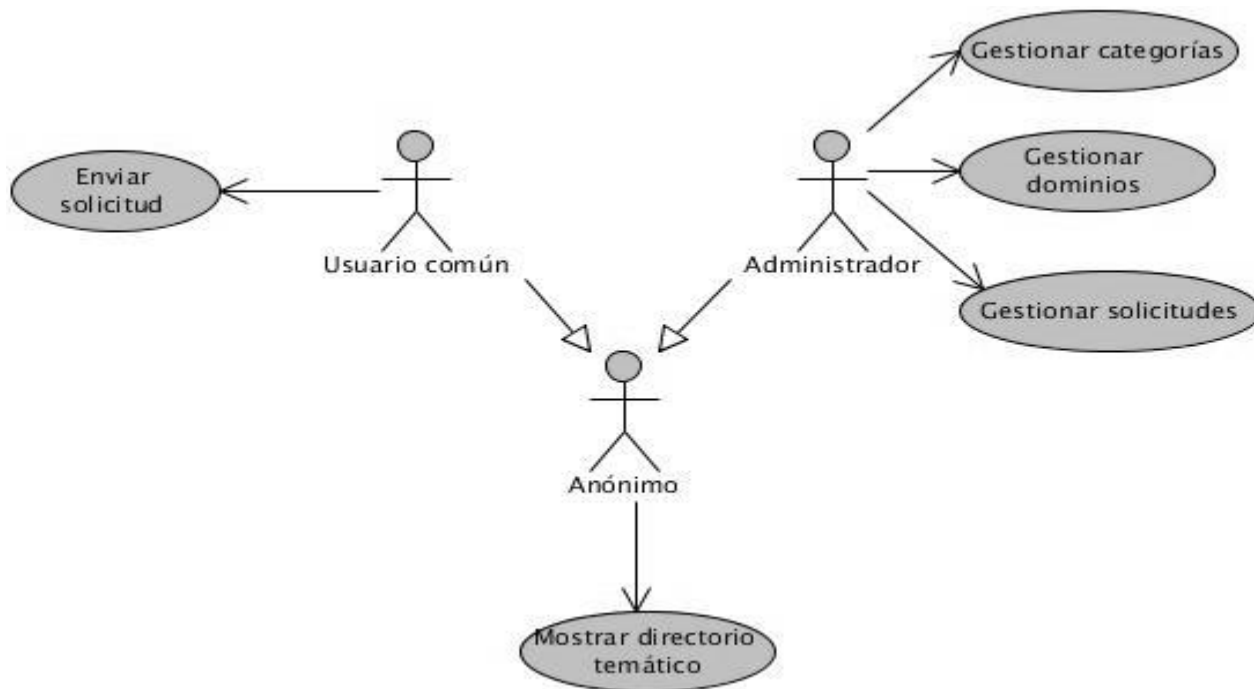


Figura 2: Diagrama de caso de uso del sistema

2.5.3 Descripción de los casos de uso del sistema.

A continuación, se realiza la descripción de uno de los principales casos de uso del sistema, el resto de las descripciones se encuentran en el **anexo #1, 2, 3, 4**

Tabla 2.0: Descripción del caso de uso del sistema “**Gestionar categorías**”.

Objetivo	Poder crear, modificar, eliminar, listar y buscar categorías en el directorio temático.	
Actores	Administrador:(Inicia)	
Resumen	El CU inicia cuando el Administrador desea crear, modificar, eliminar, listar y buscar categorías en el directorio temático para realizar alguna acción. El CU termina cuando se finaliza alguna de estas operaciones.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	El administrador debe estar autenticado.	
Pos condiciones	Se gestionan las categorías del directorio temático.	
Flujo de eventos		
Flujo básico Gestionar categorías		
	Actor	Sistema
1.	El administrador al seleccionar la opción administrar en su lista de opciones de usuario, elige el botón Directorio y elige la alternativa Gestionar categoría.	1.a) El sistema muestra la interfaz de la alternativa Gestionar categoría con las opciones de crear, modificar, eliminar, listar o buscar categorías.
.	Selecciona una de las siguientes acciones: <ul style="list-style-type: none"> • Crear (Ir a la Sección 1 Crear categoría). 	Terminar caso de uso.

	<ul style="list-style-type: none"> • Modificar (Ir a la Sección 2 Modificar categoría). • Eliminar (Ir a la Sección 3 Eliminar categoría). • Listar (Ir a la Sección 4 Listar categoría). • Buscar (Ir a la Sección 5 Buscar categoría). 	
--	--	--

Flujos alternos

1.a) No existen categorías actualmente registradas en el sistema

	Actor	Sistema
3.	Solo puede elegir la opción: <ul style="list-style-type: none"> • Crear (Ir a la Sección 1 Crear categoría). 	3.a) El sistema muestra el mensaje “ <i>No existen categorías actualmente en el sistema</i> ”.

Sección 1: “Crear categoría”

Flujo básico Crear categoría

	Actor	Sistema
4.	El administrador da clic en la opción Crear del menú Gestionar categoría.	4.a) Muestra la interfaz de Crear categoría.
5.	El administrador llena los siguientes campos: <ul style="list-style-type: none"> • Nombre de categoría • Identificador 	5.a) Valida los datos y si son correctos crea la nueva categoría.

	• Descripción	
Flujos alternos		
1.a Datos de entrada incorrectos		
	Actor	Sistema
6.	El administrador introduce incorrectamente los datos de entrada.	6.a) El sistema lanza una excepción con el siguiente mensaje <i>“Los datos introducidos poseen valores erróneos.”</i>
5.a Campos de entrada vacíos		
	Actor	Sistema
7.	El administrador deja vacío algunos de los siguientes campos: Nombre de categoría, Identificador, Descripción.	7.a) El sistema muestra el siguiente mensaje: <i>“Los datos introducidos poseen valores erróneos.”</i> Adicionalmente se muestra <i>“Este valor no debería estar vacío”</i> encima del o los campos enviados sin datos.
Sección 2: “Modificar categoría”		
Flujo básico Modificar categoría		
	Actor	Sistema
8.	El administrador da clic en la opción Modificar del menú Gestionar categoría.	8.a) Muestra la interfaz de Modificar categoría.
9.	El administrador selecciona de la lista de categorías la que desea modificar a través de la opción editar.	9.a) El sistema muestra la interfaz de editar categoría.
10.	Cambia los datos que desea modificar y los	10.a) Valida los datos y si son correctos crea la

	guarda para preservarlos	nueva categoría.
Flujos alternos		
10.a Datos de entrada incorrectos		
	Actor	Sistema
11.	El administrador introduce incorrectamente los datos de entrada.	11.a) El sistema lanza una excepción con el siguiente mensaje <i>“Los datos introducidos poseen valores erróneos.”</i> .
5.a Campos de entrada vacíos		
	Actor	Sistema
12.	El administrador deja vacío algunos de los siguientes campos: Nombre de categoría, Identificador, Descripción.	12.a) El sistema muestra el siguiente mensaje: <i>“Los datos introducidos poseen valores erróneos.”</i> Adicionalmente se muestra <i>“Este valor no debería estar vacío”</i> encima del o los campos enviados sin datos.
Sección 3: “Eliminar categoría”		
Flujo básico Eliminar categoría		
	Actor	Sistema
13.	El administrador da clic en la opción Eliminar del menú Gestionar categoría.	13.a) Muestra la interfaz de Eliminar categoría.
14.	El administrador selecciona de la lista de categorías la que desea eliminar a través de la opción habilitada.	14.a) El sistema muestra la interfaz donde confirma si se desea eliminar la categoría seleccionada o cancelar la opción.
Sección 4: “Listar categoría”		

Flujo básico Listar categoría		
	Actor	Sistema
15.	El administrador da clic en la opción Listar del menú Gestionar categoría.	15.a) Muestra la interfaz de Listar categoría.
16.		16.a) El sistema muestra la lista de categoría.
Sección 5: “Buscar categoría”		
Flujo básico Buscar categoría		
	Actor	Sistema
17.	El administrador da clic en la opción Buscar del menú Gestionar categoría.	17.a) Muestra la interfaz de Buscar categoría.
18.	Introduce los datos de la categoría a buscar	18.a) El sistema muestra la categoría deseada.

