

Universidad de las Ciencias Informáticas

Facultad 1



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

**Subsistema de gestión de perfil de usuario para el buscador
Orión**

Autora:

Arlenís González Rodríguez

Tutor(es):

MSc. Dely Lien González Hernández

MSc. Hubert Viltres Sala

La Habana, 2017



... la patria os contempla orgullosa.

DECLARACIÓN DE AUTORÍA

Yo, Arlenis González Rodríguez con carnet de identidad 92010824032 declaro ser la única autora del presente trabajo y autorizo exclusivamente a la Facultad 1 de la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Arlenis González Rodríguez

Firma de la Autora

MSc. Delly Lien González Hernández

Firma del Tutor

MSc. Hubert Viltres Sala

Firma del Tutor

DEDICATORIA

A Fidel, por enseñarme a ser revolucionaria.

AGRADECIMIENTOS

A mis padres por su amor infinito.

A mi hermana que un día me hará sentir el orgullo de llamarla licenciada.

A Fernando por su amor y por impulsar la mayor parte de mi vida.

A mi familia por su preocupación y apoyo.

A mis amigos, los buenos y los mejores: Dayan, Amanda, Emilio, Ernesto, Alexis, Rayner, Osmany, Victor M., Julio, Sergio, Marybelén, Laura, Héctor Daniel, Wendy, Oscar, Yaiselín, Lilian, Alfred.

A la FEU por ser mi mayor reto y el más grande de todos mis compromisos.

Porque la FEU fue para mí una escuela y sobre todo una familia, le agradezco a aquellos que estuvieron a mi lado: Lili, Niurka, Ively, Dashiel, Daniel, Roberto Rubén, Yoandri, Lijandy, Arlene, Maylenis, Yenishel, Oscar y Chirino, Carlos Yordan, Yoisbel, Angel, Javier Ojeda, Luis Manuel, Luis Miguel y Javier, Denet, Frank, Sandra, Eliani.

A mis profes Juana Elena, Alien, Ponce, Geidy, Consejo Dirección Fac 1, Consejo Universitario, Natalia, Rectora, Alicia, Tony, Leo, Silvano.

A Mis tutores, los asignados y los obligados: Delly, Hubert, Paúl, Chavez, Tamara, Gino, Osay.

A mis mejores amigos, Ricardo y Leyriel, por ser lo mejor que me ha pasado.

Resumen

El presente trabajo propone un subsistema de gestión de perfil de usuario para el buscador Orión. Se analizaron diferentes fuentes bibliográficas relacionadas con las características, funcionalidades y utilización de perfiles de usuario en Sistemas de Recuperación de Información a nivel nacional e internacional. La solución consiste en un subsistema capaz de gestionar perfiles de usuario en el buscador Orión definiéndose un compendio de información personal, preferencias y comportamiento de navegación en la web. El proceso de desarrollo estuvo guiado por la metodología de software AUP en su versión UCI, seleccionándose como principales tecnologías: el marco de trabajo Symfony 2.7.9, el lenguaje de programación PHP 5.6, el sistema de gestión de base de datos PostgreSQL 9.5 y Visual Paradigm 8.0 como herramienta para el modelado. Las pruebas de software aplicadas al subsistema de gestión de perfil de usuario, demostraron que es una solución funcional, segura, con un rendimiento adecuado y que se integra sin dificultad al buscador Orión. Los resultados de la investigación evidenciaron que el sistema desarrollado posee un alto valor para el buscador Orión al facilitar la identificación de las preferencias del usuario basado en su navegación.

Palabras clave: buscador, perfil de usuario, personalización, preferencias de usuario.

Índice

Introducción	1
CAPÍTULO 1. GESTIÓN DE PERFILES DE USUARIO EN SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN	7
1.1 Fundamentos teóricos asociados al dominio del problema.....	7
1.2 Estudio de sistemas de recuperación de información en la web	11
1.3 Lenguajes de programación	17
1.4 Marcos de trabajo para PHP	19
1.5 Herramienta CASE	21
1.6 Entorno de Desarrollo Integrado.....	22
1.7 Sistema de gestión de base de datos.....	23
1.8 Servidor de aplicación web.....	23
1.9 Lenguaje Unificado de Modelado (UML 8.0).....	24
1.10 Herramientas de validación	24
1.11 Metodología de desarrollo	25
1.12 Conclusiones parciales.....	26
CAPÍTULO 2. DISEÑO DEL SUBSISTEMA DE GESTIÓN DE PERFIL DE USUARIO PARA EL MOTOR DE BÚSQUEDA ORIÓN	27
2.1 Descripción de la propuesta de solución	27
2.2 Modelo del dominio	27
2.3 Especificación de los requisitos del software	28
2.4 Modelo de casos de uso del sistema.....	31
2.5 Arquitectura de la propuesta de solución.....	37
2.6 Patrones de diseño.....	38
2.7 Modelo de diseño	40
2.8 Diseño de la base de datos	41
2.9 Modelo de despliegue	42
2.10 Conclusiones parciales.....	43
CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SUBSISTEMA DE PERFIL DE USUARIO PARA EL MOTOR DE BÚSQUEDA ORIÓN	44
3.1 Modelo de componentes que integran el sistema.....	44

