Subsistema de búsqueda de videos para el buscador cubano Orión

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

Autores:
Luis Reinier Ferreiro Vázquez
Roannel Fernández Hernández

Tutores:
Ing. Tamara Betancourt Santana
Ing. Eyeris Rodríguez Rueda
Ing. Serguey González Garay

Ciudad de la Habana, 18 de junio de 2015

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Ideoinformática de la Universidad de las Ciencias Informáticas; así como a dicha universidad para que hagan el uso que estimen pertinente con el mismo.

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

Autores:

Luis Reinier Ferreiro Vázquez

Roannel Fernández Hernández

Tutores:

Ing. Tamara Betancourt Santana

Ing. Eyeris Rodríguez Rueda

Ing. Serguey González Garay

Dedicatoria

De Luis

Quiero dedicarle la tesis a mi mamá, a mi papá, a Dioni que es para mí como un padre y a mis hermanas que espero que sigan estudiando.

De Roannel

Le dedico esta tesis a mis padres y a mi hermana, por haberme dado la fuerza necesaria para cumplir mi sueño de ser ingeniero. A mi abuelita Mama y a mi tía por ser las mejores segundas mamás del mundo. A la memoria de mi abuela materna y a la de mi bisabuelo, que dieron lo mejor de sí para que yo fuera un hombre de bien.

Agradecimientos

De Luis

Lo primero que una persona debe ser en la vida, es ser agradecida con aquellas personas que siempre han estado a su lado en momentos malos y buenos. Yo quiero agradecer a mi familia, amigos, amigas, profesores, tutores y la vida, por permitirme en estos 5 años de carrera crecer como persona y formarme como profesional para enfrentarme a los retos que la misma vida nos trae. Pero quiero agradecer de forma muy especial a: mi mamá, a mi papá, a Dioni que es para mí como un padre, a mis hermanas que espero que sigan estudiando, a mis amigos Eric, Walter y a Arleni por ser una persona muy especial, a mi amigo y compañero de tesis Roannel, a mi amigo, profesor guía y tutor Serguey; a mis tutores Tamara y Eyeris. También agradecer a Paul, a Odisleysi, a Osiris, a Delly, a Damián, a Hubert, a Javier, y todos los que nos ayudaron, aconsejaron y dieron su aporte para que nuestro trabajo fuera lo mejor posible.

De Roannel

Agradezco a mis padres por el amor y la confianza que han depositado en mí. Por guiarme, aconsejarme y ser mi mayor ejemplo ante los retos de la vida. Doy un millón de gracias a mi familia, en particular a mi hermana, mis abuelas, mi tía y mis primos. Les agradezco fundamentalmente por el apoyo que me han dado y por ser la mejor familia del mundo. Le agradezco además, a Serguey, que ha sabido ser profesor, tutor, guía y compañero durante estos últimos 5 años, pero sobre todas las cosas, es mi amigo y mi hermano. Le agradezco a mis compañeros de aula: Walter, Oscar, Maikel, Eric y en especial a Frank, que más que un compañero, es mi amigo, con el que he compartido momentos buenos y malos, tristes y alegres. Muchas gracias a Odi, por estar a mi lado y servirme de apoyo en los momentos que lo necesité. Mis más sinceros agradecimientos para mis amigos de la vocacional: Yeimer, Arzuaga y Alberto, que aunque hemos tomado caminos diferentes, siempre los consideraré parte de mi familia. Muchas gracias a mi compañero de tesis, que es un ejemplo de entrega y consagración. Muchas gracias a mis tutores Tamara y Eyeris por la ayuda brindada. Le agradezco también, a todas aquellas personas que de una forma u otra han contribuido a que mi sueño de ser ingeniero se haga realidad.

Resumen

Para localizar y procesar la gran cantidad de información existente en Internet de forma rápida y automática son utilizados los motores de búsqueda. Estos sistemas para ser completos, deben ser capaces de permitir la búsqueda de todo tipo de contenidos. Sin embargo, actualmente el buscador cubano Orión no cuenta con un mecanismo automatizado para la búsqueda y análisis de videos. Por tal motivo, la presente investigación propone desarrollar el subsistema de búsqueda de videos para el buscador cubano Orión, con el objetivo de disminuir el tiempo de las búsquedas de los videos publicados en la red cubana, realizadas con dicho motor de búsqueda. La propuesta de solución está compuesta por cuatro componentes: Rastreador, Indexador, Aplicación web y Traductor de consultas; los cuales, permiten a los usuarios realizar búsquedas simples y avanzadas, filtrando los contenidos atendiendo a criterios previamente definidos. Se diseñó y aplicó un experimento puro para comprobar la reducción del tiempo de las búsquedas de los videos publicados en la red cubana, realizadas con el motor de búsqueda Orión. Dicho experimento evidenció una reducción significativa del tiempo empleado por un usuario al realizar la búsqueda de videos utilizando el subsistema desarrollado.

Palabras clave: búsqueda, indexación, rastreo, video.

Índice general

Introducción

El hombre desde su aparición, ha sentido la necesidad de contar con un sistema organizado de comunicaciones, para intercambiar de forma efectiva pensamientos, ideas y sentimientos, que a su vez constituyen la esencia de la vida en sociedad. El constante avance científico y tecnológico protagonizado por la humanidad, ha posibilitado el perfeccionamiento de los medios de comunicación y con ellos, el surgimiento de Internet (también conocida como la gran red de redes). De este modo, millones de personas tienen acceso a una cantidad extensa y diversa de información en la red.

La evolución de Internet, como plataforma para la transferencia de información globalizada e inmediata, ha posibilitado el auge de nuevas estrategias, medios virtuales de comunicación y formatos publicitarios[1] para el intercambio y difusión de la información, entre los que se pueden destacar: documentos, audio, imágenes y videos.

Los formatos anteriormente mencionados tienen una determinada función e importancia según los distintos fines para los cuales sean utilizados. No obstante se centrará la atención en el “video”, teniendo en cuenta su relevancia para el desarrollo de la presente investigación.

Un video es la reproducción en forma secuencial de imágenes, que al verse con una determinada velocidad y continuidad, dan la sensación al ojo humano de apreciar un movimiento natural. Además de la imagen, el otro componente esencial es el sonido [1].

El contenido audiovisual ayuda a lograr una Web más dinámica y le sirve a los usuarios para comprender mejor los servicios que en ella se ofrecen, debido a que es más expresivo que otros tipos de contenido. Esto se debe fundamentalmente, a la combinación de audio e imágenes y que requiere menos esfuerzo mental ver un video que leer un texto [2].

Un estudio realizado por la Oficina de Publicidad Interactiva (IAB, por sus siglas en inglés) en 2011, muestra que el mensaje lanzado en un video logra más tiempo de retención y tiene mayor proporción de recuerdo que un mensaje escrito u otro formato publicitario. Por otra parte, tiene mayor impacto y es uno de los

[1] Se refiere a los recursos utilizados para divulgar o hacer pública la información.

contenidos más compartidos en Internet, pudiendo llegar a utilizarse como un medio muy efectivo dentro de la mercadotecnia viral[2] [4].

Para localizar y procesar la información de forma rápida y automática en Internet son utilizados los motores de búsqueda o buscadores: sistemas recolectores de información que son capaces de localizar cualquier contenido existente en la Web, tales como textos, imágenes, archivos de sonido, videos, entre otros [5].

Actualmente existen diversos buscadores en Internet como es el caso de Google, Bing, Ask y Yahoo, que permiten la búsqueda de diversos formatos en dicha red, entre los que se encuentra el video. Sin embargo, algunos de estos sistemas manejan el posicionamiento de los contenidos que muestran atendiendo a políticas definidas por ellos, e incluso deniegan el acceso a Cuba a muchos de sus servicios [6].

En Cuba, se está llevando a cabo un profundo proceso de informatización de la sociedad, que persigue como objetivo, elevar el desarrollo económico, tecnológico y social de la misma [7]. Enmarcado en este proceso, en el Centro de Ideoinformática (CIDI) de la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI), se ha implementado el motor de búsqueda cubano Orión, cuyo propósito fundamental es proveer a la red nacional de una herramienta para la búsqueda y análisis de contenidos web [8].

Con la utilización de este sistema, el posicionamiento de los sitios cubanos no se maneja basado en políticas exteriores y se puede acceder a todos los servicios que se presten. Además, representa un paso más hacia la soberanía tecnológica en Cuba e implica una reducción de costos para el país, al no tener que pagar necesariamente el acceso a Internet para realizar búsquedas de contenidos en la red cubana.

Por otra parte, un buscador puede ser considerado completo si permite realizar la búsqueda de todo tipo de contenidos. Sin embargo, actualmente Orión no cuenta con un mecanismo automatizado que permita la búsqueda y análisis de videos. No obstante, algunos usuarios recurren al subsistema de búsqueda de texto de esta herramienta para localizar dichos contenidos. Este subsistema está orientado a la búsqueda del texto existente en la Web; el cual, en pocas ocasiones contiene o hace referencia a videos relacionados con los resultados esperados.

Esta forma de búsqueda resulta engorrosa y lenta para los usuarios, los cuales deben comprobar los resultados obtenidos hasta encontrar el deseado y las probabilidades de obtener resultados relevantes para un criterio de búsqueda determinado son muy bajas. Esto provoca que los contenidos publicados en la subred cubana en formato audiovisual permanezcan prácticamente “invisibles” mediante las búsquedas realizadas con el motor de búsqueda cubano Orión; viéndose limitada su capacidad de divulgación entre los propios usuarios [9].

[2]Consiste en fenómenos en los que se busca expandir un mensaje, de forma en la que cada persona que lo recibe, lo transmite a su vez a varias personas más y así sucesivamente se produce una comunicación que se extiende de forma piramidal [3].

Atendiendo a lo anteriormente planteado, en áreas como la salud y la educación se ve afectado el intercambio de información entre profesionales e investigadores de diferentes países, donde en muchas ocasiones se originan conferencias en formato de video que pueden incrementar el conocimiento. Desde el punto de vista empresarial, el crecimiento económico y la probabilidad de éxito para las empresas se ven afectados y por consiguiente, la economía del país, al no propiciar una mayor visibilidad de los productos o servicios que se prestan.

Sobre la base de los elementos expuestos, se tiene como **problema de investigación**: ¿Cómo reducir el tiempo de las búsquedas de los videos publicados en la red cubana realizadas con el buscador cubano Orión?

El **objeto de estudio** comprende el proceso de recuperación de información.

Para darle solución al problema descrito, se ha planteado el siguiente **objetivo general**: Desarrollar el subsistema de búsqueda de videos para el buscador cubano Orión para reducir el tiempo de las búsquedas de los videos publicados en la red cubana realizadas con dicho buscador.

Se define como **campo de acción**: el proceso de recuperación de información de videos.

Para cumplir la meta propuesta se han trazado los siguientes **objetivos específicos**:

1. Construir el marco teórico conceptual y estudiar el estado del arte respecto a las tecnologías y funcionalidades de los sistemas de búsqueda de videos.
2. Diseñar el subsistema de búsqueda de videos para el buscador cubano Orión.
3. Implementar el subsistema de búsqueda de videos para el buscador cubano Orión.
4. Validar el correcto funcionamiento del subsistema implementado.

Luego de haber realizado una revisión bibliográfica y desarrollado el marco teórico, se formula la siguiente **hipótesis de investigación**: el subsistema de búsqueda de videos para el buscador cubano Orión, permite reducir el tiempo de las búsquedas de los videos publicados en la red cubana realizadas con dicho buscador. Teniendo en cuenta la hipótesis anteriormente planteada, se define como **variable independiente**: el subsistema de búsqueda de videos para el buscador cubano Orión, el cual consiste en un sistema de recuperación de información de videos. Como **variable dependiente**: el tiempo de las búsquedas de los videos publicados en la red cubana realizadas con el buscador cubano Orión. Esta variable hace alusión al tiempo que demora un usuario en encontrar videos en la red cubana utilizando el buscador cubano Orión.

Para dar cumplimiento a los objetivos trazados, se hace necesario desarrollar las siguientes **tareas de investigación**:

1. Estudio de los conceptos asociados al marco teórico de la investigación.

2. Caracterización de los sistemas y tecnologías que permitan la búsqueda de videos.
3. Definición de la arquitectura del subsistema de búsqueda de videos.
4. Determinación de la estructura de almacenamiento a utilizar para persistir la información.
5. Diseño de las interfaces visuales para buscar y visualizar la información de los videos.
6. Desarrollo de las funcionalidades que permitan el rastreo de los videos publicados en la Web.
7. Desarrollo de las funcionalidades que posibiliten realizar búsquedas simples de videos.
8. Desarrollo de las funcionalidades que posibiliten realizar búsquedas avanzadas de videos.
9. Desarrollo de las funcionalidades que permitan visualizar la información de los videos resultantes de una búsqueda.
10. Validación del subsistema teniendo en cuenta su funcionamiento, seguridad, rendimiento e integración con el buscador Orión.
11. Validación de la hipótesis de investigación.

Entre los **métodos** de trabajo científico utilizados en esta investigación se destacan los siguientes:

Métodos Teóricos:

Histórico - Lógico: Es utilizado para estudiar y determinar la evolución, comportamiento y tendencias actuales de las tecnologías y herramientas a utilizar en el desarrollo del subsistema de búsqueda de videos para el buscador cubano Orión.

Analítico - Sintético: Es utilizado para el análisis de los lenguajes, tecnologías y herramientas a utilizar en el desarrollo del subsistema de búsqueda de videos. Además, es empleado para examinar los documentos consultados durante la investigación.

Métodos Empíricos:

Modelación: Es empleado en la representación mediante diagramas de las características, procesos y componentes del sistema propuesto, así como la relación existente entre ellos.

La presente investigación se desglosa en los siguientes capítulos:

Capítulo 1: En este capítulo se exponen los conceptos asociados al dominio de la investigación y se realiza un estudio de algunos motores de búsqueda reconocidos nacional e internacionalmente. Además, se estudiaron las herramientas, metodologías y técnicas utilizadas para dar solución al problema planteado, así como las librerías que permiten el procesamiento de los videos.

Capítulo 2: En este capítulo se realiza el análisis y diseño del subsistema de búsqueda de videos para el buscador cubano Orión. Para ello, se hace uso de los artefactos que propone la metodología OpenUp[3] y las políticas de calidad de CIDI[4], como lo son: el modelo de dominio, la representación de los principales procesos mediante casos de uso, diagramas de clases del diseño y de colaboración. Además, se describe la propuesta de solución, así como los requisitos funcionales y no funcionales que debe satisfacer la misma.

Capítulo 3: En este capítulo se evidencian las actividades que se llevan a cabo durante las fases de implementación y pruebas, para el subsistema de búsqueda de videos para el buscador cubano Orión. En dichas etapas, se generan los artefactos pertenecientes a las mismas, como es el caso del diagrama de componentes, los estándares de codificación y los casos de prueba.

Se espera como **posible resultado** un sistema que permita la búsqueda simple y avanzada de los videos publicados en la red cubana. Además, se obtendrá el documento de la investigación que podrá ser consultado por las personas que lo deseen, teniendo en cuenta sus intereses.

[3]Metodología de desarrollo utilizada (Ver sección 1.4.5).
[4]Centro de desarrollo de software donde se lleva a cabo la presente investigación.

Capítulo 1

Fundamentos teóricos del subsistema de búsqueda de videos para el buscador cubano Orión

Con el objetivo de lograr una mayor comprensión del alcance de la investigación y esclarecer su objeto de estudio, se exponen en el presente capítulo, los conceptos asociados al dominio de la investigación y se realiza un análisis del estado del arte que la precede. Además, se incluye un estudio de las librerías que permitirán el procesamiento de los videos; así como las herramientas, metodología y tecnologías utilizadas para dar solución al problema planteado.

1.1. Conceptos asociados al dominio de la investigación

A continuación se relacionan los principales conceptos que ayudan a entender el desarrollo de la investigación.

Los **motores de búsqueda** (también conocidos como buscadores) son sistemas de recuperación de la información, que permiten a un usuario, dado un criterio de búsqueda introducido, obtener un subconjunto de aquellos documentos que mayor relevancia tengan para dicho criterio. Esto se realiza mediante ciertas operaciones sobre una base de datos [5].

También pueden ser definidos como programas computacionales que recorren Internet, examinando la información de acceso público en la red para su indexación y almacenamiento. Con este material se generan

bases de datos en constante actualización, que permiten su interrogación por palabra clave para la recuperación de la información [10].

Se puede decir entonces, que los motores de búsqueda son sistemas que acceden a los servidores[1] de una determinada red, para el análisis y almacenamiento de la información pública contenida en ellos. De este modo, se mantiene la base de datos actualizada, para dado un criterio de búsqueda introducido por un usuario, permitir obtener un subconjunto de aquellos documentos que mayor relevancia tengan para dicho criterio.

Un motor de búsqueda sigue los enlaces presentes en las páginas o documentos e incorpora a su base de datos el resultado del análisis del contenido de las URLs[2]. A este proceso se le conoce como **rastreo** y es realizado por un componente de los motores de búsqueda al que comúnmente se le llama araña (en inglés, *spider*) o rastreador (en inglés, *crawler*).

Al proceso de almacenar la información ordenadamente para acelerar su búsqueda y seleccionarla con mayor exhaustividad y pertinencia, independientemente de que se solicite o no, se le denomina **indexación** [13].

Algunos autores también definen la indexación como: *"operación destinada a representar los resultados del análisis del contenido de un documento o de una parte del mismo, mediante elementos (términos de indexación) de un lenguaje documental o natural, orientados a facilitar la posterior recuperación de los documentos indexados"* [14].

Teniendo en cuenta los elementos antes mencionados, se puede concluir que la indexación es el proceso de representar[3] el resultado del análisis del contenido de un documento o de una parte del mismo; para facilitar y acelerar la búsqueda de la información y seleccionarla con mayor exhaustividad y pertinencia.

Otro de los aspectos a tener en cuenta para un mejor entendimiento del problema planteado es el concepto de **video**, el cual puede ser definido como: *"medio de comunicación con unos elementos simbólicos determinados, que permite la creación de mensajes por el usuario, cuya concepción técnica es la imagen electrónica configurada a partir de una serie de instrumentos tecnológicos, que posee una versatilidad de usos mayoritariamente controlados por el usuario"* [15].

[1] Componente de la arquitectura cliente - servidor que debe estar indefinidamente preparado para recibir peticiones del cliente y establecer el diálogo para el intercambio de información [11].

[2] URL es el acrónimo de Uniform Resource Locator. Es una cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos disponibles en Internet. Esto posibilita que el navegador sepa dónde y cómo solicitar la información que se encuentra en un dominio particular [12].

[3] Se refiere al modo en que se estructura la información para ser identificada posteriormente.

Algunos autores definen al video, como la tecnología de la captación, grabación, procesamiento, almacenamiento, transmisión y reconstrucción por medios electrónicos digitales o analógicos de una secuencia de imágenes que representan escenas en movimiento [16].

Por tanto, se puede afirmar que el video es un medio de comunicación, cuya concepción técnica es la reproducción secuencial de imágenes, representando escenas en movimiento, que al verse con una determinada velocidad y continuidad, dan la sensación al ojo humano de apreciar el movimiento natural.

En el aspecto gráfico, un video se compone de una secuencia de imágenes denominadas fotogramas (también llamados *"frames"* o "cuadros"), cada una de las cuales aparece en pantalla un determinado espacio de tiempo, suficiente para crear en el espectador la sensación de continuidad entre fotogramas, generando así la visión global de una única escena en movimiento [1].

Los videos, como todos los archivos pertenecientes a un sistema de ficheros en un sistema operativo poseen **metadatos**. Estos, son toda aquella información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto que tiene la finalidad de facilitar su recuperación, autentificación, evaluación, preservación o interoperatividad [17].

1.2. Arquitectura de los motores de búsqueda

Los motores de búsqueda deben tener una arquitectura que les permita manejar enormes volúmenes de datos y soportar millones de consultas diarias, manteniendo un tiempo de respuesta que haga aceptable la realización de búsquedas por parte del usuario y por otro lado, que haga posible la constante actualización de la información que se tiene almacenada. Por estos motivos es que los motores de búsqueda generalmente son sistemas distribuidos [18], de los cuales se pueden distinguir varios componentes como se muestra en la siguiente imagen:

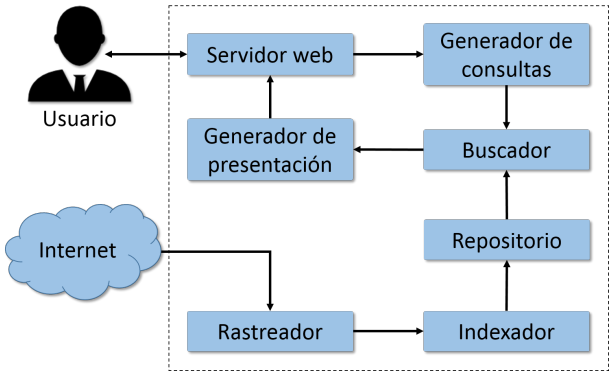

Imagen 1.1: Arquitectura genérica de un motor de búsqueda [18]

Según [18] y como se muestra en la imagen anterior, los componentes genéricos de un motor de búsqueda son:

Rastreador: Es el componente encargado del rastreo.

Indexador: Almacena los datos que recolecta el rastreador dentro de una estructura ordenada. Es el componente encargado del proceso de indexación.

Repositorio: Almacena la información utilizada para generar las respuestas a las solicitudes de los usuarios. La estructura, los datos y la forma en que se maneja este componente varían en gran medida, de acuerdo a la implementación específica del motor de búsqueda.

Servidor web: Este componente es el encargado de establecer la comunicación con el usuario a través del protocolo HTTP[4] o HTTPS[5], recibiendo consultas de este y enviándole el resultado de las mismas a través del mismo protocolo.

Generador de consultas: Recibe la consulta realizada por el usuario en forma de una cadena de texto, la interpreta y genera una consulta que pueda ser ejecutada dentro del repositorio.