2.6 Arquitectura de la propuesta de solución

La arquitectura de software demuestra la organización, funcionamiento y conexión entre las partes de un software. Su objetivo principal es garantizar un mejor desempeño en el desarrollo de las aplicaciones. Proporciona robustez, portabilidad y flexibilidad a la aplicación. Es considerado el elemento de enlace entre los requerimientos y la implementación del sistema (Gamma, 2013).

Para el desarrollo de la solución propuesta se decide utilizar el estilo arquitectónico Modelo-Vista-Controlador, en lo adelante MVC, dado que el marco de trabajo Symfony 2 basa su funcionamiento y estructura en el mismo.

Este es un estilo de llamada y retorno y tiene como propósito separar la lógica del negocio, que se encuentra en la controladora, de la interfaz gráfica, en el modelo, dándole mayor autonomía a la hora de realizar cambios en alguna de ellas sin que afecte la otra. Dentro de sus componentes encontramos la vista que es la encargada de construir todas las interfaces que posee el módulo y mostrársela al usuario a petición de la controladora y según las interacciones con este. La controladora es el cerebro del software en ella se encuentra programada toda la lógica del negocio y es la que decide que hacer en cada momento según las peticiones del usuario, interactúa con el modelo y la vista y establece lo que cada una debe hacer. El modelo representa la información que tanto el usuario como la aplicación pueden acceder y manipular, es el resultado almacenado de las iteraciones del directorio temático.

Dentro de las ventajas del empleo de este patrón arquitectónico se enuncian las siguientes (García, y otros, 2011):

- Facilita la agregación de múltiples representaciones de los mismos datos.
- Alta escalabilidad pues se puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware.
- Facilita el mantenimiento de errores.
- Crea independencia de funcionamiento.

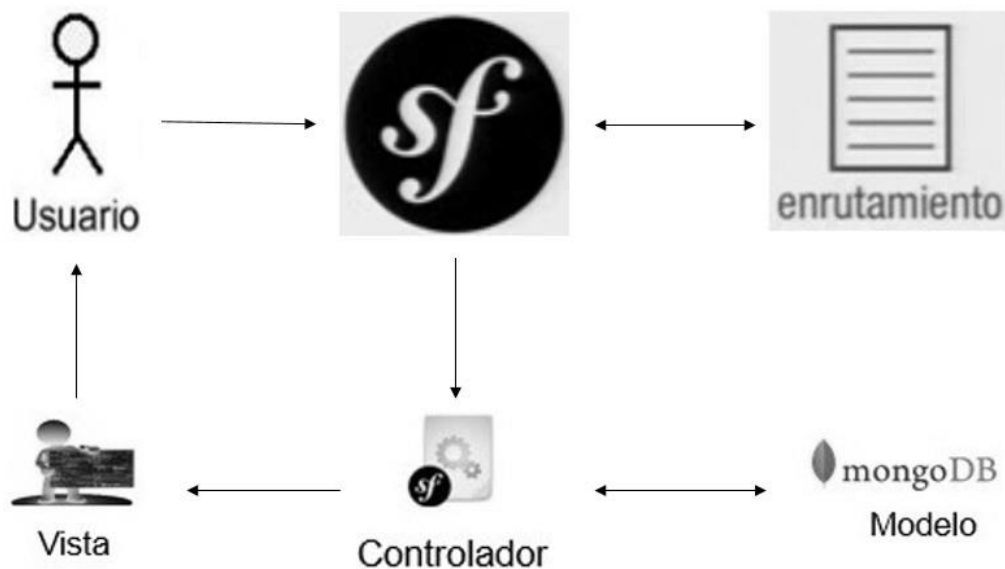


Figura 3: Patrón arquitectónico MVC

2.7 Patrones de diseño

Los patrones de diseño permiten la reutilización de diseños exitosos, basando nuevos diseños en experiencias previas. Estos patrones resuelven problemas específicos de diseño, y vuelven el diseño orientado a objetos más flexible, elegante y extremadamente reutilizable (Pressman, 2011).

Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos del diseño de interacción o interfaces. Son parejas de problema-solución, que codifican buenos principios y sugerencias relacionados generalmente con la asignación de responsabilidades. (Pressman, 2011)

Se clasifican en dos categorías: GRASP, es un acrónimo de *General Responsibility Assignment Software Patterns* (en español, patrones generales de software para asignar responsabilidades) y GoF es un acrónimo de *Gang-Of-Four* (en español, La pandilla de los cuatro). Para el diseño del Directorio temático del buscador Orión se tuvieron en cuenta los siguientes patrones: GRASP (Patrones Generales de Software para Asignación de Responsabilidades), que describen los principios fundamentales de la asignación de responsabilidades a objetos (Pressman, 2011).

Alta cohesión: Define que la información almacenada en una clase debe ser coherente y estar relacionada con esta. Propone además, que no se debe saturar una clase de métodos, sino asignar las responsabilidades a cada clase correspondiendo a la información que almacena (Larman, 2003). Este patrón se utiliza en todas las clases del módulo, permite incrementar la claridad y facilitar la comprensión del diseño. Simplificar el mantenimiento y las mejoras. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que colaboran entre sí y con otros objetos para simplificar su trabajo. Una clase con alta cohesión es relativamente fácil de mantener, entender y reutilizar.

El directorio temático implementa la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase *CategoryController*, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.

Bajo acoplamiento: El uso de este patrón garantiza que las clases estén lo menos ligadas posible entre sí, de tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases (Larman, 2003). Este patrón se utiliza en todas las clases del módulo, el

uso de este no afecta los cambios en otros componentes, fácil de entender de manera aislada y es conveniente para reutilizar.

El acoplamiento es una medida de la fuerza con que un elemento está conectado, tiene conocimiento o confía en otros elementos. El objetivo de este patrón consiste en mantener un bajo nivel de dependencia de otros elementos, por lo que constituye un principio que debe estar presente en todas las decisiones de diseño con lo que se reduce el impacto de los cambios. Se debe asignar las responsabilidades de forma tal que las clases se comuniquen con el menor número de clases que sea posible. Una clase con bajo acoplamiento no depende de muchas otras.

La clase *CategoryController* hereda únicamente de *Controller* para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Experto en información: Se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. Expresa la intuición común de que los objetos hacen las cosas relacionadas con la información que tienen. Se mantiene el encapsulamiento de la información, puesto que los objetos utilizan su propia información para llevar a cabo las tareas. Normalmente, esto conlleva un bajo acoplamiento, lo que da lugar a sistemas más robustos y más fáciles de mantener. Se distribuye el comportamiento entre las clases que contienen la información requerida, por tanto, se estimula las definiciones de clases más cohesivas y ligeras que son más fáciles de entender y mantener (Larman, 2003).

Este patrón plantea que se debe asignar una responsabilidad al experto en información, o sea, a la clase que cuenta con la información necesaria para cumplir la responsabilidad. El comportamiento se distribuye entre las clases que cuentan con la información requerida para cumplir con la responsabilidad asignada.

El uso del patrón es evidenciado en la utilización del *document manager* encargado de mapear la Base de Datos. Se utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades. Las clases de abstracción de datos, poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla de la base de datos, por ejemplo el uso de la clase *Category*, que contiene atributos, constructor y métodos para representar la información de la entidad.

2.8 Modelo de diseño

El modelo de diseño es aquel que describe la realización de los casos de uso del sistema, y sirve como una abstracción del modelo de implementación y el código fuente del software. Su objetivo fundamental es transmitir, a través de la representación mediante diagramas, una comprensión en profundidad de los aspectos relacionados con los requerimientos no funcionales y restricciones relacionados con los lenguajes de programación, en resumen, cómo cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades (Larman, 2003).

2.8.1 Diagrama de clases del diseño con estereotipos web

Un diagrama de clases del diseño es un tipo de diagrama estático que describe la estructura de un sistema; mostrando sus clases, atributos, métodos y las relaciones entre ellos. Las clases de diseño se especifican utilizando la sintaxis del lenguaje de programación elegido. Teniendo en cuenta la complejidad que el marco de trabajo utilizado incorpora en los diagramas.

