

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1

SUBSISTEMA DE RECOMENDACIÓN DE INFORMACIÓN
CON ANOTACIÓN SEMÁNTICA PARA EL BUSCADOR
ORIÓN

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

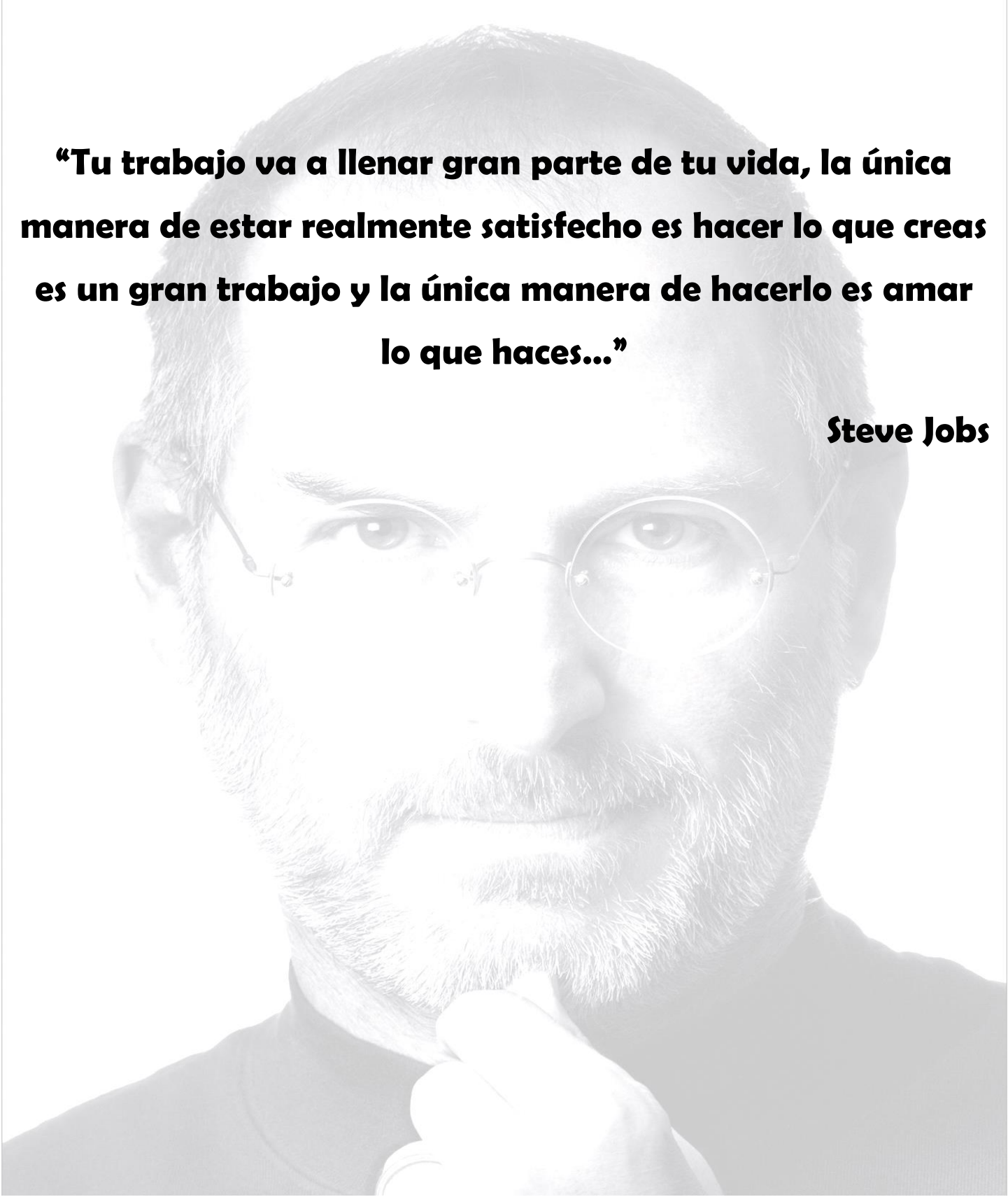
Autor: Julio Miguel López Armenteros

Tutores:

MSc. Dúnnia Castillo Galán

Ing. Eric Bárbaro Utrera Sust

La Habana, junio de 2017



“Tu trabajo va a llenar gran parte de tu vida, la única manera de estar realmente satisfecho es hacer lo que creas es un gran trabajo y la única manera de hacerlo es amar lo que haces...”

Steve Jobs

DEDICATORIA

Dedico este trabajo, fruto del sacrificio y el esmero de tantos años en la enseñanza superior, a mi familia, que tanto me ha apoyado para terminar esta etapa de mi vida. A mi madre que siempre ha estado presente para mí en todo momento, buenos y malos. Pero en especial, se lo dedico a mi padre y a mi abuela que su ausencia física no les permitió ver lo que he logrado hasta ahora y sé que estarían orgullosos de mí.

DECLARACIÓN DE AUDITORÍA

Yo, Julio Miguel López Armenteros, declaro ser el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste, se firma la presente declaración a los ____ días del mes de junio del año 2017.

Autor:

Julio Miguel López Armenteros

Firma del Autor

Tutor:

Msc. Dúnnia Castillo Galán

Firma del Tutor

Tutor:

Ing. Eric Bárbaro Utrera Sust

Firma del Tutor

AGRADECIMIENTOS

A mi familia por su apoyo incondicional.

A mi pareja por su comprensión.

A mis amigos y amigas por darme su amistad.

A mis tutores: Dúnnia y Eric porque este también es su trabajo.

A mis profesores docentes por formarme docentemente y como persona.

A mis brothers de los antiguos 1101 y 1102.

A las tardes de fútbol, y noches, con los colegas.

Y con especial mención a Cari, por pasar tantas noches sin dormir junto a mí.

RESUMEN

Los sistemas de recomendación de información son herramientas que permiten filtrar la información y determinar aquella que concuerde con las preferencias de los usuarios. En la literatura consultada, se comprobó la existencia de sistemas de recomendación basados en tecnologías de la Web Semántica, como son las ontologías y el procesamiento del lenguaje natural. Durante la investigación no se encontró ningún sistema de recomendación de información semántico que se integre a los sistemas de recuperación de información. Atendiendo a esto, se desarrolló un subsistema que integra, al buscador cubano Orión, las ontologías y el procesamiento del lenguaje natural. Para la implementación de la propuesta de solución, guiada por la metodología AUP en su variación para la Universidad de las Ciencias Informáticas, se seleccionaron las herramientas: Solr como mecanismo de indexación, Apache Jena y Protégé para el manejo de las ontologías, Stanford CoreNLP para el procesamiento del lenguaje natural, Symfony como marco de trabajo PHP, Java como lenguaje de programación para el desarrollo del componente y Visual Paradigm for UML como herramienta para el modelado. Como conclusión del proceso se obtuvo un sistema que mejora la experiencia del usuario en el buscador Orión mediante el empleo de tecnologías semánticas. Además, el componente desarrollado es reutilizable, flexible ante los cambios y de fácil mantenimiento.

Palabras clave: recomendación, ontología, sistemas de recomendación semánticos, procesamiento del lenguaje natural, buscadores.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL SUBSISTEMA DE RECOMENDACIÓN DE INFORMACIÓN CON ANOTACIÓN SEMÁNTICA	8
1.1 Introducción	8
1.2 Conceptos asociados al dominio de la investigación.....	8
1.2.1 Información	8
1.2.2 Recuperación de información	9
1.2.3 Sistemas de Recuperación de Información.....	9
1.2.4 Sistemas de Recomendación	10
1.2.5 Semántica	13
1.2.6 Procesamiento del Lenguaje Natural (PLN).....	13
1.2.7 Ontologías.....	14
1.3 Estudio de sistemas de recomendación de información semánticos	14
1.4 Herramientas y Tecnologías.....	18
1.4.1 Mecanismos existentes para la manipulación de bases ontológicas	18
1.4.2 Lenguajes de Programación	19
1.4.3 Marcos de trabajo PHP	20
1.4.4 Metodología de desarrollo	22
1.4.5 Herramientas	22
1.4.6 Bibliotecas disponibles para el PLN.....	24
1.5 Conclusiones parciales	24
CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL SUBSISTEMA DE RECOMENDACIÓN DE INFORMACIÓN CON ANOTACIÓN SEMÁNTICA PARA EL BUSCADOR ORIÓN	26
2.1 Introducción	26
2.2 Descripción de la propuesta de solución	26

2.3 Modelo del Dominio	26
2.4 Especificación de requisitos de Software	28
2.4.1 Requisitos funcionales	28
2.4.2 Requisitos no funcionales.....	29
2.5 Historias de usuario	31
2.6 Arquitectura del sistema.....	35
2.7 Patrones de diseño	36
2.8 Modelo de diseño.....	38
2.8.1 Diagrama de clases.....	38
2.9 Modelo de datos.....	38
2.10 Modelo de despliegue	39
2.11 Conclusiones parciales	40
CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SUBSISTEMA DE RECOMENDACIÓN DE INFORMACIÓN CON ANOTACIÓN SEMÁNTICA PARA EL BUSCADOR ORIÓN	41
3.1 Introducción	41
3.2 Diagramas de componentes.....	41
3.3 Estándares de codificación.....	44
3.4 Validación del Subsistema de Recomendación de Información con Anotación Semántica para el Buscador Orión	45
3.4.1 Pruebas funcionales	45
3.4.2 Pruebas de integración	48
3.4.3 Pruebas de carga y estrés.....	51
3.4.4 Pruebas de seguridad	52
3.4.5 Pruebas de usabilidad.....	53
3.4.6 Validación de la hipótesis	55
3.4.7 Conclusiones parciales	57

CONCLUSIONES.....	58
RECOMENDACIONES.....	59
REFERENCIAS BIBLIOGRÁFICAS	60
ANEXOS.....	67

ÍNDICE DE TABLAS

Tabla 1. Operacionalización de las variables	5
Tabla 2. Sistemas de Recomendación Semánticos	17
Tabla 3. <i>Frameworks</i> PHP (CodeIgniter, 2016; Symfony, 2016; Yii, 2016; Zend, 2016)	21
Tabla 4. Conceptos del modelo del dominio.....	27
Tabla 5. Requisitos funcionales.....	28
Tabla 6. Historia de usuario del RF 1	31
Tabla 7. Historia de usuario del RF 8	33
Tabla 8. Historia de usuario del RF 11	34
Tabla 9. Variables del caso de prueba de HU “Procesar consulta”.....	46
Tabla 10. Caso de prueba “Procesar Consulta”.....	46
Tabla 11. Variables del caso de prueba de HU “Mostrar recomendación”.....	46
Tabla 12. Caso de prueba “Mostrar recomendación”.....	46
Tabla 13. Resultados de las pruebas funcionales	47
Tabla 14. Caso de prueba “Integración componente_procesa_consulta – Interfaz web”	48
Tabla 15. Caso de prueba “Integración Solr – Interfaz web”.....	49
Tabla 16. Resultados de las pruebas de integración	50
Tabla 17. Resultados de las pruebas de carga y estrés	51
Tabla 18. Resultados de las pruebas de seguridad	52
Tabla 19. Candidatos seleccionados para las pruebas de usabilidad	54
Tabla 20. Métricas a evaluar en las pruebas de usabilidad.....	54
Tabla 21. Cantidad de documentos relevantes recuperados	56
Tabla 22. Resultados del indicador "Precisión".....	56
Tabla 23. Resultados del indicador "Exhaustividad"	57
Tabla 24. Anexo 1: Plan de preguntas. Entrevista al Jefe de Proyecto Orión.	67
Tabla 25. Anexo 3: Métricas de exactitud en Sistemas de Recomendación (Viltres, 2014).....	68

ÍNDICE DE FIGURAS

Figura 1. Información generada a nivel mundial, experimental y simulada. (Victoria <i>et al.</i> , 2009).	1
Figura 2. Número total de sitios web [Escala Logarítmica] (Netcraft, 2016).	2
Figura 3. Diagrama del modelo del dominio	27
Figura 4. Arquitectura del Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión	36
Figura 5. Diagrama de clases del diseño RF 7 Procesar Consulta	38
Figura 6. Diagrama del modelo de datos	39
Figura 7. Diagrama de despliegue	39
Figura 8. Diagrama de componentes correspondiente a la interfaz web.....	42
Figura 9. Diagrama de componentes correspondiente al módulo de perfil de usuario.....	43
Figura 10. Diagrama de componentes correspondiente al subsistema de procesamiento de la consulta ...	44
Figura 11. No conformidades detectadas	47
Figura 12. Anexo 2: Arquitectura de Apache Jena (Apache Jena, 2016).....	68

INTRODUCCIÓN

El mercado desarrollo de las Tecnologías de la Información y las Comunicaciones es uno de los factores principales que han influido en el progreso social que experimenta la humanidad. Los avances de las tecnologías han permitido al hombre digitalizar la información, enriqueciendo notablemente el campo del conocimiento. Como consecuencia de estos procesos, el volumen de información crece a una velocidad impensable hasta hace pocos años. Se estima que para 2025 la información generada alcance los 17374 Exabytes (2^{60}), esto se puede ver reflejado en la siguiente gráfica.

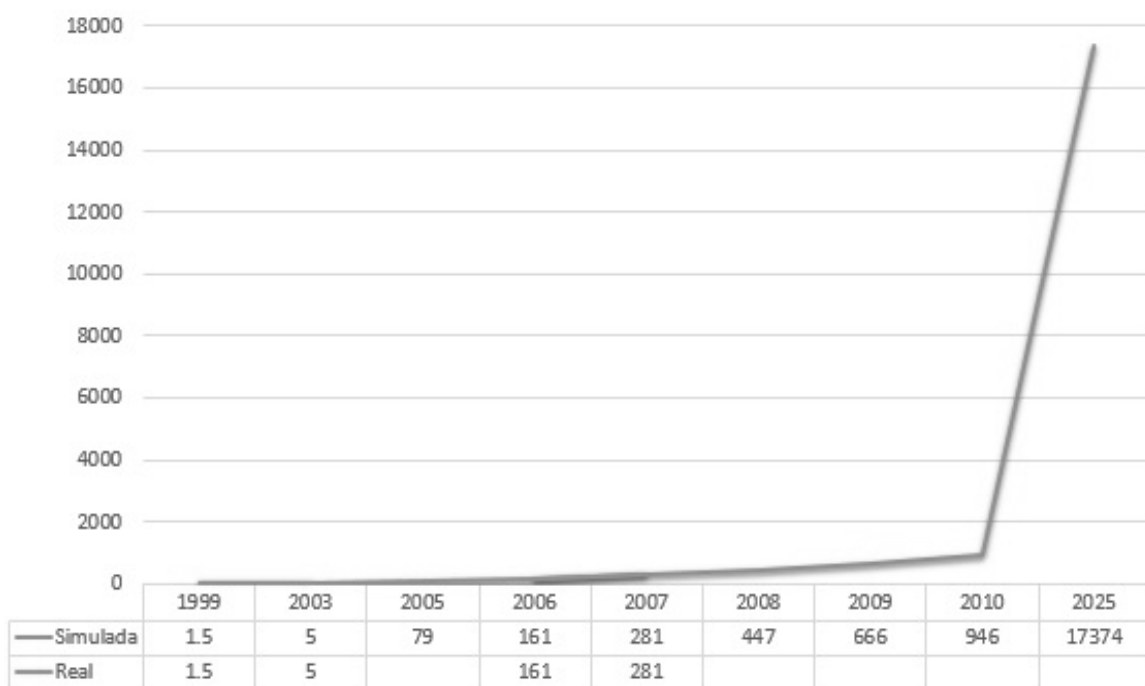


Figura 1. Información generada a nivel mundial, experimental y simulada. (Victoria *et al.*, 2009).

Una evidencia del notable crecimiento del cúmulo de información es la enorme cantidad de sitios web que se pueden encontrar en la Internet, los cuales en septiembre de 2014 ya habían superado la cifra de 1 billón (Internet Live Stats, 2016).

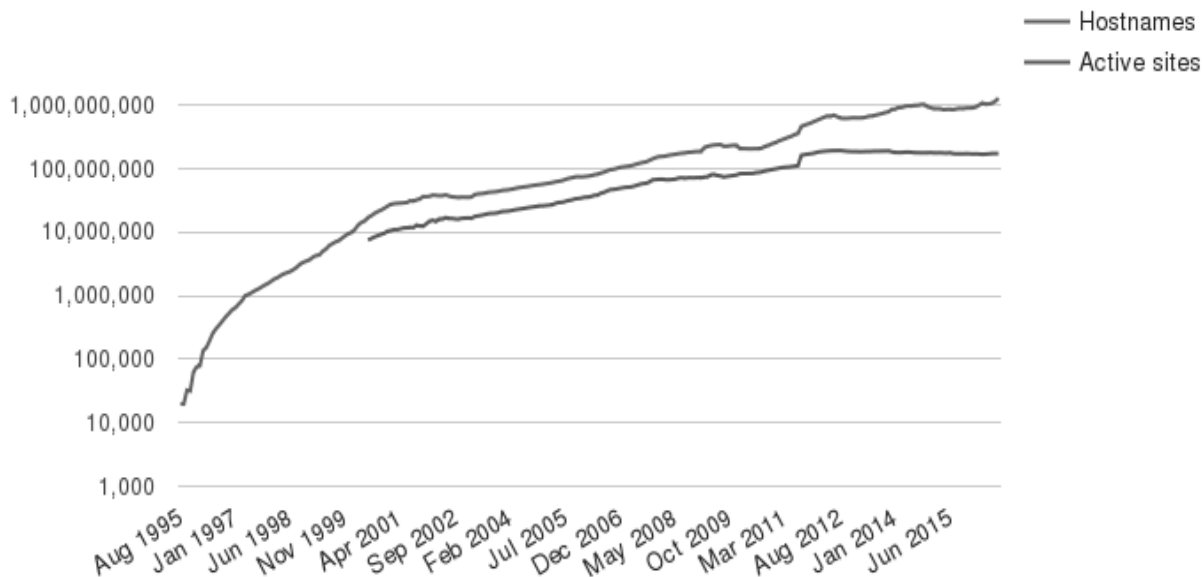


Figura 2. Número total de sitios web [Escala Logarítmica] (Netcraft, 2016).