Buscador: Es el componente encargado de acceder a la estructura interna de los datos (repositorios, índices, etc.), para satisfacer la solicitud del usuario.

Generador de presentación: Construye la vista a ser incluida en el documento HTML que recibe el usuario a través del servidor web, a partir de los datos que genera el componente Buscador.

1.3. Estudio de motores de búsqueda

En la actualidad existen numerosos sistemas de recuperación de información. Para identificar ventajas en el uso de estos sistemas y lograr una mejor comprensión de sus características y funcionalidades, se hace necesario realizar un estudio de algunos de estos motores de búsqueda. A continuación, se expone el estudio realizado de los sistemas homólogos, tanto en el ámbito nacional como internacional.

[4] Acrónimo de Hypertext Transfer Protocol. Según World Wide Web Consortium (W3C, principal organización de estándares internacionales de la WWW) e Internet Engineering Task Force (IETF, organización dedicada al desarrollo y promoción de normas que conforman el conjunto de protocolos de Internet) es un protocolo utilizado para distribuir información hipermedia.

[5] Es un protocolo basado en HTTP que protege la privacidad y la integridad de los datos que se intercambian.

1.3.1. Motores de búsqueda a nivel internacional

Google[6]

El motor de búsqueda Google, persigue como objetivo que sus usuarios encuentren la información que necesitan y consigan hacerlo de la forma más sencilla y rápida posible. Ofrece servicios como búsqueda de imágenes, libros, noticias, videos, documentos académicos, entre otros.

La búsqueda avanzada de videos en Google, incluye aspectos comunes con el resto de las búsquedas personalizadas que se ofrecen; como lo son, los criterios para encontrar todas, cualquiera o ninguna de las palabras de una frase, o incluso expresiones exactas. Dichos criterios, también se pueden incluir en el cuadro de búsqueda mediante el uso de operadores especiales[7] definidos por el propio buscador.

Específicamente para la búsqueda avanzada de videos se ofrecen filtros que permiten obtener resultados específicos por idioma, duración, fecha de publicación, calidad de los videos, sitio o dominio donde se encuentra el contenido, videos subtitulados e incluso, se incluye un filtro para evitar contenido para adultos[8] [19].

Google, considera un video de corta duración a aquellos con menos de 4 minutos, de duración media a los que tienen entre 4 y 20 minutos y de larga duración, a los de más de 20 minutos. Incluso, permite hacer búsquedas para cualquier duración, constituyendo este, el comportamiento por defecto.

El filtro para las fechas de publicación incluye rangos de fechas para la última hora, día, semana o mes, y de forma predeterminada busca videos publicados en cualquier momento. Por otra parte, permite buscar videos de cualquier calidad (comportamiento predeterminado) o solamente aquellos de alta calidad.

Bing[9]

Para una búsqueda avanzada de videos en el buscador Bing, se puede contar con filtros que limitan los resultados según la duración del contenido multimedia, la fecha en la que fue publicado en la Web, resolución con la que cuenta, fuente o lugar de origen y una búsqueda segura para excluir el contenido para adultos de los resultados.

Bing, considera un video corto a aquellos con menos de 5 minutos de duración y largo a los de más de 20 minutos, mientras que un video de mediana duración es aquel comprendido entre 5 y 20 minutos. Como comportamiento por defecto, Bing ofrece videos de cualquier duración dentro de sus resultados.

[6] Accesible en: http://www.google.com/.
[7] Palabras y símbolos que pueden ser utilizados en el cuadro de búsqueda para restringir los resultados a mostrar.
[8] Contenido pornográfico, desnudos o contenido sexual explícito.
[9] Accesible en: http://www.bing.com/.

Similar a Google, el filtro para las fechas de publicación de Bing, incluye rangos de fechas para la última hora, día, semana o mes, y busca videos publicados en cualquier momento de forma predeterminada. Por otra parte, el filtro de resoluciones limita los resultados de las búsquedas según las dimensiones de los fotogramas de los videos. Para ello, cuenta con valores de 360, 480, 720, 1080 píxeles o mayor. Como comportamiento predeterminado muestra videos de cualquier resolución.

Otro de los filtros que ofrece este buscador, está relacionado con la fuente o el origen de los videos. Para ello, se definen algunos sitios web populares como lo son: YouTube, MySpace, Dailymotion y Metacafe.

La búsqueda segura, puede ser desactivada para que se muestre el contenido para adultos o se puede activar de forma estricta o moderada. La búsqueda segura estricta evita que se muestren textos, imágenes y videos para adultos, mientras que la moderada, solamente imágenes y videos, siendo esta, la configuración predeterminada.

Ask[10]

Ask, es un motor de búsqueda que permite a los usuarios obtener respuestas a interrogantes planteadas diariamente en un lenguaje natural. Para el caso de la búsqueda de videos, el sistema incluye el criterio de relevancia para filtrar los resultados. Mediante este criterio, se pueden buscar los videos más relevantes (valor por defecto para este filtro), más populares, más vistos o los más recientes.

Además, se pueden buscar videos que hayan sido publicados en el día, la semana o el mes en curso. Dicho filtro, tiene como comportamiento predeterminado el de mostrar los resultados independientemente de la fecha en la que haya sido publicado el contenido.

Para filtrar los videos según su calidad, Ask ofrece a sus usuarios la posibilidad de obtener resultados independientemente de su calidad (comportamiento predeterminado). Incluso, permite filtrar la búsqueda para obtener videos en Alta Definición (HD por sus siglas en inglés) o en Tres Dimensiones (3D).

Según la concepción de Ask, un video es de corta duración si tiene menos de 4 minutos, de duración media los que tienen entre 4 y 20 minutos y de larga duración, los de más de 20 minutos. Estos criterios, son aplicados en el filtro que incluye el buscador para la búsqueda avanzada de videos. Sin necesidad de definir valores para este filtro, el motor de búsqueda muestra los resultados sin tener en cuenta la duración de los mismos.

Un elemento característico de este motor de búsqueda, es que permite filtrar los resultados según categorías, dígase: música, deporte, juegos, películas o noticias. Predeterminadamente, se muestran los resultados sin tener en cuenta la categoría a la que pertenecen. Por otra parte, Ask brinda a sus usuarios la posibilidad de reiniciar los filtros, es decir, que los mismos se comporten de manera predeterminada.

[10] Accesible en: http://www.ask.com/.

1.3.2. Motores de búsqueda a nivel nacional

La red cubana cuenta con algunas herramientas para la búsqueda y análisis de contenidos web, dentro de las que se encuentran: Busk2r[11], Lupa[12] y Orión[13]. Estos sistemas permiten realizar búsquedas de distintos tipos de contenidos como son imágenes y documentos, pero no incluyen la búsqueda de videos en la red cubana.

1.3.3. Resultados del estudio de los motores de búsqueda

El estudio realizado sobre los motores de búsqueda, tanto en el ámbito nacional como internacional, arrojó los siguientes resultados:

1. Los buscadores estudiados no pueden ser utilizados para dar solución al problema planteado. Esto se debe, a que los motores de búsqueda nacionales analizados no cuentan con un mecanismo que permita la búsqueda de videos y en el caso de los internacionales, su código fuente no puede ser utilizado debido a que son privativos.

2. Los motores de búsqueda internacionales estudiados, presentan filtros que facilitan la búsqueda de videos a los usuarios. Sin embargo, mediante la búsqueda básica es poco probable obtener videos publicados en la red cubana, ya que dan prioridad (mejor posicionamiento) a sitios web o redes sociales como YouTube, MySpace, entre otros.

3. Debido a que estos sistemas no le dan solución al problema planteado en esta investigación, se decidió desarrollar un subsistema de búsqueda de videos para el buscador cubano Orión, como propuesta de solución al problema planteado.

4. El análisis de los filtros presentes en los sistemas internacionales, permitió definir los filtros de búsqueda a utilizar en el subsistema propuesto; como es el caso de la fecha de rastreo, la calidad, la duración, la proporción y tamaño en disco[14] de los videos.

1.4. Lenguajes, tecnologías y herramientas

Para desarrollar el sistema informático que se propone, se hace necesario investigar sobre los lenguajes, tecnologías y herramientas a utilizar. A continuación se procede con el estudio y selección de las mismas.

[11] Accesible en: http://k2.uo.edu.cu.
[12] Accesible en: http://lupa.upr.edu.cu.
[13] Accesible en: http://orion.mes.edu.cu.
[14] Se refiere al espacio en memoria que ocupa un fichero.

1.4.1. Rastreadores

Un rastreador, según algunos autores, es el sistema que se encarga de recorrer la Web siguiendo los enlaces presentes en las páginas o documentos. Este recorrido puede ser realizado en profundidad o a lo ancho[15] y generalmente este tipo de herramienta suele estar en una computadora dispuesta para ello. Cada uno de los documentos encontrados en la Web son analizados para llevarlos a un formato común y posteriormente estos documentos son almacenados en alguna estructura de datos en forma de índices para su rápido acceso [5]. Algunos de los rastreadores de código abierto más conocidos son: Heritrix, Nutch y WIRE [20]. Seguidamente se presenta un estudio sobre dichos sistemas.

Heritrix 1.8

Heritrix es el rastreador de código abierto y extensible a gran escala de Internet Archive[16] desarrollado en el lenguaje Java. Esta herramienta, pretende recopilar y preservar los documentos digitales de la cultura en beneficio de los investigadores y las generaciones futuras. Está diseñado para respetar las directivas de exclusión de *robots*[17] [21]. Este proyecto, proporciona una lista abierta de correo para promover el intercambio de información en sus desarrolladores y los usuarios interesados [22].

Nutch 1.5

Nutch es un programa distribuido bajo la licencia Apache[18] desarrollado con el lenguaje de programación Java. La arquitectura de Nutch es muy flexible, permitiendo realizarle mejoras por parte de los usuarios a través de *plugins*[19]. Es independiente del servidor de indexación y permite la recopilación de muchos tipos de documentos utilizando componentes implementados por separado [5]. Los parámetros utilizados por Nutch para el rastreo y la indexación pueden ser configurables mediante ficheros escritos en XML (Ver sección 1.4.3).

WIRE 0.22

WIRE es una herramienta para la recuperación de información, diseñada para ser usada en la Web. Es un sistema que puede trabajar con grandes volúmenes de documentos y ha sido probado con varios millones de ellos. Está implementado con el lenguaje de programación C++ para lograr un alto rendimiento y su código

[15] Se refiere al algoritmo utilizado para recorrer la Web.

[16] Según su página oficial (accesible en: https://archive.org/index.php), es una biblioteca sin fines lucrativos de millones de libros gratuitos, películas, software, música y mucho más.

[17] Se refiere a las directivas utilizadas para evitar que los sistemas que analizan los sitios web agreguen información innecesaria a los resultados de búsqueda. Estas directivas son especificadas en un fichero llamado *robot.txt*.

[18] Contiene los términos y condiciones para el uso, reproducción y distribución definidos por Apache Software Foundation; la cual, es una organización no lucrativa que provee software y servicios para el bien público.

[19] Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

es libre bajo la licencia GPL[20]. Por otra parte, todos los parámetros para el rastreo y la indexación pueden ser configurables mediante ficheros escritos en XML (Ver sección 1.4.3). Incluye varias herramientas para el análisis, extracción de estadísticas y generación de reportes [23].

Selección del rastreador

A partir de las características de los rastreadores expuestas anteriormente, se concluye que para un adecuado rastreo de la red y una correcta recuperación de información, se utilizará Nutch 1.5. Se tuvo en cuenta para su selección, la flexibilidad con la que cuenta su arquitectura y que de los sistemas estudiados es el que más se aproxima a la búsqueda semántica[21], debido a uno de sus *plugins* para ontologías[22]. Por otra parte, cuenta con una comunidad de desarrollo que ha trabajado para lograr una solución extensible, que favorezca al desarrollo del propio sistema.

Otro elemento importante a tener en cuenta para la selección de dicho rastreador, es que es utilizado en el motor de búsqueda Orión, facilitando la integración con el mismo.

1.4.2. Indexadores

En la actualidad existen múltiples herramientas clasificadas como indexadores de documentos, las que técnicamente extraen una representación interna de los mismos y la almacenan en forma de índice en una base de datos. Existen varias técnicas para extraer la información de los documentos, variando según su complejidad e incluso según la cantidad de elementos que utilicen para crear el índice. A continuación se hace un análisis de algunos de estos sistemas.

Elasticsearch 1.4.4

Elasticsearch es un motor de búsqueda de código abierto escrito en Java. También es un indexador de documentos donde cada campo indexado puede ser utilizado para realizar búsquedas. Incluye la posibilidad de realizar análisis en tiempo real. Además, puede ser escalado a cientos de servidores y grandes cantidades de datos estructurados y no estructurados. Desde cualquier aplicación, incluso desde la línea de comandos se pueden acceder a los datos indexados en este sistema mediante una API[23] de comunicación [24].

Swish-e 2.4.7

[20] Acrónimo de GNU General Public Licence. Es una licencia que brinda a los usuarios la posibilidad de usar, compartir y modificar el software.

[21] Proceso utilizado para mejorar la búsqueda en Internet y encontrar los resultados más relevantes en relación a la demanda del usuario.

[22] Concepto empleado en la inteligencia artificial y la representación del conocimiento para facilitar la comunicación y el intercambio de información entre diferentes sistemas.

[23] Acrónimo de *Application Programming Interface*. Una API es un conjunto de instrucciones, protocolos y herramientas para la construcción de aplicaciones. Además, hace posible que los programas interactúen con otros y compartan datos.

Es un sistema de código abierto, rápido, flexible y libre para indexar colecciones de páginas web u otros archivos. Swish-e es ideal para las colecciones de un millón de documentos o más pequeños. (...) Este sistema puede indexar texto plano, correo electrónico, documentos, y casi cualquier archivo que se pueda convertir a texto XML o HTML[24]. Incluso, se pueden utilizar filtros para indexar otros tipos de documentos como PDF, gzip o PostScript.

Incluye un rastreador web para indexar documentos remotos, pero puede utilizar un programa externo que los suministre, tal como un sistema para el rastreo de servidores web o un programa para leer registros en una base de datos relacional.

Para cada búsqueda realizada se pueden obtener resúmenes de los documentos indexados y los resultados obtenidos se pueden ordenar por relevancia o por alguna otra propiedad, tanto en orden ascendente como descendente. Swish-e, también se utiliza regularmente para complementar las bases de datos (...) para una muy rápida búsqueda de texto completo. Utiliza expresiones regulares para seleccionar o excluir los documentos a indexar [25].

Solr 4.10.3

Es una plataforma de búsqueda rápida de código abierto del proyecto Apache Lucene[25]. Entre sus principales características se incluyen: una poderosa búsqueda de texto completo, (...), indexación en tiempo real, *clustering*[26] dinámico, integración de bases de datos, buena manipulación de documentos y la búsqueda geoespacial[27]. Solr es altamente fiable, escalable y tolerante a fallos, proporcionando indexación distribuida, replicación y consulta con equilibrio de carga, conmutación por error y recuperación automatizada, configuración centralizada, entre otras características.

Solr está escrito en Java y se ejecuta como un servidor de búsqueda de texto completo independiente dentro de un contenedor de *servlets*[28]. (...) La configuración externa de este sistema permite que se adapte a casi cualquier tipo de aplicación sin codificación en Java, y cuenta con una extensa arquitectura de *plugins* cuando se requiere una personalización más avanzada [26].

Selección del indexador

A partir de las características de los indexadores expuestas anteriormente se concluye que para facilitar y acelerar la búsqueda de la información se utilizará Solr 4.10.3. Se tuvo en cuenta, que es una herramienta escalable a varios servidores para búsquedas distribuidas, mejorando la rapidez de respuesta de las bús-

[24] Los lenguajes XML y HTML son explicados en la sección 1.4.3.

[25] Accesible en: `http://lucene.apache.org/`.

[26] Procedimiento de agrupación basado en un criterio, que por lo general es distancia o similitud.

[27] Es el estudio de la superficie terrestre a través de la informática, mediante la captura, tratamiento, análisis, interpretación, difusión y almacenamiento de información geográfica.

[28] Clases del lenguaje de programación Java utilizadas comúnmente para extender las aplicaciones alojadas por servidores web.

quedas. Por otra parte, está implementado con el lenguaje Java, el cual se puede utilizar para extender sus funcionalidades.

Otro elemento importante considerado en dicha selección, es que este sistema es utilizado en el motor de búsqueda Orión, facilitando de este modo su integración con dicho buscador.

1.4.3. Lenguajes de programación

Los lenguajes de programación pueden ser utilizados para crear programas. Están formados por un conjunto de reglas sintácticas y semánticas, que les permiten definir su estructura y el significado de sus expresiones. Permiten además, especificar los datos que se deben procesar, almacenar o transmitir, y las acciones que se deben realizar bajo determinadas circunstancias.

El estudio de los lenguajes a utilizar en el desarrollo de la propuesta de solución se dividirá en dos grupos: el primero dedicado a los lenguajes de programación del lado del servidor y un segundo grupo enfocado al resto de los lenguajes a utilizar. A continuación se exponen las características de cada uno de los lenguajes estudiados.

Lenguajes de programación del lado del servidor

Los lenguajes de programación del lado del servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor e independientes del cliente, en sistemas con una arquitectura cliente - servidor. Son utilizados en el procesamiento de las peticiones de los usuarios para generar páginas web dinámicamente como respuesta.

Perl 5.20.2

Es un lenguaje de programación con más de 26 años de desarrollo. Se ejecuta en más de 100 plataformas de portátiles y es adecuado para proyectos de desarrollo a gran escala y la creación rápida de prototipos. Puede ser embebido en los servidores de Internet para acelerar el procesamiento, por ejemplo el servidor web Apache (Ver sección 1.4.6) incluye un módulo para incrustar un intérprete del lenguaje. Perl incluye potentes herramientas para el procesamiento de texto que lo hacen ideal para trabajar con HTML, XML y otros lenguajes. Soporta bases de datos como Oracle, Sybase, PostgreSQL, MySQL y muchas otras [27].

PHP[29] 5.5.9

Es un lenguaje de programación (...) usado para crear aplicaciones para servidores o generar contenido dinámico para sitios web. Permite la creación de aplicaciones web muy robustas, al posibilitar la conexión a diferentes tipos de servidores de base de datos como: MySQL, PostgreSQL, Oracle y otros. Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos y puede interactuar con varios de

[29] Acrónimo recursivo de *PHP Hypertext Preprocessor*. Pasó de significar *"Personal Home Page"* a *"PHP Hypertext Preprocessor"*.

los servidores web más populares. Los principales usos de PHP son: la programación de páginas web dinámicas, la programación en consola y la creación de aplicaciones gráficas independientes del navegador.

PHP es un lenguaje multiplataforma[30]. (...) Posee gran capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones). Puede leer y manipular datos desde diversas fuentes y permite las técnicas de programación orientada a objetos. Se presenta como una alternativa de fácil acceso, debido a que es libre [28].

Python 2.7.6

Es un lenguaje de programación (...) con una sintaxis muy limpia y que ayuda a obtener un código legible. Se trata de un lenguaje interpretado, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. (...) Python también permite la programación imperativa, funcional y orientada a aspectos. (...) Sin embargo, no es adecuado para la programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico [29].

Selección del lenguaje del lado del servidor a utilizar

Teniendo en cuenta que PHP 5.5.9 es un lenguaje con la capacidad de expandir su potencial, que tiene manejo de excepciones y ha tenido una gran aceptación para el desarrollo de aplicaciones web, incluyendo el resto de las características expuestas anteriormente, se decide seleccionarlo como lenguaje del lado del servidor a utilizar.

Otros lenguajes

Para el desarrollo de la aplicación que dará solución al problema inicialmente planteado en la presente investigación, no solo es necesario tener un lenguaje de programación del lado del servidor, sino que también lo es, tener lenguajes del lado del cliente en una arquitectura cliente - servidor. Dichos lenguajes son interpretados por el propio navegador, mostrando al usuario la información proveniente del servidor con una estructura y jerarquía, facilitando una mayor comprensión de dicha información.

A continuación se hace alusión a los lenguajes utilizados, no solo para ser interpretados por el navegador de los clientes, sino también en el modelado de los artefactos del sistema y en el intercambio de información entre las aplicaciones utilizadas.

Java 1.7.0_65

Este lenguaje funciona con las principales plataformas de hardware y sistemas operativos, siendo uno de los entornos de programación más rápidos que incluye optimizaciones integradas para entornos multiproceso.

[30] Se refiere a que es un lenguaje que puede ser interpretado en múltiples plataformas informáticas, es decir en varios Sistemas Operativos.

Java alcanza un alto rendimiento nativo[31] y proporciona portabilidad en una amplia gama de procesadores y sistemas operativos integrados. Además, ofrece un entorno de aplicaciones avanzado, con un alto nivel de seguridad que es idóneo para las aplicaciones de red.

El modelo de Java para la gestión de la memoria, los procesos múltiples y la gestión de excepciones lo convierte en un lenguaje eficaz para los desarrolladores nuevos y para los más experimentados. Es una de las plataformas de aplicaciones más populares que existen y proporciona un interesante ecosistema de desarrolladores impulsado por herramientas eficaces, libros, bibliotecas, muestras de código, entro otros aspectos [30].

Lenguaje de Marcado de Hipertexto 5

El lenguaje de Marcado de Hipertexto en su versión 5 (HTML5 por sus siglas en inglés) es un lenguaje que permite especificar instrucciones especiales para indicarle al navegador cómo desplegar el contenido de los documentos, en los que se puede incluir: texto, imágenes y otros medios soportados. [31].