3.2 Estándares de codificación.....	46
3.3 Validación del sistema.....	48
3.4 Conclusiones parciales.....	64
Conclusiones.....	65
Recomendaciones.....	66
Referencias.....	67



Introducción

El acelerado desarrollo de Internet y de las Tecnologías de la Información y las Comunicaciones (TIC) es un factor determinante en el progreso social de la humanidad (Rossini, 2003). El perfeccionamiento tecnológico se ha convertido en uno de los procesos más importantes del mundo generándose la información, en su mayoría, en formato digital. Esta información, actualmente, está disponible en más de 1 billón de sitios web donde se publican noticias, fotos, videos, artículos, entrevistas, y donde más de 3.5 billones de usuarios consumen estos contenidos diariamente (Internet Live Stats, 2016).

La falta de estructuración y control sobre los contenidos publicados en la Web, trae consigo que los usuarios no encuentren de forma rápida y precisa información de calidad que satisfaga sus necesidades. Los Sistemas de Recuperación de la Información (SRI), constituyen el mecanismo ideal para resolver este tipo de problemas al encontrar, procesar y consumir la información de forma automática (Núñez, 2012). Estos sistemas son capaces de localizar casi cualquier contenido existente: textos, imágenes, archivos de sonido, videos, entre otros. Las clasificaciones de SRI más conocidas y usadas son los directorios temáticos, los metabuscadores y los motores de búsqueda o buscadores (Olivas, 2011).

La mayoría de los usuarios que buscan información, y generan la principal parte del tráfico en Internet, utilizan como puerta de acceso a los buscadores (Olivas, 2011). Un motor de búsqueda emplea diferentes algoritmos y métodos para satisfacer la necesidad de información planteada por un usuario en una consulta en lenguaje natural especificada a través de un conjunto de palabras claves ((Jaimes, 2005); (Betancourt, 2009); (Blázquez, 2013)). Sus resultados son muy exhaustivos, pero, poco precisos distando su funcionamiento de ser el más adecuado. La información que obtiene el usuario en ocasiones no es la de mayor calidad, debido principalmente a los mecanismos de ordenamientos, que favorecen o penalizan en dependencia de las políticas implementadas en el buscador.

Los principales motores de búsqueda emplean técnicas avanzadas de Recuperación de Información (RI) que, mediante el análisis de la necesidad del usuario, recuperan la información disponible y seleccionan la que consideran más relevante (Leyva, 2016). La satisfacción con los resultados mostrados por los buscadores, radica en que estos recurren al perfil de cada usuario, donde se recopila información específica que permite mostrar automáticamente contenidos y servicios ajustados a sus preferencias y necesidades.

La disponibilidad actual de información, ha permitido la transformación social, económica y política motivada por el desarrollo de las TIC a nivel mundial. Cuba no ha estado ajena a este proceso, razón por la cual, en

el 7mo. Congreso del Partido Comunista de Cuba, se aprobó el proyecto final del Plan Nacional de Desarrollo Económico y Social hasta 2030: Propuesta de Visión de la Nación, Ejes y Sectores Estratégicos. En el Capítulo 4: Ejes estratégicos, se define en el párrafo 45 como uno de los ejes el desarrollo de la infraestructura, el potencial humano, la ciencia, la tecnología y la innovación. Se traza como objetivo elevar y fortalecer la soberanía tecnológica en el desarrollo de las telecomunicaciones, tecnologías de la información y la conectividad; así como fomentar el desarrollo de nuevas plataformas tecnológicas (Partido Comunista de Cuba, 2016).