Como consecuencia de la saturación de información digital, resulta indispensable tener mecanismos que faciliten encontrar determinados datos en Internet. Con tal propósito surgen los Sistemas de Recuperación de Información (SRI), herramientas web que, a partir de una consulta y la utilización de algoritmos de recuperación y de ordenación de resultados, permiten localizar la información requerida. La rapidez de respuesta es un factor esencial en estos sistemas, por lo que su valor depende de la capacidad para identificar rápida y correctamente la información útil, así como, de su facilidad para rechazar los elementos extraños o irrelevantes (Olvera, 1999).

Una de las dificultades detectadas en los SRI es que la respuesta “ideal” es difícil de encontrar y a menudo estos brindan diversas alternativas de elección, obligando a tomar decisiones sin tener la experiencia necesaria, por lo que los usuarios tienden a buscar orientación en otras personas que se han encontrado con su misma necesidad o seleccionan los ítems de información que al parecer más semejanza tienen con su búsqueda.

Por otra parte, existen sistemas informáticos que pueden guiar a los usuarios a encontrar la información de su interés, estos son los Sistemas de Recomendación (SR), herramientas que tienen la función de ayudar en la toma de decisiones dentro de un dominio con grandes cantidades de recursos elegibles. Se puede afirmar que los sistemas de recomendación son un tipo de filtro de información que excluye toda

aquella no deseada de forma automática o semiautomática, con el propósito de mostrar los resultados que sean relevantes o de interés al usuario (Seguido, 2009).

Con la aparición del proyecto Web Semántica o como se le conoce también, Web 3.0, donde la información está dotada de un significado bien definido y permite una mejor colaboración entre humanos y máquinas (Berners-Lee, 2001), los sistemas de recomendación adoptaron también las tecnologías semánticas para analizar las consultas de los usuarios y poder brindar sugerencias más acordes a las preferencias de los mismos. Estas proporcionan una mayor compatibilidad entre sistemas debido a la homogenización de la información, y la misma al contener significado, hace posible inferirla cuando se encuentre incompleta (Seguido, 2009).

Entre las técnicas semánticas que más se utilizan en los Sistemas de Recomendación se encuentran las redes ontológicas. Las mismas funcionan como un medio de comunicación entre los sistemas computacionales y los humanos. Facilitan la reutilización del conocimiento, pues, aunque son desarrolladas con un propósito específico, pueden ser reusadas con otros fines (Abdullah *et al.*, 2012). Las ontologías permiten a los programadores especificar conceptos y relaciones que caracterizan, de forma significativa, ciertos dominios de interés (Jena Ontology API, 2016).

En Cuba, un grupo de desarrolladores del Centro de Ideo-Infornática (CIDI), perteneciente a la Universidad de las Ciencias Informáticas (UCI), creó el buscador cubano Orión, SRI encargado de indexar la información que se encuentra en la web, ponderando los contenidos nacionales. Sin embargo, Orión actualmente no presenta ningún sistema de sugerencia en las búsquedas, dejando en la experiencia de los usuarios la elección de la información a consultar, representado un obstáculo para los investigadores noveles. Por otra parte, debido a que la colección de información que indexa Orión es pequeña y al pobre posicionamiento que tiene la web cubana, los resultados de las búsquedas en ocasiones carecen de precisión y mucha de la información que es mostrada al usuario no es realmente relevante para él, pues Orión calcula la relevancia de la información según las ocurrencias de una palabra o el orden de las mismas, excluyendo por completo la importancia que puede brindar el factor humano a determinada información. Además, el sistema no cuenta con técnicas que identifiquen las preferencias de los usuarios, siéndole imposible diferir entre ellos.

Tomando en consideración la problemática anteriormente descrita, se plantea como **problema de investigación**: ¿Cómo mejorar la calidad de los resultados en las búsquedas realizadas por los usuarios

en el buscador Orión? Definiéndose como **objeto de estudio**: los procesos de Recomendación de Información; haciendo énfasis en los procesos de Recomendación Semánticos como **campo de acción**.

La presente investigación plantea como **objetivo general**: Desarrollar un subsistema de recomendación de información con anotación semántica, utilizando tecnologías semánticas (TS), para mejorar la calidad de los resultados en las búsquedas realizadas por los usuarios en el buscador Orión.

En virtud de solucionar la problemática existente, el objetivo general ha sido desglosado en los siguientes **objetivos específicos**:

- 1) Caracterizar los fundamentos teóricos relacionados con los Sistemas de Recomendación Semánticos.
- 2) Definir las tecnologías, herramientas y metodología de desarrollo para la implementación del Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión.
- 3) Diseñar el Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión.
- 4) Implementar el Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión.
- 5) Validar el Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión.

Luego de haber definido los objetivos primordiales se define la siguiente **hipótesis de investigación**: El desarrollo de un subsistema de recomendación de información con anotación semántica, basado en tecnologías semánticas, mejorará la calidad de los resultados en las búsquedas realizadas por los usuarios en el buscador Orión.

A partir de la hipótesis planteada se define como **variable independiente**: el Subsistema de Recomendación de Información con Anotación Semántica, basado en tecnologías semánticas que permite encontrar los ítems de información que busca el usuario a través de sus preferencias. Como **variable dependiente**: calidad de los resultados en las búsquedas realizadas por los usuarios en el buscador cubano Orión. Esta variable evalúa la calidad (entendida para esta investigación como: la capacidad que posee un objeto para satisfacer necesidades implícitas o explícitas) de los resultados en las búsquedas que brinda el buscador cubano Orión según las preferencias personalizadas de los usuarios.

Tabla 1. Operacionalización de las variables

Variable independiente	Dimensión	Definición	Indicadores	Unidades de medida
Subsistema de Recomendación de Información con Anotación Semántica	Sistema de Recomendación de Información Semántico	Subsistema de Recomendación de Información con Anotación Semántica, basado en tecnologías semánticas que permite encontrar los ítems de información que busca el usuario a través de sus preferencias	Proceso de recomendación de información a partir de un criterio de búsqueda.	Cualitativa
Variable dependiente	Dimensión	Definición	Indicadores	Unidades de medida
Calidad de los resultados en las búsquedas realizadas por los usuarios en el buscador cubano Orión	Calidad	Mejorar la calidad de los resultados en las búsquedas realizadas por los usuarios en el buscador cubano Orión	Cantidad de documentos relevantes recuperados	Unidades
			Precisión	Porcentuales
			Exhaustividad	Porcentuales

Para cumplir con los objetivos específicos se definieron como **tareas de investigación**:

- 1) Realización de un estudio sobre las tendencias de los Sistemas de Recomendación Semánticos.
- 2) Selección de las tecnologías, herramientas y estándares que se necesitan para implementar el Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión.

- 3) Elaboración de los artefactos requeridos por la metodología de desarrollo seleccionada para el buscador Orión.
- 4) Implementación del Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión.
- 5) Realización y documentación de las pruebas que validen el Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión.

Métodos de Investigación utilizados:

Métodos teóricos:

Analítico-Sintético: permitió la recopilación y análisis de la información sobre los elementos esenciales referentes a las teorías, documentos y literatura en general relacionada con los sistemas de recomendación.

Histórico-Lógico: permitió conocer los antecedentes e importancia de cómo han evolucionado en el tiempo los sistemas de recomendación, poniendo especial atención a los que emplean técnicas semánticas, así como las herramientas y tecnologías utilizadas en el desarrollo de aplicaciones web.

Modelación: utilizado en la representación, mediante el uso de diagramas, de las características del sistema, relaciones entre objetos; y las actividades que intervienen en los procesos implementados en el Subsistema de Recomendación de Información con Anotación Semántica para Orión.

Métodos empíricos:

Entrevista: se empleó para obtener información de los Jefes del Proyecto Orión perteneciente a CIDI mediante un modelo estructurado con preguntas predefinidas.

Estructura del documento:

Capítulo 1. Fundamentación teórica de los Sistemas de Recomendación, procesamiento semántico y las aplicaciones web: se presentan un conjunto de conceptos asociados al objeto de estudio de la investigación. Se estudian los Sistemas de Recomendación de Información y se exponen las tecnologías y metodologías a utilizar para el desarrollo del software.

Capítulo 2. Análisis y diseño del Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión: se exponen las características del sistema, patrones de diseño,

requisitos funcionales y no funcionales, arquitectura utilizada y los artefactos que requiere la metodología de desarrollo utilizada.

Capítulo 3. Implementación y validación del Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión: se presentan algunos aspectos asociados a la implementación de la solución informática, así como sus componentes. Además, se exhiben los diseños de casos de prueba a utilizar en la validación del sistema y se analizan los resultados arrojados por las pruebas realizadas.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL SUBSISTEMA DE RECOMENDACIÓN DE INFORMACIÓN CON ANOTACIÓN SEMÁNTICA

1.1 Introducción

Con el objetivo de facilitar la comprensión de la investigación, en el presente capítulo se definen los principales conceptos asociados al objeto de estudio y al campo de acción. Además, se explica qué son los Sistemas de Recomendación Semánticos y se exponen las características esenciales de las herramientas, tecnologías y metodología de desarrollo que se utilizarán para implementar la solución propuesta.

1.2 Conceptos asociados al dominio de la investigación

1.2.1 Información

La información, según (Goñi, 2000), “es el significado que otorgan las personas a las cosas. Los datos se perciben mediante los sentidos, estos los integran y generan la información necesaria para la creación del conocimiento, el cual permite tomar decisiones durante la realización de las acciones cotidianas que aseguran la existencia social”.

Los usuarios, debido a la necesidad de informarse, emplean habitualmente la web en busca de información, entendida esta como un conjunto de datos con un significado, que reduce la incertidumbre y aumenta el conocimiento. Se encuentra disponible para uso inmediato y proporciona argumentos para la toma de decisiones (Chiavenato, 2006).

Por su parte en el diccionario de la Real Academia Española (Real Academia Española, 2016), se pueden encontrar diferentes significados para este término, destacándose entre ellos: “Comunicación o adquisición de conocimientos que permiten ampliar o precisar los que se poseen sobre una materia determinada”.

El concepto de información presenta diferentes acepciones como lo evidencian los brindados anteriormente. Por lo que a partir del estudio de estos el autor del presente trabajo define como información al conjunto de datos con un sentido y propósito bien definido, los cuales se pueden obtener a partir de diversas formas, que favorecen la creación del conocimiento y esclarecen la incertidumbre en el proceso de toma de decisiones.

Es importante resaltar que, aunque se pueden obtener diversidad de datos de todo, estos por si solos no constituyen información. Es necesario un procesamiento de los mismos: recolección y manipulación de los elementos para producir información con un significado y propósito (French, 1996).

1.2.2 Recuperación de información

Con el crecimiento de la web es cada vez más grande el cúmulo de información que se evidencia en la red. Para consultar una temática en específico es necesario contar con mecanismos que posibiliten obtener información sobre la misma, para esto se aplica la Recuperación de Información (RI): “disciplina que estudia la representación, la organización y el acceso eficiente a la información que se encuentra registrada en documentos” (Abadal *et al.*, 2005). También (Korfhage, 2008) definió la RI como “la localización y presentación a un usuario de información relevante a una necesidad de información expresada como una pregunta”.

El proceso de recuperación de información puede ser modelado desde distintos enfoques, estadístico, algebra de Boole, algebra de vectores, lógica difusa, procesamiento de lenguaje natural y otros (*Bordignon et al.*, 2010). Dentro de estos modelos se describen tres clásicos: booleano, probabilístico y vectorial (*Baeza-Yates et al.*, 1999; *Martínez*, 2006), los que son utilizados por los Sistemas de Recuperación de Información para su funcionamiento.

1.2.3 Sistemas de Recuperación de Información

Los Sistemas de Recuperación de Información (SRI) son herramientas que recuperan una lista de documentos que han sido indizados con anterioridad, dándole respuesta a una consulta realizada por el usuario. El objetivo fundamental de estos sistemas es brindar al usuario un conjunto de documentos ordenados por relevancia, para lograrlo esto se combinan diferentes formas de calcular la relevancia de la información. Algunas de las formas más usadas son las etiquetas de juicio y los datos de clic (*Zoeter et al.*, 2011).

Entre los diferentes tipos de SRI, se encuentran los directorios, los metabuscadores y los buscadores. Los directorios presentan las páginas web organizadas por temática y su búsqueda es basada en palabras clave (*Olvera*, 2000). Los metabuscadores son servidores web que realizan búsquedas en muchos buscadores y/o directorios y presentan un resumen de los resultados, eliminando duplicaciones (*Jiménez*, 2005). Los buscadores permiten encontrar sitios web, documentos, archivos, imágenes y cualquier fuente

de información que se requiera, para esto combinan tres componentes principales: web crawlers, generadores de índice y servidores de consulta (Zoeter *et al.*, 2011).

Web Crawlers: rastrea la web haciendo uso de los enlaces entre sitios y envía los sitios web identificados para que posteriormente sean indizados.

Generadores de Índice: incluye información de ocurrencias de un término en una página y los términos del título. Además, almacena otras informaciones como el lenguaje de la página y la cantidad de sitios que enlazan la página en cuestión.

Servidores de consulta: posiciona los documentos indizados en base a cuan parecidos son a los términos de la consulta del usuario.

La cantidad de información que rescatan los SRI puede llegar a ser inmensa, saturando a los usuarios y llevándolos a elegir elementos que no presentan relevancia para él, para mitigar esto se hace uso de los sistemas de recomendación.

1.2.4 Sistemas de Recomendación

Los Sistemas de Recomendación (SR) son técnicas y herramientas de software que sugieren elementos para los usuarios. Estos buscan satisfacer las necesidades de información de los usuarios cuando se enfrentan a un gran cúmulo de esta. Para lograr este objetivo, los SR intentan seleccionar un subconjunto de elementos que mejor concuerden con los gustos y preferencias de los usuarios (Di Noia, 2016).

Existen diversos tipos de SR y aunque difieren en su funcionamiento, todos presentan la necesidad de interactuar con los usuarios para obtener resultados, siendo una de las características fundamentales, su capacidad para obtener retroalimentación (*feedback*). Una forma característica de obtener este *feedback* es cuando los usuarios clasifican los ítems en forma de puntuaciones (*ratings*), por ejemplo, un sistema de clasificación numérico o un sistema de clasificación de 5 estrellas. Sitios comerciales como *Amazon.com* obtienen *feedback* a través a los historiales de navegación de sus usuarios (Aggarwal, 2016).

Los modelos clásicos de sistemas de recomendación funcionan con dos tipos de datos: las interacciones usuario-elemento y los atributos de información sobre el usuario. Algunos de estos modelos clásicos son:

Filtrado Colaborativo: este modelo, para mostrar recomendaciones, se centra en la clasificación colaborativa que proveen múltiples usuarios. Véase el caso de dos usuarios de gustos parecidos, la

clasificación de un elemento por parte de uno de ellos, será, muy probablemente, similar a la clasificación del otro. La mayoría de los modelos de filtrado colaborativo, para mostrar resultados, se centran en las correlaciones inter-elementos o las inter-usuarios, incluso algunos unen ambos tipos de correlaciones. Las primeras buscan lo que le gusta al usuario y entonces encuentra elementos similares. Las segundas buscan usuarios con criterios similares y obtiene que les gusta.

El modelo filtrado colaborativo es, actualmente, uno de los más usados y exitosos métodos de recomendación. Cuando son bien implementados proveen una exactitud y escalabilidad para una gran cantidad de información. Sin embargo, presenta algunos inconvenientes: el sistema desconoce la puntuación de los elementos para un nuevo tema; el número de relaciones conocidas es pequeño comparado con el número de relaciones que deberían ser pronosticadas (Madhushree, 2016).

Basado en Contenido: la idea detrás de este modelo es que, si el usuario prefirió un elemento en el pasado, entonces probablemente preferirá un elemento similar en el futuro (Shah *et al.*, 2016). Los sistemas de *ratings* y comportamiento del usuario en la web son combinados con la información descriptiva del elemento para realizar recomendaciones. A los usuarios con determinadas preferencias sobre alguna clasificación, les serán sugeridos todos aquellos elementos que dentro de su descripción aparezca esa clasificación.

Los sistemas de recomendación basados en contenido tienen ciertas ventajas cuando se realizan recomendaciones sobre elementos nuevos sin un historial de puntuaciones. Sin embargo, si un usuario nunca ha consumido elementos con determinada clasificación, estos no tienen ninguna oportunidad de ser recomendados. Otra desventaja es que, aunque estos modelos son muy efectivos recomendando elementos nuevos, presentan dificultades cuando interactúan con usuarios nuevos, esto se debe a que no poseen un historial sobre él.

Existe una forma de mitigar las desventajas que sufre el modelo ante los nuevos usuarios, el perfil de usuario. Este obtiene las preferencias de los usuarios de dos maneras diferentes, explícitamente (mediante un formulario o cuestionario) o implícitamente (analizando los registros de compra, enlaces visitados, artículos vistos o votados, entre otros) (De Campos *et al.*, 2011).