A continuación, se relaciona el diagrama de clases del diseño correspondiente al CU Gestionar categoría. En el mismo se muestran las relaciones entre las clases participantes, teniendo como característica particular sobre los diagramas de clases tradicionales, el uso de estereotipos UML especiales que son específicos para los diagramas de clases web, los cuales permiten modelar la arquitectura y funcionamiento de aplicaciones de este tipo. El resto de los diagramas de clases del diseño se encuentran en el **Anexo # 5, 6, 7**.

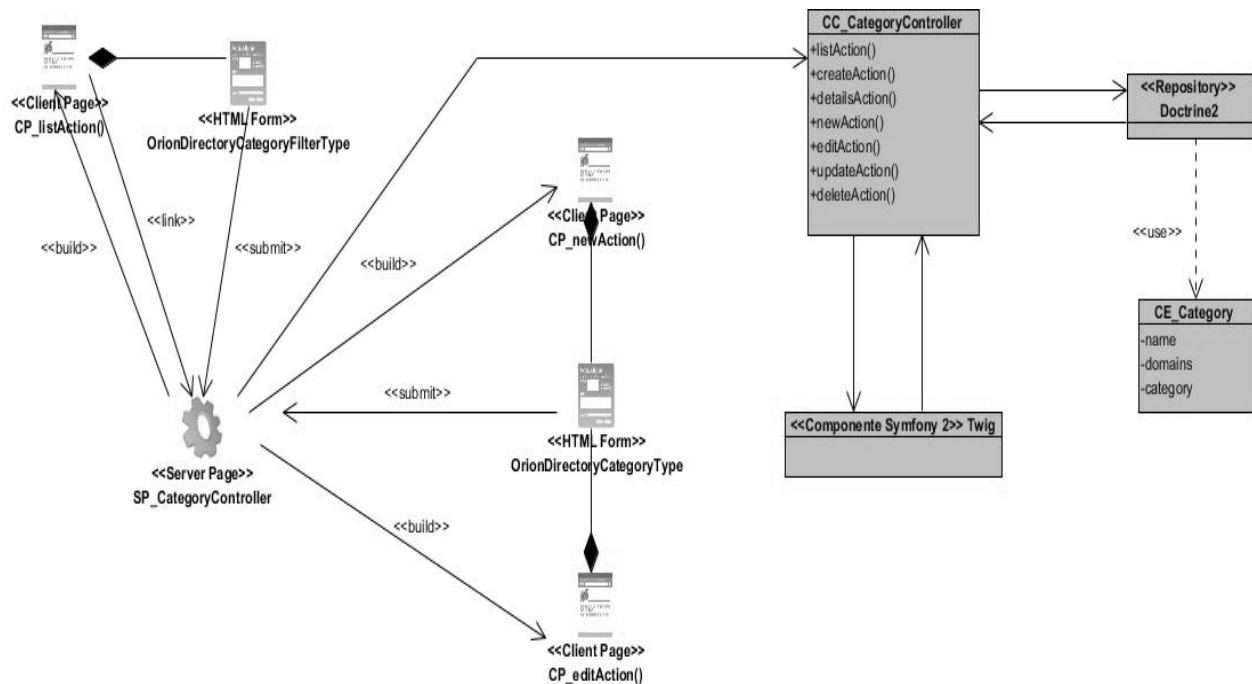


Figura 4: Diagrama de Clases de Diseño utilizando estereotipos web del Caso de Uso Gestionar categoría.

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

2.9 Diseño de la base de datos

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

El módulo que se propone hace uso de una base de datos ya que es necesario persistir una cantidad considerable de información relacionada con los dominios registrados, sus categorías y las solicitudes enviadas por los usuarios.

Como ya se ha definido anteriormente, el gestor encargado de almacenar la información capturada por la aplicación es MongoDB, una base de datos documental escalable y de alto rendimiento.

A continuación, se muestra el modelo Entidad Relación de la base de datos.

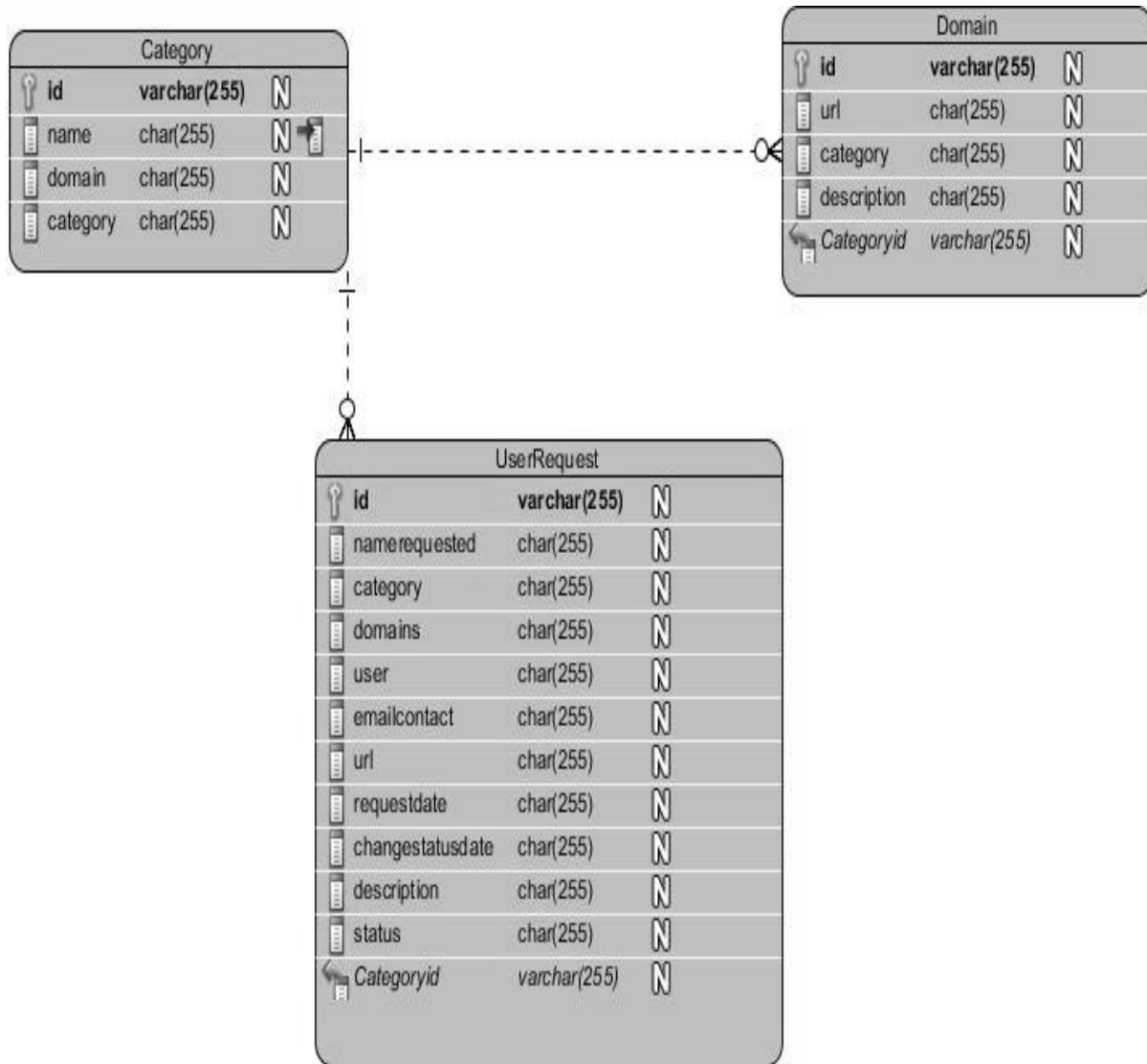


Figura 5: Diagrama Entidad Relación de la base de datos

Aunque no se encuentra representada en el diagrama, existe una relación directa entre la entidad *UserRequest* de la propuesta y la entidad *User* del módulo de gestión de usuario del sistema Orión. Esta relación implica una dependencia directa entre dichos módulos por lo que se necesita tener acceso al código fuente del módulo de gestión de usuario de Orión.