Parte de ese proceso se ha evidenciado en la web cubana que, desde los últimos tres años, ha alcanzado una cifra de más de 20 600 000 páginas web distribuidas en 6714 dominios únicos bajo .cu registrados oficialmente (Cubanic, 2017). La búsqueda y consulta de información en el país en estas páginas, mediante buscadores en Internet, representan un grupo de desventajas para los intereses de los usuarios cubanos a causa del bloqueo económico:

- 1. Manipulación de la información:** Se evidencia en la manipulación del posicionamiento de la información indexada según los intereses de sus compañías o gobiernos representativos. (Mexidor, 2011).
- 2. Soberanía tecnológica:** La mayoría de los buscadores en Internet obedecen las leyes del gobierno norteamericano, lo que implica que un grupo de servicios estén bloqueados, dificultando el acceso a los mismos para los usuarios cubanos ((Salomón, 2014); (Martínez, 2013)).
- 3. Posicionamiento web:** La información presente en sitios web cubanos en ocasiones no es indexada por estos buscadores internacionales y por ende no aparece en los resultados. Por otra parte, la información que llegan a ser indexada por lo general proviene de sitios mal posicionados, provocando que aparezca al final de los resultados (Carrillo, 2012).

Enmarcado en este ámbito, en la Universidad de las Ciencias Informáticas (UCI), en la Facultad 1 el Centro de Ideoinformática (CIDI), ha implementado el motor de búsqueda cubano Orión, cuyo propósito fundamental es proveer a la red nacional de una herramienta para la búsqueda y análisis de contenidos web (Hidalgo, 2010). Actualmente, la plataforma que integra los servicios web disponibles en la red cubana Contenidos Unificados para Búsqueda Avanzada (c.u.b.a.)¹ está basada en la tecnología del motor de

¹ www.redcuba.cu



búsqueda Orión. Surge como respuesta a la necesidad de sugerir a los usuarios la información existente en el dominio cubano y brindar la posibilidad de acceder a sitios de interés cultural, informativos e investigativos.

Cuando un usuario realiza una búsqueda utilizando Orión, solamente dispone de las palabras introducidas y las fuentes de datos almacenadas, y como consecuencia el resultado que se brinda al usuario es limitado y no está personalizado acorde a sus preferencias o búsquedas anteriores. No se dispone de un seguimiento sobre los principales temas que les interesan a los usuarios para identificar sus necesidades de información y así reconocer la de mayor interés para ellos, imposibilitando brindar contenidos y servicios ajustados a sus preferencias.

No se dispone de un sistema que permita almacenar las trazas de navegación y su posterior análisis, por tanto, no se recolecta, procesa y guarda información de las acciones del usuario que posibilite identificar sus preferencias y tendencias de comportamiento. La inexistencia de un mecanismo para definir las preferencias del usuario, en el buscador Orión, ocasiona que los criterios de búsqueda no estén personalizados. En tal sentido, si los resultados mostrados no satisfacen las necesidades del usuario, este puede llegar a sentirse insatisfecho y abandonar el uso sostenido del sistema.

Por lo anteriormente expuesto, se plantea como **problema de investigación**: ¿Cómo gestionar los perfiles de usuario para identificar sus preferencias en el buscador Orión?

Se establece para la realización de la investigación como **objeto de estudio**: proceso de gestión de perfiles de usuario; enfocado en el proceso de gestión de perfiles de usuario en el buscador Orión como **campo de acción**.

Para darle solución al problema descrito, se ha planteado el siguiente **objetivo general**: Desarrollar un subsistema que permita gestionar los perfiles de usuario para identificar sus preferencias en el buscador Orión.

Para darle cumplimiento al mismo se definen los siguientes **objetivos específicos**:

1. Construir los referentes teóricos fundamentales que sustentan la investigación relacionados con la gestión de perfiles de usuarios.
2. Diagnosticar el estado de la gestión de perfiles de usuarios.
3. Diseñar las funcionalidades del subsistema de gestión de perfil de usuario para el buscador Orión.

4. Implementar las funcionalidades del subsistema de gestión de perfil de usuario para el buscador Orión.
5. Validar las funcionalidades del subsistema de gestión de perfil de usuario para el buscador Orión.

Luego de haber tratado los elementos fundamentales del área de la ciencia a incidir y los objetivos primordiales, se formula la siguiente **hipótesis de investigación**: Con el desarrollo de un subsistema para gestionar los perfiles de usuario se identificarán sus preferencias en el buscador Orión.

Se define como **variable independiente**: subsistema de gestión de perfiles de usuario, como **variable dependiente**: identificación de las preferencias del usuario en el buscador Orión.