Basado en Conocimiento: proveen recomendaciones basados en especificaciones explícitas sobre un tipo de contenido. Su principal característica es que fomenta su funcionamiento en una base de

conocimientos. Para mostrar sugerencias, se buscan similitudes entre las especificaciones del usuario y los atributos de información del elemento. Son particularmente efectivos sobre aquellos productos que no son consumidos regularmente: joyas, automóviles de lujo, servicios financieros, etc. Este modelo es muy similar al modelo basado en contenido, la diferencia es que el basado en contenido aprende del historial del comportamiento del usuario en la web, mientras que el modelo basado en conocimientos utiliza las especificaciones de intereses y necesidades de los usuarios (Aggarwal, 2016).

Sistemas de Recomendación Semánticos: Los Sistemas de Recomendación Semánticos son aquellos que utilizan tecnologías de la Web Semántica y establecen su funcionamiento sobre una base de conocimiento, normalmente definida sobre esquemas jerárquicos de conceptos (taxonomías) u ontologías. También se pueden encontrar sistemas que añaden filtros de información adicionales como los sistemas adaptables al contexto y sistemas basados en redes de confianza. En los últimos años se han desarrollado diversos sistemas de recomendación que emplean tecnologías semánticas y proponen soluciones de aplicación en diferentes contextos. (Peis *et al.*, 2008; Seguido, 2009).

Los Sistemas de Recomendación Semánticos presentan diversas clasificaciones, dentro de las mismas, las más marcadas son: los sistemas basados en ontologías, los sistemas basados en redes de confianza y los sistemas sensibles al contexto. Cada uno de estos sistemas utilizan diferentes criterios o herramientas para brindar recomendaciones.

- **Sistemas basados en ontologías:** utilizan ontologías para representar la información o modelado de los elementos. Existen diferentes lenguajes para representar ontologías tales como RDF (*Resource Description Framework*), RDF-Schema (RDFS) y OWL (*Ontology Web Language*).
- **Sistemas basados en redes de confianza:** utiliza redes de confianza entre las partes que componen el sistema. Los usuarios aportan credibilidad a los resultados de las recomendaciones, implicando un aumento de calidad, precisión y usabilidad del sistema.
- **Sistemas sensibles al contexto:** se enfocan en diversos factores para determinar la situación o contexto del usuario y así brindar recomendaciones adaptadas a ese contexto o situación. Los factores pueden ser: temporales, de lugar, experiencia del usuario o dispositivo utilizado (Wang *et al.*, 2015).

Sistemas de Recomendación Híbridos: estos modelos aprovechan los puntos fuertes de otros modelos y los combinan para desarrollar una solución más robusta. Los diferentes modelos anteriormente vistos, para su funcionamiento, explotan diferentes tipos de fuentes de información, estas pueden trabajar bien en varios escenarios. En los casos que hay disponible una amplia variedad de entradas, surge la oportunidad de usar diferentes tipos de SR para realizar una misma tarea. Un Sistema de Recomendación Híbrido flexible y extensible que pueda hacer un buen uso de la información cobra cada vez más importancia (Aggarwal, 2016; Kouki *et al.*, 2015).

En la investigación que se presenta se utilizará un sistema de recomendación semántico atendiendo a las bondades que brinda el procesamiento semántico de la información. Sujeto a esto resulta de vital importancia estudiar las tecnologías semánticas para una mejor comprensión de los sistemas de recomendación que tratan de explotar sus beneficios.

1.2.5 Semántica

La semántica es el estudio del sentido de las palabras y esta participa de tres ciencias distintas, la psicología, la lógica y la lingüística, donde cada una estudia de forma independiente la significación y sentido de los signos. La semántica léxica, semántica por excelencia, estudia las palabras en el seno del lenguaje: su significado, relaciones entre su forma y sentido, y entre una palabra y otra (Guiraud, 1994).

Para (Real Academia Española, 2016), semántica es: “Del gr. σημαντικός *sēmantikós* 'significativo’”. Semántica léxica: “Rama de la semántica que estudia el significado de las palabras, así como las diversas relaciones de sentido que se establecen entre ellas”.

1.2.6 Procesamiento del Lenguaje Natural (PLN)

El Procesamiento del Lenguaje Natural es una de las tecnologías de la Web Semántica estudiada en el campo de las ciencias de la computación y la inteligencia artificial. Se centra principalmente en las interacciones entre las computadoras y el lenguaje natural. Este mitiga la necesidad de procesar la amplia cantidad de información que se encuentra almacenada en lenguaje natural (libros, periódicos, reportes, revistas y mucha más información). Existen 5 fases involucradas en el procesamiento del lenguaje natural: análisis léxico y morfológico, análisis sintáctico, análisis semántico, integración del discurso y análisis pragmático (Chopra *et al.*, 2013).

El Procesamiento del lenguaje natural, NLP (por sus siglas en inglés), se aplica en diferentes disciplinas como la lingüística, informática, telecomunicaciones, inteligencia artificial, incluso psicología y da respuesta a tareas tales como (Alcázar *et al.*, 2013):

- Análisis del lenguaje.
- Reconocimiento del habla.
- Síntesis de voz.
- Traducción automática.
- Respuestas a preguntas.
- Extracción de información.
- Reconocimiento del idioma.
- Análisis de sentimiento.
- Clasificación de documentos.

1.2.7 Ontologías

Muchas aplicaciones de la Web Semántica emplean ontologías como esquemas conceptuales para dar significados a términos usados. La necesidad de las ontologías en las ciencias de la computación viene dada porque, de manera general, las personas usan su lenguaje para comunicarse y crear modelos del mundo que les rodea, pero este lenguaje natural no es adecuado para construir modelos que interprete la computadora, pues es muy ambiguo (Maedche, 2012).

En las dos últimas décadas, esta palabra (ontología) se ha vuelto relevante para la comunidad de Ingeniería del Conocimiento. En este campo, el término se refiere a un modelo de conocimiento restringido a un dominio específico (Albacete, 2016). Una ontología define un vocabulario común para los investigadores que necesitan compartir información en un dominio. Incluye definiciones interpretables por la máquina de los conceptos básicos en el dominio y las relaciones entre ellos (Noy *et al.*, 2001).

1.3 Estudio de sistemas de recomendación de información semánticos

En la actualidad existen en la web diversas aplicaciones que realizan recomendaciones de información sobre un dominio determinado, haciendo uso del procesamiento semántico de la información. Algunas de las aplicaciones que se identificaron son: AVATAR, AFRICA BUILD Portal - *Mobility Brokerage Service*,

IRS-T2D, RAMCAT y FESCADEU (Filtrado espacial, semántico y colaborativo para apoyar decisiones en entornos ubicuos desde ahora).

AVATAR (Blanco-Fernández, 2004)

Es un sistema de recomendación personalizado de contenidos de televisión que combina diferentes estrategias de inferencia del conocimiento con el objetivo de proponer contenidos televisivos acordes a las preferencias de los usuarios. Entre sus características se encuentran:

- Propone una arquitectura multi-agente.
- Basa su recomendación final (G-REC) en el criterio brindado por tres agentes de recomendación: un agente recomendador basado en técnicas bayesianas (B-REC), uno basado en relaciones semánticas (S-REC) y otro basado en coincidencia de perfiles (P-REC).
- Hace uso de perfiles de usuario.
- Utiliza una red ontológica sobre el dominio de la televisión digital.
- El lenguaje utilizado en el trabajo con ontologías es el *Ontology Web Language* (OWL).
- La información sobre los contenidos es obtenida a través del formato TV-Anytime.

AFRICA BUILD Portal - Mobility Brokerage Service (Muiño, 2013)

Es un sistema de recomendación semántico sobre la gestión de recursos humanos (movilidad de estudiantes e investigadores entre instituciones) en el entorno específico de AFRICA BUILD Portal. El sistema de recomendación ayuda en la gestión de recursos humanos en lo referente a la selección de las mejores ofertas y demandas de movilidad. El sistema cuenta con:

- Un enfoque de los sistemas de *Matchmaking* semánticos.
- Una arquitectura modular.
- Un razonador de lógica descriptiva que ofrece inferencias útiles en el cálculo de las recomendaciones.
- Herramientas de procesamiento de lenguaje natural para dar soporte al proceso de recomendación
- Utiliza ontologías para crear las formalizaciones lógicas de las ofertas y demandas; y generar las recomendaciones siguiendo la semántica que establecen.
- El lenguaje utilizado en el trabajo con ontologías es el *Ontology Web Language* (OWL).

IRS-T2D (Mahmoud *et al.*, 2016)

Individualize Recommendation System for Type2 Diabetes (IRS-T2D) es un sistema de recomendación para el tratamiento individual de la diabetes tipo 2. Está desarrollado principalmente para ayudar a los doctores a controlar la enfermedad y evitar los condicionantes de la medicación. El sistema presenta:

- Ontologías para los perfiles de los pacientes y para los medicamentos contra la diabetes.
- *Semantic Web Rule Language* (SWRL) para representar las restricciones de la medicación.
- El sistema recomienda los medicamentos para cada paciente haciendo uso de JESS, motor de reglas de inferencia para la plataforma Java (JESS, 2016).
- El lenguaje utilizado en el trabajo con ontologías es el *Ontology Web Language* (OWL).

RAMCAT (Olivencia *et al.*, 2015)

RAMCAT (Realidad Aumentada Móvil Contextual Aplicada al Turismo) es un SR híbrido que recomienda puntos de interés, teniendo en cuenta factores como preferencias personales y atributos contextuales. Está orientado a turistas que visitan el destino Costa del Sol, España.

El sistema a partir de los intereses y preferencias del turista, selecciona las actividades más adecuadas teniendo en cuenta el contexto en el que se solicita la recomendación. Entre sus características principales aparecen:

- El sistema cuenta con una arquitectura modular.
- Emplea perfiles de usuario.
- Utiliza varios módulos de recomendación; filtrado colaborativo, sensible al contexto y basado en contenido.
- Utiliza una ontología que permite categorizar las distintas actividades.

FESCADEU (González, 2015)

Filtrado espacial, semántico y colaborativo para apoyar decisiones en entornos ubicuos. Es un sistema híbrido que recomienda puntos de interés que puedan interesar a un usuario a partir de sus preferencias. Está compuesto por:

- Un sistema de filtrado sensible al contexto.
- Un sistema de filtrado colaborativo.
- Una ontología donde se representan los puntos de interés.

- Un motor de consultas para RDQL (RDF Data Query Language).

Todos estos sistemas comparten un grupo de características que se pueden apreciar en la siguiente Tabla.

Tabla 2. Sistemas de Recomendación Semánticos

Sistemas	Ontologías	Dominio del Sistema	Herramienta para el trabajo con Ontologías.	Procesamiento del lenguaje natural
AVATAR	Si	Programas Televisivos	Protégé	No
AFRICA BUILD Portal - Mobility Brokerage Service	Si	Gestión de RRHH	Protégé	Si
IRS-T2D	Si	Diabetes Tipo 2	Protégé	No
RAMCAT	Si	Puntos de Interés	Protégé	No
FESCADEU	Si	Puntos de Interés	Protégé	No

Después de haber realizado un análisis exhaustivo sobre los sistemas de recomendación semánticos estudiados, se llega a la conclusión de que todos emplean redes ontológicas para realizar recomendaciones sobre los dominios en los que se desempeñan. Además, se detecta que la cantidad de dominios de la información son escasos y el procesamiento del lenguaje natural es una buena práctica para categorizar de forma automática textos que no entiende la máquina.

Es importante resaltar que durante el proceso de investigación no se encontraron Sistemas de Recomendación Semánticos en buscadores, resultado al que se arriba a partir del estudio realizado sobre los Sistemas de Recomendación en buscadores.

1.4 Herramientas y Tecnologías

Para desarrollar la propuesta de solución, surge la necesidad de estudiar varias tecnologías, lenguajes y herramientas disponibles para dar cumplimiento al objetivo general planteado.

1.4.1 Mecanismos existentes para la manipulación de bases ontológicas

Protégé

Protégé es un *Framework* y editor de ontologías libre y de código abierto, basado en Java, para el desarrollo de sistemas inteligentes. La arquitectura de Protégé permite adaptarse al desarrollo de aplicaciones basadas en ontologías tanto simples como complejas. Cuenta con una interfaz completamente personalizable y herramientas de visualización que permiten la navegación interactiva de relaciones ontológicas. Entre las funcionalidades que brinda se encuentran: unión de ontologías, mover axiomas entre ontologías y renombrado de múltiples entidades. Los desarrolladores cuentan con la ayuda de la vasta comunidad de Protégé y del equipo de Stanford. Protégé soporta completamente las especificaciones OWL 2 y RDF de la W3C (Protégé, 2016).

Apache Jena

Apache Jena es un *Framework* Java libre y de código abierto para la construcción de la Web Semántica y aplicaciones basadas en *Linked Data* (datos entrelazados). Está compuesto por diferentes APIs (*Application Programming Interface*) que interactúan juntas para procesar datos RDF. Provee bibliotecas Java para apoyar el trabajo de los desarrolladores con el manejo de RDF, RDFS, RDFa, OWL y SPARQL de acuerdo a las recomendaciones publicadas por la W3C. Jena incluye un motor de inferencias basado en reglas para realizar el razonamiento utilizando ontologías OWL y RDFS, y una variedad de estrategias de almacenamiento para guardar tripletas RDF (Apache Jena, 2016). La Arquitectura del *Framework* Jena se puede observar en el Anexo 1.

Atendiendo a que las herramientas antes presentadas son libres, de código abierto y desarrolladas en Java, el autor determinó que la herramienta a utilizar para el gestionar (crear, editar) las ontologías sería Protégé por las facilidades que brinda en este aspecto. Sin embargo, la integración del buscador cubano sería con el *Framework* Apache Jena por presentar más facilidades de implementación.

1.4.2 Lenguajes de Programación

Java

Java es un lenguaje de programación orientado a objetos. Surgió en 1995, desarrollado por James Gosling de *Sun Microsystems*. Es definido como un lenguaje de alto nivel, simple, distribuido, multihilo, dinámico, portable, robusto, seguro y de alto rendimiento. El diseño del lenguaje está influenciado por otros lenguajes de programación como C y C++, de los que toma elementos sintácticos, sin embargo, elimina herramientas de bajo nivel de estos. El lenguaje se encuentra vinculado a la plataforma Java, la cual está compuesta por la Máquina Virtual de Java y la API Java (Gosling, 2005).

PHP

Es un lenguaje de código abierto de alto nivel e interpretado, utilizado en su mayoría para el procesamiento dinámico de información en la web. Puede ser incrustado en documentos HTML, pero solo puede ejecutarse en el lado del servidor (Bakken, 2013). La comunidad internacional que posee, colabora en el mejoramiento del código fuente, el desarrollo y actualización de librerías para el lenguaje y la traducción de la documentación del proyecto. Puede ser compilado y ejecutado en diversas plataformas, incluyendo diferentes versiones de Unix, Windows y Mac. Su sintaxis es similar a la del C.

Teniendo en consideraciones que PHP es un lenguaje libre, expandible y cuenta con una gran variedad de módulos, se decide seleccionarlo como lenguaje del lado del servidor a utilizar, mientras que Java se utilizará para el trabajo con el procesamiento del lenguaje natural (PLN), debido a que la plataforma Java cuenta con una serie de APIs populares en este ámbito.

Del lado del cliente

Para el trabajo en la web se utilizará HTML (*HyperText Markup Language*), CSS y JavaScript, todos estos integrados en el *Framework* Bootstrap.

Bootstrap

Bootstrap es un *Framework*, creado por Twitter, permite realizar interfaces web con CSS y JavaScript. Emplea una técnica de diseño conocida como “responsive design” o diseño adaptativo. Esta permite adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Este *Framework* trae varios elementos con estilos predefinidos fácilmente configurables: Botones, Menús desplegables, formularios incluyendo todos sus elementos e integración con jQuery para ofrecer ventanas y *tooltips* dinámicos (Otto

y Thornton, 2013). Bootstrap utiliza HTML, lenguaje de publicación especificado como un estándar por el W3C (*World Wide Web Consortium*), CCS para aplicar diferentes estilos a los documentos creados (W3C, 2016) y JavaScript para brindarle dinamismo a las páginas web (Wright, 2012).

1.4.3 Marcos de trabajo PHP

El concepto de *Framework* se emplea en muchos ámbitos del desarrollo de sistemas de software. En general se hace referencia a una estructura de software compuesta por componentes personalizables para el desarrollo de aplicaciones. Un *Framework* se puede considerar como una aplicación genérica incompleta y configurable a la que es posible añadirle las últimas piezas para construir una aplicación concreta (Gutiérrez, 2006).

CodeIgniter

CodeIgniter es un poderoso *Framework* PHP creado para desarrolladores que necesitan una herramienta simple y elegante para crear potentes aplicaciones web. Presenta elementos que ayudan a mejorar la productividad, mientras que, al mismo tiempo, permite incorporar complementos de terceros que permite adicionar funcionalidades a las aplicaciones web. CodeIgniter es fácil de aprender y permite sacarle un rápido provecho con un excepcional rendimiento (CodeIgniter, 2016).

Symfony

Symfony es un *Framework* PHP de alto rendimiento de código abierto para el desarrollo de aplicaciones web, construido sobre una serie de componente web reutilizables. Posee la característica de que es fácil de instalar y puede ser usado tanto en Linux como en Windows. Symfony tiene una corta curva de aprendizaje y una comunidad de desarrollo comprometida para llevar PHP al próximo nivel. El proyecto tiene más de 2000 contribuidores y más de 300000 desarrolladores (Symfony, 2016).