2.10 Modelo de despliegue

Es una colección de nodos y arcos donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar (Jacobson, y otros, 2000).

Los diagramas de despliegue son capaces de describir la arquitectura física del sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software. Permite una mejor comprensión entre la correspondencia de la arquitectura de software y la arquitectura de hardware.

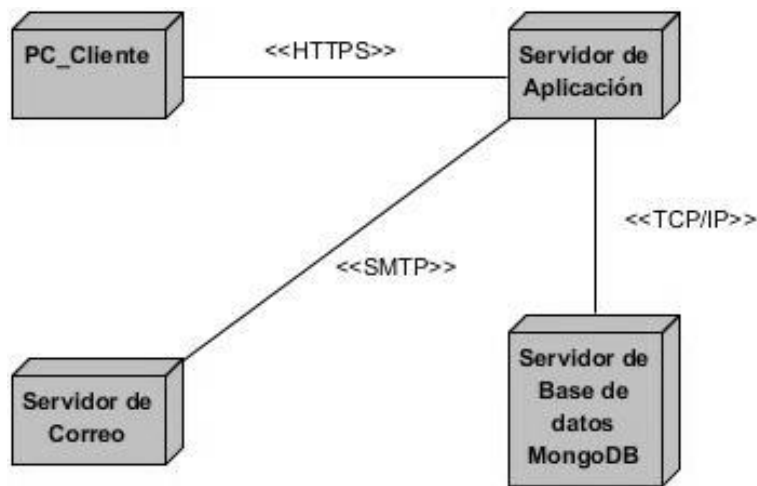


Figura 6: Diagrama de Despliegue

Se describe a cada nodo y su interrelación del Diagrama de Despliegue:

PC_Cliente: Contiene la estación de trabajo cliente, necesita un navegador web que conecte con la aplicación web alojado en el servidor de aplicaciones, utiliza el protocolo de comunicación HTTP.

Servidor de Aplicación: Aquí se encuentra el código fuente de la aplicación, brinda a los usuarios las interfaces necesarias durante todos los procesos del negocio. Esta estación se comunica con el servidor de Base de datos MongoDB donde se almacenan los datos del directorio temático realizando la comunicación a través del protocolo TCP.

Servidor de Bases de datos MongoDB: Este nodo es el encargado del almacenamiento de los datos del directorio temático. Se comunica con el servidor de aplicaciones a través del protocolo TCP, permite el acceso a la información que tanto el usuario como la aplicación pueden manipular, es el resultado almacenado de las iteraciones del software.

Servidor de Correo: Este nodo es el encargado del almacenamiento de los datos del envío y recepción del correo electrónico que genera los avisos de la gestión del directorio temático. La comunicación se establece a través del protocolo SMTP.

Conclusiones del capítulo

La solución propuesta utiliza las mismas tecnologías que emplea el proyecto de desarrollo de Orión, lo que garantiza una compatibilidad arquitectónica entre Orión y el Directorio Temático, y una futura integración a partir de cambios mínimos. La realización del modelo de dominio ofrece una visión más palpable de los componentes y los conceptos asociados a la creación del directorio temático del buscador Orión, así como las relaciones entre estos. La asociación de funcionalidades con diferentes tipos de usuarios, implicó un tratamiento arquitectónico intencionado para permitir el desarrollo de funcionalidades de forma agrupadas, y la reutilización de componentes y métodos propios. La arquitectura MVC seleccionada permitió definir la estructura del software y su interrelación entre los diferentes componentes. El diagrama entidad relación permite identificar una dependencia directa con la entidad *user* que pertenece a un módulo de Orión, por tanto, para el desarrollo de la propuesta es necesario contar con el código fuente del módulo de Orión encargado de la gestión de usuarios.

Capítulo 3: Implementación y validación del Directorio temático de Orión

3.1 Introducción

La fase de implementación en el desarrollo de un producto de software, es el mecanismo donde se ponen en práctica todas las descripciones y arquitecturas propuestas en las fases de análisis y diseño, es el complemento del trabajo de las fases que lo preceden dentro del proceso de desarrollo de software. La implementación ofrece una materialización precisa de los requisitos, con el objetivo de conformar el producto final requerido por el cliente.

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

3.2 Diagrama de componentes

Este tipo de diagrama ilustra la relación que existe entre componentes de software, así como la ubicación de cada uno de ellos dentro del módulo, como se implementan las clases en término de componentes. Describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponible en el entorno de implementación y en el lenguaje de programación utilizado. Además, muestra las dependencias entre componentes.

A continuación se muestra la primera de las tres partes que componen el diagrama de componentes propuesto, el cual está acorde con los requerimientos del sistema a implementar y con el patrón arquitectónico MVC propuesto por Symfony 2. El resto de las partes del componente se encuentran en los **anexos 13, 14**.

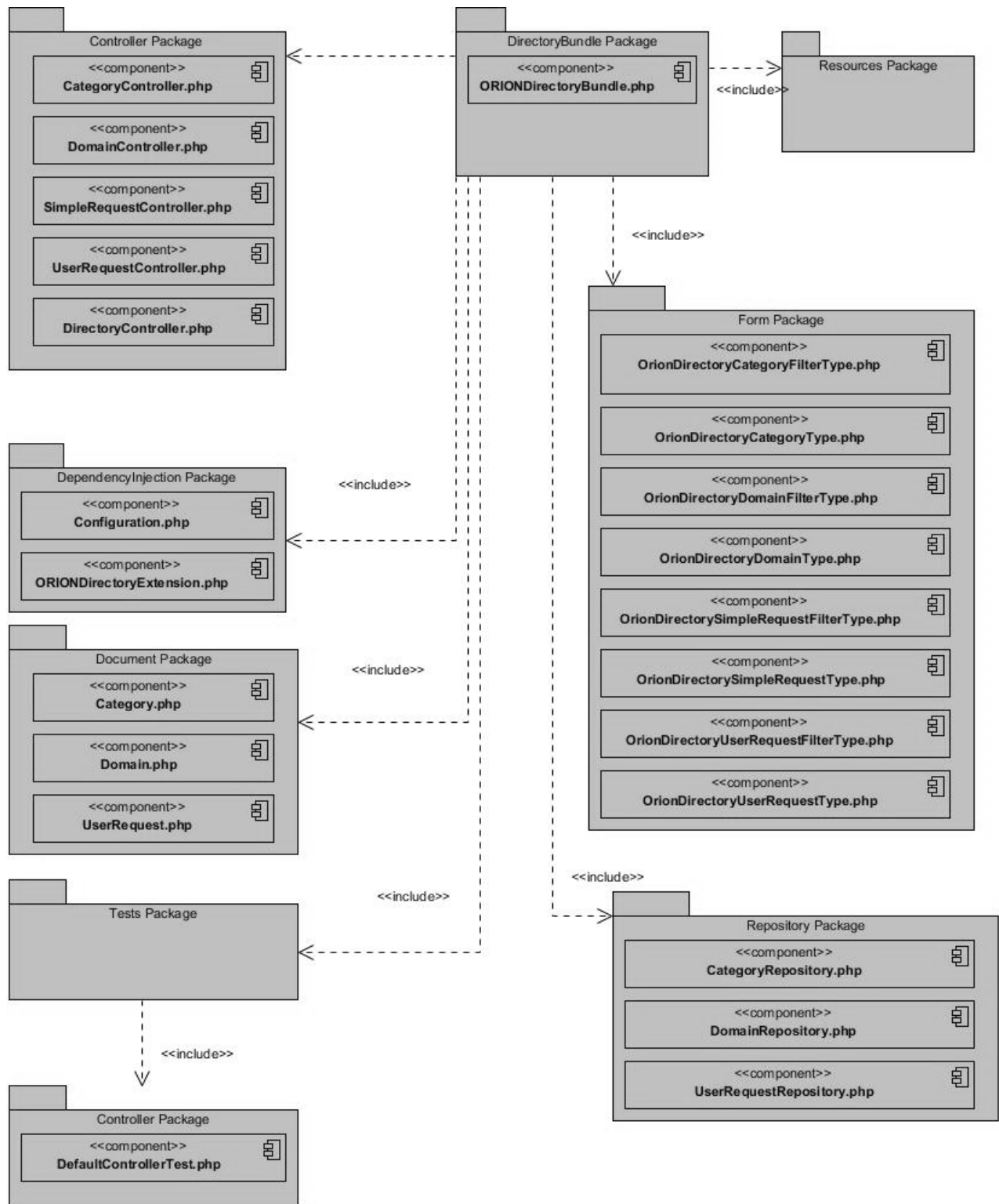


Figura 7: Diagrama de componentes. Parte 1.