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

Tabla 1. Operacionalización de las variables (Elaboración propia)

Variable independiente	Dimensión	Definición conceptual	Indicadores	Unidades de medida
Subsistema de gestión de perfiles de usuario	Perfil de usuario	Involucra el proceso de recopilación de información específica del usuario durante la interacción con el buscador	Grado de satisfacción con la información recopilada	MA (Muy adecuado) BA (Bastante adecuado) A (Adecuado) PA (Poco adecuado) I (Inadecuado)

Variable dependiente	Dimensión	Definición conceptual	Indicadores	Unidades de medida
Identificación de las preferencias del usuario en el buscador Orión	Preferencias del usuario	Se refiere a la identificación de las preferencias implícitas y explícitas del usuario	Nivel de actualización de las preferencias del usuario	MA (Muy adecuado) BA (Bastante adecuado) A (Adecuado) PA (Poco adecuado) I (Inadecuado)

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

Métodos Teóricos:

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

Analítico – Sintético: Permite concretar y resumir el conocimiento obtenido de los materiales consultados sobre el tema de gestión de perfiles de usuario y del propio conocimiento como usuarios de motores de búsqueda con el fin de utilizarlo en el desarrollo de esta investigación.

Métodos Empíricos:

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

Entrevista: Es utilizado para obtener información que tribute a la definición de los requisitos del módulo; los problemas existentes y la satisfacción de los servicios que brinda actualmente el buscador Orión. La entrevista se realizó a una muestra de especialistas del centro CIDI que han colaborado con el proyecto Orión y a un grupo de usuarios de dicho buscador (Ver Anexo 1).

Capítulo 1. Gestión de perfiles de usuario en Sistemas de Recuperación de Información: En este capítulo se analizan los referentes teóricos relacionados con perfiles de usuario. Además, se realiza un



estudio homólogo de los perfiles de usuario en SRI más utilizados. Finalmente, se seleccionan las tecnologías a utilizar en el desarrollo del subsistema, así como la metodología de desarrollo de software.

Capítulo 2. Diseño del subsistema de gestión de perfil de usuario para el motor de búsqueda Orión:

Se abordan las temáticas relacionadas con el dominio, la caracterización del subsistema a desarrollar y los requerimientos planteados. Se formula la propuesta de la solución a implementar, incluyendo todos los diagramas necesarios para ello.

Capítulo 3. Implementación y validación del subsistema de perfil de usuario para el motor de búsqueda Orión:

Aborda la implementación y se presentan los diseños de casos de prueba a utilizar en la validación del sistema. Se analizan los resultados de las pruebas realizadas que permiten evaluar la calidad de la propuesta de solución.

1

CAPÍTULO 1. GESTIÓN DE PERFILES DE USUARIO EN SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN

En este capítulo con el objetivo de favorecer una mejor comprensión de la investigación que se presenta, se realiza una revisión bibliográfica acerca de los elementos y áreas del conocimiento que abarcan los perfiles de usuario de los SRI. Además, se realiza un análisis y evaluación de los principales perfiles de usuario y se exponen las características esenciales de las herramientas, las tecnologías y la metodología de desarrollo de software que se utilizarán para la implementación de la solución.

1.1 Fundamentos teóricos asociados al dominio del problema

La información constituye hoy la base del desarrollo de las esferas de la sociedad. Según la Real Academia Española (2016) la información es: "Comunicación o adquisición de conocimientos que permiten ampliar o precisar lo que se posee sobre una materia determinada". La información "comprende los datos y conocimientos que se usan en la toma de decisiones" (Ferrell, 2010).

La recuperación de información, en lo adelante (RI), es un término que abarca el saber de diferentes ciencias, desde las ciencias informáticas, la documentación y la ciencia de la información. Se plantea que, expresada en un determinado lenguaje de consulta, la RI consiste en encontrar y *rankear* documentos relevantes que satisfagan la necesidad de información de un usuario (Tolosa, 2013). Según Baeza-Yates (1999) "...es ordenar los documentos de más a menos relevantes para esa necesidad y presentar un conjunto de aquellos con mayor relevancia".

Al responder a las consultas realizadas por los usuarios, como se muestra en la Figura 1, se necesitan diferenciar todos los documentos relevantes y al mismo tiempo rechazar todos los documentos irrelevantes. Para ello, se utilizan modelos o técnicas que ayudan a realizar este proceso, entre los más utilizados están los modelos booleano, probabilístico y vectorial (Ellis, 1996).