Lenguaje de Marcado Extensible 1.0

El lenguaje de Marcado Extensible (XML por sus siglas en inglés) es un simple formato basado en texto para representar información estructurada: documentos, datos, configuraciones, libros, transacciones, facturas y mucho más. (...) Es un lenguaje muy abundante y copioso de palabras. Por ejemplo, cada etiqueta[32] de cierre debe ser suministrada, lo cual le permite a la computadora capturar errores comunes como anidamiento incorrecto. Por otra parte, la legibilidad de XML y la presencia de nombres de elementos y atributos, ayudan a que las personas que buscan en un documento XML, a menudo pueden obtener una ventaja en la comprensión del formato y en la detección de errores [32].

Hojas de Estilo en Cascada 3

Hojas de Estilo en Cascada en la versión 3 (CSS3 por sus siglas en inglés) es un lenguaje creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. (...) Además, permite visualizar un mismo documento en infinidad de dispositivos diferentes. (...) Es utilizado para definir el aspecto de cada elemento de un documento HTML o XHTML, como: color, tamaño, tipo de letra del texto, separación horizontal y vertical entre los elementos, posición de cada elemento dentro de la página, etc. [33].

[31] Se refiere al rendimiento del código generado para un procesador específico. A este tipo de código se le conoce como código nativo o lenguaje máquina.

[32] Hace alusión a los fragmentos de texto que permiten la definición de las distintas instrucciones en lenguajes basados en XML.

Lenguaje Unificado de Modelado 2.4.1

Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es el lenguaje estándar especificado por el Object Management Group (OMG por sus siglas en inglés)[33] para visualizar, especificar, construir y documentar los artefactos de un sistema y además, sirve para el modelado del negocio y sistemas de software.

Ofrece un estándar para describir los modelos, incluyendo aspectos conceptuales como procesos de negocio, funciones del sistema, expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

UML cuenta con un conjunto de notaciones y diagramas para modelar sistemas orientados a objetos y describe la semántica esencial de lo que significan estos diagramas y símbolos. Puede utilizarse en las diferentes etapas del ciclo de vida del desarrollo de sistemas y es independiente del proceso o metodología de desarrollo y del lenguaje de implementación. Con UML es posible extender la funcionalidad de la notación gráfica mediante estereotipos y proveer una base formal para los diagramas [34].

1.4.4. Marcos de trabajo para PHP

Un marco de trabajo (comúnmente llamado *framework*), según algunos autores, *"hace alusión a una estructura de software compuesta por componentes personalizables e intercambiables para el desarrollo de una aplicación. Además está asociado a un determinado tipo de aplicaciones, lo que implica que su alcance esté acotado; o sea que está ligado a un dominio concreto. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se puede añadir las últimas piezas para construir una aplicación concreta"* [35].

A continuación se realizará un estudio de algunos marcos de trabajo para PHP y de este modo realizar la selección que se considere más apropiada para el desarrollo del sistema propuesto. Estos marcos de trabajo se encuentran entre los más utilizados según [36].

Zend Framework 2 (ZF2) 2.4.1

ZF2 es un marco de trabajo de código abierto para el desarrollo de aplicaciones y servicios web utilizando PHP 5.3 o superior. Usa en su totalidad código orientado a objetos y utiliza la mayor parte de las nuevas características de la versión del lenguaje. (...)

La estructura de componentes de ZF2 es única; cada componente está diseñado con pocas dependencias de otros componentes. (...) Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar cualquier componente que deseen. (...) Mientras que pueden ser utilizadas por separado, los componentes de ZF2

[33] Consorcio dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos.

en la librería estándar forman un potente y extensible marco de trabajo de aplicaciones web cuando son combinados.

También, ofrece una implementación robusta y de alto rendimiento del patrón arquitectónico Modelo Vista Controlador (MVC, por sus siglas en inglés)[34], una abstracción de la base de datos que es simple de utilizar, un componente de formularios que implementa una forma de representación utilizando HTML5, validación y filtrado de modo que los desarrolladores puedan consolidar todas estas operaciones. (...)

ZF2 no podría entregar y brindar soporte a todas estas características sin la ayuda de la enérgica y entusiasmada comunidad de Zend Framework 2. Miembros de la comunidad, incluidos los contribuyentes, se ponen a disposición en listas de correo, canales de IRC[35] y foros [38].

CodeIgniter 2.2.0

CodeIgniter es un marco de trabajo de código abierto de aplicaciones web para el lenguaje PHP. Tiene numerosas características que lo hacen destacar entre el resto de sus semejantes, ya que a diferencia de la mayoría es muy minucioso y completo. (...)

Es compatible con PHP4 y PHP5, por lo que es posible ejecutarlo en la mayoría de los servidores web. Implementa el patrón arquitectónico MVC y hace extensivo el uso del patrón de diseño Singleton; el cual, indica la forma de cargar las clases, de modo que si ellas son llamadas varias veces, la misma instancia es retornada. Esto es muy útil para conexiones a las bases de datos, ya que solo se requeriría una conexión cada vez que la clase es usada.

CodeIgniter también tiene una implementación del patrón Active Record. Esto lo hace fácil para escribir complejas consultas SQL y hace la aplicación más legible. Active Record también permite fácilmente permutar y cambiar controladores de las bases de datos. (...) También trae un número muy útil de librerías y otro conjunto de funciones que ayudan en la construcción de aplicaciones [39].

Symfony 2.3

Es un completo marco de trabajo diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la del servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (...) Por otra parte, es compatible con la mayoría de gestores de bases de datos [40].

[34] MVC es uno de los patrones de arquitectura de software más conocidos, y provee la infraestructura para sistemas interactivos [37] (Ver sección 2.7.3).

[35] Según la IETF, el protocolo IRC (Internet Relay Chat) ha sido diseñado para su uso en conferencias basadas en texto. Puede ser visto como un sistema de teleconferencias.

Symfony2 ha sido ideado para aprovechar al máximo todas las nuevas características de PHP 5.3 y por eso es uno de los marcos de trabajo PHP con mejor rendimiento. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no se ajustan a un proyecto. Además, es el marco de trabajo que más ideas incorpora del resto de sus semejantes, incluso de aquellos que no están programados con PHP [33].

Selección del marco de trabajo

Teniendo en cuenta las características y beneficios expuestos anteriormente, se decide seleccionar para el desarrollo de la aplicación web del sistema propuesto el marco de trabajo Symfony 2.3. Además, se tuvo en cuenta para dicha selección que es rápido, flexible, adaptable, altamente funcional, favorece al rendimiento de las aplicaciones y es el utilizado para la implementación de la aplicación web del buscador cubano Orión, facilitando esto último su integración.

1.4.5. Metodología de desarrollo

Una metodología de desarrollo de software es un conjunto de procedimientos utilizados para alcanzar un determinado objetivo, pero enmarcado en la ingeniería de software. Además, permite estructurar, planificar y controlar el proceso de desarrollo de un software determinado.

Para el desarrollo de la propuesta de solución se decide utilizar la metodología OpenUp; la cual, está dirigida a la gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental. Es apropiada para proyectos pequeños y de bajos recursos. Posee varias iteraciones dentro del ciclo de vida del proyecto, que no superan las pocas semanas de duración, en dependencia de los acuerdos que se toman en el equipo de trabajo. Se debe tener en cuenta que cada iteración concluye obligatoriamente con una muestra concreta del producto, que necesariamente tiene que ser “demostrativa” o “explotable”, ya que es la forma que tiene la metodología de desarrollo de demostrarle el valor agregado al cliente [34].

Por otra parte, permite detectar errores tempranos a través de un ciclo iterativo y evita la elaboración de documentación, diagramas e iteraciones innecesarias, requeridos en metodologías con enfoque tradicional. Además, es la metodología que guía el proceso de desarrollo de software en CIDI.

1.4.6. Herramientas

Servidor web Apache 2.4.7

Es un servidor web libre y multiplataforma reconocido en muchos ámbitos empresariales y tecnológicos. Es altamente configurable y de diseño modular, por lo que es muy sencillo ampliar sus capacidades. Tiene una

alta configuración en la creación y gestión de *logs*[36], garantizando un mayor control sobre lo que sucede en el servidor.

Apache permite personalizar la respuesta ante los posibles errores que puedan suceder en el servidor. Es posible configurarlo para que ejecute un determinado *script*[37] cuando ocurra un error en concreto [41].

Entorno de Desarrollo Integrado NetBeans 8.0

Es un proyecto de código abierto dedicado a proveer un sólido desarrollo de software, dirigido fundamentalmente a las necesidades de los desarrolladores y los usuarios; dotándolos de una herramienta para el desarrollo rápido, eficiente y fácil de productos de software [42].

NetBeans es una herramienta modular de desarrollo para un amplio rango de tecnologías para el desarrollo de aplicaciones. Incluye un avanzado editor para varios lenguaje, editor de perfiles y un detector de errores, además de herramientas para el control de versiones y el desarrollo colaborativo. Proporciona aplicaciones de muestra en forma de plantillas de proyectos para todas las tecnologías que soporta, (...) así como una amplia variedad de *plugins* para todos los tipos de desarrollo [43].

Visual Paradigm para UML 8.0

Es una herramienta que soporta el ciclo de vida completo en el desarrollo de software: análisis y diseño orientado a objetos, construcción, prueba y despliegue. Permite dibujar todo tipo de diagrama de clases, generar código fuente a partir de diagramas y generar documentación [44].

Soporta el intercambio de diagramas UML y modelos con otras herramientas, así como la importación y exportación a formatos XMI[38], XML y archivos Excel[39]. Permite importar proyectos de Rational Rose[40] y la integración con Microsoft Office Visio[41].

Permite la captura de requisitos mediante diagramas de requisitos, modelado de caso de uso y análisis textual. Posee un entorno para la especificación de detalles de casos de uso, incluyendo la especificación del modelo general y las descripciones de los casos de uso. Además, posibilita la generación de código e ingeniería inversa para diversos lenguajes, entre los que se encuentra PHP y Java [46].

[36] Se refiere a un fichero con una lista de las acciones que han ocurrido en el servidor.
[37] Es una lista de comandos que pueden ser ejecutados sin la interacción de un usuario.
[38] Es un estándar de gran alcance para la representación de objetos en XML [45].
[39] Es una aplicación distribuida por Microsoft Office (suite ofimática que contiene aplicaciones de escritorio, servidores y servicios para los sistemas operativos Microsoft Windows y Mac OS X) para hojas de cálculo. Es utilizado normalmente en tareas financieras y contables.
[40] Actualmente es conocida como una familia de software de International Business Machines Corp (IBM, empresa multinacional estadounidense de tecnología y consultoría) para el despliegue, diseño, construcción, pruebas y administración de proyectos en el proceso desarrollo de software.
[41] Es un software de dibujo vectorial que permite realizar diagramas de oficinas, de bases de datos, de flujo de programas, UML, entre otros.

Mediainfo 0.7.67

Según el manual de esta herramienta[42], es una utilidad en línea de comandos[43] para mostrar información acerca de los ficheros multimedia y suministrar información técnica. Posibilita la obtención de información atendiendo a metadatos generales, referentes al video, al audio o al texto. Soporta numerosos formatos y codificadores de video y audio, así como variados formatos de subtítulos. Permite obtener y almacenar la información relacionada con los metadatos en diferentes formatos como los son: HTML, XML o texto plano.

Avconv 9.16

A partir de la descripción presente en el manual de la herramienta[44] se puede decir que es un convertidor de audio y video. Puede convertir entre frecuencias de muestreo arbitrarias y cambiar el tamaño de video durante su reproducción.

Acunetix Web Vulnerability Scanner 9.5

Es una herramienta para el análisis de la seguridad en aplicaciones web. Incluye un analizador automático para pruebas de seguridad para AJAX[45], desarrollado con el lenguaje de programación JavaScript. Además, permite realizar pruebas para ataques de inyección de código, secuencia de comandos en sitios cruzados (XSS por sus siglas en inglés) y falsificación de peticiones. Posee un componente que facilita la realización de pruebas a formularios y a áreas protegidas por contraseña. Permite realizar varias peticiones simultáneamente, siendo capaz de explorar cientos de páginas sin interrupciones [47].

Apache Jmeter 2.8.20130705

Es una aplicación de código abierto diseñada para medir el rendimiento de las aplicaciones a partir de comportamientos funcionales. Puede ser utilizado para probar recursos estáticos y dinámicos, servicios web, simular una carga pesada en un servidor, grupo de servidores, en la red y para hacer un análisis gráfico de rendimiento [48].

1.5. Conclusiones del capítulo

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

[42] Una vez instalada la herramienta se puede acceder a su manual mediante el comando: ***man mediainfo***.

[43] Entiéndase por interfaz de línea de comandos (CLI por sus siglas en inglés) como un método que permite a los usuarios dar instrucciones a algún programa informático por medio de una línea de texto simple.

[44] Una vez instalada la herramienta se puede acceder a su manual mediante el comando: ***man avconv***.

[45] Es una técnica de desarrollo web que permite realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

1. Las relaciones existentes entre los principales conceptos asociados al dominio de la presente investigación, permitieron una mayor comprensión de la propuesta de solución.
2. Las deficiencias encontradas en los motores de búsqueda nacionales, el limitado acceso a los internacionales y la manipulación de los resultados presente en ellos, hacen necesario la creación de un subsistema que permita la búsqueda de videos en la red cubana.
3. El análisis de los diferentes criterios para filtrar los resultados en los motores de búsqueda internacionales, permitió identificar los diferentes filtros a incorporar en el subsistema a desarrollar.
4. El estudio realizado de las principales herramientas, tecnologías y lenguajes de programación a utilizar, permitió realizar la selección de los más adecuados para dar solución al problema planteado.

Capítulo 2

Análisis y diseño del subsistema de búsqueda de videos para el buscador cubano Orión

Para el desarrollo de un software se debe partir de la comprensión de los objetivos a alcanzar y las funcionalidades con las que debe contar, así como las necesidades a las que dará respuesta una vez concluido el mismo. Para lograr un mayor entendimiento del subsistema a desarrollar, en este capítulo se realiza el análisis y diseño del software haciendo uso de los artefactos que propone la metodología OpenUp y las políticas de calidad de CIDI, como lo son el modelo de dominio, la representación de los principales procesos mediante casos de uso, diagramas de clases del diseño, de colaboración y de despliegue. En aras de satisfacer el objetivo de la presente investigación se describe la propuesta de solución, así como los requisitos que debe cumplir la misma.

2.1. Modelo de dominio

Un modelo de dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, que contiene conceptos propios de la realidad física. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno del sistema. (...) El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema, por lo cual puede ser tomado como el punto de partida para su diseño [34].

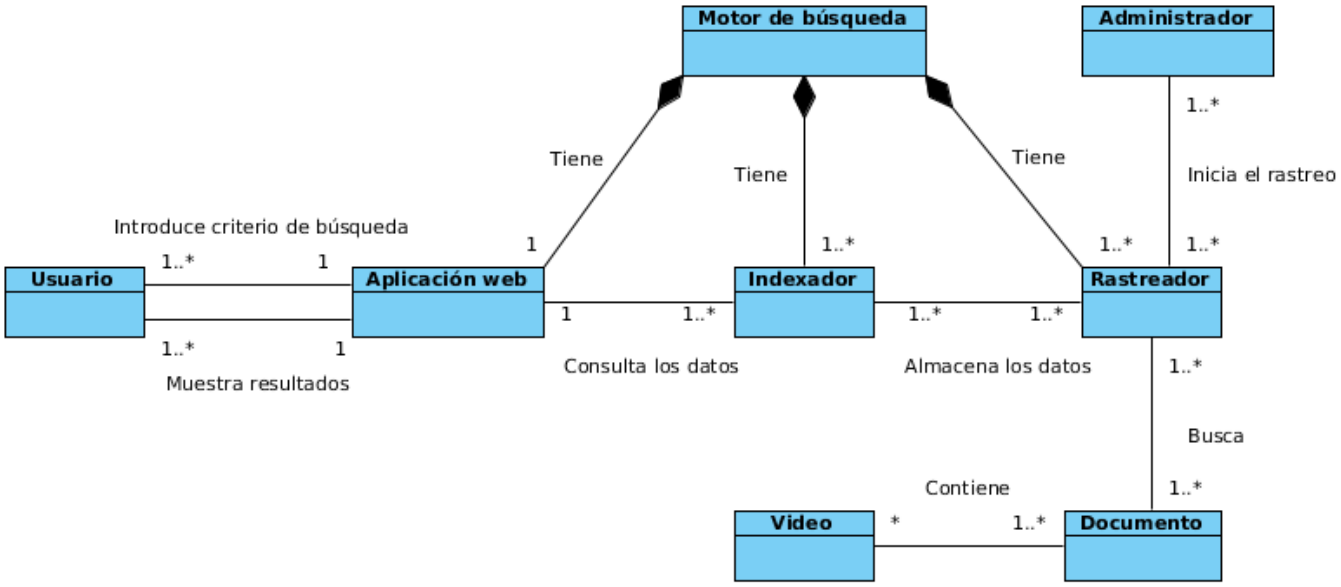

Imagen 2.1: Diagrama del modelo de dominio

2.1.1. Descripción de las clases del dominio

Usuario: Persona que introduce un criterio de búsqueda en la aplicación web.

Administrador: Persona que configura los rastreadores e inicia el rastreo de los videos en la red.

Aplicación web: Componente del motor de búsqueda encargado de recibir las peticiones de los usuarios, consultar los datos en los indexadores y mostrarle al usuario los resultados de su búsqueda.

Indexador: Componente del motor de búsqueda que almacena los resultados del rastreo en forma de índice para una mejor selección de los datos solicitados por la aplicación web. Pueden existir varios componentes de indexación.

Rastreador: Componente del motor de búsqueda que accede a la información pública en la red y procesa los videos encontrados. Pueden existir varias instancias de este componente, es decir, varios rastreadores del mismo tipo escaneando la red.

Motor de búsqueda: Sistema de recuperación de información compuesto por una aplicación web, varios indexadores y rastreadores.

Documento: Tipo de contenido electrónico que puede o no contener videos.

Video: Tipo de contenido del cual se almacenarán sus datos para mostrarlos al usuario.

En el modelo de dominio mostrado anteriormente (Ver imagen 2.1) se parte de las acciones realizadas por un usuario o un administrador. Un usuario puede introducir uno o varios criterios de búsqueda en la aplicación web, la cual consulta los datos en los indexadores existentes, atendiendo a lo que se desea buscar. Cada indexador le devuelve a la aplicación web todos los datos que coinciden con la búsqueda y esta a su vez, se los muestra al usuario. Por otro lado, un administrador inicia el rastreo de los documentos presentes en la red en uno o varios rastreadores. De estos documentos, pueden ser analizadas las imágenes y el texto contenido en ellos. Sin embargo, pueden contener referencia a uno o varios videos, los cuales no pueden ser analizados por el rastreador, ya que este no cuenta con las funcionalidades necesarias para ello. Una vez obtenida la información de los recursos rastreados, es enviada a los indexadores disponibles para su persistencia.

2.2. Descripción del sistema propuesto

El subsistema que se propone desarrollar, pretende disminuir el tiempo de las búsquedas de los videos publicados en la red cubana, realizadas con el buscador cubano Orión. Este sistema debe permitir a cualquier usuario (sin previa autenticación) realizar búsquedas avanzadas, donde se podrán filtrar los contenidos atendiendo a los siguientes criterios:

Calidad de los videos: El sistema debe permitir buscar videos en HD, para ello solamente mostraría aquellos videos con más de 720 píxeles de alto. Esto se debe, a que un video HD es aquel con 720, 1080 o 2160 líneas de escaneo progresivo[1] (720p, 1080p o 2160p respectivamente) o 1080 de entrelazado[2] (1080i) [49].

Tamaño de los videos: Los usuarios podrán filtrar los resultados de sus búsquedas para obtener videos pequeños (con un tamaño menor de 256 MB), medianos (mayores de 256 MB y menores de 512 MB) y grandes (mayores de 512 MB).

Proporción: Este filtro permitirá a los usuarios buscar videos atendiendo a la relación de aspecto (RA)[3] que posean. Para ello, el sistema deberá permitir buscar videos que tengan 4:3, 16:9 y 16:10[4] de RA.

Formato: El sistema deberá permitir a los usuarios buscar videos atendiendo a su formato (por ejemplo: avi o webm). Para ello, el administrador podrá seleccionar los formatos que desea que el sistema rastree, indexe y muestre a los usuarios.

[1] El escaneo progresivo es un método utilizado para componer cada uno de los fotogramas de un video proyectando secuencialmente todas las líneas horizontales de la imagen. Este método permite obtener una mejor visualización que el escaneo entrelazado.

[2] El escaneo entrelazado es un método para la representación de videos donde cada fotograma es proyectado por la mitad de las líneas (pares o impares alternamente).

[3] La relación de aspecto (*aspect ratio* en inglés) es la proporción existente entre el ancho y el alto de una imagen [49].

[4] 4:3, 16:9 y 16:10 son las relaciones de aspecto más comunes [49].

Audio: Con este filtro los usuarios podrán buscar videos que tengan audio mono[5] o estéreo[6].

Duración: Haciendo uso de este filtro los usuarios podrán buscar videos cortos (menos de 5 minutos), con una duración media (entre 5 y 20 minutos) y largos (más de 20 minutos).

Fecha del rastreo: Este filtro permitirá a los usuarios buscar videos que hayan sido encontrados por el componente de rastreo, un día, una semana, un mes o año antes del momento en que se realiza la búsqueda.

El sistema debe incluir además, la posibilidad de buscar en el título o en la URL, tanto del video como de la página web que lo referencia.

Los usuarios podrán realizar también búsquedas simples, desde las cuales se podrán filtrar los contenidos (si lo desea) mediante operadores definidos para ello. El subsistema debe contar con una ayuda que tiene como objetivo orientar al usuario en la realización de las búsquedas y la obtención de información.

Por otra parte, un administrador podrá ser capaz de configurar el rastreador mediante ficheros de configuración e iniciar el rastreo de la red cuando lo considere necesario.

2.2.1. Arquitectura del sistema propuesto

Atendiendo a la arquitectura genérica de los motores de búsqueda descrita en la sección 1.2 y las particularidades del sistema propuesto, se sugiere la arquitectura que se muestra en la siguiente imagen.