Se describen los paquetes que componen el Directorio temático del buscador Orión:

ORIONDirectoryBundle: agrupa en su interior todos los componentes, estableciendo una estructura organizativa acorde al patrón de arquitectura MVC.

Controller: contiene las clases controladoras encargadas de procesar las peticiones de las páginas cliente y devolver las respuestas con la información requerida.

Form: alberga los formularios HTML utilizados para el envío de información por parte de los usuarios.

Document: en este paquete residen las clases documentos que se corresponden con las colecciones de la base de datos en MongoDB.

Repository: concentra aquellas clases que auxilian a las clases documentos, y que contienen las funciones especiales para la obtención de información desde la base de datos.

DependencyInjection: se utiliza para la configuración de inyecciones de dependencias, contiene los ficheros encargados de precargar los elementos de los cuales depende el módulo.

Resources: metapaquete que contiene los paquetes relacionados con las vistas y las configuraciones del módulo.

3.3 Estándares de codificación

Los estándares de codificación son especificaciones o estilos que establecen la forma de generar el código de las aplicaciones informáticas. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. En muchas ocasiones, los sistemas de cómputo son implementados por varios programadores, la adopción inicial de un único estilo de codificación constituye uno de los factores de mayor peso en la calidad, rendimiento, legibilidad y capacidad de mantenimiento del producto final (Revisiones de código y estándares de codificación, 2016).

Para facilitar el entendimiento del código y precisar un modelo a seguir, se adoptaron determinados estándares de codificación que a continuación se describen, agrupados por los aspectos en los que fueron utilizados. Cuando se trabaja con Symfony2 es aconsejable seguir su estándar de codificación. Esto implica imitar el código ya existente.

Estructura

1- Hacer uso de las etiquetas cortas (<?);

- 2- Finalizar las clases con la etiqueta usual de cierre (`?>`);
- 3- La indentación se debe realizar utilizando cuatro espacios;
- 4- Utilizar el carácter de salto de línea (`0x0A`) para finalizar las líneas;
- 5- Agregar un único espacio después de cada delimitador coma;
- 6- No poner espacios después de la apertura de un paréntesis y antes del cierre del mismo;
- 7- Agregar un único espacio alrededor de operadores (`==, &&,...`);
- 8- Agregar un único espacio antes de los paréntesis de apertura de una palabra clave de control;
- 9- Agregar una línea en blanco antes de la sentencia *return*;
- 10- No agregar espacios al final de las líneas;
- 11- Utilizar llaves para indicar el cuerpo de las estructuras de control sin importar el número de sentencias que éstas contengan;
- 12- Colocar las llaves en sus propias líneas para clases, métodos y declaración de funciones;
- 13- Separar las sentencias condicionales y las llaves de apertura con un único espacio sin dejar una línea en blanco;
- 14- Declarar explícitamente la visibilidad de clases, métodos y propiedades (el uso de `var` está prohibido);
- 15- Utilizar constantes de tipo PHP nativas en minúsculas: `false`, `true` y `null`. Lo mismo aplica para `array()`;
- 16- Utilizar letras mayúsculas para constantes, con palabras separadas por guiones bajos;
- 17- Definir una clase por archivo;
- 18- Declarar las propiedades de las clases antes de los métodos;
- 19- Declarar los métodos públicos primero, luego los protegidos y finalmente los privados.

Convención de Nombres

- 1- Utilizar camelCase y no guiones bajos, para variables, funciones y nombres de métodos;

- 2- Utilizar guiones bajos para definir opciones, argumentos y nombres de parámetros;
- 3- Utilizar los *namespace* para todas las clases;
- 4- Añadir como sufijo Interface a las interfaces;
- 5- Utilizar caracteres alfanuméricos y guiones bajos para nombres de archivos.

Documentación

- 1- Agregar los bloques PHPDoc para todas las clases, métodos y funciones;
- 2- Las anotaciones @package y @subpackage no son utilizadas.

3.4 Validación del sistema

Las pruebas son los procesos que permiten verificar y revelar la calidad de un producto de software, una vez terminado la implementación del producto es necesario realizarle pruebas, con el objetivo de detectar errores en la aplicación y la documentación, se identifican posibles fallos de implementación, calidad, o usabilidad de un software, este proceso resulta de gran importancia ya que da una medida de la calidad del mismo, siempre que se lleve a cabo de la forma correcta. En este proceso se ejecutan pruebas dirigidas al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. Las mismas persiguen como objetivo fundamental, la detección de las no conformidades que este procedimiento arroja y que son de vital importancia en el proceso de desarrollo del software.

3.4.1 Pruebas funcionales

Las pruebas funcionales son aquellas que se aplican a un software determinado, con el objetivo de validar que las funcionalidades implementadas funcionen de acuerdo a las especificaciones de los requisitos definidos con anterioridad, también denominadas pruebas de comportamiento o de caja negra se centran en los requisitos funcionales del software. Estas permiten al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (Mañas, 2016). Al ejecutarlas el especialista solo se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

Se comprueban las funcionalidades diseñando casos de prueba que definen cómo proceder. Estos casos de prueba incluyen las combinaciones de datos a usar que son los correctos o esperados por la

lógica del sistema y los no válidos o no esperados por el programa. Las pruebas se le deben dedicar una gran parte del esfuerzo total del desarrollo. Se deben planificar correctamente desde el inicio y establecer qué hacer, cómo hacer, quién va a hacer y en qué condiciones hacer las comprobaciones (Pressman, 2011).

El método de prueba que se utilizará será Partición equivalente, el mismo consiste en diseñar un caso de prueba por cada caso de uso, para los casos de uso incluidos no se diseñarán casos de pruebas, sino que estos se agregarán en el caso de uso que los contenga y se probarán cuántas veces parezcan en el software.

A continuación se presenta el caso de prueba correspondiente al caso de uso Gestionar categoría. Para consultar el resto ver el documento “Descripción de casos de prueba del directorio temático”.

Condiciones de ejecución:

El administrador se ha autenticado en el sistema Orión.

Tabla 3.0: Caso de prueba del EC1.1 CU “Gestionar categorías”

Escenario	Descripción	Categoría	Categoría padre	Descripción de la categoría	Respuesta del sistema	Flujo Central
EC1.1 Crear categoría de forma correcta.	El administrador crea una nueva categoría de forma correcta.	V Matemática	V Ciencias	V Esto es una categoría de ciencias.	Valida los datos y si son correctos crea la nueva categoría y muestra el siguiente mensaje: "Completado, La categoría se ha añadido satisfactoriamente"	1. El administrador selecciona la opción Administrar Categorías dentro del menú desplegable Administrar. 2. El sistema muestra la página Administrar Categorías. En esta página aparece el listado de las categorías existentes, en forma de tabla, dividido en las columnas Categoría, Categoría Padre, Descripción de la categoría (donde aparecen las opciones Crear, Editar y

						<p>Eliminar); y el formulario para filtrar los datos de dicho listado, conformado por el campo Categoría y el botón Buscar.</p> <p>3. El administrador selecciona la opción Crear categoría.</p> <p>4. El sistema muestra la página Adicionar Categoría, donde aparece un formulario con los campos Categoría, Categoría Padre y Descripción de la Categoría y los botones Adicionar y Cancelar.</p>
--	--	--	--	--	--	--

A continuación se muestra un gráfico donde se evidencian los resultados obtenidos durante las 4 iteraciones de pruebas funcionales realizadas al módulo desarrollado.