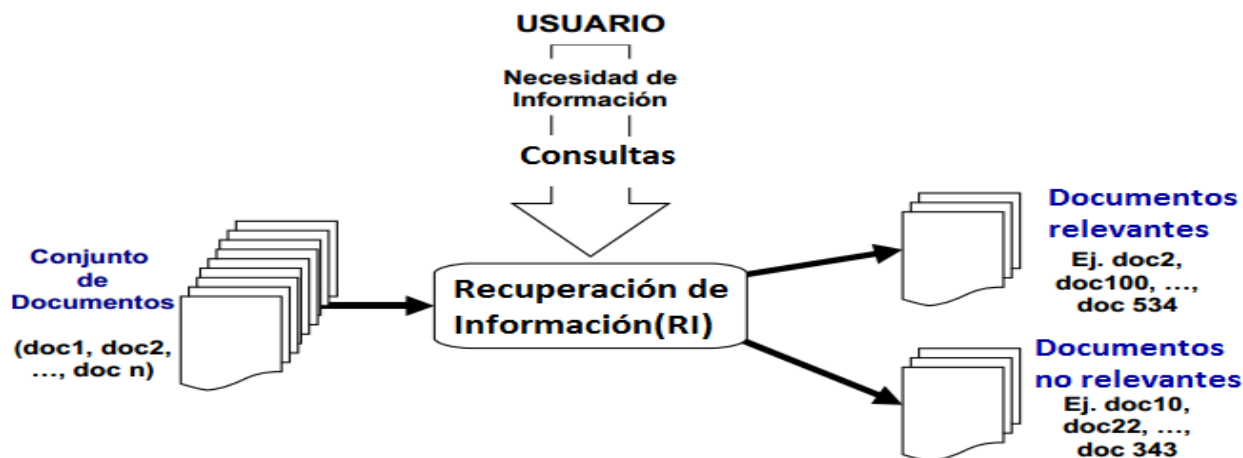


Figura 1. Esquema de Recuperación de Información (Tolosa, 2013)

En la presente investigación se define la RI como la condición de encontrar información variada, sobre cualquier ciencia del saber humano, en las distintas fuentes existentes, con la finalidad de procesarla y ordenarla según su importancia o índice de relevancia, para la satisfacción de las necesidades de conocimiento del usuario.

Los sistemas encargados de buscar información, dentro de un grupo de documentos que han sido indexados previamente, son los Sistemas de Recuperación de Información (SRI). De acuerdo con Baeza–Yates (2005) los SRI “deben de alguna manera interpretar el contenido de la información dentro de una colección de documentos y establecer con ellos, un orden de acuerdo al grado de relevancia que estos posean, para recuperar exactamente lo que se quiere para las consultas de los usuarios”.

Los SRI se clasifican teniendo en cuenta la forma en que indexan el contenido de la web, el alcance que poseen, los tipos de documentos que recuperan y la forma que muestran el resultado. Las clasificaciones más conocidas y usadas son: los directorios temáticos o web, los buscadores o motores de búsqueda y los metabuscadores (Olivas, 2011).

Los buscadores o motores de búsqueda constituyen el SRI más idóneo para la implementación de sistemas autónomos dedicados a la recopilación, procesamiento y recuperación de grandes volúmenes de información. Según Camiño (2003): “Los buscadores, motores de navegación o motores de búsqueda (MB), son aquellos programas o herramientas interactivas que facilitan la búsqueda y recuperación de información en Internet. Los motores de búsqueda ofrecen formularios para introducir los datos mediante una interfaz

de fácil comprensión para el usuario, el cual teclea una palabra clave o frase y recupera una lista de recursos que se corresponden con el criterio indicado”.

Los buscadores pueden llegar a ofrecer millones de resultados para una sola búsqueda, aunque esto carece de importancia. Desde la perspectiva del usuario final es indiferente que para una palabra clave, el buscador obtenga decenas o varios millones de resultados, ya que solamente examinará los primeros. Así que, para cualquier búsqueda, los resultados que importan son los que aparecen en las primeras posiciones de la página de resultados o SERP (*Search Engine Results Page*) (Gonzalo, 2015).

Los motores de búsqueda, para la obtención de mejores resultados que se adecuen a las preferencias y necesidades de los usuarios, recurren a la utilización de perfiles de usuario. Bonnet (2001), considera que el proceso de recopilación de información específica del usuario durante la interacción con él, es usado para entregar contenidos y servicios apropiados, hecho a la medida de sus necesidades. Las técnicas de investigación social que sirven para determinar el comportamiento de los mismos, son denominadas estudios de usuarios. El objetivo fundamental de dichos estudios consiste en suministrar datos verídicos que caractericen a los usuarios. Se obtiene como resultado un compendio de información personal y rasgos distintivos de un individuo denominado perfil de usuario (Salazar, 1993).