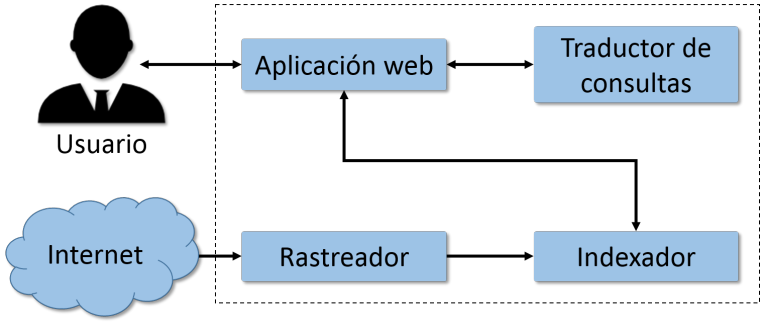

Imagen 2.2: Arquitectura del sistema propuesto

Como se muestra en la imagen anterior, el sistema que se propone debe tener un componente llamado **Rastreador** el cual, sería el encargado del proceso de rastreo. El **Indexador** incluye los componentes Indexador y Repositorio de la arquitectura genérica de los motores de búsqueda mostrada en la sección 1.2 y es el encargado del proceso de indexación. A su vez, **Aplicación web** incluye los componentes Servidor

[5]El sonido monoaural (abreviado frecuentemente como mono) es el sonido que está definido por un solo canal de audio.
[6]El sonido estereofónico o estéreo se refiere al grabado y reproducción en dos canales de audio.

web, Buscador y Generador de presentación de la arquitectura genérica y el componente **Traductor de consultas** es el equivalente a Generador de consultas.

Según esta arquitectura, el componente de rastreo sigue los enlaces presentes en Internet y obtiene información de los videos que visita, la cual es enviada a Indexador para ser almacenada. Por otra parte, el usuario envía a Aplicación web un criterio de búsqueda, esta a su vez, se comunica con Traductor de consultas y le solicita la consulta equivalente al criterio introducido por el usuario para enviarla a Indexador. Luego de esto, el Indexador devuelve los videos que cumplan con la consulta formulada para mostrarlos al usuario.

2.3. Especificación de los requisitos de software

Los requisitos o requerimientos de software son una descripción de las necesidades o deseos que satisface un producto. La meta primaria (...) es identificar y documentar lo que en realidad se necesita, en forma que claramente se le comunique al cliente y a los miembros del equipo de desarrollo [50]. El cliente del sistema propuesto, es el departamento de Soluciones Informáticas para Internet (SINI) del centro CIDI de la UCI. Los requisitos identificados para el subsistema de búsqueda de videos para el buscador cubano Orión se relacionan a continuación.

2.3.1. Requisitos funcionales

Los requisitos funcionales (RF) o funciones del sistema son lo que este tiene que hacer [51]. A continuación se muestran los requisitos funcionales de la propuesta de solución y la prioridad que poseen de acuerdo a su importancia en el sistema.

Número	Requisito funcional	Prioridad
RF1	Usar operadores especiales en la búsqueda simple de videos.	Baja
RF2	Filtrar contenidos por fecha de rastreo.	Alta
RF3	Filtrar contenidos por calidad del video.	Alta
RF4	Filtrar contenidos por duración del video.	Alta
RF5	Filtrar contenidos por el tamaño del video.	Alta
RF6	Filtrar contenidos por la proporción del video.	Alta
RF7	Filtrar contenidos por el formato del video.	Alta
RF8	Determinar la calidad de los videos.	Alta
RF9	Identificar el formato de los videos.	Alta
RF10	Identificar la dimensión de los videos.	Alta
RF11	Identificar el tamaño de los videos.	Alta
RF12	Identificar la proporción de los videos.	Alta

Número	Requisito funcional	Prioridad
RF13	Identificar la duración de los videos.	Alta
RF14	Obtener la miniatura de los videos.	Alta
RF15	Identificar la fecha de rastreo de los videos.	Alta
RF16	Identificar el título de la etiqueta video.	Alta
RF17	Identificar el nombre del recurso.	Alta
RF18	Identificar los videos publicados en la red.	Alta
RF19	Mostrar el título del video.	Alta
RF20	Mostrar el título de la página donde se encuentra el video.	Alta
RF21	Mostrar el tiempo de duración del video.	Alta
RF22	Mostrar la URL de la página donde se encuentra el video.	Alta
RF23	Mostrar una miniatura del video.	Alta
RF24	Visualizar el recurso en el navegador.	Alta
RF25	Mostrar el tamaño del video.	Alta
RF26	Mostrar página de ayuda al usuario.	Baja
RF27	Mostrar el tiempo en que el sistema demora en mostrar los resultados.	Baja

Tabla 2.1: Requisitos funcionales del sistema

2.3.2. Requisitos no funcionales

Los requisitos no funcionales (RNF) describen las características y limitaciones que debe tener el sistema para alcanzar el éxito [52]. A continuación se muestran los requisitos no funcionales de la propuesta de solución, agrupados fundamentalmente atendiendo a las categorías: usabilidad, confiabilidad, eficiencia, soporte, restricciones de diseño, documentación de usuarios en línea y ayuda del sistema, interfaz y seguridad.

Usabilidad

RnF 1. El sistema deberá ser utilizado por los usuarios que tengan acceso a la red nacional y deseen realizar una búsqueda de videos.

RnF 2. El subsistema de búsqueda de videos para el buscador cubano Orión será una aplicación web.

RnF 3. El subsistema de búsqueda de videos para el buscador cubano Orión facilitará y reducirá el tiempo de las búsquedas de los videos publicados en la red cubana realizadas con dicho motor de búsqueda.

RnF 4. Los servidores donde estarán desplegados el componente de rastreo, el de indexación, la aplicación web y el traductor de consultas deben tener como recursos mínimos de hardware: 4 GB de RAM y un microprocesador Core i3 con una velocidad de 3.30 GHz.

RnF 5. Los servidores destinados para el traductor de consultas y la aplicación web deben poseer un disco duro con una capacidad mínima de 80 GB.

RnF 6. Los servidores para el componente de rastreo y el de indexación deben contar con un disco duro que posea más de 500 GB de almacenamiento.

RnF 7. Los dispositivos de los usuarios deben contar con navegadores web que soporten HTML5 y CSS3.

RnF 8. Los servidores donde se instalarán cada uno de los componentes del sistema deben tener una distribución GNU/Linux como sistema operativo, recomendándose CentOS 7.

RnF 9. Se requiere la instalación de un servidor web (recomendándose, Apache en su versión 2.4) y PHP 5.5 o superior para el correcto funcionamiento de la aplicación web.

RnF 10. Se requiere la instalación de un servidor de *servlets* para el correcto funcionamiento del componente de indexación y el traductor de consultas, recomendándose Apache Tomcat 7.

RnF 11. Se requiere la instalación de la Máquina Virtual de Java (JVM por sus siglas en inglés) para el correcto funcionamiento del rastreador, el indexador y el traductor de consultas.

Confiabilidad

RnF 12. El sistema no debe gestionar ni requerir información de usuarios para su uso.

Eficiencia

RnF 13. El sistema debe permitir que los usuarios interactúen con él de manera concurrente.

RnF 14. El sistema debe ser capaz de responder 5000 peticiones en 5 segundos como máximo.

Soporte

RnF 15. El soporte del sistema se debe gestionar mediante el Centro de Soporte de la UCI.

Restricciones de diseño

RnF 16. En la elaboración del diseño de diagramas y artefactos se deberá utilizar Visual Paradigm para UML 8.0.

RnF 17. Para el componente de rastreo se deberá utilizar la herramienta Nutch.

RnF 18. Para el componente de indexación se deberá utilizar la herramienta Solr.

RnF 19. Como lenguaje de programación para la aplicación web se deberá utilizar PHP en su versión 5.5 o mayor.

RnF 20. Como lenguaje de programación para los *plugins* del componente de rastreo se deberá utilizar Java 1.7.0_65 o superior, así como para el traductor de consultas.

RnF 21. Para el desarrollo de la aplicación web se deberá utilizar Symfony2 como marco de trabajo.

Requisitos para la documentación de usuarios en línea y ayuda del sistema

RnF 22. El sistema deberá contar con una ayuda para los usuarios, la cual explicará detalladamente la utilización del sistema.

Interfaz

RnF 23. La interfaz gráfica de la aplicación debe cumplir con las pautas de diseño definidas por el cliente.

Seguridad

RnF 24. El sistema debe estar protegido contra ataques de suplantación de peticiones en sitios cruzados (CSRF por sus siglas en inglés), XSS e inyecciones de código.

2.4. Modelo de casos de uso del sistema

Un caso de uso (en lo adelante CU) cuenta una historia estilizada de la manera en que un usuario final (el cual desempeña uno de varios papeles posibles) interactúa con el sistema en un conjunto específico de circunstancias [53]. A continuación se muestra el modelo de casos de uso que describe las funcionalidades propuestas para el subsistema de búsqueda de videos del buscador cubano Orión.

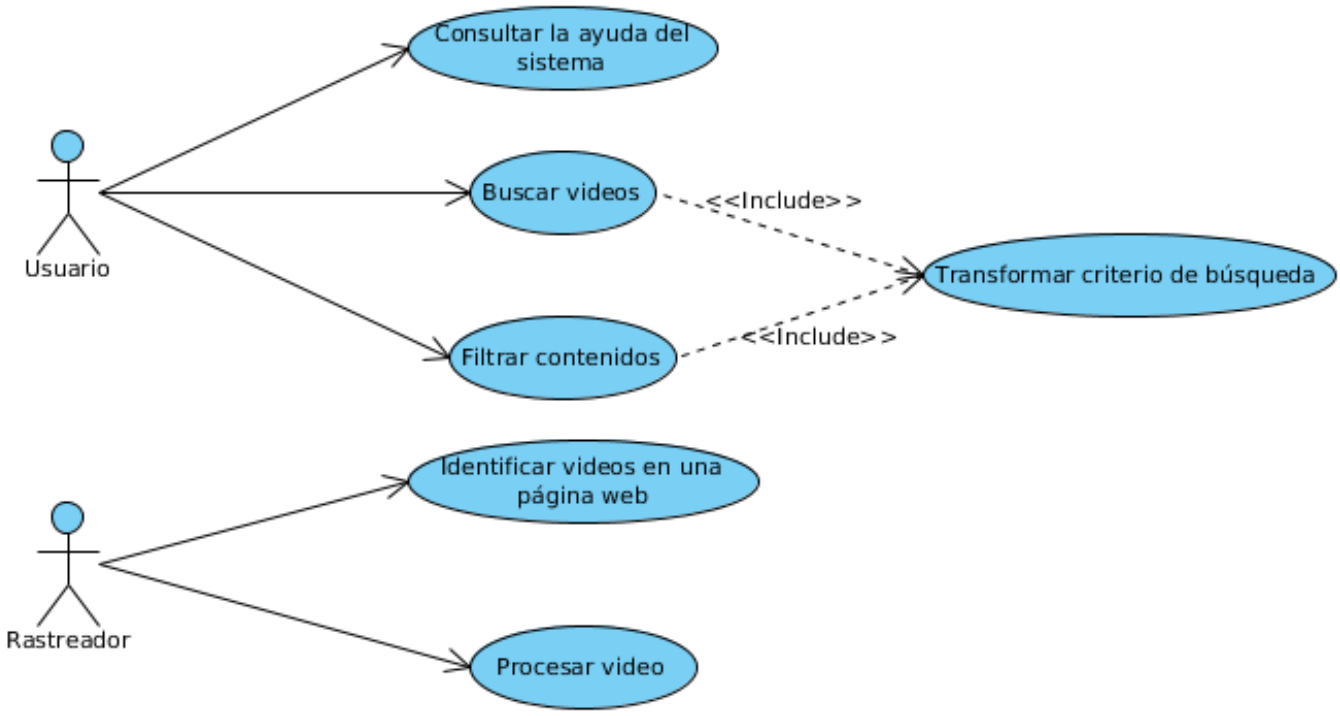

Imagen 2.3: Diagrama de casos de uso

A continuación se muestra la especificación de los casos de uso “Identificar videos en una página web” y “Procesar video”, los cuales serán los CU tratados en el resto del documento debido a la relevancia que tienen para el desarrollo del sistema propuesto.

2.4.1. CU Identificar videos en una página web

Objetivo	Identificar las URLs de los videos existentes en una página web.	
Actores	Rastreador.	
Resumen	El sistema identifica las URLs de los videos existentes en una página web.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	Un administrador ha configurado el sistema e iniciado el rastreo. El contenido de la página web ha sido obtenido y el árbol de su estructura ha sido creado.	
Postcondiciones	Los videos fueron identificados.	
Flujo de eventos		
Flujo básico Identificar videos en un documento.		
	Actor	**Sistema**

1.	Ejecuta el filtro que analiza el contenido.	
2.		Realiza un recorrido en profundidad[7] por el árbol mientras no se alcance el límite máximo de videos por página establecido en la configuración: • Si un nodo corresponde a una etiqueta A se dirige al flujo alterno 1. • Si un nodo corresponde a una etiqueta VIDEO se dirige al flujo alterno 2.
3.		Si se alcanza el límite máximo de videos por página se continua recorriendo el árbol en profundidad: • Si un nodo corresponde a una etiqueta A se chequea la URL del recurso. Ver sección 1 Chequear URL. Si la URL es válida la elimina de la cola para próximas visitas.
4.		Analiza cada video identificado: • Si el video no tiene una miniatura definida, le extrae un fotograma.
5.		Devuelve los metadatos de los videos encontrados.
6.	Ejecuta el filtro para almacenar los metadatos.	
7.		Guarda los metadatos de cada video.
8.		Termina el caso de uso.
Flujos alternos.		
1 Evento El nodo corresponde a una etiqueta A.		
	Actor	**Sistema**
1.		Obtiene la URL del recurso asociado a la etiqueta y le asocia un video: • Si existe un video con la URL, obtiene una referencia a él. • Si no existe un video, chequea la URL del recurso. Ver sección 1 Chequear URL. Si la URL es válida, crea un nuevo video con dicha URL.
2.		Si el video no tiene una miniatura definida, obtiene la primera imagen (en caso que exista alguna) dentro de la etiqueta.

[7] Se refiere al modo de visitar los nodos de una estructura de datos, en este caso un árbol.

3.		Si aún no tiene miniatura definida le extrae un fotograma al video.
4.		Obtiene el texto alrededor de la etiqueta.
2 Evento El nodo corresponde a una etiqueta VIDEO.		
	Actor	**Sistema**
1.		Obtiene el atributo *"title"* de la etiqueta.
2.		Obtiene el atributo *"poster"* de la etiqueta.
3.		Obtiene las URLs de los recursos asociados a la etiqueta.
4.		Analiza cada URL de los recursos encontrados: • Si existe un video con la URL, obtiene una referencia a él. • Si no existe un video, chequea la URL del recurso. Ver sección 1 Chequear URL. Si la URL es válida, la agrega a la cola para próximas visitas y crea un nuevo video con dicha URL.
5.		Analiza cada video: • Si no tiene título le asocia el valor del atributo *"title"* de la etiqueta. • Si no tiene miniatura definida le asocia el valor del atributo *"poster"* de la etiqueta. • Si aún no tiene miniatura definida le extrae un fotograma al video.
6.		Obtiene el texto alrededor de la etiqueta y lo asocia a cada video relacionado con ella.
Sección 1: "Chequear URL".		
Flujo básico Chequear URL.		
	Actor	**Sistema**
1.		Normaliza la URL.
2.		Filtra la URL teniendo en cuenta la configuración del rastreador. Si no es una URL válida se dirige al flujo alterno 1.
3.		Comprueba si es una URL externa a la página que se está analizando. Si es una URL externa y en la configuración no están permitidas las URLs externas, se considera una URL no válida y se dirige al flujo alterno 1.

4.		Comprueba que la URL esté permitida por el *robot.txt* del sitio. Si no está permitida se considera una URL no válida y se dirige al flujo alterno 1.
5.		Chequea el estado del contenido. Si el contenido ha sido movido temporal o permanentemente redirecciona tantas veces como la configuración del rastreador lo permita. Si excede la cantidad de redireccionamientos permitidos se considera una URL no válida y se dirige al flujo alterno 1.
6.		Se obtiene el *content-type*[8] del contenido y se comprueba si está permitido teniendo en cuenta la configuración del sistema. Si no está permitido se considera una URL no válida y se dirige al flujo alterno 1.
7.		Devuelve la URL.
8.		Termina la sección.
Flujos alternos.		
1 Evento La URL no es válida.		
	Actor	**Sistema**
1.		Devuelve nulo.
2.		Termina la sección.

Relaciones	**CU incluidos**	
	CU extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 2.2: CU Identificar videos en una página web

2.4.2. CU Procesar video

Objetivo	Extraer los metadatos de un video.
Actores	Rastreador.
Resumen	El sistema extrae los metadatos de un video previamente identificado en la red.
Complejidad	Alta.

[8]Se refiere al identificador del tipo de dato contenido en un fichero.

Prioridad	Crítico.	
Precondiciones	El administrador ha configurado el sistema e iniciado el rastreo. El contenido del video ha sido obtenido y se ha identificado como tal.	
Postcondiciones	Se han extraído los metadatos del video y se ha creado un documento para ser indexado en Solr.	
Flujo de eventos		
Flujo básico Identificar videos en un documento.		
	Actor	**Sistema**
1.		Crea un fichero temporal (si no existiese) con las opciones que requiere la librería Mediainfo (Ver sección 1.4.6) para extraer los metadatos del video.
2.		Extrae los metadatos.
3.		Obtiene los datos del video almacenados en la base de datos de Nutch.
4.		Crea el documento a indexar en Solr.
5.		Termina el caso de uso.
Relaciones	**CU incluidos**	
	CU extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 2.3: CU Procesar video

2.5. Diagramas de clases del diseño

Un diagrama de clases del diseño (en lo adelante CD) describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. (...) A diferencia del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en lugar de conceptos del mundo real.

UML no define concretamente un elemento denominado “diagrama de clases del diseño”, sino que se sirve de un término más genérico: “diagrama de clases” [51]. Por tal motivo, se decide utilizar este tipo de diagrama para la representación gráfica de las clases y sus relaciones. A continuación se muestran los diagramas de clases del diseño de los CU que se están analizando; los cuales, representan subprocesos independientes[9] dentro del proceso de rastreo.

[9]Se refiere a que no existe relación directa entre las clases de los subprocesos. Sin embargo, el segundo subproceso sí depende de los resultados que se obtienen en el primero, los cuales son facilitados por el actor del CU.

Para un mayor entendimiento del diagrama de CD correspondiente al CU “Identificar videos en una página web”, se decidió dividir la imagen en dos partes, las cuales representan procesos independientes dentro del propio CU. A continuación se muestra la primera parte de dicho diagrama.

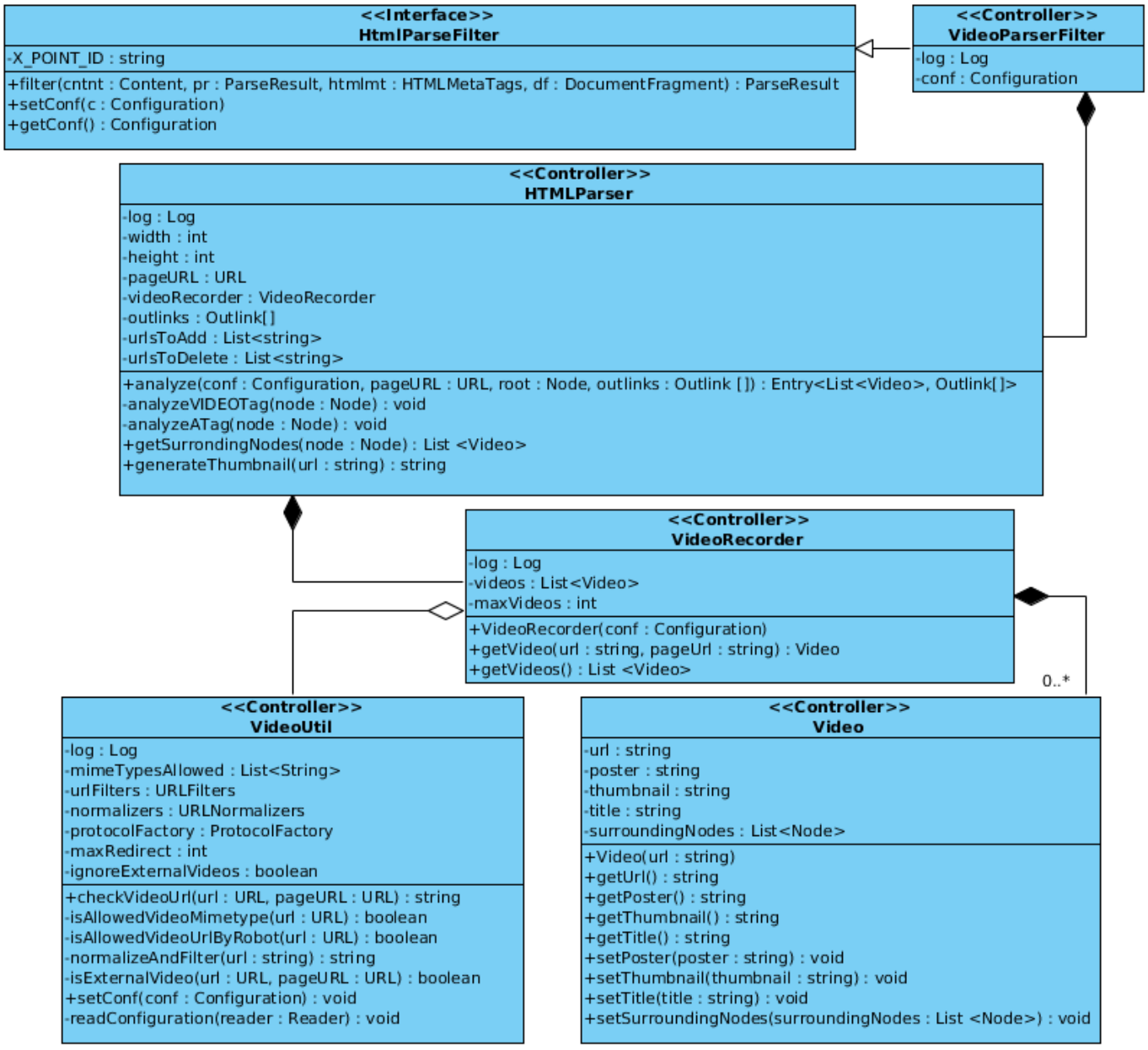

Imagen 2.4: Primera parte del diagrama de clases del diseño del CU “Identificar videos en una página web”

En la imagen anterior, se muestran las clases que intervienen en la primera parte del proceso de identificar los videos existentes en una página web. Para ello, la interfaz HtmlParserFilter actúa como punto de comunicación entre el rastreador y la clase VideoParserFilter, la cual es la encargada de devolver la información de los videos encontrados. Esta última clase, depende de HTMLParser, la cual contiene los procedimientos necesarios para identificar los videos y obtener la información alrededor de ellos. La clase VideoRecorder registra todos los videos encontrados durante el proceso, que en términos de programación serían las instancias de la clase Video; la cual, registra toda la información que se obtiene del video que representa. VideoRecorder depende de VideoUtil para comprobar si una URL representa a un video en la red.