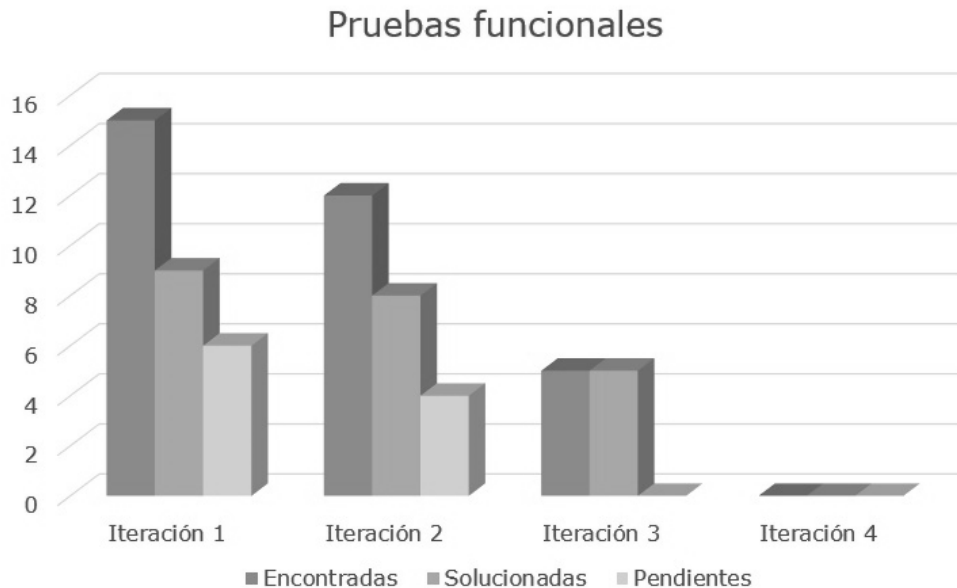


Figura 8: Resultado de las pruebas de funcionales

3.4.2 Pruebas estructurales

La prueba de unidad es la prueba enfocada a los elementos probables más pequeños del software, consiste en una prueba estructural (o caja blanca), lo cual requiere conocer el diseño interno de la unidad puesto que verifica la lógica interna y el flujo de datos (Andrews, y otros, 2003). Estas definen un conjunto amplio de acciones de comprobación que abarcan todas las características que determinan la calidad de un software.

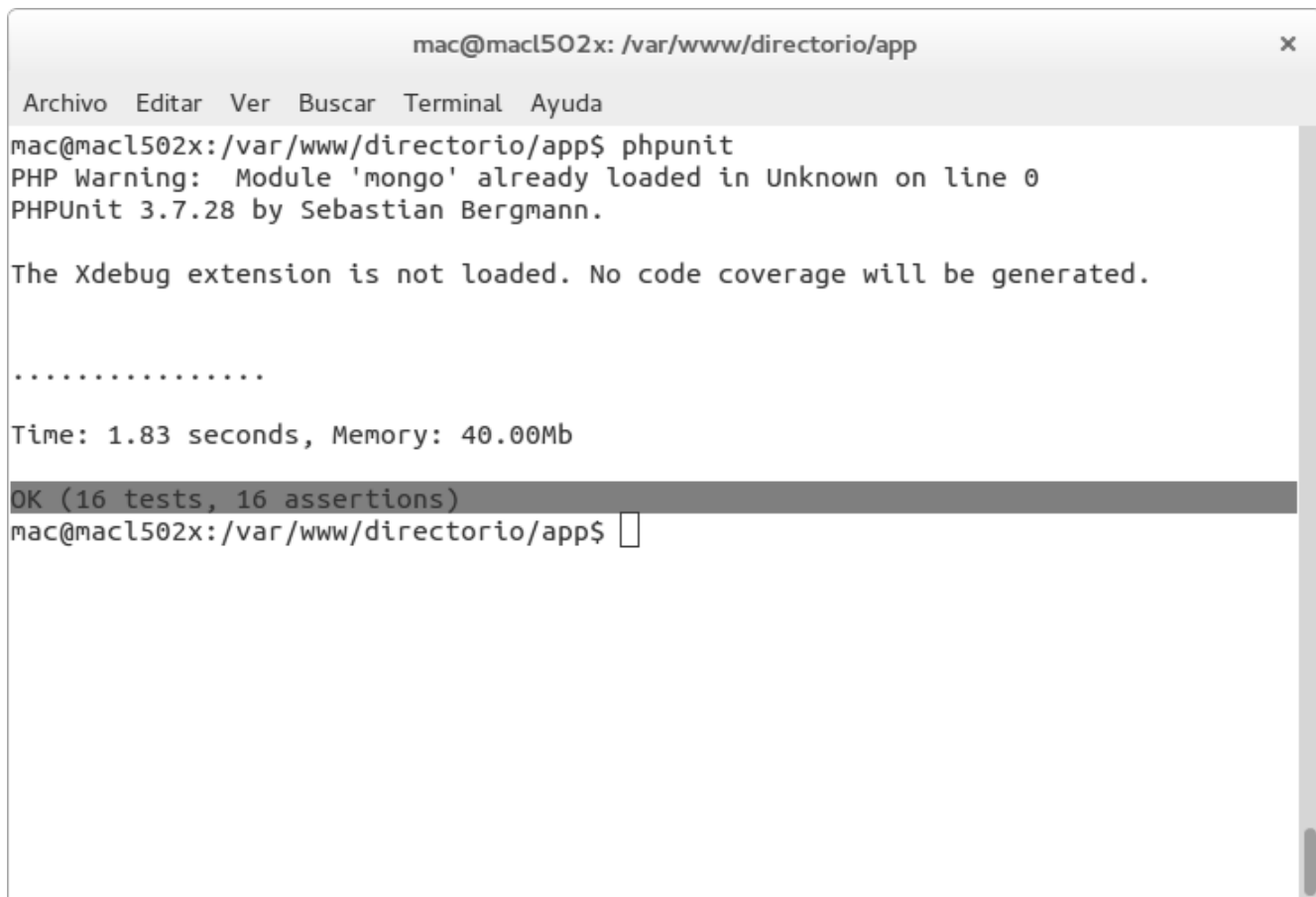
Las pruebas de caja blanca se ejecutaron mediante el sistema de integración continua Jenkin. En la configuración de las pruebas se habilitaron las herramientas Pdepend, phpmid-ci, phpcpd, phpcs-ci.

A partir de la herramienta PMD Warnings del Jenkins para mostrar los resultados de las pruebas se identificó que en una primera iteración un total de 80 *warnings* de prioridad normal, separadas en 3 categorías (*Code Size Rules*, *Naming Rules*, *Unused Code Rules*), distribuidas en 4 tipos (*ShortVariable*, *TooManyPublicMethods*, *UnusedFormalParameter*, *UnusedLocalVariable*). El resultado de la tercera iteración de pruebas muestra 15 alertas asociados al uso de estándares propios del trabajo en Orión o clases del propio *framework* que no deben ser modificadas y que no constituyen no conformidades. El desarrollo de las pruebas puede verse mediante los **anexos 8, 9, 10, 11**.

En la ejecución de pruebas estructurales se evidenció que el entorno Jenkins utilizado posee configuraciones que generan alertas asociadas al uso de estándares definidos en el proyecto Orión;

tales como el empleo de variables de dos caracteres para la definición de los identificadores de las clases que almacenan su información en base de datos. Estas configuraciones globales generan falsos positivos, por lo que se recomienda que el entorno de prueba del proyecto tenga configuraciones específicas que sobrescriban las globales para evitar la generación de este tipo de alertas.

Adicionalmente para comprobar el correcto funcionamiento del módulo se aplicaron pruebas unitarias mediante el uso de PHPUnit. En la figura 9 se observa la ejecución satisfactoria de las 16 pruebas ejecutadas.



```
mac@mac1502x: /var/www/directorio/app
Archivo Editar Ver Buscar Terminal Ayuda
mac@mac1502x:/var/www/directorio/app$ phpunit
PHP Warning: Module 'mongo' already loaded in Unknown on line 0
PHPUnit 3.7.28 by Sebastian Bergmann.

The Xdebug extension is not loaded. No code coverage will be generated.

.....

Time: 1.83 seconds, Memory: 40.00Mb

OK (16 tests, 16 assertions)
mac@mac1502x:/var/www/directorio/app$
```

Figura 9: Resultado de pruebas de PHPUnit.