Yii

Yii es un *Framework* PHP de alto rendimiento para el desarrollo de aplicaciones web. Presenta una corta curva de aprendizaje apoyada en una amplia gama de tutoriales, ejemplos y guías de usuario. La seguridad es un standard con Yii, incluye validación de entradas, filtrado de salida y prevención frente inyección SQL y *Cross-Site Scripting*. Yii está explícitamente diseñado para trabajar eficientemente con AJAX (Yii, 2016).

Zend

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

En la siguiente tabla comparativa se presentan los marcos de trabajo estudiados.

Tabla 3. *Frameworks* PHP (CodeIgniter, 2016; Symfony, 2016; Yii, 2016; Zend, 2016)

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

Como se puede apreciar en la tabla anterior, todos los *Frameworks* de desarrollo comparados se ajustan a las necesidades de desarrollo, pero, se decide seleccionar Symfony en su versión 2.7 porque el buscador web cubano Orión, contiene una interfaz desarrollada con este popular *Framework*, permitiendo incorporar la solución de propuesta de una forma más sencilla y natural.

1.4.4 Metodología de desarrollo

Una metodología de desarrollo de software propone un proceso para la producción organizada de software. Guían, en todo momento, las actividades a realizar para lograr el producto deseado, así como definen el papel a desempeñar por las personas que deben participar en el proceso de desarrollo. Puede incluir modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos (Sommerville, 2005).

Para el desarrollo de la solución se usará AUP en variación de la metodología para la universidad (AUP-UCI), por ser la metodología empleada en el centro CIDI para el desarrollo de proyectos.

AUP-UCI

Es una variación de la metodología de desarrollo AUP diseñada por la UCI. Fue creada con el objetivo de adaptarse al ciclo de vida definido para la actividad productiva de la universidad. Describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles. Define tres fases para componer el ciclo de vida de un proyecto:

- Inicio: se mantiene de AUP, pero se modifica el objetivo; se llevan a cabo las actividades relacionadas con la planeación del proyecto.
- Ejecución: se unifican las tres restantes fases que propone AUP (Elaboración, Construcción y Transición). Se ejecutan las actividades requeridas para obtener el software.
- Cierre: esta fase se incorpora a la metodología en relación a AUP. Se analizan los resultados del proyecto, su ejecución y se llevan a cabo actividades formales de cierre del proyecto.

1.4.5 Herramientas

Visual Paradigm for UML

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

del negocio. La herramienta proporciona abundante documentación en forma de tutoriales, demostraciones iterativas de UML y proyectos UML (Visual Paradigm, 2016).

Entorno de Desarrollo Integrado (IDE) NetBeans

El IDE NetBeans permite un rápido y fácil desarrollo de aplicaciones Java de escritorio y móviles, así como aplicaciones web haciendo uso de HTML, JavaScript y CSS. El IDE también provee para los desarrolladores un gran conjunto de herramientas para PHP y C/C++. Promueve la programación libre de bugs: se integra con herramientas como la popularmente usada *FindBugs Tool*, además el *Debugger* de NetBeans permite analizar el código línea a línea en busca de bugs. Es de código abierto y tiene una amplia comunidad de usuarios y desarrolladores en todo el mundo (NetBeans, 2016).

PostgreSQL

PostgreSQL es un sistema gestor de base de datos relacionales de código abierto y multiplataforma. Surge en 1996 y desde entonces ha evolucionado en cuanto a estabilidad, poder y robustez. Implementa la mayoría de las características que brinda el estándar ISO/IEC 9075:2011, también conocido como SQL:2011 (PostgreSQL, 2017). Su desempeño es mejor en ambientes con altas cargas de usuario y consultas complejas y donde la integridad de los datos es muy importante. En la actualidad es mantenido por *PostgreSQL Global Development Group*, un grupo internacional de compañías y desarrolladores (Lindblom *et al.*, 2015).

JMeter

JMeter es un proyecto de código abierto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el rendimiento. En un principio fue diseñado para probar aplicaciones web, pero se ha expandido hacia otros tipos de servicios. Tiene la capacidad de extraer datos de formatos de respuestas como: HTML, JSON, XML o cualquier formato textual. JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas (The Apache Software Foundation-JMeter, 2016).

Acunetix

Acunetix es una herramienta de seguridad de aplicaciones web. Esta comprueba los sistemas en busca de múltiples vulnerabilidades (por ejemplo, inyección de SQL y *Croos Site Scripting*) que un atacante podría aprovechar para obtener acceso a los sistemas y datos. Acunetix puede escanear todos los sitios web accesibles a través del protocolo HTTP/HTTPS (DSA, 2016). Escanea vulnerabilidades en

aplicaciones web y para introducir pruebas de acceso frente a los problemas identificados. La herramienta provee sugerencias para mitigar las vulnerabilidades identificadas y puede utilizarse para incrementar la seguridad de los sistemas que se analizan (Acunetix, 2016).

1.4.6 Bibliotecas disponibles para el PLN

Apache OpenNLP

El proyecto Apache OpenNLP maneja tareas comunes del PLN. Consiste en una serie de componentes que realizan tareas específicas, permitiendo la formación de modelos y el soporte para pruebas de estos modelos. El enfoque general, utilizado por OpenNLP, es instanciar un modelo que soporta la tarea desde archivo y luego ejecuta métodos contra el modelo para realizar una tarea (Reese, 2015).

Stanford CoreNLP

Stanford CoreNLP es un conjunto de herramientas para las tareas del PLN propuesto por el Grupo de NLP de Stanford. Entre los idiomas que admite se pueden encontrar el inglés, chino y el español. Esta herramienta trabaja bajo la licencia GPL, pero no permite ser usada en aplicaciones comerciales, aunque existe una licencia de carácter comercial (Reese, 2015).

Aunque las herramientas antes presentadas son libres, de código abierto y son APIs para la plataforma Java, el autor determina que la herramienta a utilizar para el procesamiento del lenguaje natural será Stanford CoreNLP, pues presenta modelos de soporte en varios idiomas, entre ellos la lengua española, además de ser una herramienta que goza de gran popularidad.

1.5 Conclusiones parciales

Después de analizar los elementos teóricos que se trataron en este capítulo y que dan sustento a la propuesta de solución del problema planteado, se arriba a las conclusiones siguientes:

- El estudio de los conceptos asociados a la recuperación de información, sistemas de recuperación de información y sistemas de recomendación de información permitió lograr un mejor entendimiento de la investigación que se realiza.
- El análisis de las funcionalidades y características de algunos de los sistemas de recomendación de información, permite identificar que el uso de ontologías y el procesamiento del lenguaje natural son elementos a tener en consideración para dar cumplimiento a los objetivos trazados.

- El estudio realizado en los sistemas de recomendación de información permitió identificar la importancia de gestionar elementos referentes al factor humano, a través de perfiles de usuario y/o acciones del mismo, para obtener resultados más acordes a la búsqueda realizada.
- Las herramientas y tecnologías basadas en software libre, permitieron obtener una base tecnológica enfocada en los componentes que utilizan los sistemas de recomendación semánticos estudiados.

CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL SUBSISTEMA DE RECOMENDACIÓN DE INFORMACIÓN CON ANOTACIÓN SEMÁNTICA PARA EL BUSCADOR ORIÓN

2.1 Introducción

En este capítulo se presentan aspectos relacionados con el diseño del subsistema a desarrollar. Se identifican los requisitos funcionales y no funcionales que deberá presentar el software. Se generan los artefactos relacionados con estos requisitos identificados; así como la modelación de diagramas. En el proceso de diseño de la aplicación se definen los patrones de arquitectura y diseño. A lo largo del capítulo se muestran los principales artefactos de ingeniería de software asociados a los requisitos críticos.

2.2 Descripción de la propuesta de solución

El subsistema contará con:

- Un sistema de perfiles de usuarios que permita identificar los gustos y preferencias de cada usuario.
- Un panel de sugerencias de temas relacionados a las búsquedas realizadas por el usuario.
- Un mecanismo de monitoreo y control de las recomendaciones realizadas a los usuarios.
- Un panel de administración de sugerencias que se le realizan al usuario.
- Seguridad y confiabilidad de los datos que son proporcionados por los usuarios.

Observaciones

Se define, por parte del autor, que el subsistema se desarrolla para el *Core General (Web)* del buscador cubano Orión. No se descarta que se pueda adaptar en un futuro y se pueda emplear en otros *Cores* del buscador.

2.3 Modelo del Dominio

Un modelo del dominio, también llamado modelo conceptual, es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Es importante resaltar que un modelo del dominio no representa componentes de software. No se trata de un conjunto de diagramas que describen clases de software, u objetos software con responsabilidades (Larman, 2004).

Diagrama del Modelo del Dominio

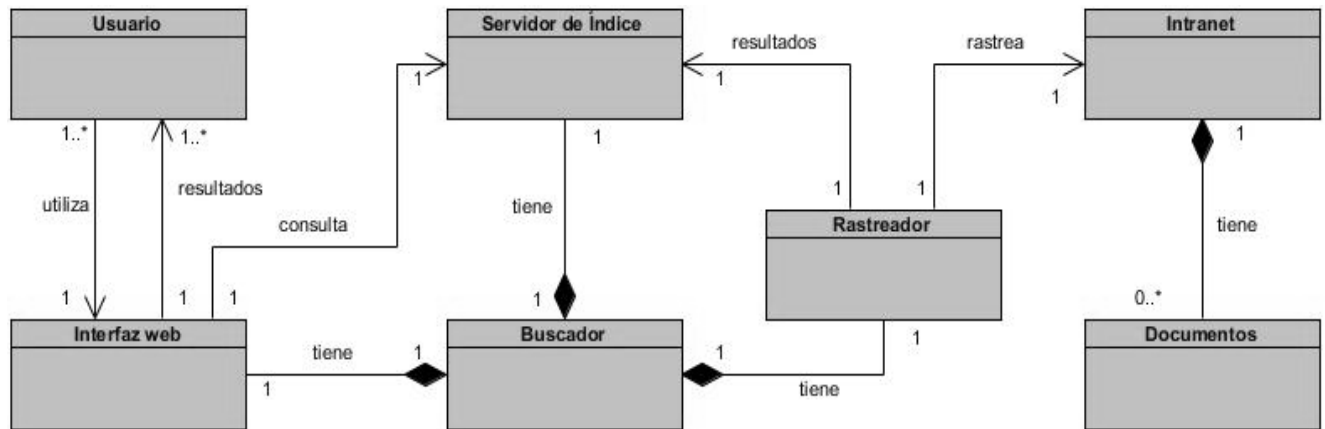


Figura 3. Diagrama del modelo del dominio

Conceptos del Modelo del Dominio

Tabla 4. Conceptos del modelo del dominio

Concepto	Descripción
Buscador	Herramienta de recuperación de información en la web.
Servidor de índice	Herramienta que indexa los documentos.
Rastreador	Herramienta que recopila los documentos en la web.
Interfaz web	Vista con la que interactúa el usuario.
Usuario	Entidad que realiza las búsquedas.
Intranet	Red informática que almacena los documentos.
Documentos	Elementos vigentes en la web (imágenes, videos, páginas web, documentos ofimáticos).

2.4 Especificación de requisitos de Software

Los requisitos del software permiten establecer lo que el sistema debe hacer, sus características fundamentales y las restricciones en el funcionamiento del sistema y los procesos de desarrollo del software. De manera general, estos requisitos expresan las necesidades objetivas que presentan los usuarios, ante un sistema que resuelve un problema en particular de un determinado dominio (Sommerville, 2005).

2.4.1 Requisitos funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer (Sommerville, 2005).

Tabla 5. Requisitos funcionales

No.	Requisito Funcional	Prioridad
RF 1.	Crear perfil de usuario	Alta
RF 2.	Editar perfil de usuario	Medio
RF 3.	Listar perfiles de usuario	Baja
RF 4.	Eliminar perfil de usuario	Media
RF 5.	Similitud entre perfiles de usuario	Alta
RF 6.	Almacenar consulta	Baja
RF 7.	Procesar consulta	Alta
RF 8.	Mostrar recomendación	Alta

RF 9.	Compartir resultado	Alta
RF 10.	Quitar recomendación	Baja
RF 11.	Listar recomendaciones	Media
RF 12.	Ordenar recomendaciones por categoría	Baja
RF 13.	Ordenar recomendaciones por fecha	Baja
RF 14.	Ordenar recomendaciones por usuario	Baja

2.4.2 Requisitos no funcionales

Los requisitos no funcionales son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (Sommerville, 2005).

A continuación, se agrupan los requisitos no funcionales obtenidos por una serie de criterios definidos a conveniencia del autor.

Requerimientos de operaciones

- RNF 1. Se requiere del sistema operativo CentOS 7.
- RNF 2. Se requiere la instalación del servidor web y de servlets Tomcat 7 para el correcto funcionamiento del servidor de Solr.
- RNF 3. Se requiere la instalación de la Máquina Virtual de Java (JVM, por sus siglas en inglés) para el correcto funcionamiento del rastreador.
- RNF 4. Se requiere la instalación del servidor web Nginx en su versión 1.10.3 y PHP 5.5 o superior para poder visualizar la interfaz web.

Requerimientos de hardware

- RNF 5. Para el servidor de índice se necesita como mínimo: 4 GB RAM, CPU de 4 núcleos.
- RNF 6. Para el servidor de rastreo: 4 GB RAM, CPU de 4 núcleos y al menos 80 GB de Disco Duro.
- RNF 7. Para la interfaz web: 4 GB RAM, CPU de 4 núcleos y al menos 40 GB de Disco Duro.

Requerimientos de software

- RNF 8. Los dispositivos clientes que utilizarán la herramienta deben contar con navegadores web que soporten HTML5 y CSS3.

Requerimientos de apariencia e interfaz de usuario

- RNF 9. En la página de resultados siempre estará visible el menú de opciones y el formulario de búsqueda, para resoluciones superiores a 800x600.
- RNF 10. En el formulario de perfil se brindará una descripción textual para cada campo.

Requerimientos de usabilidad

- RNF 11. La plataforma podrá ser empleada por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

Requerimientos de eficiencia

- RNF 12. El sistema debe ser capaz de responder en un tiempo medio de 5 segundos 2000 peticiones de 400 usuarios concurrentes.

Requerimientos de seguridad

- RNF 13. Disponibilidad: el sistema debe permitir una instalación distribuida contribuyendo al balanceo de carga y la redundancia.
- RNF 14. Los formularios deben ser protegidos por un Token de seguridad.
- RNF 15. Los campos de entrada deben ser validados y escapados.
- RNF 18. El sistema contará con diferentes niveles de acceso:
 - ✓ Invitado
 - ✓ Usuario

✓ Administrador

2.5 Historias de usuario

Las historias de usuario son una técnica utilizada en las metodologías de desarrollo para la especificación de requisitos. Una historia de usuario es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario (Suaza, 2013). En estas el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Canós *et al.*, 2003).

Tabla 6. Historia de usuario del RF 1

Historia de usuario Crear perfil de usuario	
Número: 1	Nombre del requisito: Crear perfil de usuario
Programador: Julio M. López Armenteros	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 48h
Riesgo en Desarrollo: Baja	Tiempo Real: 37h
<p>Descripción: Muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none">▪ Nombre(s) (Obligatorio. Campo de texto. Permite caracteres alfabéticos y espacios).▪ Apellidos (Obligatorio. Campo de texto. Permite caracteres alfabéticos y espacios).<ul style="list-style-type: none">▪ Correo electrónico (Obligatorio. Campo de tipo e-mail).▪ Contraseña (Obligatorio. Campo de tipo password. Longitud mínima de 6 caracteres).▪ Confirmar contraseña (Obligatorio. Campo de tipo password. Longitud mínima de 6 caracteres).	

- Preferencias (Opcional. Campos de selección múltiple).
- Recomendaciones (Opcional. Campos de selección múltiple).

Observaciones:

- Al completar el registro, si no se incumplen las condiciones del mismo, el sistema notifica el registro satisfactorio, en caso contrario se muestra un mensaje de error y señalan los campos que incumplen las condiciones requeridas.