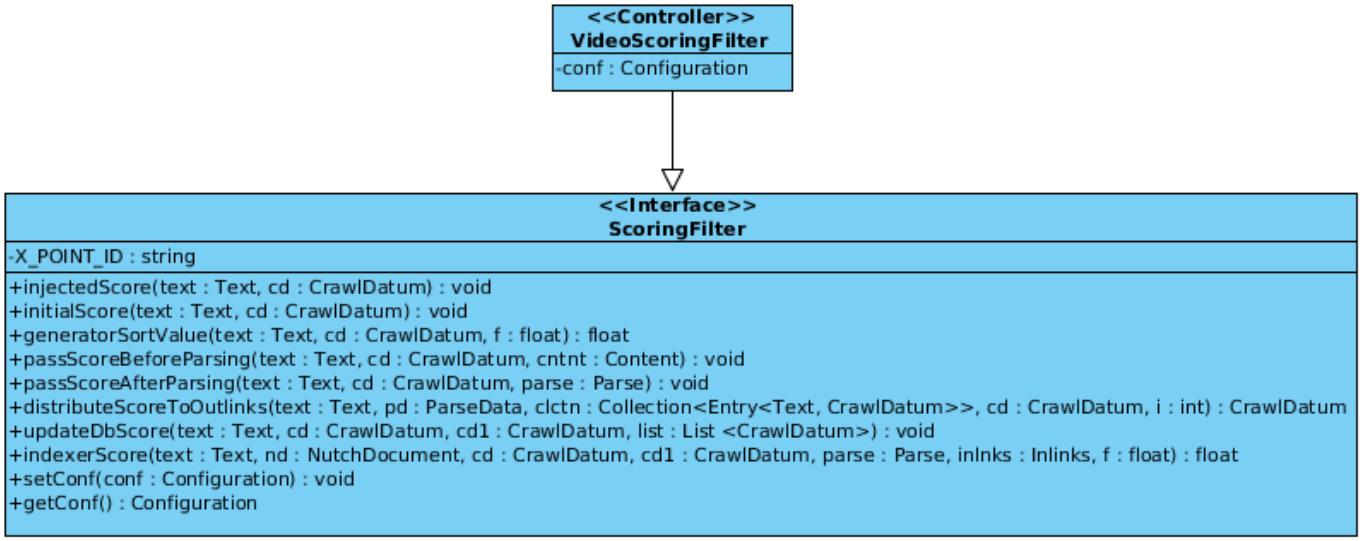

Imagen 2.5: Segunda parte del diagrama de clases del diseño del CU "Identificar videos en una página web"

En una segunda etapa del proceso de "Identificar videos en una página web", la cual, es un proceso independiente al mostrado en la imagen 2.4, la clase VideoScoringFilter tiene la responsabilidad de asociar la información extraída alrededor de cada video a su URL. La comunicación con el rastreador se lleva a cabo a través de la interfaz ScoringFilter.

En la siguiente imagen se muestran las clases que intervienen en el proceso de analizar un video. Para ello, el rastreador se comunica a través de la interfaz Parse con la clase VideoParser, la cual es la encargada de devolver los metadatos extraídos del video. Esta clase depende de VideoMetadataExtractor para la extracción de los metadatos; los cuales, en un subproceso independiente son procesados por VideoIndexingFilter para decidir cuáles metadatos y cómo serán indexados. Esta última clase se comunica con el rastreador mediante la interfaz IndexingFilter.

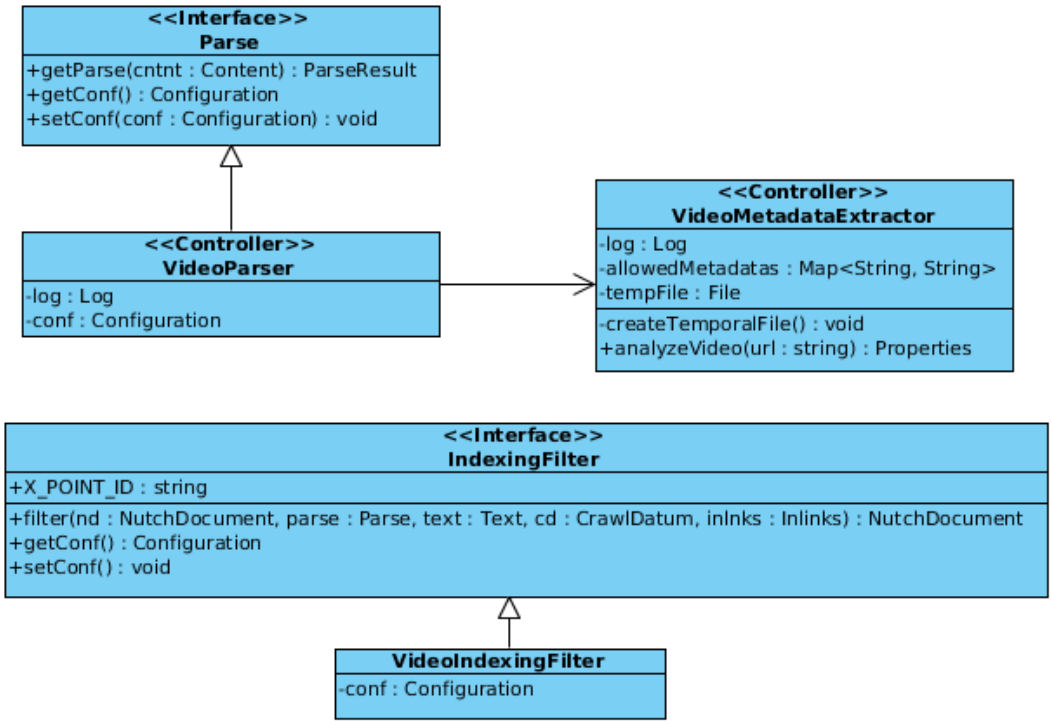

Imagen 2.6: Diagrama de clases del diseño del CU "Procesar video"

2.6. Diagramas de interacción

Según [51] un diagrama de interacción explica gráficamente las interacciones existentes entre las instancias y las clases del modelo de estas. (...) Los tipos de estos diagramas que define UML son: colaboración y secuencia. Ambos son utilizados para expresar interacciones semejantes o idénticas de mensaje. Teniendo en cuenta que en los diagramas de colaboración no es necesario representar el tiempo como una dimensión en el intercambio de mensajes, se deciden utilizar los mismos para el diseño del sistema propuesto.

2.6.1. Diagramas de colaboración

Los diagramas de colaboración describen las interacciones entre los objetos en un formato de grafo o red [51]. A continuación se muestran los diagramas de colaboración para los CU especificados anteriormente.

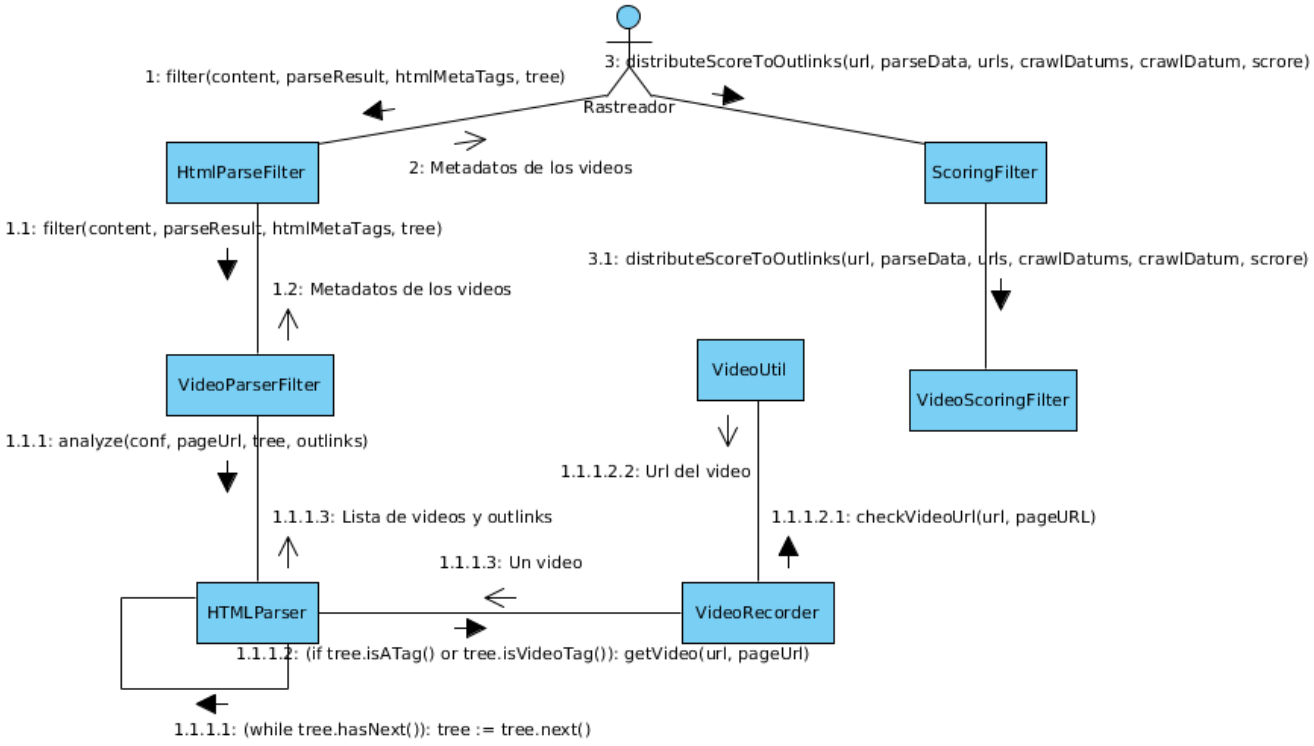

Imagen 2.7: Diagrama de colaboración del CU "Identificar videos en una página web"

En la imagen anterior, se muestra el intercambio de mensajes entre las clases que intervienen en el proceso de identificar los videos existentes en una página web. Para ello, el rastreador se comunica con la clase VideoParserFilter enviándole: el contenido obtenido, el resultado de los análisis realizados en fases[10] anteriores, las metaetiquetas[11] de la página web y el árbol generado. Esta comunicación se realiza mediante la interfaz HtmlParserFilter. Después, VideoParserFilter le solicita a la clase HTMLParser los videos y el listado actualizado de las URLs que contiene la página web. Para que esta clase sea capaz de realizar el análisis, se le envía la configuración del sistema, la URL de la página, el árbol generado y las URLs encontradas hasta el momento.

HTMLParser realiza un recorrido a lo ancho[12] del árbol y por cada URL encontrada le solicita a la clase VideoRecorder el video asociado a ella. Si no existiese, esta última se comunica con la clase VideoUtil para que compruebe la URL que le fue enviada. Si se corresponde con un video y tiene un formato permitido, VideoRecorder registra un nuevo video y se lo envía a HTMLParser para que continúe el proceso.

[10] Se refiere a las fases del proceso de rastreo definidas por Nutch.

[11] Las metaetiquetas son etiquetas HTML que se incorporan en el encabezado de una página web y que resultan invisibles para un usuario, pero de gran utilidad para navegadores u otros programas que puedan valerse de esta información.

[12] Se refiere al modo de visitar los nodos de una estructura de datos, en este caso un árbol.

Una vez obtenida la información que se encuentra alrededor de los videos, el rastreador se comunica con la clase VideoScoringFilter (a través de la interfaz ScoringFilter) para asociarle a cada video encontrado la información que le corresponde. Para ello, le envía la URL de la página web analizada, la información obtenida durante el proceso anterior y la URL de cada enlace contenido en la página.

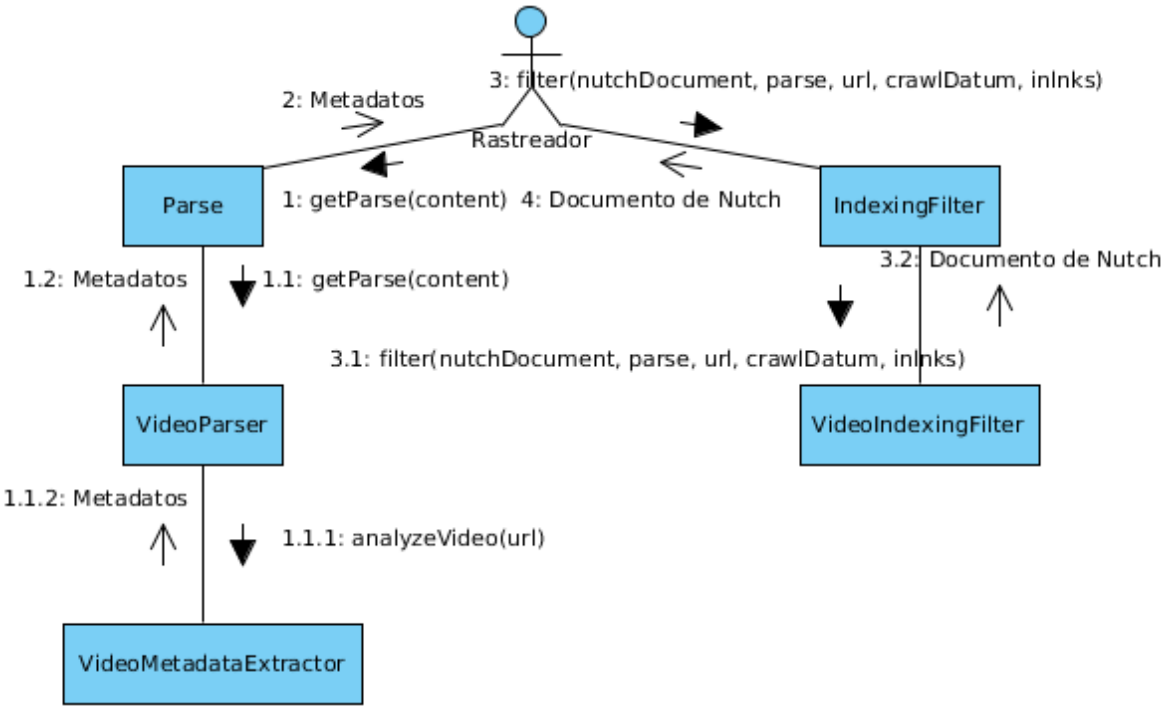

Imagen 2.8: Diagrama de colaboración del CU "Procesar video"

El diagrama de colaboración mostrado anteriormente representa el intercambio de mensajes durante el proceso de analizar un video. El rastreador una vez que ha obtenido el contenido de un video se lo envía a la clase VideoParse mediante la interfaz Parse para que sea procesado. Esta clase se comunica con VideoMetadataExtractor para que obtenga los metadatos del video, facilitándole la URL del mismo.

Una vez obtenidos los metadatos, el rastreador le solicita a la clase VideoIndexingFilter (a través de la interfaz IndexingFilter) que seleccione los metadatos que son necesarios para indexar la información del video que se está analizando. Para ello, le envía la información obtenida del video y su URL.

2.7. Patrones utilizados en el desarrollo del software

En terminología de objetos, un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos. También puede ser visto como una pareja de problema/solución con una sugerencia sobre la manera de utilizarlo en situaciones nuevas [51].

2.7.1. Patrón de casos de uso

Para el modelado de los casos de uso del sistema y su correcta estructuración y organización se utilizan los patrones de casos de uso, específicamente: **"Inclusión concreta"**. Este patrón indica que cuando se tiene algún comportamiento parcial común a varios casos de uso, es conveniente separarlo en uno propio e indicar su inclusión para evitar duplicaciones [50].

Como se muestra en la imagen 2.3, este patrón es utilizado para incluir el CU Transformar criterio de búsqueda en los CU Buscar videos y Filtrar contenidos. Estos últimos, presentan un comportamiento común, el cual consiste en transformar un criterio de búsqueda en una consulta para el indexador. Es por ello, que se decide separar dicho comportamiento en un CU independiente.

2.7.2. Patrones Generales de Software para la Asignación de Responsabilidades (GRASP)

Para el diseño de la propuesta de solución se utilizaron fundamentalmente patrones GRASP, los cuales describen los principios fundamentales de la asignación de responsabilidades a los objetos y generalmente son asignados en el momento de modelar los diagramas de interacción [51]. A continuación se relacionan los principales patrones GRASP utilizados en la realización de los diagramas de colaboración para los CU analizados.

Experto en información

La solución que propone este patrón es la de asignar una responsabilidad a la clase (experto) que cuenta con la información necesaria para cumplirla. Ofrece una analogía con el mundo real ya que da origen al diseño donde el objeto de software realiza las operaciones que normalmente se aplican al elemento real que representa [51].

El siguiente fragmento del diagrama de colaboración del CU "Identificar videos en una página web" indica que la clase VideoScoringFilter es experto, ya que conoce la información necesaria y su adecuada manipulación para distribuirla a todos los enlaces salientes de una página web.

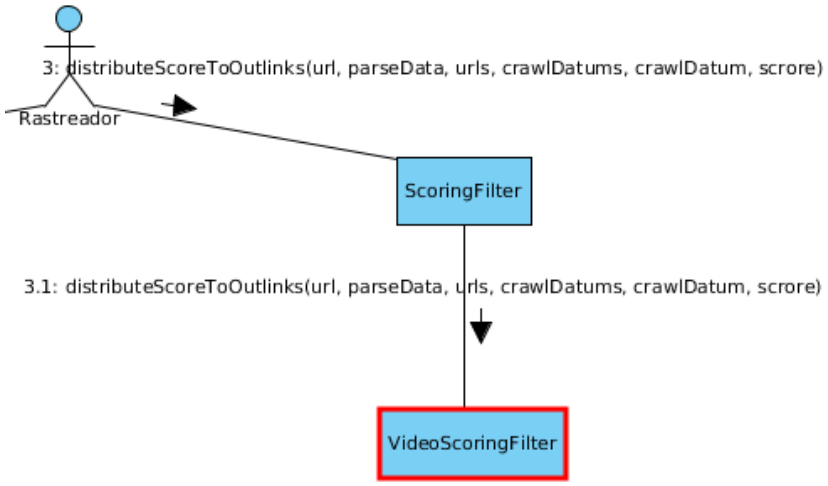

Imagen 2.9: Ejemplo del patrón Experto en información

Bajo acoplamiento

El **acoplamiento** es una medida de la fuerza con que una clase está conectada a otras. (...) Una clase con alto (o fuerte) acoplamiento recurre a muchas otras y no es conveniente su existencia ya que son más difíciles de reutilizar porque se requiere la presencia de las clases de las que dependen. Por otra parte, cambios en las clases afines pueden ocasionar cambios locales y son más difíciles de entender cuando están aisladas [51]. El patrón que se analiza propone asignar las responsabilidades de tal manera que permita lograr un bajo acoplamiento entre las clases involucradas.

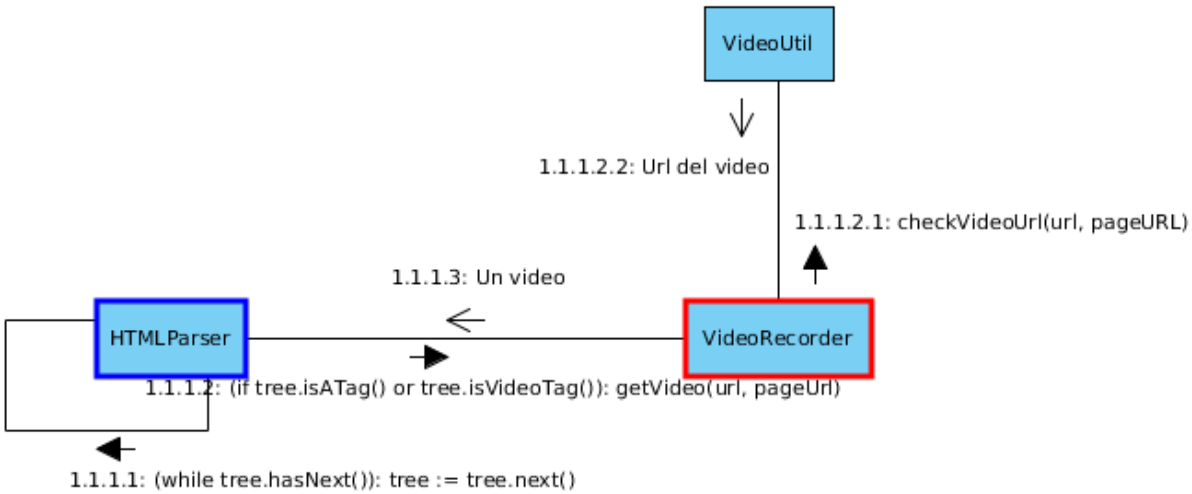

Imagen 2.10: Ejemplo del patrón Bajo acoplamiento

Según el patrón Experto un buen candidato para comprobar si una URL encontrada en una página web pertenece a un video es la clase HTMLParser (se muestra marcada con color azul en la imagen anterior) en el CU “Identificar videos en una página web”. Esta clase cuenta con la información necesaria para tener esta responsabilidad; sin embargo, un diseño con intercambio de mensajes entre las clases HTMLParser y VideoUtil no se ajusta a lo que indica el patrón que se está analizando. Por tal motivo, se decide utilizar el diseño que se muestra en la imagen anterior, el cual conserva un menor acoplamiento global.