3.4.3 Pruebas de integración

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

Una vez realizadas las pruebas funcionales a cada componente interno de manera independiente, y verificado que las funcionalidades implementadas se corresponden de acuerdo a los requisitos funcionales y no funcionales establecidos; se pudo comprobar el correcto funcionamiento de los componentes mediante el estudio del flujo de datos entre ellos. Posterior a estas pruebas, se hace necesaria la realización de pruebas de integración, con la finalidad de validar la compatibilidad y el funcionamiento de las interfaces que comunican las diferentes partes que componen la solución informática.

Algunas de las acciones que se llevaron a cabo para la realización de estas pruebas fueron:

- ✓ Comprobación del funcionamiento del enlace entre el motor de búsqueda Orión y el directorio temático implementado.
- ✓ Verificación de la conexión entre el módulo y la base de datos en MongoDB.
- ✓ Correspondencia entre los datos manejados por los usuarios en el directorio temático y los existentes en las respectivas instancias del bundle de usuarios de Orión (ORIONUserBundle).
- ✓ Validación de las rutas de acceso, en el *routing* principal del sistema Orión.
- ✓ Utilización de los layout del bundle en que se definen las vistas principales de Orión (UIBundle).

Mediante la ejecución de las pruebas de integración se logró verificar la operación conjunta de cada uno de los componentes del Directorio temático del buscador Orión, junto a los componentes que conforman el bundle de usuarios de Orión (ORIONUserBundle) y layout del bundle en que se definen las vistas principales de Orión (UIBundle), haciendo énfasis en la interacción entre estos y comprobando el correcto funcionamiento de los mismos.

3.4.4 Pruebas de carga y Estrés

En el diseño del plan de pruebas de rendimiento para el módulo, se tuvo en cuenta, las diferentes acciones que los usuarios pueden realizar al conectarse para gestionar categorías, dominios y solicitudes, y cuando un usuario realiza la navegación a través del directorio. En una muestra de 100 usuarios conectados concurrentemente, con un período de subida de 1 segundo (tiempo de espera de cada usuario para realizar una petición) la herramienta JMeter generó el reporte mostrado en la tabla 4.0.

Tabla 4.0: Resultados de las pruebas de rendimiento mediante el uso de Jmeter

Total de muestras	6161
Rendimiento (peticiones/segundo)	65,2
Porcentaje de errores	0%
Media de respuesta	707
Tiempo de respuesta (milisegundos/petición)	707

Como se puede observar en el análisis del resumen arrojado por la herramienta JMeter, para un total de 6161 muestras que se le realizaron al módulo, se alcanzó un rendimiento de 65,2 peticiones por segundo, con un porcentaje de 0% de errores para cada petición realizada. Como se puede apreciar en la tabla 4.0 la media de respuesta es de 707 lo que significa un tiempo de respuesta de 707 milisegundos por cada petición. A partir de este resultado se considera que el directorio temático responde correctamente ante situaciones de carga y estrés en función de la muestra utilizada.

Para realizar la validación de los resultados de las pruebas se verificó, además, la influencia que ejercen algunos parámetros sobre el tiempo de respuesta de la aplicación, elementos que pueden provocar cierto retraso en la generación de la carga en las pruebas e interferencia en la obtención de los resultados, dichos parámetros a tener en cuenta son:

Ambiente de prueba: El despliegue fue simulado en una máquina virtual a partir de una computadora con Proxmox instalado como sistema operativo. En el mismo se creó una máquina virtual con Ubuntu 14.04 de 32 bits, con 2.00 GB de memoria RAM y 4 procesadores (2 *sockets*, 2 *cores*) a velocidad 3.2 GHz.

Utilización del CPU: Una utilización del 98% demuestra un uso completo del procesador, lo que representa que la configuración establecida en entornos de alta concurrencia debe utilizar un mayor número de procesadores.

Utilización de la memoria: Un uso de la memoria física de la máquina menor al 50% es aceptable. Una configuración de la máquina con menor cantidad de procesadores mostró como resultado un menor uso de CPU pero un peor rendimiento en la aplicación cuando se le vuelven a aplicar las mismas

pruebas mediante el sistema JMeter, por lo que se deduce que con las características del servidor para lograr un mayor rendimiento la configuración óptima es la probada.

Conclusiones del capítulo

La elaboración de los diagramas de componentes, permitió una mejor comprensión de la estructura de los componentes del módulo implementado. En el desarrollo se respetaron los estándares arquitectónicos y de diseños definidos en el proyecto Orión con vistas a lograr una adecuada integración del módulo con el resto de los componentes del proyecto. Las pruebas estructurales realizadas permitieron la aplicación de métricas y estándares de codificación internacionales, a partir de las diferentes herramientas de pruebas unitarias. Como resultado de estas pruebas se evidencia que el modulo implementado cumple con las métricas y estándares aplicados. La propuesta fue probada mediante varios tipos de pruebas funcionales garantizando el correcto funcionamiento del mismo, así como su completa integración con el resto de los componentes de Orión. La simulación de conexiones concurrentes mediante las pruebas de carga y stress, garantizan una respuesta acertada a las peticiones que pueden realizar 100 usuarios consumiendo datos generados por el módulo, en un tiempo medio inferior a 1 segundo. Aunque las pruebas de rendimiento garantizan adecuados tiempos de respuesta, se recomienda la realización de estas pruebas en un servidor dedicado; ya que uno de los resultados es la utilización del 98% de CPU en el entorno de prueba, lo que puede producir sobrecarga en el servicio en un entorno real.

Conclusiones generales

De manera general se puede concluir sobre la presente investigación:

1. El estudio del estado del arte de los directorios web permitió demostrar que los directorios temáticos existentes a nivel nacional e internacional no satisfacen las necesidades requeridas para la recuperación de información restringida a ámbitos temáticos del buscador Orión.
2. La elaboración de los artefactos propuestos por la metodología de desarrollo y el levantamiento de requisitos permitieron una mayor comprensión del módulo a desarrollar, así como la identificación de los procesos y características del mismo.
3. El directorio temático permite la completa gestión de categorías, dominios y solicitudes por parte de los usuarios interesados en incluir sus dominios en alguna de las categorías, así como la navegación interactiva del directorio a partir de la búsqueda de categorías, subcategorías y dominios en la interfaz de búsqueda de Orión.
4. Las pruebas realizadas garantizan un correcto funcionamiento del módulo implementado, así como una total integración al resto de los componentes del sistema Orión.

Recomendaciones

Durante la realización del módulo y como resultado de la investigación realizada, a pesar de haberse cumplido los objetivos de este trabajo, surgieron algunas ideas que pueden ampliar las prestaciones del software creado. El autor de este trabajo recomienda al equipo de desarrollo del proyecto Orión profundizar en:

- ✓ La aplicación de algoritmos de minería de datos que permitan establecer la posición de los dominios en categorías de forma automática.
- ✓ El desarrollo de funcionalidades que utilicen la información almacenada en la base de datos Solr del buscador Orión, para recomendar dominios que pueden ser insertados en el directorio temático.