Los perfiles de usuario se pueden obtener de diferentes fuentes y a través de diferentes procesos. Nasraoui (2003) define el perfil de usuario como “la información acerca de los atributos demográficos de los usuarios y preferencias que son obtenidos explícita o implícitamente”.

El método explícito o manual responde principalmente a la introducción de datos mediante formularios como la fecha de nacimiento, edad, nombre, entre otros. El método colaborativo o de composición a partir de otros perfiles, recurre al conocimiento específico del dominio y heurísticas inteligentes para crear o actualizar el perfil a través de la interacción con otros perfiles con los que se relaciona. Por último, el método implícito, recurriendo normalmente a técnicas de inteligencia artificial, crea o modifica automáticamente los perfiles (Mobasher, 2005) utilizando información dinámica como los intereses del usuario que cambian con regularidad.

Los perfiles de usuarios se enmarcan dentro del proceso de personalización como una estructura de almacenamiento de información sobre las preferencias del usuario (Escobar, 2007). “La capacidad de proporcionar información diferente en función de los distintos estereotipos definidos para clasificar a los usuarios, sobre la base del conocimiento de sus preferencias y comportamientos a la hora de interactuar,

se refiere a la personalización” (Hagen, 1999). En el caso de la web, el proceso de personalización obtiene el conocimiento a partir de un conjunto de acciones realizadas por el usuario en su navegación, basándose en un proceso previo de minería web de uso. Este proceso permite obtener tendencias en el comportamiento de navegación o identificar grupos de usuarios con características similares y así generar recomendaciones.

A efectos de la presente investigación se define el perfil de usuario como la obtención de las preferencias y datos identificativos del usuario. El modelado del perfil conlleva un proceso de aprendizaje continuo de la información proporcionada por el usuario de manera implícita o explícita derivada de su actividad en la web.

Un campo importante de aplicación en Internet es la minería web, la cual facilita el acceso, recuperación y organización de información. Esta se subdivide en tres clases fundamentales: minería web de contenido, de estructura y de uso web. Esta última clase tiene como principal objetivo extraer patrones de uso de la web por parte de los usuarios. Para esto, se utilizan los archivos Log (registros de sucesos/eventos) de los servidores web. Este tipo de minería permite obtener perfiles de los distintos tipos de usuarios a partir de su comportamiento y navegación, para ofrecer una atención más personalizada (Reyes, 2007).

Todos los procesos de personalización basados en minería web de uso se estructuran en tres fases como muestra la Figura 2. La fase de preparación de datos transforma los archivos del servidor web (archivos Log) en la forma intermedia adecuada para ser procesados (Justicia, 2004); incluye, además, la integración de datos de múltiples recursos o fuentes como aplicaciones del servidor, base de datos de *back-end* y el contenido del sitio. En la siguiente fase se aplican a las transacciones de datos técnicas de minería web de uso (agrupamiento o *clustering*, asociación, secuencias frecuentes) en la búsqueda de patrones. Los resultados de la fase de minería pueden ser transformados en perfiles de usuario, adecuados para ser utilizados en la fase de recomendación (Escobar, 2007).

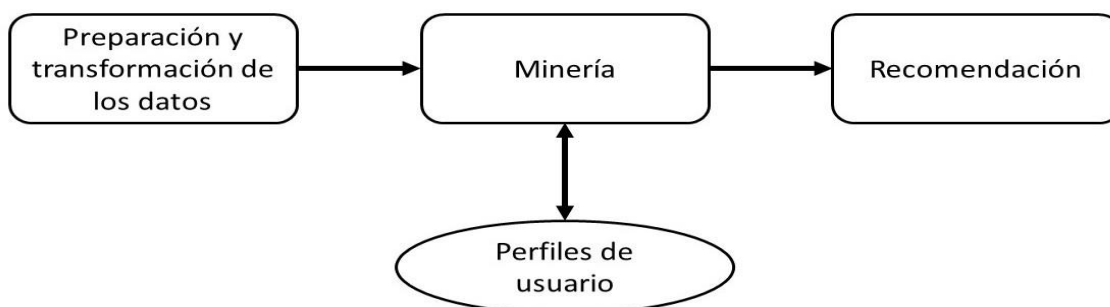


Figura 2. Etapas del proceso de personalización (Escobar, 2007)

Los sistemas de recomendación son utilizados para sugerir productos y/o contenido a los visitantes de sitios web a partir de sus preferencias. El comportamiento de navegación que presentan los usuarios es un indicador de estas preferencias y es la base para extraer patrones de uso frecuente (Corso, 2011). Estos patrones ayudan a las personas a encontrar elementos en dominios regidos por un gran cúmulo de información. Tales herramientas tienen como objetivo proporcionarles a los usuarios elementos útiles de acuerdo a sus necesidades o gustos (Remedios, 2014).