Prototipo de interfaz:

Registro

Nombre(s):

Apellidos:

Correo electrónico:

Contraseña:

Confirmar contraseña :

Preferencias:
 Tema 1 Tema 2 Tema 3 Tema 4 Tema 5

Recomendaciones:
 Si No

Tabla 7. Historia de usuario del RF 8

Historia de usuario Mostrar recomendación	
Número: 8	Nombre del requisito: Mostrar recomendación
Programador: Julio M. López Armenteros	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 72h
Riesgo en Desarrollo: Baja	Tiempo Real: 92h
<p>Descripción: En la interfaz de resultados del buscador se mostrará en la parte derecha un panel con las recomendaciones agrupadas por categoría. Al lado de cada categoría se encontrará el botón para quitar sugerencia.</p> <p>Adjunto a cada resultado se muestra el botón de compartir para sugerir el resultado deseado a los demás usuarios que comparten preferencias en sus perfiles (Similitud de perfiles).</p>	
<p style="text-align: center;">Observaciones:</p> <ul style="list-style-type: none"> ▪ Las recomendaciones solo le serán realizadas a aquellos usuarios que especificaron en su perfil que querían recibir recomendaciones. <ul style="list-style-type: none"> ▪ El botón de compartir solo se le mostrará a usuarios registrados. 	
Prototipo de interfaz:	

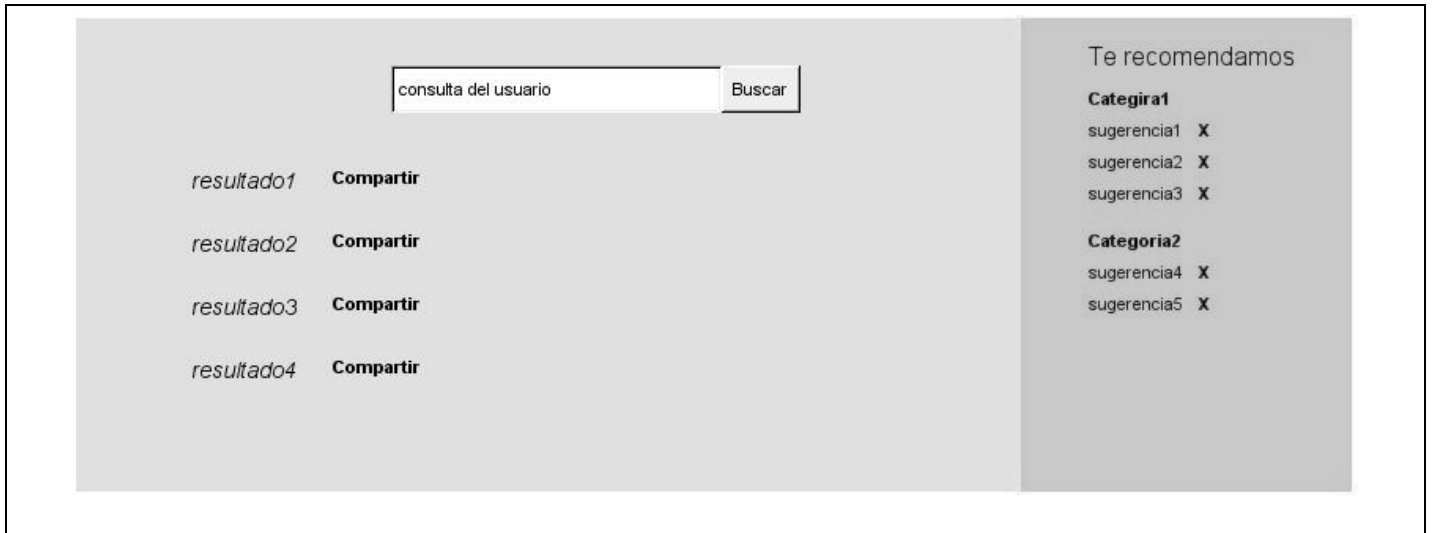


Tabla 8. Historia de usuario del RF 11

Historia de usuario Listar recomendaciones	
Número: 11	Nombre del requisito: Listar recomendaciones
Programador: Julio M. López Armenteros	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 24h
Riesgo en Desarrollo: Baja	Tiempo Real: 3h
<p>Descripción: El sistema muestra un listado con las recomendaciones que ha realizado. De las mismas se presentarán metadatos como:</p> <ul style="list-style-type: none"> ▪ Usuario al que fueron realizadas. ▪ Fecha en que fueron realizadas. ▪ Categoría a la que pertenecen. 	

Observaciones:

- El usuario deberá estar registrado con permisos administrativos.

Prototipo de interfaz:

Administración de Recomendaciones

	Usuario	Categoría	Fecha
recomendacion1	usuario1	categoria1	fecha1
recomendacion2	usuario2	categoria3	fecha2
recomendacion3	usuario1	categoria3	fecha2
recomendacion4	usuario1	categoria2	fecha1
recomendacion5	usuario3	categoria4	fecha3

2.6 Arquitectura del sistema

El sistema que se diseña sigue una arquitectura basada en componentes porque este estilo arquitectónico es para aplicaciones compuestas de componentes individuales. Esta se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor que los principios de orientación por objetos y no se enfoca en asuntos específicos de los objetos como los protocolos de comunicación y la forma como se comparte el estado (Foukalas *et al.*, 2005).

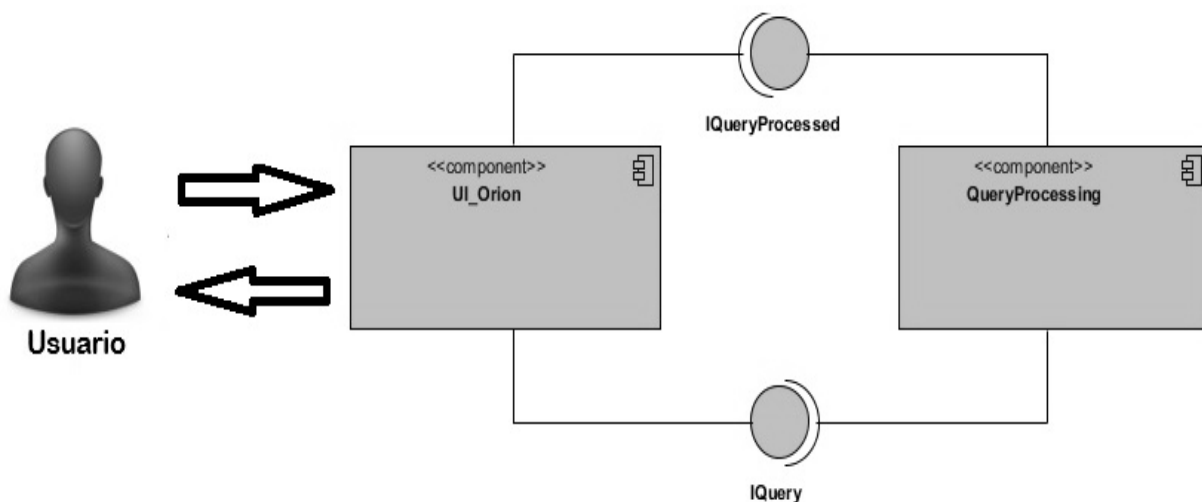


Figura 4. Arquitectura del Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión

Como se observa en la Figura 4, el componente encargado de recibir y atender las peticiones del usuario es “UI_Orion”, el cual interiormente sigue la arquitectura que propone Symfony (Modelo-Vista-Controlador (MVC)) (Potencier, 2011). Cuando el controlador recibe una petición del usuario, este inmediatamente ejecuta el servicio que da respuesta a la solicitud vigente. Entre estos servicios se encuentra el servicio de recomendación de información, el cual es el encargado de proporcionarle al componente “QueryProcessing” los datos necesarios para su funcionamiento, y posteriormente obtener los datos procesados. Este servicio solo se ejecuta para usuarios conocidos (registrados). Los datos son almacenados en el mecanismo de indexación y en la base de datos, la cual es gestionada por un ORM.

2.7 Patrones de diseño

Los patrones de diseño representan la descripción de un problema particular y recurrente, que aparece en contextos específicos y presenta un esquema genérico demostrado con éxito para su solución; este último se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí (Larman, 2004).

En el diseño del Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión se tuvieron en cuenta los siguientes patrones GRASP (Patrones Generales de Software para Asignación de Responsabilidades):

Controlador: este patrón tiene como objetivo asignar la responsabilidad a una clase de recibir o manejar un mensaje de evento del sistema generado por un actor externo, por lo general a través de una interfaz gráfica de usuario a la que accede un usuario para realizar ciertas operaciones en el sistema (Larman, 2004). Este patrón se evidencia en la clase *DefaultController*, encargada de atender y ofrecer respuesta a las peticiones realizadas por el usuario mediante la interfaz web. Esta clase se encuentra en el paquete *Controller*, que se aprecia en el diagrama de clases de la Figura 5.

Experto: este patrón plantea que se debe asignar la responsabilidad de experto en información a las clases que cuentan con los datos necesarios para cumplir la responsabilidad. Conserva el encapsulamiento de la información, puesto que los objetos ejecutan las tareas que le corresponden de acuerdo a la información que poseen (Larman, 2004). Este patrón se puede apreciar en la clase *Query* (Figura 5), clase que tiene la responsabilidad de conocer el contenido de una consulta.

Creador: la instanciación de una clase es una de las actividades fundamentales en un sistema orientado a objetos. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, con lo que se logra menos dependencia y mayores oportunidades de reutilización de código (Larman, 2004). La clase *QueryProcessing* (Figura 5) es la encargada de procesar la consulta y para ello utiliza instancias de las clases *Query* y *Mapping Ontology*.

Alta cohesión: en el diseño orientado a objetos, la cohesión es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento (clase o subsistema). Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que colaboran entre sí y con otros objetos para simplificar su trabajo (Larman, 2004). Este patrón se puede apreciar en la clase *QueryProcessing* (Figura 5), la encargada de procesar las consultas de los usuarios, sin embargo, esta delega responsabilidades en *CoreNLPProcessing* y *MappingOntology*.

2.8 Modelo de diseño

El objetivo fundamental del modelo de diseño es transmitir, mediante diagramas, una comprensión de los aspectos relacionados con los requisitos del sistema. Este modelo sirve de abstracción entre la implementación y el sistema, razón por la cual es muy utilizado en las actividades de implementación (Jacobson *et al.*, 2000).

2.8.1 Diagrama de clases

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software, así como también sus relaciones (Booch *et al.*, 1999). A continuación, se muestra el diagrama de clases relaciona con el requerimiento de procesar consulta RF 7.

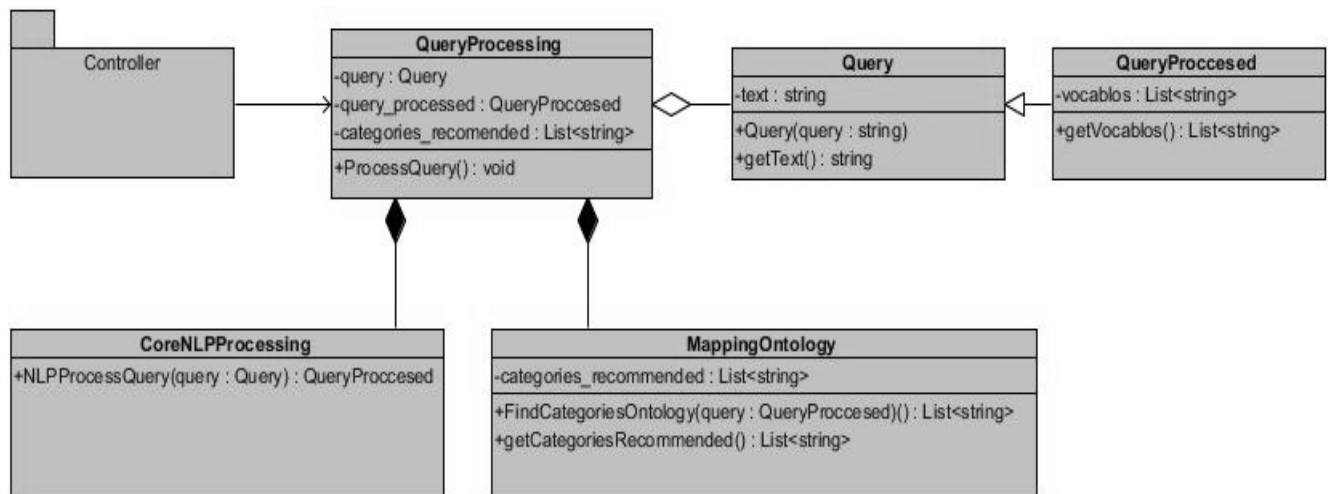


Figura 5. Diagrama de clases del diseño RF 7 Procesar Consulta

2.9 Modelo de datos

Un modelo de datos está orientado a representar los elementos 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 (Pressman, 2010). A continuación, se muestra el modelo de datos que se propone para el desarrollo del sistema.

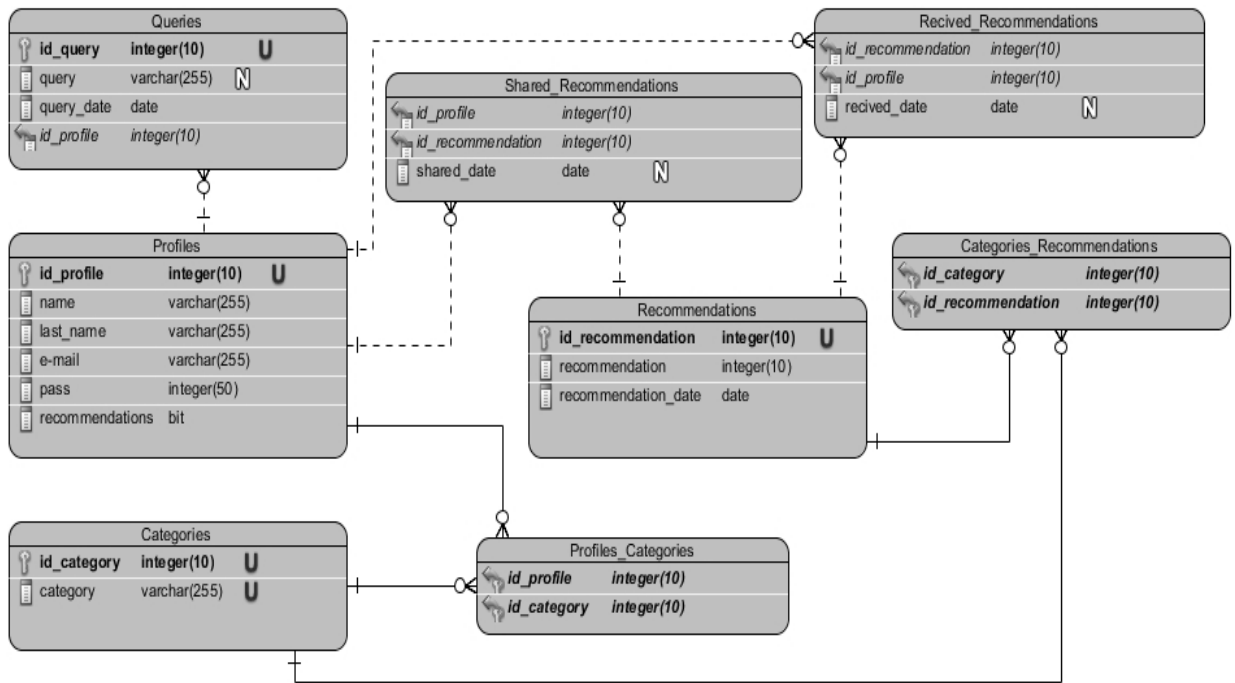


Figura 6. Diagrama del modelo de datos

2.10 Modelo de despliegue

Un modelo de despliegue representa la relación física que se establece entre los distintos componentes o nodos que describen la topología de un sistema. Estos muestran la configuración de los elementos de hardware y muestra cómo los elementos y artefactos del software se relacionan en esos nodos (SparxSystems, 2014).

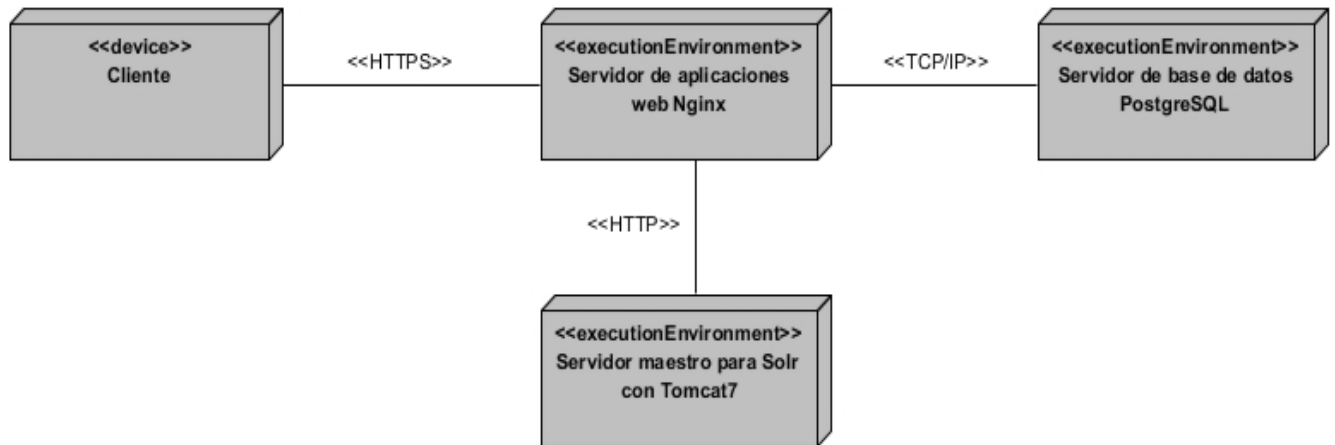


Figura 7. Diagrama de despliegue

Se aprecia en la Figura 7 que, el nodo “Cliente” representa el dispositivo desde el cual el usuario, haciendo uso de un navegador web, podrá realizar las búsquedas. El nodo “Servidor de aplicaciones web Nginx” es el encargado de atender las solicitudes del cliente. Los otros dos nodos representan al servidor maestro para Solr con Tomcat7 como contenedor de servlets y el servidor de base de datos PostgreSQL. La comunicación entre los componentes Cliente y el Servidor de aplicaciones web Nginx, se realiza a través del protocolo HTTPS (puerto 443). En el Servidor de base de datos PostgreSQL y el servidor de aplicaciones la comunicación se establece a través del protocolo TCP/IP (puerto 5432); mientras que la conexión con el Servidor maestro para Solr es mediante el protocolo HTTP (puerto 80). Para el despliegue, se sugiere que se encuentren en servidores independientes aprovechando al máximo las características de hardware y software de estos, aunque pudieran alojarse en uno solo.

2.11 Conclusiones parciales

En desarrollo de este capítulo se trataron aspectos correspondientes al análisis y diseño del Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión, llegándose a las siguientes conclusiones:

- La especificación de los requisitos funcionales y no funcionales del sistema, así como la representación y descripción de los artefactos generados, permitieron una mayor comprensión del negocio y obtener una visión más clara de los resultados que se pretenden obtener.
- La definición de la arquitectura del sistema y los patrones de diseño a utilizar, permitieron establecer buenas prácticas de programación en la fase de implementación, así como disminuir el impacto de los cambios futuros en los componentes del sistema o en el código fuente.
- Se pudo identificar la disposición de hardware y de software de los componentes de la herramienta que se pretende desarrollar.