Alta cohesión

En la perspectiva del diseño orientado a objetos, la **cohesión** es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. El reto fundamental del patrón “Alta cohesión” es mantener la complejidad de una clase dentro de los límites manejables para que sea fácil de comprender, reutilizar y conservar [51].

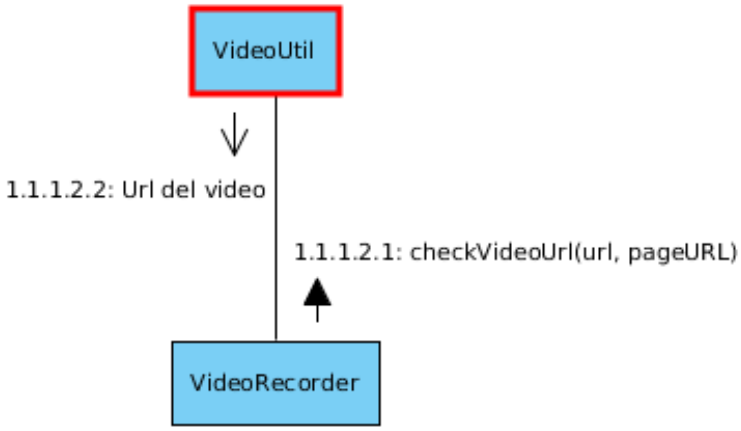

Imagen 2.11: Ejemplo del patrón Alta cohesión

Analizando el CU “Identificar videos en una página web” se puede observar que la clase VideoUtil (marcada en rojo en la imagen anterior) es creada para lograr una alta cohesión en el diseño, ya que sus responsabilidades pueden ser asignadas a la clase VideoRecorder también; sin embargo, estaría asumiendo responsabilidades que no están relacionadas con el propósito para el cual fue creada.

Fabricación pura

Este patrón propone asignar un conjunto cohesivo de responsabilidades a una clase artificial que no representa nada en el dominio del problema, pero da soporte a una alta cohesión, bajo acoplamiento y reutilización [50].

En la imagen siguiente se muestra el uso de este patrón en el diagrama de colaboración del CU “Identificar videos en una página web” donde la clase VideoRecorder fue creada para gestionar y controlar los videos encontrados en una página web y suprimirle esta responsabilidad a la clase HTMLParser logrando un diseño con bajo acoplamiento y alta cohesión.

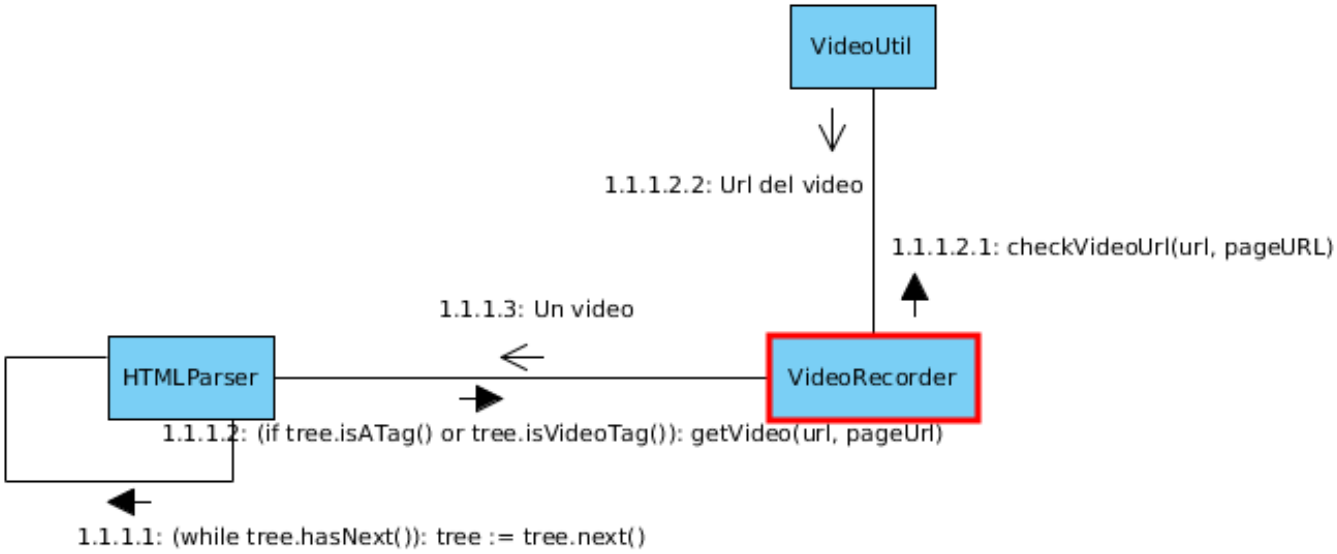

Imagen 2.12: Ejemplo del patrón Fabricación pura

Creador

La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. El patrón “Creador” propone asignarle a una clase B la responsabilidad de crear una instancia de la clase A cuando B agrega, contiene, registra o utiliza las instancias de A o tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado [51].

En la imagen anterior se muestra la utilización del patrón analizado en la elaboración del diagrama de colaboración del CU “Identificar videos en una página web”, donde la clase VideoRecorder tiene la responsabilidad de registrar todos los videos encontrados durante el análisis de la página web.

2.7.3. Patrón arquitectónico implementado

Los patrones de arquitectura expresan un esquema fundamental de organización estructural para sistema de software, donde provee una serie de subsistemas predefinidos, especificando sus responsabilidades, e incluye reglas y guías para organizar las relaciones entre ellos. Estos patrones son utilizados durante la etapa de definición de arquitectura de un sistema [37]. A continuación se muestra el patrón arquitectónico a utilizar en la implementación de la aplicación web del sistema que se propone.

Modelo Vista Controlador

El patrón Modelo Vista Controlador separa la lógica de negocio de la interfaz del usuario y es el más utilizado en aplicaciones web, ya que facilita la mantenibilidad y escalabilidad del sistema, de forma simple y sencilla, a la vez que permite “no mezclar lenguajes de programación en el mismo código” [54].

Atendiendo a lo anteriormente planteado y a que el marco de trabajo seleccionado para el desarrollo de la aplicación web establece el uso de este patrón, se decide utilizarlo en el desarrollo del componente Aplicación web del subsistema de búsqueda de videos para el buscador cubano Orión.

2.8. Modelo de datos

El modelado de datos se basa en la identificación de los objetos primarios que va a procesar el sistema, la composición y atributos de los mismos. Además de dónde se encuentran almacenados actualmente dichos objetos, la relación entre ellos y los procesos que los transforman [55]. A continuación se muestra el modelo de datos para el sistema que se propone desarrollar.

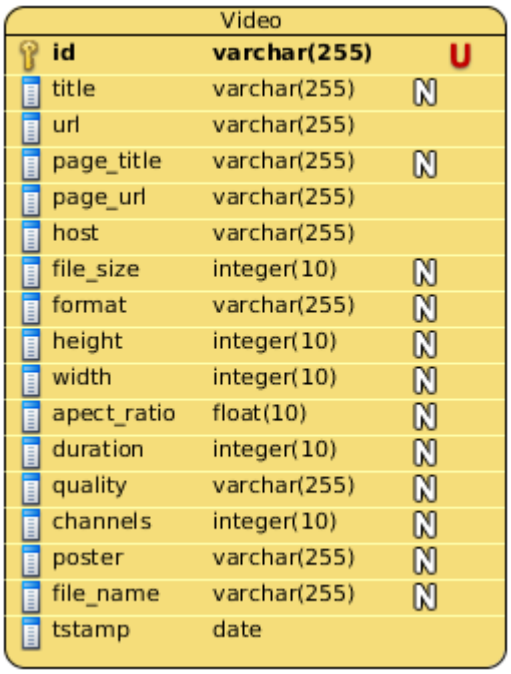

Imagen 2.13: Diagrama de modelo de datos

El modelo de datos mostrado anteriormente es la representación de un documento en Solr[13] y sus atributos. Los cuales, son los metadatos que le fueron extraídos a un video en el proceso de rastreo como es el caso del identificador (id), el título (title), la URL (url), el servidor donde se encuentra (host), el tamaño del recurso (file_size), ancho y alto (width y height respectivamente), proporción (aspect_ratio), duración (duration) y calidad (quality), además del título y la URL de la página web donde se encuentra (page_title y page_url respectivamente), así como el nombre que posee el recurso (file_name).

2.9. Diagrama de despliegue

Según OMG en 2007, un diagrama de despliegue es un tipo de diagrama de UML que se utiliza para modelar la disposición física de los artefactos de software en nodos. A continuación se muestra el diagrama de despliegue propuesto para el subsistema de búsqueda de videos para el buscador cubano Orión.

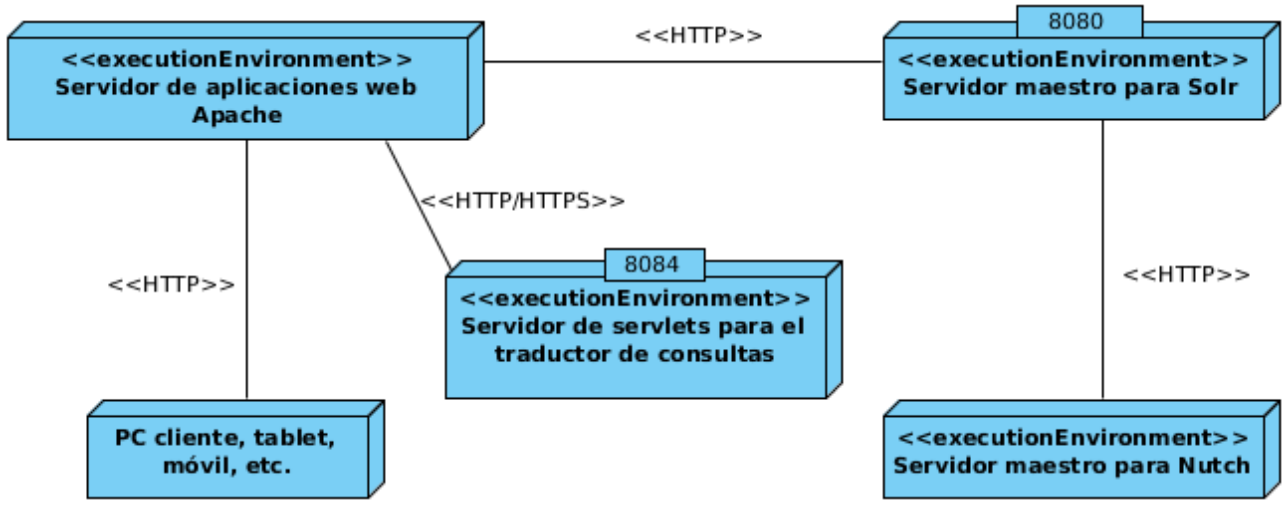

Imagen 2.14: Diagrama de despliegue

El diagrama anteriormente mostrado, representa la distribución física de los componentes del sistema propuesto para su despliegue. Para lo cual, se sugiere que cada uno de ellos se encuentre en servidores independientes con el objetivo de utilizar al máximo las características de software y hardware de éstos.

Cada uno de los nodos, debe cumplir con las características especificadas en los requisitos no funcionales 3, 4, 5, 6, 7, 8, 9, 10 y 11. Por otra parte, el intercambio de los mensajes entre cada uno de ellos se debe realizar mediante el protocolo HTTP o HTTPS y los servidores para el traductor de consultas y el componente de indexación deben utilizar puertos que no sean los predeterminados, por ejemplo: 8084 y 7835 como se muestra en la imagen anterior.

[13] Se refiere a la estructura interna que Solr utiliza para almacenar la información.

2.10. Conclusiones del capítulo

En este capítulo se han abordado los elementos del análisis y diseño del subsistema de búsqueda de videos para el buscador cubano Orión, arribando a las siguientes conclusiones:

1. La elaboración del modelo de dominio y su descripción permitió una mayor comprensión del sistema que se propone desarrollar.
2. La identificación de los requisitos funcionales, permitió agrupar las funcionalidades del sistema en casos de uso, los cuales contribuyeron a una mayor comprensión de los principales procesos del subsistema propuesto.
3. La identificación de los requisitos no funcionales permitió definir las características y condiciones del sistema a desarrollar.
4. La elaboración de los diagramas de clases del diseño y los diagramas de colaboración propició una mayor comprensión de la distribución y asignación de responsabilidades de cada una de las clases involucradas en los casos de uso analizados.
5. La elaboración del diagrama de despliegue permitió identificar la disposición física de los componentes del sistema que se propone.

Capítulo 3

Implementación y pruebas del subsistema de búsqueda de videos para el buscador cubano Orión

El proceso de programación involucra la conversión del diseño en un código de programa. Esto significa que las clases definidas en el diseño deben ser convertidas en clases expresadas en un lenguaje de programación [56]. Una vez terminado este proceso, el código fuente debe ser probado para descubrir y corregir el máximo de errores posibles antes de su entrega al cliente [55]. En estas etapas se generan los artefactos pertenecientes a las mismas, como es el caso del diagrama de componentes, los estándares de codificación y los casos de prueba. De igual modo se valida el correcto funcionamiento del sistema, así como el cumplimiento con los requisitos funcionales y no funcionales identificados en la etapa de análisis, mediante la aplicación de diferentes pruebas.

3.1. Diagrama de componentes

Los diagramas de componentes representan las partes o componentes físicos y reemplazables de un sistema, así como las relaciones entre ellos [57]. A continuación se muestra el diagrama de componentes del subsistema de búsqueda de videos para el buscador cubano Orión.

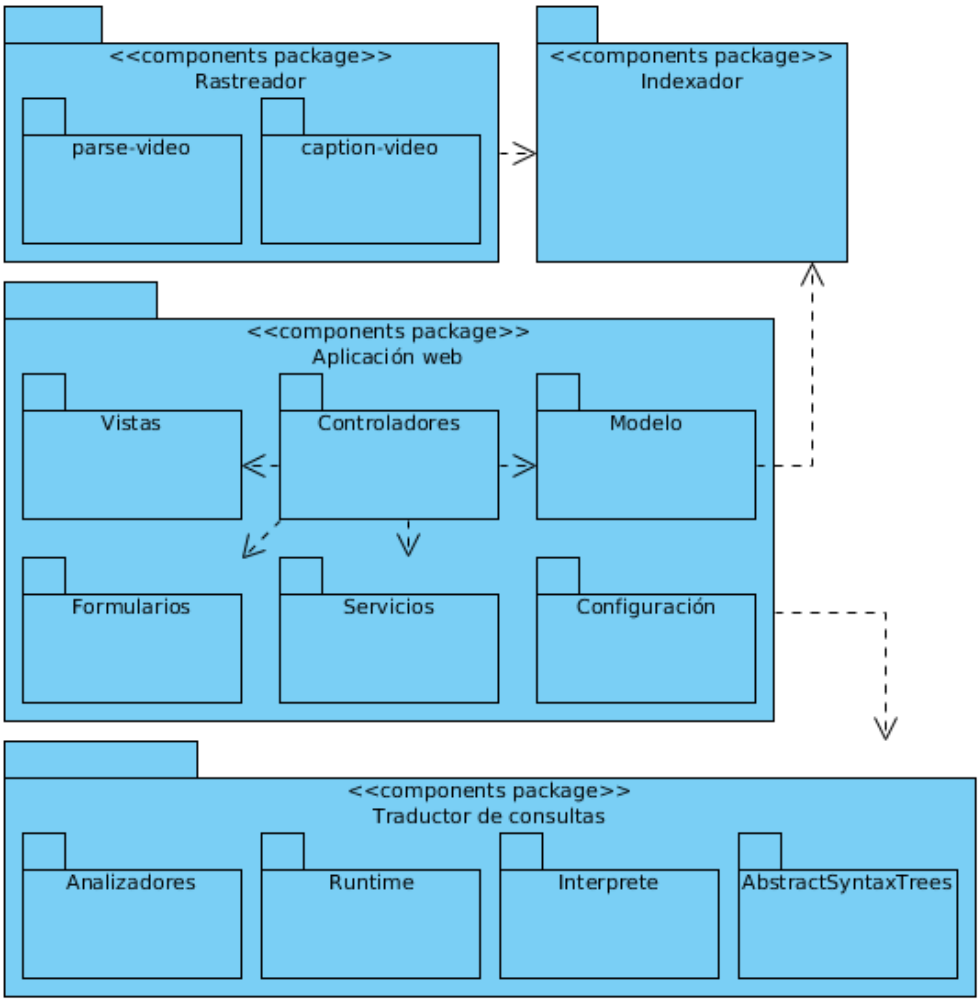

Imagen 3.1: Diagrama de componentes del sistema

El diagrama mostrado anteriormente está compuesto por cuatro paquetes, los cuales representan los componentes de la arquitectura del subsistema de búsqueda de videos para el buscador cubano Orión:

Rastreador: Implementa la lógica de la búsqueda de videos en la red.

Indexador: Contiene los componentes de configuración que permitirán tanto la indexación de documentos como la comunicación con el rastreador y la aplicación web.

Aplicación web: Incluye los paquetes que contienen los controladores, vistas, servicios, formularios y ficheros de configuración de la aplicación web.

Traductor de consultas: Contiene los componentes del sistema encargados de transformar el criterio de búsqueda introducido por el usuario en una consulta para el indexador.

El paquete Rastreador está dividido en dos subpaquetes: parse-video y caption-video, los cuales representan los *plugins* desarrollados para el componente de rastreo. A continuación se muestra la estructura interna de dicho paquete.

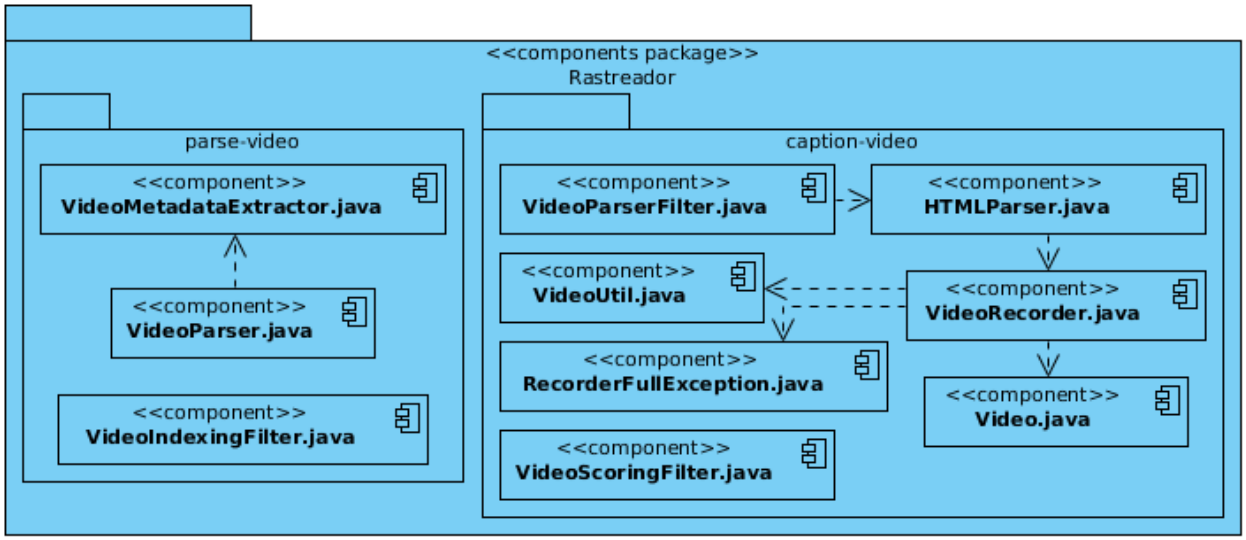

Imagen 3.2: Diagrama de componentes del paquete Rastreador

El subpaquete parse-video incluye tres componentes:

VideoMetadataExtractor.java: Es el fichero que contiene la clase VideoMetadataExtractor, la cual es utilizada para la extracción de los metadatos de los videos.

VideoParser.java: Representa al fichero que contiene la clase VideoParser, cuya función principal es interactuar con el rastreador y enviarle los metadatos extraídos de un video.

VideoIndexingFilter.java: Componente físico que incluye la clase VideoIndexingFilter, la cual tiene la responsabilidad de decidir cuáles metadatos y cómo serán indexados.

El subpaquete caption-video está compuesto por siete componentes, los cuales son:

VideoParserFilter.java: Representa al fichero que contiene la clase VideoParserFilter, la cual contiene los procedimientos necesarios para gestionar la información de los videos encontrados en una página web.

HTMLParser.java: Es el fichero que contiene a la clase HTMLParser, utilizada para identificar los videos en una página web y obtener la información alrededor de ellos.

VideoRecorder.java: Componente que contiene la clase VideoRecorder, la cual registra todos los videos encontrados durante el proceso de análisis de una página web, que en términos de programación serían las instancias de la clase Video (incluida en el componente Video.java y utilizada para intercambiar la información de los videos encontrados).

VideoUtil.java: Representa al fichero que contiene la clase VideoUtil, la cual es utilizada para comprobar si una URL representa a un video en la red.

RecorderFullException.java: Componente físico que contiene la clase RecorderFullException, utiliza para indicar que no se pueden registrar más videos nuevos, ya que la configuración del sistema no lo permite.