Bibliografía

- Andrews, A. 2003.** *Test adequacy criteria for UML design models. Software Testing, Verification and Reliability.* 2003.
- Arias, J. 2006.** *Web semántica: significado, componentes y funcionamiento.* 2006.
- Arroyo, N. 2007.** La biblioteca como usuaria de la web 2.0. [En línea] 2007. [Citado el: 3 de Diciembre de 2015.] eprints.rclis.org/archive/00009787/01/Arroyo&Merlo_FESABID07.pdf.
- Ayala, J. 2007.** *Modelo de recuperación booleano.* 2007.
- Baeza-Yates, R. 2005.** *Crawling a Country: Better Strategies than Breadth-First for web Page Ordering.* 2005.
- Bannister, T. 2012.** *What is Apache?* 2012.
- Bonilla, S. 2011.** Web semántica, marcadores discursivos y metarepresentación. [En línea] 2011. <http://dialnet.unirioja.es/servlet/articulo?codigo=2254309>.
- Castells, P. 2008.** La web semántica. [En línea] 2008.
- Castrillon, J. 2008.** *Bases de datos, motores de búsqueda e índices temáticos: herramientas fundamentales para el ejercicio médico.* 2008.
- Chiavenato, I. 2006.** *Introducción a la Teoría General de la Administración.* s.l. : McGraw-Hill Interamericana, 2006.
- Cobo, J. 2008.** *Planeta web 2.0. Inteligencia colectiva o medios fast food.* 2008.
- Consortium, W. 2014.** World Wide Web Consortium(W3C). *CSS Specifications.* [En línea] 2014. [Citado el: 4 de Diciembre de 2015.] <http://www.w3.org/Style/CSS/current-work>.
- Cruz, J. 2004.** *Buscador web.* 2004.
- Cubadebate.** [En línea] 21 de Octubre de 2015. [Citado el: 7 de Diciembre de 2015.] <http://www.cubadebate.cu/prensa-cuba/>.
- Cubarte.** [En línea] 2014. [Citado el: 6 de Diciembre de 2015.] <http://archivo.cubarte.cult.cu>.

- Cubawebdirectory.** [En línea] 2013. [Citado el: 7 de Diciembre de 2015.]
<http://www.cubawebdirectory.com>.
- Delgado, A. 2012.** [En línea] 15 de 10 de 2012. [Citado el: 16 de Noviembre de 2015.] <http://dmi.uib.es>.
- Flores, E. 2014.** *METODOLOGIAS AGILES PROCESO UNIFICADO AGIL (AUP)*. 2014.
- Gamma, E. 2013.** *Design Patterns: Elements of Reusable Object-Oriented Software*. s.l. : Pearson Education, 2013.
- García, R. 2006.** *Modelos de Recuperación de la Información*. 2006.
- García, J. 2011.** *Un estudio comparativo de dos herramientas MDA: OptimalJ y ArcStyler*. s.l. : Departamento de informática y sistemas, Universidad de Murcia., 2011.
- Gómez, R. 2002.** *Estudio de la incidencia del conocieminto lingüístico en los Sistemas de Recuperación de información*. s.l. : Salamanca: Universidad de Salamnaca, 2002.
- GoogleDirectorio.** [En línea] 2014. [htt://www.cubadirectory.com](http://www.cubadirectory.com).
- Jacobson, I. 2000.** *El proceso unificado de desarrollo de software*. 2000.
- Java, T. 2012.** Conozca más sobre la tecnología Java. [En línea] 2012. [Citado el: 4 de Diciembre de 2015.] <http://www.java.com/es/about/>.
- Kiva, B. 2011.** [En línea] 2011. [Citado el: 28 de 11 de 2015.]
<http://www.searchoptimization.es/buscadores-search-engines/buscadores-search-engines.htm>.
- Larman, C. 2003.** *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. . s.l. : Pearson Educación, 2003.
- Liu, G. 2001.** *Guide to meta-search engines*. 2001.
- López, A. 2006.** *Modelos de Sistemas de Recuperación de Información Lingüística Difusa*. 2006.
- López, D. 2011.** PHP4 Y PHP5: ¿Cuál elegir? ¿Migrar o no Migrar? El advenimiento de PHP6 . [En línea] 2011. [Citado el: 4 de Diciembre de 2015.] <http://www.mnogosearch.org/doc33/msearch-intro.html#features>.
- Mañas, J. 2016.** Pruebas de Programas. [En línea] 28 de Abril de 2016.
<http://www.it.uc3m.es/tsps/testing.html>.

Méndez, E. 2002. *Metadatos y recuperación de información: estándares, problemas, y aplicabilidad en bibliotecas digitales.* 2002.

Hernández, H. 2000. *Metodologías Águiles en el Desarrollo de Software.*

Características de MongoDB. [En línea] 2013. [Citado el: 4 de Diciembre de 2015.]
<http://www.mongodb.com>].

Netbeans. 2014. NetBeans IDE 8.0 Release Information. . [En línea] 2014.
<http://netbeans.org/community/releases/70/>.

Núñez, M. 2002. Criterios para la evaluación de la calidad de las fuentes de información sobre salud en internet. *Acimed.* [En línea] 2002. [Citado el: 15 de Noviembre de 2015.]
http://scielo.sld.cu/scielo.php?pid=S1024-94352002000500005&script=sci_arttext.

Olivas, J. 2011. *Búsqueda Eficaz de información en la web.* Buenos Aires : EDUNLP, 2011.

Oller, J. 2003. *Acimed.* [En línea] 16 de 11 de 2003. [Citado el: 3 de Diciembre de 2015.]
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352003000600007&lng=es.

OpenUP. 2015. [En línea] Diciembre de 2015. [Citado el: 19 de Enero de 2016.]
<http://openup.blogspot.es/tags/open-up/>.

Visual Paradigm for UML. [En línea] 2012. [Citado el: 4 de Diciembre de 2015.]
<http://www.visualparadigm.com>.

PHP, Group. 2015. [En línea] 2015. [Citado el: 3 de Diciembre de 2015.] <http://php.net/>.

Potencier, F. 2014. Web PHP Framework. [En línea] 2014. [Citado el: 4 de Diciembre de 2015.]
<http://www.symfony-project.org/>.

Pressman, R. 2011. *Software Engineering: A Practitioner's Approach.* 2011.

OpenUP, Basic Lifecycle. 2014. Proceso de entrega: OpenUP, Basic Lifecycle. [En línea] Noviembre de 2014. [Citado el: 19 de Enero de 2016.]
http://epf.eclipse.org/wikis/openupsp/openup_basic/deliveryprocesses/openup_basic_process_0uyGoMlgEdmt3adZL5Dmdw_desc.html.

Mozilla. 2013. Java Scrip Overview. [En línea] 2013. [Citado el: 4 de Diciembre de 2015.]
<https://developer.mozilla.org/enUS/docs/Web/JavaScript/Overview>.

- Riehle, D. 2012.** Framework design. [En línea] 25 de Septiembre de 2012. [Citado el: 6 de Abril de 2016.] <http://dirkriehle.com/computer-science/research/dissertation/diss-a4.pdf>.
- Salton, G. 2006.** *Introduction to Modern Information Retrieval*. s.l. : McGraw-Hill, 2006.
- Santos, R. 2009.** *Introducción a los sistemas de recuperación de información*. La Habana : s.n., 2009.
- SOAP. 2015.** SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). [En línea] 17 de Abril de 2015. [Citado el: 22 de Marzo de 2016.] <http://www.w3.org/TR/soap12-part1/>.
- Sommerville, I. 2005.** *Ingeniería del software. 7 ed.* s.l. : Pearson Educación, 2005.
- Stanley, T. 2001.** [En línea] Febrero de 2001. [Citado el: 3 de Diciembre de 2015.] <http://www.ariadne.ac.uk/issue14/search-engines/>.
- Tolosa, G. 2008.** *Introducción a la Recuperación de Información*. 2008.
- Universidad Carlos III de Madrid. 2012.** Portal Universidad Carlos III de Madrid. [En línea] 29 de Febrero de 2012. [Citado el: 3 de Diciembre de 2015.] http://portal.uc3m.es/portal/page/portal/biblioteca/aprende_usar/calendario/bib_politecnica/internet_ingenieros/indices_directorios.
- Veron, H. 2004.** *Performance of four World Wide Web (WWW)*. 2004.
- Villota, O. 2004.** *Internet y redes: Herramientas de Búsqueda*. 2004.
- Wallis, J. 2014.** *Towards a Classification-based Approach to Resource Discovery on the Web*. s.l. : School of Computing & Information Technology, University of Wolverhampton, UK, 2014.
- Zayas, S. 2002.** Elementos conceptuales básicos útiles para comprender las redes de telecomunicaciones. [En línea] 2002. [Citado el: 3 de Diciembre de 2015.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352002000600003&lng=es&nrm=iso.