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SUBSISTEMA DE RECOMENDACIÓN DE INFORMACIÓN CON ANOTACIÓN SEMÁNTICA PARA EL BUSCADOR ORIÓN

3.1 Introducción

La etapa de implementación del software es el proceso de convertir una especificación del sistema en un sistema ejecutable (Sommerville, 2005). Esta fase comprende la materialización, en forma de código, de todos los artefactos, descripciones y arquitectura propuestos en la etapa de análisis y diseño; con el objetivo de conformar el producto final requerido por el cliente (Larman, 2004).

Al tiempo que se desarrolla el software, este debe ser sometido a una serie de pruebas que muestren que el sistema se ajusta a su especificación y que cumple con las expectativas del usuario. Esta etapa es conocida como la validación e implica una serie de procesos de comprobación, como las inspecciones y revisiones (Sommerville, 2005).

3.2 Diagramas de componentes

El diagrama de componentes permite visualizar la estructura de un sistema informático atendiendo a las partes que lo conforman. Cada componente debe ser tratado como como una unidad de composición independiente e indispensable dentro de un sistema, pudiéndose encontrar dependencias entre estos. Algunos ejemplos de componentes físicos lo constituyen los archivos, módulos, librerías, ejecutables y binarios (Sommerville, 2005).

A continuación, se describen los principales paquetes que componen los diagramas de componentes pertenecientes a la interfaz web y al módulo de perfil de usuario de la aplicación.

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

UserProfileBundle: agrupa en su interior todos los componentes del módulo de perfil de usuario.

Controller: contiene las clases controladora encargada de procesar las peticiones de las páginas clientes y del usuario.

Config: contiene los archivos donde se definen las rutas de la aplicación y los servicios disponibles para las vistas.

DependencyInjection: contiene las clases que manejan la inyección de dependencias propias del *Framework Symfony*.

Entity: contiene las clases que se encargan de mapear las tablas de la base de datos.

Form: contiene las clases para construir los formularios utilizados para el envío de información por parte de los usuarios.

Public: en su interior contiene diferentes componentes de apoyo a las vistas de la aplicación (CSS, JavaScripts, imágenes, entre otros recursos).

Translations: contiene las traducciones de los contenidos que visualizan los usuarios.

Twig: Contiene las extensiones del motor de plantillas (Twig) que utiliza el *Framework Symfony*.

Views: agrupa las páginas referentes a las vistas de la aplicación, así como las plantillas bases.

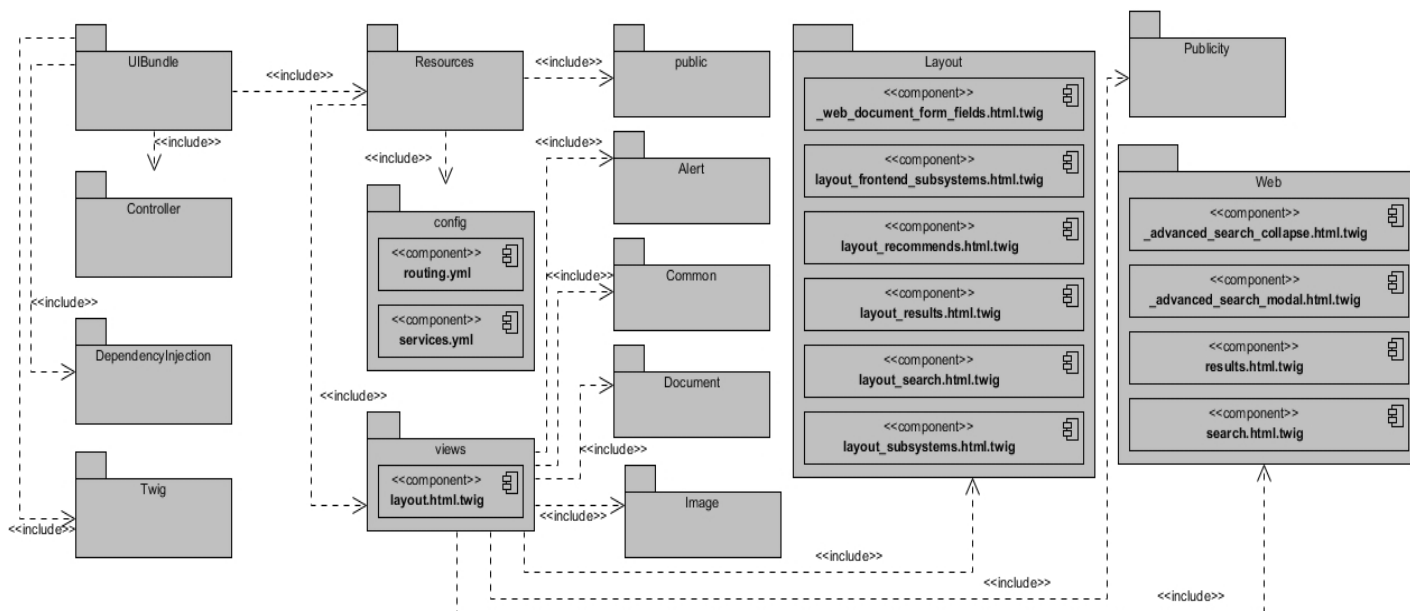


Figura 8. Diagrama de componentes correspondiente a la interfaz web

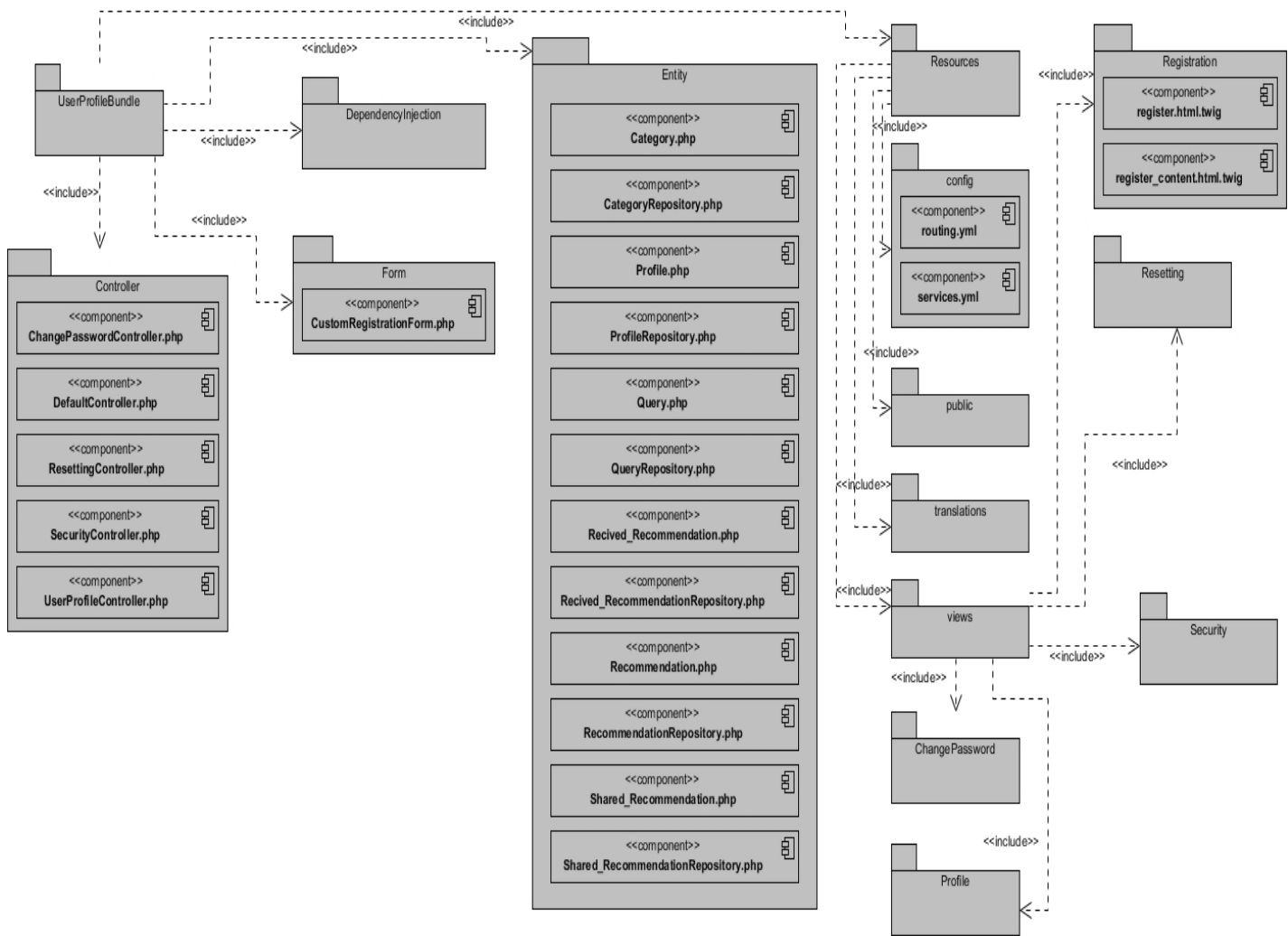


Figura 9. Diagrama de componentes correspondiente al módulo de perfil de usuario

Se describen a continuación los paquetes principales que agrupa el componente de procesamiento semántico de la consulta.

QueryProcessing Package: contiene los componentes asociados al subsistema de procesamiento de la consulta, estableciendo una estructura como la que tienen los proyectos Java.

Apache Jena Package: contiene las bibliotecas que proporciona Apache Jena para el trabajo con las ontologías.

Stanford CoreNLP Package: contiene las bibliotecas proporcionadas por Stanford Natural Language Processing para el trabajo con el procesamiento del lenguaje natural de las consultas.

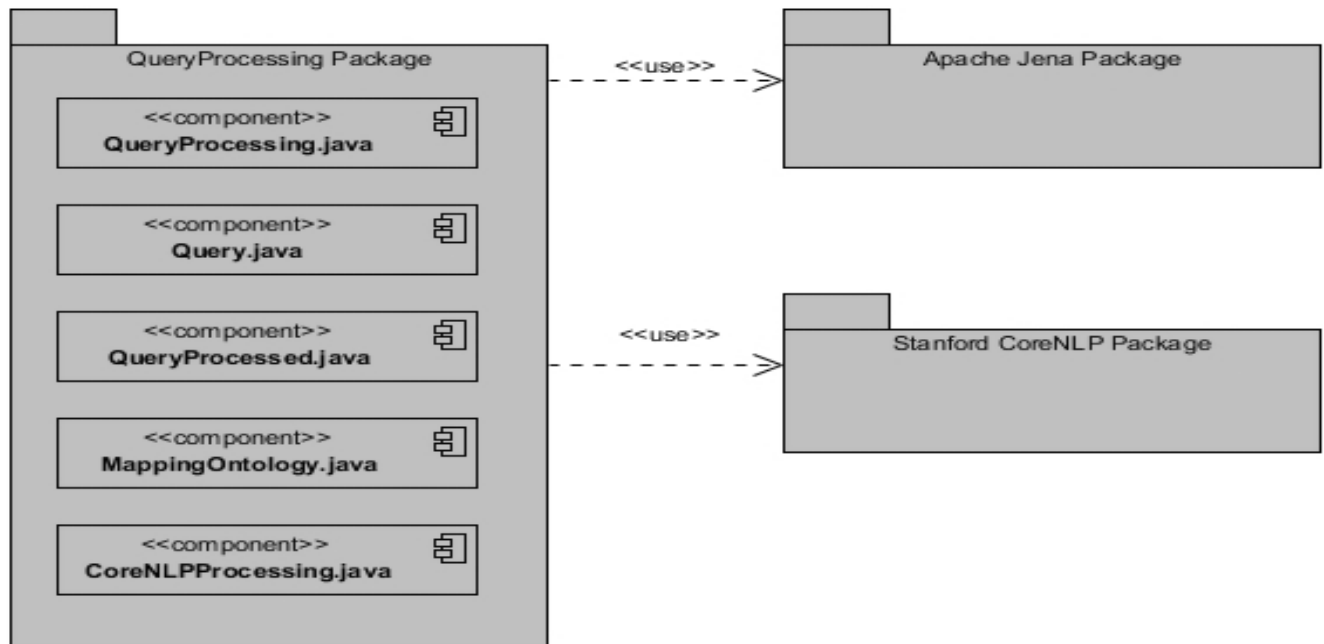


Figura 10. Diagrama de componentes correspondiente al subsistema de procesamiento de la consulta

3.3 Estándares de codificación

Los estándares de codificación son un conjunto de buenas prácticas reconocidas por la industria que proporcionan una variedad de directrices para desarrollar código. El cumplimiento de las normas de codificación en el desarrollo de software puede mejorar la comunicación del equipo, reducir los errores y mejorar la calidad del código (Li *et al.*, 2005). Un estilo de programación homogéneo permite que todos los programadores noveles puedan entender el código en menos tiempo y en consecuencia sea mantenible.

En el lenguaje PHP se utilizó el estándar de codificación que establece el *Framework* Symfony, el cual sigue los estándares definidos en los documentos PSR-0, PSR-1, PSR-2, PSR-4 (SensioLabsNetwork, 2017; PHP-Fig (PSR-0), 2017; PHP-Fig (PSR-1), 2017; PHP-Fig (PSR-2), 2017; PHP-Fig (PSR-4), 2017).

El estándar de codificación que se empleó en el lenguaje JAVA fue el propuesto por ORACLE (Type-Safe, 2014). Entre sus elementos más relevantes y comunes se encuentran:

- Una sentencia por línea de código.
- Los nombres de clases deben ser mezclas de mayúsculas y minúsculas, con la primera letra de cada palabra interna esté en mayúsculas (*CamelCase*).
- La declaración de las variables locales a una clase, método o bloque de código se realizan al principio del mismo y no justo antes de necesitarse la utilización de la variable. La única excepción a esta regla son las variables que gestionan los bucles *for*.
- Son obligatorias las tres condiciones del bucle *for*: inicialización, condición de finalización y actualización del valor de la variable de avance.
- Todo bloque de sentencias entre llaves, aunque sea una sola sentencia después de un *if*.
- El acceso/modificación de las propiedades de una clase (no constantes) siempre mediante métodos de acceso *get/set*.

3.4 Validación del Subsistema de Recomendación de Información con Anotación Semántica para el Buscador Orión

A continuación, se detallan los tipos de pruebas aplicadas al subsistema desarrollado. Las mismas persiguen detectar las no conformidades respecto a las funcionalidades de la aplicación, la medición del grado de usabilidad de las funcionalidades implementadas, así como la correcta integración entre los diferentes componentes de la arquitectura del sistema.

3.4.1 Pruebas funcionales

Las pruebas funcionales se aplican a un software determinado, con el objetivo de validar que las funcionalidades implementadas se desempeñen de acuerdo a las especificaciones de los requisitos definidos con anterioridad. Al conocer las funciones específicas que se le asignaron a una aplicación para su realización, pueden llevarse a cabo casos de pruebas que demuestren que cada función es completamente operativa mientras que al mismo tiempo se buscan errores en cada función (Pressman, 2010).

Tabla 9. Variables del caso de prueba de HU “Procesar consulta”

No.	Nombre la variable	Clasificación	Valor Nulo	Descripción
1	Consulta	Cadena de texto	No	Cadena de texto.

Tabla 10. Caso de prueba “Procesar Consulta”

Escenario	Descripción	V1	R/ del sistema	Flujo central
EC1: Ejecución de una consulta en el componente.	Se ejecuta una consulta en el componente encargado del procesamiento de la misma.	flor	Elementos similares semánticamente al criterio de búsqueda.	Se ejecuta en una Shell el comando “java -jar ruta_componente (V1)”.

Tabla 11. Variables del caso de prueba de HU “Mostrar recomendación”

No.	Nombre la variable	Clasificación	Valor Nulo	Descripción
1	Consulta	Cadena de texto	No	Cadena de texto.

Tabla 12. Caso de prueba “Mostrar recomendación”

Escenario	Descripción	V1	R/ del sistema	Flujo central
EC1: Ejecución de una consulta en el buscador.	Un usuario registrado y con las recomendaciones habilitadas ejecuta una consulta en el buscador.	El perro de Messi	Recomendaciones calculadas según la ontología y las preferencias del usuario.	Un usuario acreditado que tiene habilitado recibir recomendaciones introduce un criterio de búsqueda.

Resultados de las pruebas funcionales

Con el objetivo de probar el correcto desempeño de las funcionalidades del sistema se realizaron las pruebas funcionales. Como resultado de las mismas se detectaron un total de 13 no conformidades (ver figura 11), las cuales se clasifican en funcionales, de validación y errores ortográficos.

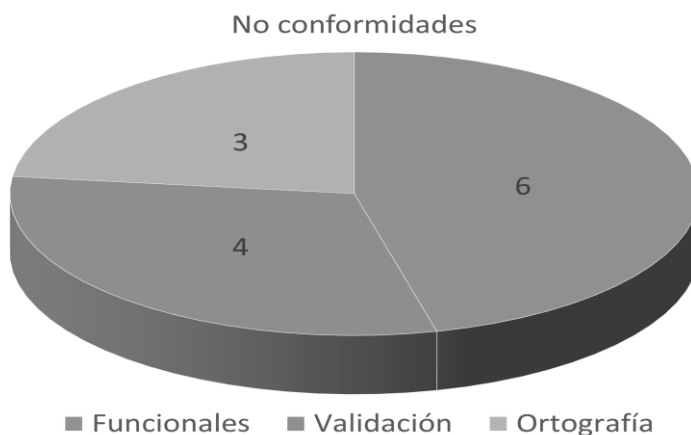


Figura 11. No conformidades detectadas

Durante el proceso se realizaron tres iteraciones de pruebas a la herramienta. En la Tabla 13 se muestran los resultados obtenidos en cada iteración de prueba.