VideoScoringFilter.java: Indica el fichero que contiene la clase VideoScoringFilter, la cual tiene la responsabilidad de asociar la información extraída alrededor de cada video a su URL.

El paquete Indexador incluye tres componentes, los cuales son ficheros de configuración que garantizan el correcto almacenamiento de los datos, así como la comunicación con el resto de los componentes de la arquitectura del sistema.

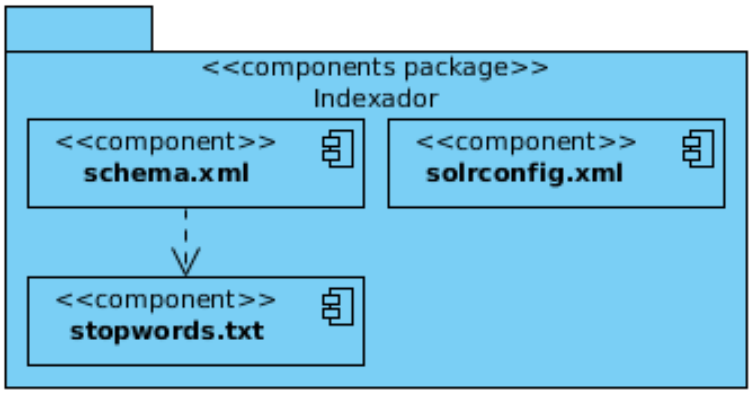

Imagen 3.3: Diagrama de componentes del paquete Indexador

schema.xml: Contiene los campos o atributos de un documento, así como los filtros y tokenizadores[1] que se le aplican a dichos campos, los cuales ayudan a mejorar y facilitar la indexación y la búsqueda.

stopwords.txt: Contiene palabras que no son incluidas como *tokens*[2] cuando se realiza la indexación de documentos o la búsqueda de los mismos.

[1] Componentes de Solr encargados de dividir los datos en unidades léxicas llamadas *tokens*.

[2] Un *token* o también llamado componente léxico es una cadena de caracteres que tiene un significado coherente en cierto lenguaje de programación.

solrconfig.xml: Fichero de configuración principal de Solr.

Como se muestra en la siguiente imagen, los componentes físicos del paquete Aplicación web están divididos en seis subpaquetes: Vistas, Controladores, Formularios, Servicios, Modelo y Configuración.

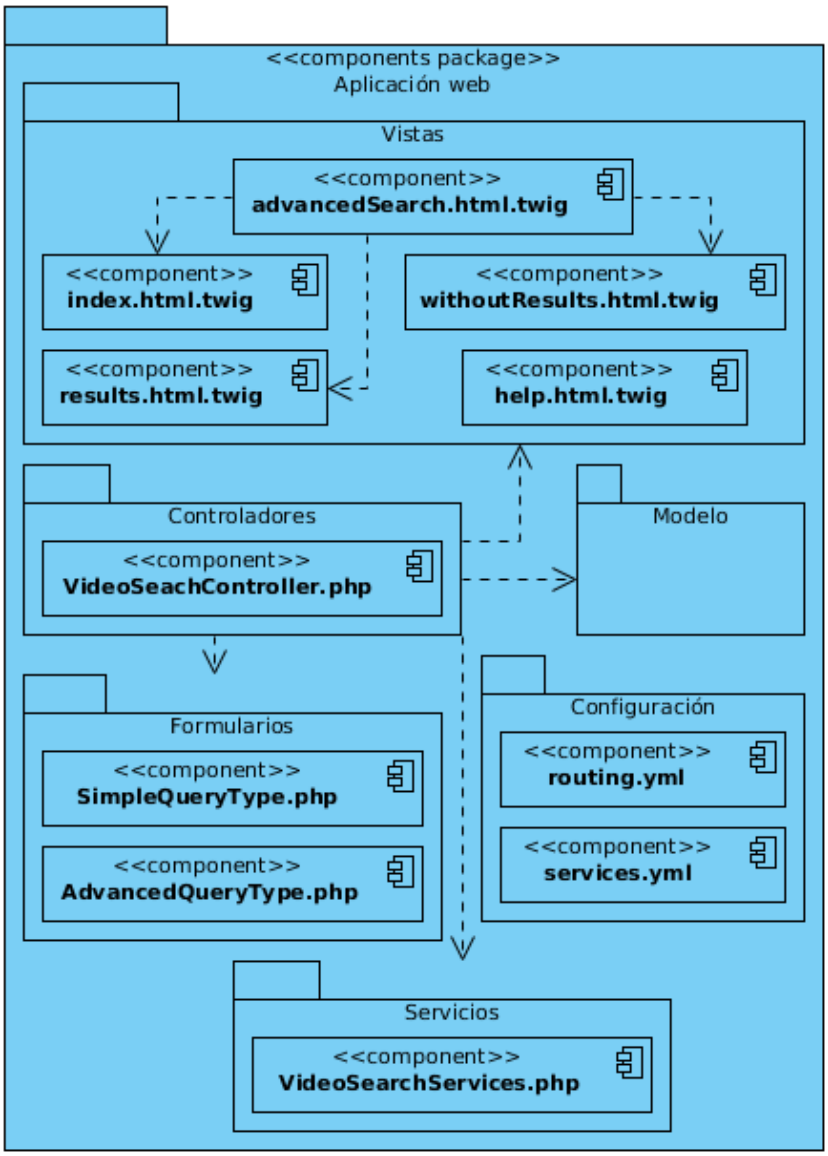

Imagen 3.4: Diagrama de componentes del paquete Aplicación web

En el paquete Controladores se incluye el componente **VideoSearchController.php**, el cual contiene la clase VideoSearchController, encargada de controlar las peticiones de los usuarios, crear las vistas correspondientes a cada una de ellas y manejar los servicios del sistema, así como los formularios de búsqueda.

Incluido en el paquete Servicios, se encuentra el componente **VideoSearchServices.php**, el cual contiene la clase VideoSearchServices, encargada de realizar las peticiones a Solr a partir de la consulta traducida por el Traductor de consultas.

Modelo, es el paquete encargado de la comunicación con el componente de indexación y la representación de la información almacenada en él. Este paquete, no es desarrollado por los autores de la presente investigación pero si es utilizado para la realización de consultas al indexador.

Dentro del paquete Formularios se encuentran los siguientes componentes:

SimpleQueryType.php: Contiene la clase SimpleQueryType, la cual representa la estructura del formulario de búsqueda simple.

AdvancedQueryType.php: Contiene la clase AdvancedQueryType, la cual representa la estructura del formulario de búsqueda avanzada.

En el paquete Configuración se encuentran los ficheros de configuración correspondientes a la aplicación web:

routing.yml: Contiene las rutas que utilizará el sistema.

services.yml: Contiene la configuración de los servicios del sistema.

El paquete Vistas incluye las vistas que muestra el sistema al usuario, las cuales son:

index.html.twig: Es la vista principal del sistema, la cual muestra los formularios de búsqueda simple y búsqueda avanzada.

results.html.twig: Es la vista donde se muestran los resultados, así como los formularios de búsqueda simple y de búsqueda avanzada.

advancedSearch.html.twig: Es la vista que contiene el formulario de búsqueda avanzada, la cual es incluida en la vista principal y la vista de resultados del sistema.

withoutResults.html.twig: Es la vista que se muestra cuando el usuario no ha introducido criterios de búsqueda mostrándole en la misma un mensaje de error, así como los formularios de búsqueda simple y de búsqueda avanzada.

help.html.twig: Es la vista de ayuda del sistema.

Como se muestra en la siguiente imagen, los componentes del paquete Traductor de consultas están divididos en cuatro subpaquetes: Analizadores, Intérprete, Runtime y AbstractSyntaxTrees.

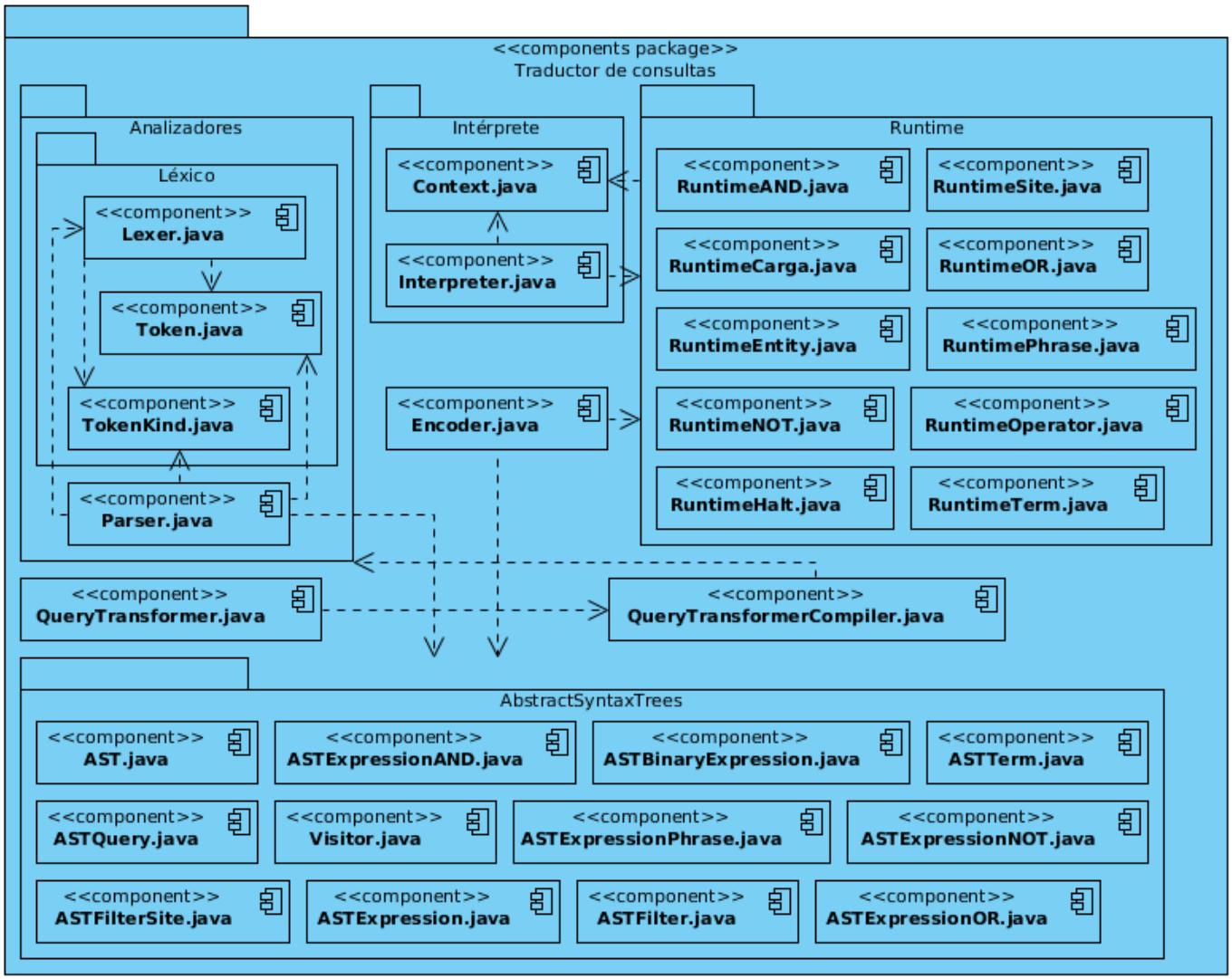

Imagen 3.5: Diagrama de componentes del paquete Traductor de consultas

Dentro del paquete mostrado anteriormente se incluyen los componentes:

Encoder.java: Es el fichero que contiene la clase Encoder, la cual es la encargada de convertir un árbol de sintaxis abstracta (AST por sus siglas en inglés)[3] en la estructura interna utilizada para convertir la consulta del usuario.

QueryTransformer.java: Contiene la clase QueryTransformer, la cual es la encargada de recibir las peticiones y obtener las consultas que le son enviadas al traductor.

[3]Se refiere a la representación en forma de árbol de la estructura sintáctica.

QueryTransformerCompiler.java: Este componente incluye la clase QueryTransformerCompiler, la cual tiene la responsabilidad de gestionar y controlar el proceso de traducción.

El subpaquete Analizadores incluye los componentes encargados de analizar las consultas y detectar errores léxicos y sintácticos. El subpaquete Léxico agrupa los componentes que intervienen en el proceso de análisis léxico. A continuación se mencionan cada uno de ellos:

Lexer.java: Es el fichero que contiene la clase Lexer, la cual analiza lexicográficamente la consulta del usuario para identificar los *tokens* presentes en la misma.

Token.java: Es el fichero que contiene la clase Token, la cual representa los *tokens* encontrados durante el análisis léxico.

TokenKind.java: Es el fichero que contiene la clase enumerativa TokenKind, la cual incluye los tipos de *tokens* permitidos.

Parser.java: Es el fichero que contiene la clase Parser, la cual analiza sintácticamente los *tokens* proporcionados por el analizador léxico.

El paquete Intérprete contiene los componentes necesarios para la traducción de la consulta del usuario, los cuales se describen a continuación:

Context.java: Es el fichero que incluye la clase Context, la cual contiene la información necesitada por el paquete Runtime.

Interpreter.java: Contiene la clase Interpreter, la cual es la encargada de ejecutar los componentes incluidos en el paquete Runtime para transformar la consulta del usuario.

Por otra parte, el paquete AbstractSyntaxTrees contiene los componentes utilizados para representar la estructura sintáctica de la consulta que se desea traducir; es decir, contiene los AST. Además de esto, el paquete Runtime incluye los componentes utilizados en la representación de la estructura interna utilizada para convertir la consulta del usuario.

3.2. Estándares de codificación

La más evidente de las prácticas comunes es probablemente la propiedad colectiva del código, característica esencial para la libertad del software. Para facilitar el trabajo conjunto, se siguen estándares de codificación que permiten una lectura rápida y simple del código [58], además de influir en la calidad del software contribuyendo de esta manera a una adecuada gestión del código fuente [59], logrando así mantenibilidad y legibilidad en el mismo.

En el caso de la aplicación web se utiliza el estándar definido por la comunidad de desarrollo de Symfony[4] y para la implementación de los *plugins* para Nutch se emplea el estándar definido para el lenguaje Java[5].

3.3. Validación del sistema

La validación del sistema incluye un conjunto de actividades para asegurar que el software desarrollado se corresponde con los requisitos del cliente. Dentro de estas actividades se encuentran las pruebas de validación, las cuales tienen como objetivo evaluar la calidad y de manera más pragmática, descubrir errores en el sistema [53]. A continuación se muestran algunas pruebas realizadas al subsistema de búsqueda de videos para el buscador cubano Orión, así como los resultados obtenidos.

3.3.1. Pruebas funcionales

Con el objetivo de identificar situaciones que no se ajustan a las especificaciones funcionales, establecidas en la fase de análisis del proceso de desarrollo del software propuesto, se realizan las pruebas funcionales. Estas, son descritas en artefactos de la Ingeniería de Software conocidos como casos de prueba, los cuales son especificaciones de las entradas y la salida esperada por el sistema [60]. A continuación, se muestran fragmentos de algunos casos de pruebas elaborados para los CU “Identificar videos en una página web” y “Procesar videos”.

[4]Accesible en: https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-2-coding-style-guide.md.
[5]Accesible en: http://systempix.com/descargas/Convenciones_Codigo_Java.pdf.

Descripción	Var. 4	Var. 10	Var. 11	Var. 12	Var. 13	Var. 14	Var. 15	Var. 16	Var. 17	Respuesta del sistema
Se le envía al sistema el árbol de la página web y una configuración válida para que realice el análisis.	V	V	NA	V	V	NA	V	NA	V	Metadatos de 14 videos diferentes. Los metadatos incluyen el título y la URL de la página que se analiza, además de una miniatura del video con 229 píxeles de ancho y 172 píxeles de alto. Incluye para cada video el texto de 1 etiqueta HTML tanto a la izquierda como a la derecha.
	Árbol que contiene enlaces a 20 videos diferentes (4 videos .avi, 10 videos .webm, 6 videos .mpg y todos pertenecen al mismo dominio de la página que se analiza) utilizando la etiqueta A de HTML sin atributos.	-1		229	172		1		video/ (webm| avi)	

Tabla 3.1: Muestra del escenario: Identificar los enlaces a videos en una página web con una configuración válida

En la tabla anterior, se muestran los valores que deben tomar las principales variables que intervienen en el proceso de identificación de videos en una página web utilizando una configuración válida. Además de esto, se evidencia la respuesta correcta del sistema para cada uno de los juegos de datos de entrada. Las principales variables que intervienen en este proceso son:

Var. 4: Representa la estructura de la página web mediante un árbol.

Var. 10: Representa la cantidad máxima de videos a identificar en una página web. Si el valor es -1 se identifican todos los videos.

Var. 11: Si es falso, los videos que no están ubicados en el mismo dominio son incluidos y si es verdadero sucede lo contrario.

Var. 12: Ancho de las miniaturas de los videos.

Var. 13: Alto de las miniaturas de los videos.

Var. 14: Si es verdadero se extrae como metadato del video el atributo *"title"* de las etiquetas A de HTML y si es falso no se extrae.

Var. 15: Cantidad de nodos a obtener alrededor de las etiquetas A de HTML.

Var. 16: Cantidad de nodos a obtener alrededor de las etiquetas VIDEO de HTML.

Var. 17: Expresión regular con los tipos de videos a identificar en una página web.

La siguiente tabla contiene un fragmento de los casos de prueba para el proceso de identificación de videos en una página web utilizando una configuración inválida. En esta tabla se utilizan las mismas variables que en el escenario mostrado anteriormente.

Descripción	Var. 4	Var. 10	Var. 11	Var. 12	Var. 13	Var. 14	Var. 15	Var. 16	Var. 17	Respuesta del sistema
Se le envía al sistema el árbol de la página web y una configuración inválida para que realice el análisis.	V	I	I	I	I	I	I	I	V	Establece valores por defecto para cada una de las variables inválidas y extrae los metadatos de 14 videos diferentes. Los metadatos incluyen el título y la URL de la página que se analiza, además de una miniatura del video con 229 píxeles de ancho y 172 píxeles de alto. Incluye para cada video el texto de 1 etiqueta HTML tanto a la izquierda como a la derecha.
	Árbol que contiene enlaces a 20 videos diferentes (4 videos .avi, 10 videos .webm, 6 videos .mpg y todos pertenecen al mismo dominio de la página que se analiza) utilizando la etiqueta A de HTML sin atributos.	cuatro	FALSO	cien	veinte	FALSO	uno	dos	video/ (webm| avi)	

Tabla 3.2: Muestra del escenario: Identificar los enlaces a videos en una página web con una configuración inválida

La siguiente tabla contiene un caso de prueba para el escenario relacionado con el procesamiento de un video. En este, la **variable 1** (variable que interviene en este proceso) contiene información referente al contenido de la página web que se está analizando; tal como: URL, el binario del contenido, metadatos propios del protocolo de la URL y el tipo de contenido. Además, para este caso de uso se muestra la respuesta correcta del sistema.

Descripción	Variable 1	Respuesta del sistema
Se le envía al sistema la URL del video para ser procesado.	V	Metadatos del video.
	Contiene la URL: http://internos.uci.cu/reproductor_am/video?type=descarga&nid=24437	

Tabla 3.3: Muestra del escenario: Procesar videos

Como se muestra en el siguiente gráfico se realizaron dos iteraciones de pruebas funcionales al sistema. En la primera se detectaron 13 no conformidades, relacionadas fundamentalmente con: la introducción de caracteres no alfanuméricos[6] en los formularios de búsqueda, la obtención de todos o gran parte de los documentos indexados mediante la ejecución de consultas directas al componente de indexación, la incorrecta extracción de metadatos de los videos rastreados y la inadecuada gestión de videos duplicados durante el proceso de rastreo. Estas no conformidades fueron resueltas en su totalidad y en una segunda iteración de pruebas no se detectaron no conformidades. Esto muestra, que el sistema se ajusta a las necesidades del cliente y cumple con los requisitos funcionales definidos por este.

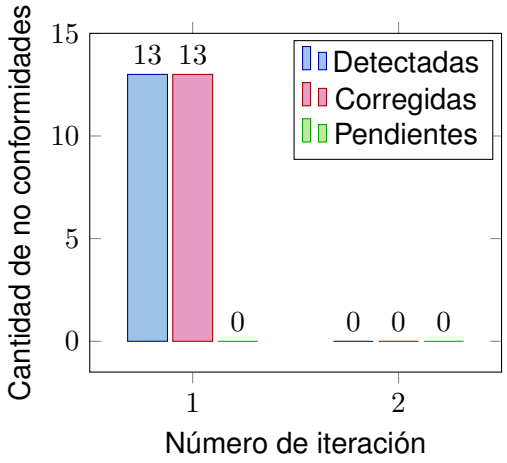

Imagen 3.6: Resultados estadísticos de las pruebas funcionales realizadas

3.3.2. Pruebas de integración

El proceso de integración del sistema implica construirlo a partir de sus componentes y probar el sistema resultante para encontrar problemas que pueden surgir debido a su integración. (...) Las pruebas asociadas a este proceso comprueban que los componentes realmente funcionan juntos, son llamados correctamente y transfieren los datos correctos en el tiempo preciso a través de sus interfaces [60].

El motor de búsqueda cubano Orión, cuenta con una estructura que facilita la integración de nuevos subsistemas para incrementar sus funcionalidades. Por tal motivo, se decidió realizar pruebas de integración descendentes. Las cuales, consisten en desarrollar la infraestructura del sistema en su totalidad y luego añadirle los componentes funcionales [60].

[6]Un carácter alfanumérico es un término informático referente al conjunto de caracteres numéricos y alfabéticos de los cuales dispone una computadora.

Las pruebas de integración realizadas permitieron identificar una URL duplicada en diferentes subsistemas del buscador cubano Orión. Luego de solucionar esta no conformidad, el subsistema de búsqueda de videos desarrollado se integró correctamente con dicho motor de búsqueda.