En dependencia de la fuente de información y la técnica que se utilice para clasificar los elementos se pueden mencionar varios diseños de sistemas de recomendación: Sistema de Recomendación Basado en el Contenido (SRBC)², Sistema de Recomendación Colaborativo, Sistema de Recomendación Demográfico, Sistema de Recomendación Basado en el Conocimiento, Sistema de Recomendación Híbrido.

El SRBC incluye un componente de Filtrado que utiliza el perfil de usuario para sugerir elementos relevantes, comparando la representación del perfil de usuario contra la del elemento que se pretende recomendar. El resultado es un valor de relevancia que puede ser binario o continuo (este es determinado haciendo uso de alguna métrica de similitud (Holte, 1996)). En este sentido, los elementos determinados de mayor relevancia, son incluidos en la lista de elementos que será presentada al usuario (Remedios, 2014).

1.2 Estudio de sistemas de recuperación de información en la web

En la actualidad existen numerosos sistemas de recuperación de información. Para identificar ventajas en el uso de estos sistemas y lograr una mejor comprensión de sus características y funcionalidades, se hace necesario realizar un análisis de ellos. A continuación, se expone el estudio realizado de los sistemas homólogos más utilizados (NetMarketShare, 2017), tanto en el ámbito nacional como internacional.

Google

Google Inc. es una empresa multinacional estadounidense especializada en productos y servicios relacionados con Internet, software, dispositivos electrónicos y otras tecnologías (Google, 2016). El motor de búsqueda Google³, persigue como objetivo que sus usuarios encuentren la información que necesitan y

² Este tipo de sistema de recomendación computa las recomendaciones de acuerdo a las características de los elementos que el usuario prefirió en el pasado (Pazzani, 1996).

³ www.google.com

consigan hacerlo de la forma más sencilla y rápida posible. Ofrece servicios como búsqueda de imágenes, libros, noticias, videos, documentos académicos, entre otros.

A partir de los sucesivos cambios en los algoritmos de búsqueda y ordenación que Google implementó entre 2012 y mediados de 2015, los resultados aparecidos en sus respuestas fueron cada vez menos predecibles y, por tanto, menos manipulables (Gonzalo, 2015). Entre estos cambios hubo dos que se centraron en el usuario: uno de ellos en su perfil (procedencia, edad, intereses, etc.) y el otro en la intención de búsqueda expresada de forma directa o indirecta. Es necesario tener en cuenta que Google se expresa a partir de lo que se denomina la *Search Engine Results Page* (SERP), por lo que, los resultados que importan son los que aparecen en las primeras posiciones de la SERP.

Google analiza entre los parámetros más importantes del perfil de usuario la ubicación geográfica, la cual determina mediante la IP de su conexión y, en el caso de las búsquedas móviles, por geolocalización. Además, tiene presente su actividad anterior a través de las *cookies* instaladas y otros aspectos de sus intereses cuando es un usuario que ha iniciado sesión y, por tanto, se ha identificado. Aquí Google puede utilizar sus búsquedas anteriores, así como su perfil demográfico y profesional gracias a datos procedentes de otras cuentas, como Google+ o Gmail (Gonzalo, 2015).

Twitter

Twitter⁴, con sede en Estados Unidos desde 2006, es una red de información en tiempo real a base de mensajes que le permiten al usuario estar actualizado y en contacto con las personas que más le interesan. Las relaciones optativas (seguir/ ser seguido) no requiere el consentimiento mutuo entre los usuarios (Twitter, 2016).