Tabla 13. Resultados de las pruebas funcionales

No conformidades	1ra. Iteración	2da. Iteración	3ra. Iteración
Detectadas	7	4	2
Resueltas	7	4	2
Pendientes	0	0	0

La realización de las pruebas funcionales posibilitó la detección temprana de errores en la aplicación, así como la corrección de cada uno de estos. Entre las faltas más comunes se encontraban los errores ortográficos en la interfaz de usuario y la incorrecta validación de los datos.

3.4.2 Pruebas de integración

Las pruebas de integración comprueban que los componentes del software funcionen juntos y transfieren los datos a través de sus interfaces en el tiempo preciso. Estas pruebas implican construir la aplicación desde sus componentes independientes y probar el sistema resultante en busca de errores que puedan surgir debido a al proceso de integración de los mismos. Los componentes que se integran pueden ser reutilizables o componentes nuevos desarrollados para un sistema en particular (Sommerville, 2005).

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

Para la realización de las pruebas de integración se realizaron una serie de acciones, a continuación, se mencionan las fundamentales:

- Verificación de la conexión del componente encargado de procesar la consulta y la interfaz web del buscador. Ver tabla 14
- Verificación de la conexión de Solr y la interfaz web del buscador. Ver tabla 15

Tabla 14. Caso de prueba “Integración componente_procesa_consulta – Interfaz web”

Caso de prueba “Integración componente_procesa_consulta – Interfaz web”	
Código de caso de prueba	Nombre de historia de usuario:
Nombre de la persona que realiza la prueba: Julio M. López Armenteros.	
Descripción: Se comprueba la correcta interacción entre el componente que se encarga de procesar la consulta y la interfaz web del buscador.	

<p>Entrada/Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Mediante la función exec de PHP se ejecuta el comando "java -jar ruta_componente/QueryProcessingProject.jar Argumento". 2. Se almacena en una variable la respuesta del componente. 3. Se pasa hacia la vista la variable antes definida. 4. Haciendo uso de la etiqueta <i>dump()</i> de Twig se muestra el valor de la variable. <p>ruta_componente: ruta donde se encuentra el componente.</p> <p>Argumento: parámetro que se pasa al componente.</p>
<p>Argumento: flor</p>
<p>Respuesta del sistema: El sistema muestra el contenido de la variable que recibe la vista desde el controlador.</p>
<p>Evaluación de la prueba: Satisfactoria.</p>

Tabla 15. Caso de prueba "Integración Solr – Interfaz web"

Caso de prueba "Integración Solr – Interfaz web"	
Código de caso de prueba	Nombre de historia de usuario:
Nombre de la persona que realiza la prueba: Julio M. López Armenteros.	
Descripción: Se comprueba la correcta interacción entre Solr y la interfaz web del buscador.	
Entrada/Pasos de ejecución:	

<ol style="list-style-type: none"> 1. Haciendo uso del vendor solarium se ejecuta una consulta a Solr. 2. Se almacena en una variable la respuesta de Solr. 3. Se pasa hacia la vista la variable antes definida. 4. Haciendo uso de la etiqueta <i>dump()</i> de Twig se muestra el valor de la variable.
Argumento: Cuba
Respuesta del sistema: El sistema muestra el contenido de la variable que recibe la vista desde el controlador.
Evaluación de la prueba: Satisfactoria.

La tabla 16 describe las no conformidades detectadas durante las pruebas de integración del sistema:

Tabla 16. Resultados de las pruebas de integración

No.	No conformidad	Detectada	Resuelta	Pendiente
1	Conflicto de permisos entre el servidor web y el componente encargado de procesar la consulta	X	X	-

Resultados de las pruebas de integración

La ejecución de las pruebas de integración permitió verificar el trabajo conjunto de los componentes de la herramienta en cuestión. Se realizó especial énfasis en la integración entre la interfaz web, Solr y el componente encargado de procesar la consulta, con el objetivo de detectar incoherencias en el funcionamiento de la aplicación, permitiendo encontrar no conformidades en el proceso. Tras solventar estas no conformidades se llega a la conclusión de que existe una correcta integración entre los componentes internos de la herramienta.

3.4.3 Pruebas de carga y estrés

Las pruebas de carga y estrés se diseñan para enfrentar los programas a situaciones anormales. Evalúan la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremas. Ejecutan un sistema, de manera que demande recursos en cantidad, frecuencia o volúmenes extremos (Pressman, 2010).

Para realizar las pruebas de carga y estrés al Subsistema de Recomendación de Información con Anotación Semántica para el Buscador Orión se utiliza la herramienta Apache JMeter en su versión 3.1, pues resulta necesario comprobar el rendimiento del sistema soportando una cantidad máxima de usuarios que soliciten recursos en la web.

La prueba define 400 hilos de concurrencia, los cuales simulan 400 usuarios accediendo concurrentemente cada "1 segundo". La prueba fue realizada en un servidor Core-i3 a 2.4Ghz de velocidad de procesador, una memoria RAM de 4.00Gb.

Tabla 17. Resultados de las pruebas de carga y estrés

Total	Muestras	Media	Min	Max	%Error	Rendimiento	Kb/s Recibidos
	2000	4891	507	23798	0.00%	11.0/s	22.22

Descripción de los parámetros evaluados:

Muestras: cantidad de hilos utilizados para la URL.

Media: tiempo promedio en milisegundos para un conjunto de resultados.

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

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

%Error: porciento de error de las respuestas de las peticiones.

Rendimiento: rendimiento medido en los requerimientos por segundo / minuto / hora.

Kb/s Recibidos: rendimiento medido en Kbytes por segundo.

De manera general los resultados arrojados son satisfactorios, pues muestran que el servidor, sometido a 2000 peticiones de 400 usuarios concurrentes, atendió, con 0% de error en la respuesta de la URL, en un tiempo medio menor de 5 segundos.

3.4.4 Pruebas de seguridad

Las pruebas de seguridad se diseñan para sondear las vulnerabilidades del lado del cliente, del entorno del lado del servidor, y las comunicaciones que ocurren conforme viajan los datos entre el cliente y el servidor. Cada uno de estos dominios pueden ser atacados, siendo de vital importancia descubrir las debilidades que puedan explotar quienes tengan intención de hacerlo (Pressman, 2010).

La realización de pruebas de seguridad contribuye a la detección temprana de vulnerabilidades y la toma de medidas para la disminución de amenazas de ataque, y con ello proveer sistemas de cómputo más seguros y confiables.

A la herramienta desarrollada se le realizaron una serie de pruebas de seguridad mediante el software Acunetix, las cuales se presentan a continuación:

- Ataques de inyección SQL.
- *Cross-Site Scripting (XSS)*.
- Falsificación de petición (CSRF).

Tabla 18. Resultados de las pruebas de seguridad

Tipo	Cantidad	Descripción
Software	1	La versión de PHP presenta vulnerabilidades de seguridad que han sido resultas en actualizaciones posteriores.
Configuración	3	La configuración del servidor de aplicaciones permite la visualización de información sensible del negocio.

Las vulnerabilidades detectadas a partir de la ejecución de las pruebas de seguridad fueron solventadas y se recomienda que se tenga en cuenta la nueva configuración para un futuro despliegue del sistema en su entorno de producción.

3.4.5 Pruebas de usabilidad

La usabilidad es la capacidad de un producto de software de ser entendido, aprendido, utilizado y atractivo al usuario, cuando es utilizado bajo condiciones especificadas (Pérez, 2006). Es un atributo intangible del software, por lo que es difícil de visualizar, medir y reconocer como un factor determinante de su calidad. Una mayor atención a este elemento contribuiría a incrementar la calidad del producto percibida por el usuario, sin un aumento excesivo en el costo de desarrollo (Mascheroni, 2012).

Las pruebas de usabilidad intentan encontrar problemas de usabilidad del sistema en su interacción con humanos. Según el enfoque tradicional, estas pruebas se aplican sobre el producto software para garantizar o determinar si el mismo alcanza un nivel aceptable de usabilidad. Entre los modelos que presenta para evaluar se encuentran los heurísticos y los empíricos. Los primeros implican la participación de expertos especialistas en usabilidad, mientras que los segundos constan de técnicas que implican la participación de los usuarios (Mascheroni, 2012).

Para la realización de las pruebas de usabilidad a la herramienta se definieron los siguientes pasos:

- Confección de una lista de especificaciones de usabilidad.
- Selección de un conjunto de usuarios.
- Especificación de las métricas a evaluar durante las pruebas.
- Evaluación de la usabilidad del sistema mediante un test de usabilidad.

Para la confección de la lista de especificaciones de usabilidad se toma en consideración algunos de los requisitos funcionales que se identificaron durante el proceso de levantamiento de requisitos de la herramienta.

El proceso de selección de los usuarios que participan en las pruebas se realiza de forma aleatoria. Se seleccionaron 5 individuos de un conjunto de posibles candidatos.

Tabla 19. Candidatos seleccionados para las pruebas de usabilidad

No.	Candidatos	Edad
1	Carlos René Fuentes Herrera	25
2	Liena Hernández Picart	26
3	Carlos Miguel Rodríguez Armenteros	30
4	Lázaro Placencia González	27
5	Alberto Quintana Roig	18

Tabla 20. Métricas a evaluar en las pruebas de usabilidad

Métrica	Descripción	Unidades de medida	Valores
Facilidad de Aprendizaje	Indica qué tan fáciles aprender la funcionalidad básica del sistema.	Cualitativa	Muy Adecuado
			Adecuado
			Inadecuado
Presentación visual apropiada	Indica la correcta selección de colores, tipos de letra, la disposición de los elementos en una ventana, entre otros aspectos de diseño gráfico.	Cualitativa	Muy Adecuado
			Adecuado
			Inadecuado

Resultados de las pruebas de usabilidad

Luego de seleccionados los individuos de pruebas, se sometieron a un test de usabilidad definido para evaluar el Subsistema de Recomendación de Información con Anotación Semántica para el buscador Orión.

Métricas evaluadas	Valores	Frecuencia Absoluta	Frecuencia Relativa
Facilidad de Aprendizaje	Muy Adecuado	13	0.86
	Adecuado	2	0.14
	Inadecuado	0	0
Presentación visual apropiada	Muy Adecuado	11	0.73
	Adecuado	4	0.27
	Inadecuado	0	0

Los resultados obtenidos, tras aplicar las pruebas de usabilidad a la herramienta desarrollada, muestran que, de manera general, la aplicación presenta un grado de usabilidad muy adecuado de acuerdo a las especificaciones requeridas por el cliente. Esto lo corroboran los niveles de satisfacción de los individuos involucrados en las pruebas.

3.4.6 Validación de la hipótesis

Para la validación de la hipótesis planteada en la presente investigación se define un experimento que evalúa los indicadores: cantidad de documentos relevantes recuperados, precisión y exhaustividad (*recall*) en el Subsistema de Recomendación de Información con Anotación Semántica para el Buscador Orión, correspondiente a la variable dependiente definida como parte de la hipótesis de investigación.

Para el experimento se definió 1 colección con 100 documentos indexados, 30 con la categoría de naturaleza y 70 con diversas clasificaciones. Se toma como criterio de búsqueda la consulta “flor” y,

primeramente, se analizan los resultados arrojados por el buscador sin el Subsistema de Recomendación de Información con Anotación Semántica y posteriormente con la herramienta integrada.

Observaciones:

Se considera que un documento relevante no solo es aquel que responde directamente a una consulta determinada, sino también presenta relevancia el documento que presente un criterio similar a la consulta hecha por el usuario. Ejemplo: una flor es una planta, por lo que se infiere que presentan similitud semántica, aportando que a alguien que le gustan las flores, también le gustan las plantas en un porcentaje muy alto.

Tabla 21. Cantidad de documentos relevantes recuperados

Buscador / Indicador	Cantidad
Orión nativo	5
Orión con la herramienta desarrollada	12

Tabla 22. Resultados del indicador "Precisión"

Buscador / Indicador	Factor de relevancia
Orión nativo	0.8
Orión con la herramienta desarrollada	0.75

Tabla 23. Resultados del indicador "Exhaustividad"

Buscador / Indicador	Factor de relevancia
Orión nativo	0.8
Orión con la herramienta desarrollada	0.9

Para interpretar las tablas se debe tomar en cuenta que los resultados arrojados por Orión Nativo no tienen en cuenta la similitud del criterio de búsqueda con otros términos asociados al dominio donde se relacionan, lo que provocó que los resultados del indicador "Precisión" fueran ligeramente superior. Por lo que, teniendo en cuenta lo anteriormente explicado, se evidencia que utilizando la herramienta desarrollada se obtienen resultados con mayor calidad de relevancia para el usuario con respecto a Orión, siendo esto un resultado satisfactorio que apoya la hipótesis de la presente investigación.

3.4.7 Conclusiones parciales

En este capítulo se abordaron aspectos correspondientes a la implementación y validación del Subsistema de Recomendación de Información con Anotación Semántica para el Buscador Orión, llegándose a las siguientes conclusiones:

- Mediante la representación de los diagramas de componentes se facilita la comprensión de la estructura general de la herramienta a través de sus componentes físicos.
- La implantación del código fuente de la herramienta es mucho más clara y comprensible gracias a la utilización de estándares de codificación asegurando mayor calidad y facilitando el mantenimiento.
- La ejecución de pruebas a la herramienta permitió detectar y corregir deficiencias presentes en la solución y ofrecer una aplicación con mayor calidad, seguridad y usabilidad.

CONCLUSIONES

Finalizada la presente investigación, se arriba a las siguientes conclusiones:

- A partir del estudio realizado de los fundamentos teóricos relacionados con los sistemas de recomendación de información semánticos se determinó que, a pesar de la existencia de una serie de herramientas de este tipo, durante la investigación no se pudieron encontrar en buscadores web. Sin embargo, se detectaron funcionalidades a tener en cuenta para confeccionar una propuesta de solución de acuerdo a las necesidades existentes.
- El enfoque propuesto por la metodología AUP en su variación para la universidad y el uso de las tecnologías y herramientas seleccionadas, permitieron analizar los procesos que se debían implementar, estableciéndose una correcta correspondencia entre las especificaciones del cliente y las características que debía tener la herramienta.
- Una vez detectados los aspectos que intervienen en el proceso de recomendación de información utilizando tecnologías semánticas, fue posible la modelación de los artefactos que contribuyeron al diseño de la propuesta de solución posibilitando el soporte a la implementación de la herramienta.
- Llevar a cabo la implementación del Subsistema de Recomendación de Información con Anotación Semántica para el Buscador Orión utilizando las herramientas, las tecnologías y la metodología de desarrollo seleccionada, permitió dar solución a la problemática existente.
- La realización de pruebas al Subsistema de Recomendación de Información con Anotación Semántica para el Buscador Orión permitió mitigar las insuficiencias detectadas, logrando así un producto más seguro y funcional conforme a las necesidades de los usuarios finales.
- La validación de la hipótesis de la investigación, a través de un experimento, demostró la calidad de la herramienta desarrollada, obteniéndose resultados satisfactorios para el autor del presente trabajo.

RECOMENDACIONES

Una vez finalizada la investigación y el desarrollo de la propuesta de solución, el autor del presente trabajo recomienda:

- Incrementar y perfeccionar los dominios de información para aumentar la calidad de la red ontológica y poder brindar recomendaciones más precisas a los usuarios.
- Añadir funcionalidades a la herramienta que permitan perfeccionar el procesamiento de la consulta para lograr mejores resultados en el mapeo con la ontología.
- Integrar Subsistema de Recomendación de Información con Anotación Semántica en el entorno de Red Cuba.

REFERENCIAS BIBLIOGRÁFICAS

ABADAL, Ernest; CODINA, Lluís. *Recuperación de Información. Bases de Datos Documentales: Características, funciones y método*, 2005, p. 29-92.

ABDULLAH, Norris Syed; SADIQ, Shazia; INDULSKA, Marta. *A Study of Ontology Construction: The Case of. Ontology-Based Applications for Enterprise Systems and Knowledge Management*, 2012, p. 276.

ACUNETIX. *Audit your website security with Acunetix Web Vulnerability Scanner*. [En línea]. Acunetix, 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<http://www.acunetix.com/>].

AGGARWAL, Charu. *Recommender Systems: The Textbook*. Springer, 2016.

ALBACETE GARCÍA, Esperanza. *An ontology for human-like interaction systems*. [Tesis]. Universidad Carlos III de Madrid. Madrid. 2016.

ALCÁZAR JAÉN, Samuel, *et al.* *Diseño e implementación de un sistema para el análisis y categorización en Twitter mediante técnicas de clasificación automática de textos*. [Tesis]. Universidad Carlos III de Madrid. Madrid. 2013.

APACHE JENA. *Apache.org*. [En Línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<https://jena.apache.org/>].

BAEZA-YATES, Ricardo, *et al.* *Modern information retrieval*. New York: ACM press, 1999.

BAKKEN, Stig Sæther, *et al.* *Manual de PHP*. s.l.: The PHP Documentation Group, 2013.

BOOCH, Grady, *et al.* *El lenguaje unificado de modelado*. Addison-Wesley, 1999.

BORDIGNON, Fernando; TOLOSA, Gabriel. *Recuperación de información: un área de investigación en crecimiento*. Telematique, 2010, vol. 6, no 1, p. 51-73.

BERNERS-LEE, Tim, *et al.* *The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*. Scientific American, 2001, vol. 284, no 5, p. 1-5.