3.3.3. Pruebas de seguridad

Las pruebas de seguridad se realizan para comprobar que los mecanismos de protección integrados en el sistema realmente lo protejan de irrupciones inapropiadas [53]. Para ello, se utilizó la herramienta Acunetix Web Vulnerability Scanner, caracterizada en la sección 1.4.6.

Mediante estas pruebas, se detectaron 6 vulnerabilidades: una de ellas posee un nivel de criticidad medio, cuatro son de nivel bajo y una es informacional[7]. Dichas vulnerabilidades, no están asociadas al subsistema de búsqueda de videos desarrollado. Sin embargo, fueron identificadas algunas relacionadas con la versión de PHP instalada, por lo que se recomienda para el despliegue del sistema utilizar la versión 5.5.12 o superior, ya que en estas versiones se solucionan dichas vulnerabilidades. Además, se detectaron vulnerabilidades relacionadas con la configuración del servidor web utilizado, tales como:

Ausencia del encabezado X-Frame-Options: Esta vulnerabilidad permite a un atacante realizar ataques Clickjacking[8].

Habilitado el método OPTIONS: Provee la lista de métodos[9] soportados por el servidor web, lo cual expone información sensible que puede ayudar a un atacante a realizar ataques avanzados.

Revelación de la versión del servidor web en las páginas de error: Las páginas de error contienen la versión del servidor web y una lista de los módulos habilitados en este. Esta información puede conducir a la realización de ataques.

Las vulnerabilidades mostradas anteriormente deben tenerse en cuenta para la configuración del servidor web utilizado en el despliegue del sistema.

3.3.4. Pruebas de carga y estrés

Una vez que un sistema se ha integrado correctamente, es posible probar las propiedades emergentes del sistema tales como rendimiento y fiabilidad. Las pruebas de rendimiento (también conocidas como pruebas de carga y estrés) tienen que diseñarse para asegurar que el sistema pueda procesar la carga esperada. Esto

[7] Esta categoría no representa un nivel de criticidad en sí. Es utilizada para agrupar recomendaciones que permitan optimizar la seguridad de los sistemas.

[8] Técnica maliciosa para engañar a usuarios de Internet, con el fin de que revelen información confidencial o tomar control de su computadora cuando hacen clic en páginas web aparentemente inocentes.

[9] Se refiere a los métodos que ofrece el protocolo HTTP para ejecutar acciones en el servidor web.

normalmente implica planificar una serie de pruebas en las que la carga se va incrementando regularmente hasta que el rendimiento del sistema se haga inaceptable [60].

Para la realización de las pruebas de carga y estrés se utilizó la herramienta Apache Jmeter, descrita en la sección 1.4.6. Las pruebas se realizaron desde una computadora con 4GB de RAM, microprocesador Intel Core i3 con 3.30 GHz y sistema operativo Ubuntu 14.04. A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al sistema.

Muestra: Cantidad de peticiones realizadas para cada URL.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Mediana: Tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

Min.: Tiempo mínimo que demora un hilo en acceder a una página.

Max.: Tiempo máximo que demora un hilo en acceder a una página.

Línea 90 %: Máximo tiempo utilizado por el 90 % de la muestra, al resto de la misma le llevo más tiempo.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): El rendimiento se mide en cantidad de solicitudes por segundo.

KB/s: El rendimiento se mide en cantidad de kilobytes[10] por segundo.

Como se muestra en la siguiente tabla, se simularon las peticiones realizadas al sistema por un total de 100, 500 y 1000 usuarios simultáneamente, obteniéndose los siguientes resultados:

Usuarios	Muestra	Media	Mediana	Línea 90 %	Mín.	Máx.	% Error	Rend.	KB/s
100	8200	1259	1397	1724	1	2666	0	75.7	655.3
500	41000	4167	2113	3110	5	321789	0.48	73.3	663.7
1000	69396	9770	2260	5142	2	506901	2.09	44.2	346.7

Tabla 3.4: Resultados obtenidos a partir de las pruebas de carga y estrés realizadas

Las pruebas realizadas muestran que el sistema es capaz de responder a 8200 peticiones de 100 usuarios conectados simultáneamente en un tiempo promedio de 1259 milisegundos (1.3 segundos aproximadamente) con 0 % de error. Esto evidencia que el sistema puede procesar la carga esperada, cumpliéndose de este modo el requisito no funcional 14.

[10] Unidad de almacenamiento de información equivalente a 1024 bytes.

Por otra parte, se realizaron 41000 peticiones iniciadas por 500 usuarios y en este caso el sistema respondió en 4167 milisegundos (4.2 segundos aproximadamente) como tiempo promedio. Esto demuestra que el sistema responde en el tiempo esperado a un conjunto de peticiones 8.2 veces mayor que el propuesto en el requisito no funcional 14, aunque no fue capaz de responder correctamente el 0.48 % de las peticiones realizadas.

Por último, y con el objetivo de analizar el comportamiento del sistema en condiciones extremas, se realizó una prueba de estrés para un conjunto de 1000 usuarios conectados simultáneamente. En este caso, el sistema no responde adecuadamente, siendo capaz de responder solamente a 69396 peticiones de las 82000 esperadas, en aproximadamente el doble del tiempo propuesto en el requisito no funcional 14.

3.3.5. Evaluación del tiempo del proceso de búsqueda

Con el objetivo de determinar si la hipótesis de la presente investigación es apoyada o refutada de acuerdo a los resultados obtenidos con el desarrollo del subsistema propuesto, se decide realizar un experimento puro. Este, es *"un estudio de investigación en el que se manipulan deliberadamente una o más variables independientes (supuestas causas) para analizar las consecuencias de esa manipulación sobre una o más variables dependientes (supuestos efectos), dentro de una situación de control para el investigador"* [61].

Mediante dicho estudio se analizó el comportamiento de la variable dependiente: "tiempo de las búsquedas de los videos publicados en la red cubana realizadas con el buscador cubano Orión" planteada en la hipótesis de investigación. Para ello, se realizó una manipulación de grado dos[11] de la variable independiente, donde se tomó una muestra de 10 personas que se encuentran familiarizadas con la utilización de motores de búsqueda, incluido el propio Orión.

Estas personas fueron sometidas a 5 escenarios, donde se evidencian diferentes necesidades de información (Ver anexo 4). En un primer momento, se realizaron búsquedas de videos utilizando el buscador Orión sin integrarle el subsistema de búsqueda de videos desarrollado y luego se utilizó el mismo buscador pero con dicho subsistema integrado.

Para realizar dicho experimento, se estableció una cuota máxima de 30 minutos, donde los usuarios debían abandonar la búsqueda una vez excedido dicho tiempo. En la siguiente tabla se muestran los resultados del experimento realizado.

[11] La manipulación o variación de la variable independiente de grado dos (también conocida como presencia - ausencia) implica que un grupo (en este caso de personas) se exponga a la presencia de la variable independiente y el otro no. Luego, los dos grupos son comparados para ver si el grupo que se expuso a la variable independiente difiere del grupo no expuesto a esta [61].

	Escenario 1	Escenario 2	Escenario 3	Escenario 4	Escenario 5	Promedio
Sin subsistema	1619	1518.1	1495	1180.1	830.6	1328.74
Con subsistema	78.8	16.2	17.6	14.9	43.9	34.28

Tabla 3.5: Resultados de la medición de la variable "tiempo de las búsquedas de los videos publicados en la red cubana realizadas con el buscador cubano Orión" (en segundos)

A partir de la comparación de los tiempos medios obtenidos en el experimento realizado (Ver tabla 3.5), se evidencia una reducción significativa del tiempo empleado por un usuario al realizar la búsqueda de videos mediante el uso del subsistema desarrollado. Por tal motivo, la hipótesis de investigación anteriormente planteada es apoyada.

3.4. Conclusiones del capítulo

En este capítulo se han abordado los elementos de la implementación del subsistema de búsqueda de videos para el buscador cubano Orión, así como las pruebas realizadas al mismo y los resultados obtenidos; arribando a las siguientes conclusiones:

1. La elaboración de los diagramas de componentes, permitió una mejor comprensión de la estructura de los componentes del sistema implementado.
2. El correcto uso de los estándares de codificación permitió que el código del sistema desarrollado fuera legible para lograr una fácil y mejor comprensión del mismo, la cual es de utilidad para el mantenimiento del sistema.
3. El proceso de validación de software arrojó como resultado que el sistema implementado responde a los requerimientos definidos por el cliente.

Conclusiones generales

De manera general se puede concluir sobre la presente investigación:

1. El estudio de las relaciones existentes entre los principales conceptos asociados al dominio de la presente investigación, permitieron una mayor comprensión de la propuesta de solución.
2. El estudio del estado del arte de los sistemas de recuperación de información de videos permitió demostrar la necesidad de crear una herramienta para la búsqueda de videos, la identificación de los filtros a incorporar en el subsistema y las tecnologías a utilizar en su desarrollo.
3. La elaboración de los artefactos propuestos por la metodología de desarrollo y el levantamiento de requisitos permitieron una mayor comprensión del sistema que se propone desarrollar, así como la identificación de los procesos y características del mismo.
4. La utilización de los estándares de codificación permitió lograr una mayor legibilidad y mantenibilidad del código creado.
5. El proceso de validación de software arrojó como resultado que el sistema implementado responde a los requerimientos definidos por el cliente.

Recomendaciones

1. Incluir al subsistema de búsqueda de videos un mecanismo para la identificación de desnudos.
2. Incorporar al sistema desarrollado un mecanismo para la identificación de objetos en los videos.
3. Incorporar al subsistema de búsqueda de videos un mecanismo para la identificación del idioma a partir del audio de los videos.

Referencias Bibliográficas

1 GARCÍA DUCONGÉ, O. *Biblioteca para la manipulación de videos digitales en Sistemas de Realidad Virtual.* 2007. 70 págs. ⟨http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_1571_08⟩.

2 CABERO ALMENARA, J. Esfuerzo mental y percepciones sobre la televisión/vídeo y el libro. Replicando un estudio de Salomon. *Universidad de Sevilla*. 1993. ⟨http://edutec.rediris.es/documentos/1993/6.htm⟩. ISSN 0210-5934.

3 NAVARRO, E. *Negocios en Internet y el comercio electrónico.* 2007.

4 IAB. *Video marketing y publicidad en vídeo online: aproximación desde la perspectiva del usuario.* 2011. dirección: ⟨http://www.iabspain.net/wp-content/uploads/downloads/2012/05/Informe_vIdeo_IAB_15_septiembre_2011.pdf⟩.

5 RODRÍGUEZ RUEDA, E. e HIDALGO DELGADO, Y. Los spiders y su función en los motores de búsqueda. *UCIENCIA*. 2012. ⟨http://repositorio_institucional.uci.cu//jspui/handle/ident/4130⟩. ISSN 978-959-286-019-3.

6 CUBADEBATE. *Escandalosa censura de Google a Cuba: Medios y blogueros de la Isla no pueden acceder a estadísticas.* 2012. dirección: ⟨http://www.cubadebate.cu/noticias/2012/06/19/escandalosa-censura-de-google-a-cuba-medios-y-blogueros-de-la-isla-no-pueden-acceder-a-estadisticas/⟩.

7 PÉREZ, L. y ELIZALDE, R. M. *Ailyn Febles: La política de Informatización de Cuba partirá de una visión inclusiva, moderna y sostenible.* 2015. dirección: ⟨http://www.granma.cu/cuba/2015-02-13/ailyn-febles-la-politica-de-informatizacion-de-cuba-partira-de-una-vision-inclusiva-moderna-y-sostenible⟩.

8 HIDALGO DELGADO, Y. *Orión, un motor de búsquedas para la web de la UCI.* 2010. ⟨http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_03440_10/1/TD_03440_10.pdf⟩.

9 BETANCOURT GONZÁLEZ, J. L. *Búsqueda de videos en el Orión actual.* 2015. Comunicación personal.

10 CABRERA GUERRA, I. y VEGA PRIETO, R. *Propuesta de un modelo base para un sistema de búsqueda de videos digitales a través de metadatos.* 2009. 104 págs. ⟨http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2756_09⟩.

11 ZAFRA, A.; GIBAJA, E; LUQUE, M. y VENTURA, S. Diseño de aplicaciones cliente/servidor para el aprendizaje de las tecnologías de comunicación. *Iniciación a la Investigación*. 2013, págs. 12. ⟨http://revistaselectronicas.ujaen.es/index.php/ininv/article/view/1748/1528⟩. ISSN 1988-415X.

12 HARRISON, M.; BEIDEMAN, R.; BARTHEL, H.; GRAY, S. y TRAUB, K. HTTP Uniform Resource Identifiers to associate a web resource with a GS1 key and optional Application Identifiers. *Auto-ID Labs*. 2014, págs. 13. ⟨http://autoidlabs.org/uploads/media/AUTOIDLABS-WP-SWNET-032.pdf⟩.

13 GUZMÁN GONZÁLEZ, G. y SANTOS SANABRIA, C. *Componente para la indexación y búsqueda contextual de información audiovisual.* 2011. 102 págs. ⟨http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_04322_11⟩.

14 BRITO SALAZAR, C. y MACIAS SALAZAR, E. *Sisweb. Sistema Bot-Web Buscador e Indexador de Información.* 2008. 98 págs. ⟨http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_1607_08⟩.

15 RUIZ MATEO, A. La utilización educativa del video en educación primaria. *Innovación y experiencias educativas*. 2009, págs. 13. ISSN 1988-6047.

16 CAIZA MÉNDEZ, D. G. y PÉREZ INSUASTI, J. J. *Implementación de un Prototipo de Grabación Automatizado de Señal de Televisión Abierta.* 2011. 198 págs. ⟨http://dspace.espoch.edu.ec/handle/123456789/1940?mode=simple&submit_simple=Muestra+el+registro+sencillo+del+%C3%83%C2%ADtem⟩.

17 SENSO, J. A. y ROSA PIÑERO, A. The metadata concept: something more than description of electronic resources.: The metadata concept. *Ciência da Informação*. 2003, vol. 32, n.º 2, págs. 95-106. ⟨http://dx.doi.org/10.1590/S0100-19652003000200011⟩. ISSN 0100-1965.

18 WAINERMAN, E. *Motores de búsqueda en Internet.* 2001. ⟨http://www.unlu.edu.ar/~tyr/tyr/TYR-motor/wainerman-motor.pdf⟩.

19 MILLER, M. *Using Google Advanced Search.* 2012. ISBN 978-0-7897-4365-7.

20 CAMARGO SARMIENTO, F. I. y ORDÓÑEZ SALINAS, S. Evolución y tendencias actuales de los Web crawlers. *Ingeniería*. 2013, vol. 18, n.º 2, págs. 17. ⟨http://dx.doi.org/10.14483/udistrital.jour.reving.2013.2.a02⟩. ISSN 0121-750X.

21 JACK, P. *Heritrix - Heritrix - IA Webteam Confluence.* 2004. dirección: ⟨https://webarchive.jira.com/wiki/display/Heritrix/Heritrix;jsessionid=483B1610008E0564DB61A8D096AE48BC⟩.

22 MOHR, G.; STACK, M.; RANITOVIC, I.; AVERY, D. y KIMPTON, M. *An Introduction to Heritrix: An open source archival quality web crawler.* 2004.

23 *WIRE - Web Information Retrieval Environment.* 2011. dirección: ⟨http://www.cwr.cl/projects/WIRE/⟩.

24 *Elasticsearch: The Definitive Guide.* 2014. dirección: ⟨http://www.elastic.co/guide/en/elasticsearch/guide/current/intro.html⟩.

25 *Swish-e :: Home Page.* 2007. dirección: ⟨http://swish-e.org/index.html⟩.

26 *Apache Lucene - Apache Solr.* 2011. dirección: ⟨http://lucene.apache.org/solr/index.html⟩.

27 *About Perl.* 2014. dirección: ⟨http://www.perl.org/about.html⟩.

28 MACÍAS RODRÍGUEZ, J. y VÁZQUEZ RODRÍGUEZ, A. *Estudio de Sistemas de Gestión de Contenidos basados en lenguaje PHP.* 2007. 70 págs. ⟨http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_0882_07⟩.

29 GONZÁLEZ DUQUE, R. *Python para todos.* 2008. 115 págs. ⟨http://sunshine.prod.uci.cu/book/4e67b7b305717436a9000007/⟩.

30 *Oracle y Java - Características | Tecnologías | Oracle ES.* 2014. dirección: ⟨http://www.oracle.com/es/technologies/java/features/index.html⟩.

31 MUSCIANO, C. y KENNEDY, B. *HTML. La Guía Completa.* 2004. 546 págs. ⟨http://sunshine.prod.uci.cu/book/4e78ab41057174105600005b/⟩. ISBN 1-56592-235-2.

32 *XML Essentials - W3C.* 2010. dirección: ⟨http://www.w3.org/standards/xml/core⟩.

33 EGUÍLUZ PÉREZ, J. *Desarrollo web ágil con Symfony2.* 2013. 518 págs.

34 CABRERA GONZÁLEZ, L. y POMPA TORRES, E. R. *Extensión de Visual Paradigm for UML para el Desarrollo Dirigido por Modelos de aplicaciones de gestión de información.* 2012. 80 págs. ⟨http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_05815_12⟩.

35 REYES VIADA, E. y JIMÉNEZ RAMÍREZ, C. Y. *Framework para la arquitectura de los sistemas registrales del proyecto Registros y Notarias Fase II.* 2011. 79 págs. ⟨http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_04261_11⟩.

36 ACOSTA, J.; GREINER, C.; DAPOZO, G. y ESTAYNO, M. Medición de atributos POO en frameworks de desarrollo PHP. En. *XVIII Congreso Argentino de Ciencias de la Computación*. 2012, págs. 10. ⟨http://sedici.unlp.edu.ar/handle/10915/23734⟩.

37 MARQUINA, E. *Guía de Patrones, Prácticas y Arquitectura .NET.* 2006. ⟨http://sunshine.prod.uci.cu/gridfs/sunshine/books/PPArquitecturaNET.pdf⟩.

38 *Zend Framework.* 2014. dirección: ⟨http://framework.zend.com/about/⟩.

39 GRIFFTITHS, A. *CodeIgniter 1.7 Professional Development.* 2010. 301 págs. ⟨http://sunshine.prod.uci.cu/gridfs/sunshine/books/1849510903.pdf⟩. ISBN 978-1-849510-90-5.

40 POTENCIER, F. y ZANINOTTO, F. *Symfony. La guía definitiva.* 2008. 425 págs. ⟨http://sunshine.prod.uci.cu/book/4e682816057174073d00004c/⟩.

41 CABRERA RODRÍGUEZ, L. *Sistema de gestión de perfiles de tesis para la facultad 1 de la Universidad de las Ciencias Informáticas.* 2011.

42 *An Introduction to NetBeans.* 2013. dirección: ⟨https://netbeans.org/about/index.html⟩.

43 *NetBeans IDE - Base IDE Features.* 2013. dirección: ⟨https://netbeans.org/features/ide/index.html⟩.

44 GAMMA, E.; HELM, R.; JOHNSON, R. y VLISSIDES, J. *Design Patterns - Elements of Reusable Object Oriented Software.* 1.ª ed., 2005. 431 págs. ISBN 0-201-63361-2.

45 GROSE, T. J.; DONEY, G. C. y BRODSKY, S. A. *Java Programming with XMI, XML and UML.* 2002. 85 págs. ⟨http://www.google.com.cu/books?hl=en&lr=&id=6B5-Wz6WbIIC&oi=fnd&pg=PR5&dq=XMI&ots=JzsCiQpK8u&sig=mNWCZk8UN3ObeFHMjan1dv-pYhA&redir_esc=y#v=onepage&q=XMI&f=false⟩.

46 LEDESMA RODRÍGUEZ, Y. y BOZA ROGET, Y. *Extensión de la herramienta Visual Paradigm for UML para el soporte al Desarrollo Dirigido por Modelos con Ext JS.* 2011. 74 págs. ⟨http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_04358_11⟩.

47 *Web Application Security with Acunetix Web Vulnerability Scanner.* 2015. dirección: ⟨https://www.acunetix.com/vulnerability-scanner/⟩.

48 *Apache JMeter.* 2015. dirección: ⟨http://jmeter.apache.org/⟩.

49 JACK, K. *Digital Video and DSP.: Instant Access*. 2008. ISBN 978-0-7506-8975-5.

50 LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado.* 2da Edición, 2003.

51 LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* 1ra Edición, 1999.

52 FERNÁNDEZ ALARCÓN, V. *Desarrollo de sistemas de información: una metodología basada en el modelado.* 2006. ISBN 86-8301-862-4.

53 S. PRESSMAN, R. *Ingeniería del software. Un enfoque práctico.* 6ta Edición, 2006. ISBN 970-10-5473-3.

54 BAHIT, E. *POO y MVC en PHP. El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC.* 2011. 66 págs. ⟨http://sunshine.prod.uci.cu/book/4eb204eb0571745ff5000007/⟩.

55 S. PRESSMAN, R. *Ingeniería del software. Un enfoque práctico.* 5ta Edición, 2005.

56 S. PRESSMAN, R. *Ingeniería del software. Un enfoque práctico.* 5ta Edición, 2005.

57 IEEE. *Guide to the Software Engineering Body of Knowledge.* 2004. ISBN 0-7695-2330-7.

58 ROBLES, G. y FERRER, J. Programación eXtrema y Software Libre. 2002, págs. 23.

59 OMAÑA, M. y CADENAS, J. Manufactura Esbelta: una contribución para el desarrollo de software con calidad. *Revista Venezolana de Información, Tecnología y Conocimiento*. 2010, págs. 16. ISSN 1690-7515.

60 SOMMERVILLE, I. *Ingeniería del software.* 7ma Edición, 2005. 712 págs. ISBN 84-7829-074-5.

61 HERNÁNDEZ SAMPIER, R. *Metodología de la investigación.* 2008.