- BLANCO-FERNÁNDEZ, Yolanda, *et al.* *AVATAR: An advanced multi-agent recommender system of personalized TV contents by semantic reasoning*. En: International conference on web information systems engineering. Springer Berlin Heidelberg, 2004. p. 415-421.
- CANÓS, José H.; LETELIER, Patricio; PENADÉS, M^a Carmen. *Metodologías ágiles en el desarrollo de software*. Metodologías Ágiles en el Desarrollo de Software, 2003, vol. 1, no 10, p. 1-8.
- CHIAVENATO, Idalberto. *Introducción a la Teoría General de la Administración*. s.l.: McGraw-Hill, 2006. 589p.
- CHOPRA, Abhimanyu; PRASHAR, Abhinav; SAIN, Chandresh. *Natural language processing*. International Journal of Technology Enhancements and Emerging Engineering Research, 2013, vol. 1, no 4, p. 131-134.
- CODEIGNITER. *CodeIgniter Web Framework*. [En Línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<https://www.codeigniter.com/>].
- DE CAMPOS, Luis M., *et al.* *Uso de conocimiento estructurado en un sistema de recomendación basado en contenido*. Boletín de la Sociedad Cubana de Matemática y Computación, 2011, p. 1-6.
- DI NOIA, Tommaso. *Recommender Systems Meet Linked Open Data*. En International Conference on Web Engineering. Springer International Publishing, 2016. p. 620-623.
- DSA. *Acunetix*. [En línea], 2016. [Consultado el: 2 de diciembre 2016]. Disponible en: [<http://dsav.net/wp-content/uploads/2013/02/Acunetix-WVS-castellano.pdf>].
- FOUKALAS, Fotis, *et al.* Protocol reconfiguration using component-based design. En IFIP International Conference on Distributed Applications and Interoperable Systems. Springer Berlin Heidelberg, 2005. p. 148-156.
- FRENCH, Carl. *Data Processing and Information Technology*. Cengage Learning EMEA, 1996.
- GONZÁLEZ SUÁREZ, Guillermo, *et al.* *Filtrado espacial, semántico y colaborativo para apoyar decisiones en entornos ubicuos*. Revista Cubana de Ciencias Informáticas, 2015, vol. 9, no 2, p. 66-80.
- GOÑI CAMEJO, Ivis. *Algunas reflexiones sobre el concepto de información y sus implicaciones para el desarrollo de las ciencias de la información*. Acimed, 2000, vol. 8, p. 201-207.

- GOSLING, James, *et al.* *Java (TM) Language Specification, The (Java Series)*. 2005.
- GUIRAUD, Pierre; HASLER, TR Juan A. *La semántica*, Volumen 143. Breviarios del Fondo de Cultura Económica, 1994. 142p.
- GUTIÉRREZ, Javier J. *¿Qué es un framework Web?*. [En línea], 2006. [Consultado el: 2 de diciembre 2016]. Disponible en: [http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf].
- INTERNET LIVE STATS. *Total Number of Websites*. [En Línea] 2016. [Consultado el: 19 de octubre 2016.]. Disponible en: [<http://www.internetlivestats.com/total-number-of-websites/>].
- JACOBSON, IvarBooch, *et al.* *El proceso unificado de desarrollo de software/The unified software development process*. Pearson Educación. 2000.
- JENA ONTOLOGY API. *Ontology*. [En Línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<https://jena.apache.org/documentation/ontology/>].
- JESS. *Jess, the Rule Engine for the Java Platform*. [En Línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<http://www.jessrules.com/jess/docs/>].
- JIMÉNEZ, Pilar María Moreno. *Estrategias y mecanismos de búsqueda en la web invisible*. [En Línea] 2005. [Consultado: 3 de noviembre 2016] Disponible en: [http://biblio.colmex.mx/recelec/web_invisible.htm].
- KORFHAGE, Robert Roy. *Information storage and retrieval*. Wiley. 2008. 368 p.
- KOUKI, Pigi, *et al.* *HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems*. En Proceedings of the 9th ACM Conference on Recommender Systems. ACM, 2015. p. 99-106.
- LARMAN, Craig. *UML y Patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda. s.l.: Prentice Hall, 2004. pág. 520.
- LI, Xiaosong; PRASAD, Christine. *Effectively teaching coding standards in programming*. En Proceedings of the 6th conference on Information technology education. ACM, 2005. p. 239-244.
- LINDBLOM, Martin; STRANDIN, Fredrik. *Performance analysis and improvement of PostgreSQL*. [Tesis]. Department of Computer Science, Lund University. 2015.

- MADHUSHREE, Basavarajaiah. *A Novel Research Paper Recommendation System*. Technology, 2016, vol. 7, no 1, p. 07-16.
- MAEDCHE, Alexander. *Ontology learning for the semantic web*. Springer Science & Business Media, 2012.
- MAHMOUD, Nesma; ELBEH, Heba. *IRS-T2D: Individualize Recommendation System for Type2 Diabetes Medication Based on Ontology and SWRL*. En: Proceedings of the 10th International Conference on Informatics and Systems. Giza, Egypt: INFOS 2016, 2016, p 203-209.
- MARTÍNEZ COMECHE, Juan Antonio. *Los modelos clásicos de Recuperación de información y su vigencia*. En: III Seminario Hispano-Mexicano de investigación en Bibliotecología y Documentación. México DF. 2006.
- MASCHERONI, Maximiliano Agustín, et al. *Calidad de software e ingeniería de usabilidad*. En: XIV Workshop de Investigadores en Ciencias de la Computación. 2012.
- MUIÑO MIGALLÓN, Carlos. *Sistema de recomendación semántico para la movilidad en el entorno AFRICA BUILD Portal*. [Tesis]. Universidad Politécnica de Madrid. Madrid. 2013.
- NETBEANS. *NetBeans IDE*. [En Línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<https://netbeans.org/>].
- NETCRAFT. *September 2016 Web Server Survey*. [En Línea] 2016. [Consultado el: 19 de octubre 2016.]. Disponible en: [<https://news.netcraft.com/archives/2016/09/19/september-2016-web-server-survey.html>].
- NOY, Natalya, et al. *Ontology development 101: A guide to creating your first ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. 2001.
- OLIVENCIA, José Luis Leiva; PLAZA, Antonio Guevara; JIMENEZ, Carlos Rossi. *RAMCAT: Modelo para generar recomendaciones en un sistema de realidad aumentada contextual basándose en las preferencias del turista*. PASOS. Revista de Turismo y Patrimonio Cultural, 2015, vol. 13, no 3, p. 649-668.

OLVERA LOBO, M^a Dolores. *Evaluación de sistemas de recuperación de información: aproximaciones y nuevas tendencias*. Profesional de la Información, El: Information World en Español, 1999, vol. 8, no 11, p. 4-14.

OLVERA LOBO, M^a Dolores. *Rendimiento de los sistemas de recuperación de información en la world wide web: revisión metodológica*. Revista española de documentación científica, 2000, vol. 23, no 1, p. 63-77.

OTTO, Mark; THORNTON, Jacob. *Bootstrap 3, el manual oficial*. [En Línea] 2013. [Consultado el: 2 de diciembre de 2016.]. Disponible en: [http://librosweb.es/libro/bootstrap_3/].

PEIS, Eduardo; MORALES-DEL-CASTILLO, José Manuel; DELGADO-LÓPEZ, José Andrés. *Sistemas de Recomendación Semánticos. Un análisis del estado de la cuestión*. Hipertext.net, 2008, vol. 6.

PÉREZ, Beatriz. *Proceso de testing funcional independiente*. [Tesis]. Facultad de Ingeniería, Universidad de la República. Montevideo, Uruguay. 2006.

PHP FRAMEWORK INTEROP GROUP (PSR-0). *PSR-0: Autoloading Standard*. [En línea]. www.phpfig.org, 2016. [Citado el: 21 de marzo de 2017.] Disponible en: [<http://www.php-fig.org/psr/psr-0/>].

PHP FRAMEWORK INTEROP GROUP (PSR-1). *PSR-1: Basic Coding Standard*. [En línea]. www.phpfig.org, 2016. [Citado el: 21 de marzo de 2017.] Disponible en: [<http://www.php-fig.org/psr/psr-1/>].

PHP FRAMEWORK INTEROP GROUP (PSR-2). *PSR-2: Coding Style Guide*. [En línea]. www.phpfig.org, 2016. [Citado el: 21 de marzo de 2017.] Disponible en: [<http://www.php-fig.org/psr/psr-2/>].

PHP FRAMEWORK INTEROP GROUP (PSR-4). *PSR-4: Autoloader*. [En línea]. www.phpfig.org, 2016. [Citado el: 21 de marzo de 2017.] Disponible en: [<http://www.php-fig.org/psr/psr-4/>].

POTENCIER, Fabien. *What is Symfony2?*. [En línea]. SensioLabsNetwork, 2011. [Citado el: 20 de febrero de 2017.]
Disponible en: [<http://fabien.potencier.org/article/49/what-is-symfony2>].

POSTGRESQL. *Appendix D. SQL Conformance*. [En línea]. PostgreSQL: Documentation, 2017. [Citado el: 22 de mayo de 2017.]
Disponible en: [<https://www.postgresql.org/docs/9.4/static/features.html>].

- PRESSMAN, Roger S. *Ingeniería del software. Un enfoque práctico*. 7ta Edición, 2010.
- PROTÉGÉ. *Protégé Stanford*. [En Línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<http://protege.stanford.edu/>].
- REAL ACADEMIA ESPAÑOLA. *Diccionario de la lengua española*. [En línea]. Real Academia Española, 2016. [Consultado el: 14 de noviembre de 2016.]. Disponible en: [<http://dle.rae.es>].
- REESE, Richard M. *Natural language processing with Java*. Birmingham. Packt Publishing Ltd, 2015. 262 p.
- SEGUIDO FONT, Miguel. *Sistemas de recomendación para webs de información sobre la salud*. [Tesis]. Universidad Politécnica de Cataluña. Barcelona. 2009.
- SENSIOLABSNETWORK. *Coding Standards*. [En línea] SensioLabs Product, 2016. [Citado el: 21 de marzo de 2017.]. Disponible en: [<http://symfony.com/doc/current/contributing/code/standards.html>].
- SHAH, Lipi; GAUDANI, Hetal; BALANI, Prem. *Survey on Recommendation System*. *International Journal of Computer Applications*, 2016, vol. 137, no 7.
- SOMMERVILLE, Ian. *Ingeniería de Software*. s.l.: Pearson Educación, 2005. 712p.
- SPARXSYSTEMS. *Diagrama de Despliegue UML 2*. [En línea]. Sparx Systems - Tutorial UML 2 - Diagrama de Despliegue, 2014. [Citado el: 20 de febrero de 2017.]. Disponible en: [http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html].
- SUAZA, Katerine Villamizar. *Definición de equivalencias entre historias de usuario y especificaciones en UN-LENCEP para el desarrollo ágil de software*. [Tesis]. Universidad Nacional de Colombia: Medellín, Colombia, 2013.
- SYMFONY. *Symfony PHP Framework*. [En Línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<https://symfony.com/>].
- THE APACHE SOFTWARE FOUNDATION-JMETER. *Apache JMeter*. [En línea]. Apache JMeter, 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<http://jmeter.apache.org/>].
- TYPE-SAFE, P. H. P. *Java Code Conventions*. Sun Microsystems. California. 2014.

VICTORIA, José Alfredo. y ÁVILA, Manuel. *Patrones de crecimiento en la generación de información en discos duros*. Revista unam.mx, 2009, vol. 10, No.06, p. 3 –3.

VILTRES, Hubert. *Procedimiento para recomendar problemas en el Juez en Línea Caribeño*. [Tesis]. Universidad de las Ciencias Informáticas. La Habana. 2014.

VISUAL PARADIGM. *Visual Paradigm for UML - Software design tools for agile software development*. [En línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<http://www.visual-paradigm.com/features/>].

W3C. *Cascading Style Sheets*. [En línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<https://www.w3.org/Style/CSS/>].

WANG, Xiangyu, et al. *Semantic-based location recommendation with multimodal venue semantics*. IEEE Transactions on Multimedia, 2015, vol. 17, no 3, p. 409-419.

WRIGHT, Tim. *Learning JavaScript: a hands-on guide to the fundamentals of modern JavaScript*. Addison-Wesley, 2012.

YII. *Yii PHP Framework*. [En Línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<http://www.yiiframework.com/>].

ZEND. *Zend Framework*. [En Línea] 2016. [Consultado el: 2 de diciembre 2016.]. Disponible en: [<https://framework.zend.com/>].

ZOETER, Onno, et al. *Information retrieval system*. U.S. Patent No 8,037,043, 11 oct. 2011.

ANEXOS

Tabla 24. Anexo 1: Plan de preguntas. Entrevista al Jefe de Proyecto Orión.

No.	Preguntas
1	¿Qué es Orión?
2	¿Qué visión presenta el proyecto?
3	¿Qué estructura tiene Orión?
4	¿Cómo rastrea los documentos en la web?
5	¿Cómo almacena los documentos encontrados?
6	¿Qué criterio analiza para brindar un ranking a los documentos?
7	¿Qué herramientas tecnológicas se usaron en su implementación y posterior mantenimiento?
8	¿Qué metodología de desarrollo empleó el proyecto?
9	¿Qué número de usuarios haciendo peticiones se calcula puedan soportar los servidores?
10	¿Está actualmente desplegado?
11	¿Se está trabajando en versiones posteriores del proyecto?

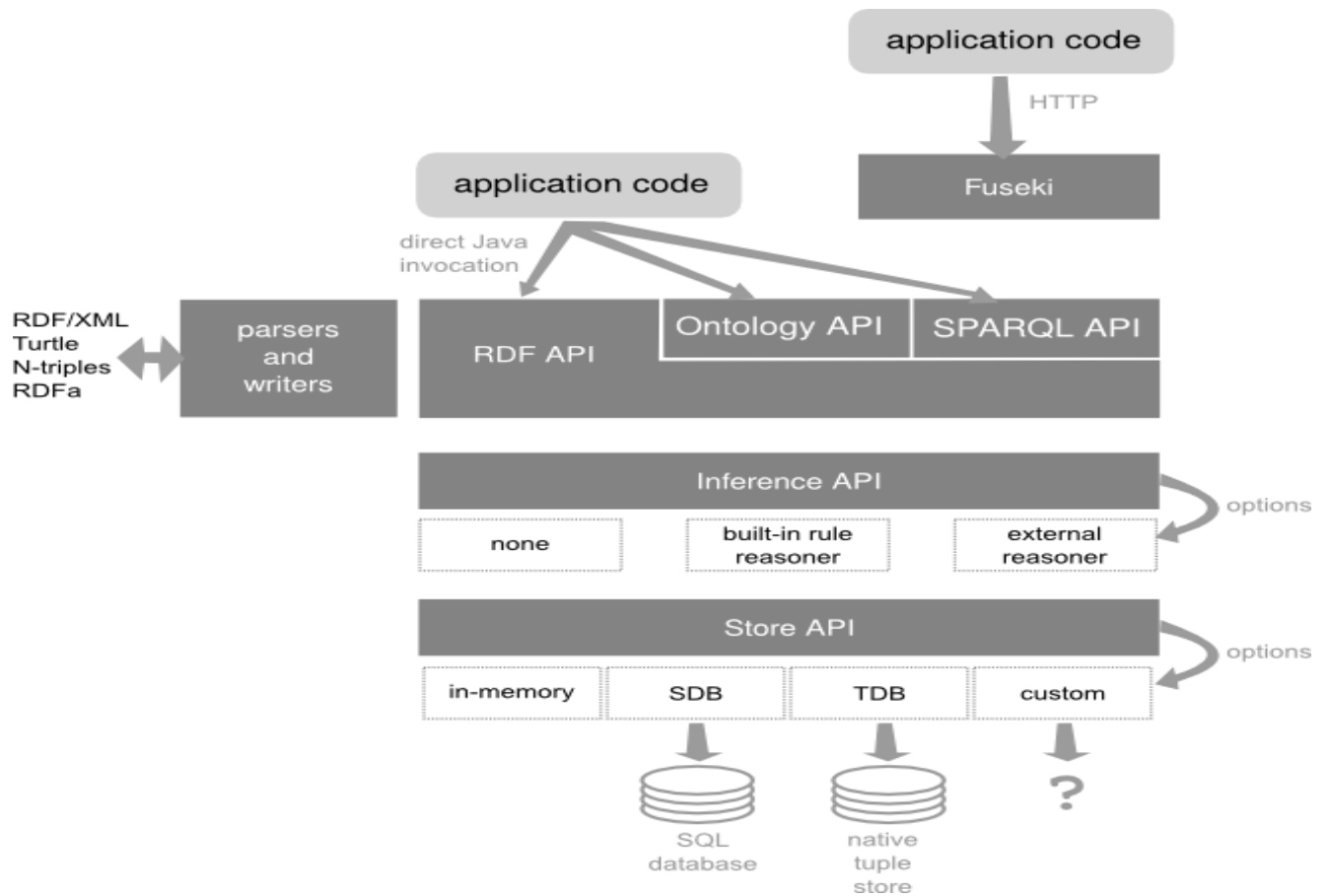


Figura 12. Anexo 2: Arquitectura de Apache Jena (Apache Jena, 2016).

Tabla 25. Anexo 3: Métricas de exactitud en Sistemas de Recomendación (Viltres, 2014).

	Recomendados	No recomendados
Relevante	Verdadero Positivo (VP)	Falso Negativo (FN)
No relevante	Falso Positivo (FP)	Verdadero Negativo (VN)

$P = \frac{VP}{VP + FP}$ <p>Precisión</p>	$R = \frac{VP}{VP + FN}$ <p>Exhaustividad (<i>Recall</i>)</p>
---	---