Twitter es un entorno de lecto-escritura en el que cada mensaje contiene enlaces por defecto, el uso del símbolo de la @ y del # genera enlaces de manera automática. Para crear una cuenta en esta red social el usuario asume un nombre antecedido por "@". Luego de abrir la cuenta, se le asigna un perfil donde puede estar al pendiente de sus seguimientos también llamados en inglés *following profile* y de sus seguidores o *followers profile*, también puede buscar amigos, familiares, artistas u otras personas de su interés. Twitter ofrece otras opciones como invitar amigos por email, buscar en otras redes sociales o seleccionar a usuarios recomendados. A continuación, se explican algunas de ellas (Twitter, 2016):

⁴ www.twitter.com

- **A quién seguir:** En función de los usuarios que sean seguidos, la descripción que se haya puesto en el perfil, los seguidores que se tengan, entre otros, Twitter sugiere usuarios que podrían resultar de interés.
- **Información seleccionada:** Al decidir a quién se sigue en Twitter se decide también qué se quiere ver en el *timeline*. Además, Twitter ayuda a organizar esa información mediante listas, que cada usuario puede formar según temas a elección (Noticias/Amigos/Negocios/Famosos/Clientes).
- **Rapidez en búsqueda de información:** Permite encontrar las publicaciones de temas deseados o recurrentes para el usuario con prontitud.
- **Crear listas.** Se pueden crear listas de usuarios de Twitter por tema o interés para organizar la información. Teniendo en cuenta que cada vez hay más contenidos en la red social, resulta una función muy práctica.

La plataforma de Twitter ofrece el acceso a los datos almacenados a través de varias interfaces de programación. Cada Interfaz de Programación de Aplicaciones (por sus siglas en inglés API) representa una faceta de Twitter que facilita a los desarrolladores construir y ampliar sus aplicaciones en formas nuevas y creativas (Twitter, 2016).

Facebook

Facebook⁵ es una red social creada por Mark Zuckerberg, en Estados Unidos. Originalmente era un sitio para estudiantes de la Universidad Harvard, pero actualmente está abierto a cualquier persona que tenga una cuenta de correo electrónico (Facebook, 2016).

A través del perfil, en primera instancia, se comparte información sobre el usuario accediendo mediante la opción "Registrarse" donde se solicitan datos de información básica (nombre, apellidos, correo electrónico, contraseña, sexo y fecha de nacimiento). Se puede comprobar que el perfil se divide en diferentes sub-apartados:

- **Muro:** donde aparecerán las publicaciones.
- **Información:** aquí se podrán encontrar los datos que se han querido compartir en el perfil (información básica, filosofía, actividades e intereses, etc.).
- **Notas:** desde dónde se comparten publicaciones más extensas que las del muro con los amigos.

⁵ www.facebook.com

- **Amigos:** se encuentra un listado completo de los contactos en Facebook.
- **Sugerencias:** El sistema suele mostrar como sugerencias a personas que piensa que puedes conocer: por ejemplo, a los amigos de tus amigos, de forma que puedas enviarles solicitudes de amistad.

Los usuarios de Facebook muestran al resto del mundo virtual una visión de su vida. Seleccionan las páginas que más les gustan, siguen a sus bandas de música favorita, etiquetan a sus amigos y comentan los mensajes más divertidos. A partir de aquí, Facebook muestra publicidad que considera que puede ser de interés del usuario, recomienda amigos de amigos y muestra páginas que podrían gustar.

A veces, los usuarios no muestran su comportamiento real. Mediante la minería de dato se puede detectar rastreando palabras claves que aparecen en su perfil o la generación de contenido en la plataforma, el comportamiento de un usuario que no ha sido especificado explícitamente. De esta manera, se define un perfil mucho más auténtico del usuario permitiendo mostrar información y promociones que si son de su interés, más allá de lo que este ha indicado (Espinoza, 2014).

La red social cuenta con una plataforma que contiene una serie de API para la integración con sus servicios. El núcleo de integración de Facebook reside en la Graph API, este mecanismo de integración permite insertar y recuperar información de los recursos de la red social, la comunicación se basa en el estilo arquitectónico REST y presenta soporte para varios formatos de resultados (Facebook, 2016).

LUPA

LUPA⁶ es un buscador de contenidos en la REDUNIV⁷. Ha sido desarrollado en la Universidad de Pinar del Río por el Grupo de Desarrollo y Diseño Web de la Dirección de Informatización con el propósito de localizar contenidos dispersos en sitios web y repositorios públicos de la red MES⁸ y en sus enlaces transversales de alta velocidad (MINED⁹, INFOMED¹⁰, Cultura, Joven Club, UCI). El empleo de esta herramienta tiene un impacto directo en el aprovechamiento de los contenidos dentro del país que disminuye el tráfico de los

⁶ www.lupa.upr.edu.cu

⁷ Red Nacional de Datos del Ministerio de Educación Superior de Cuba disponible en <http://reduniv2.mes.edu.cu>

⁸ Siglas del Ministerio de Educación Superior

⁹ Siglas del Ministerio de Educación

¹⁰ Red de salud cubana disponible en <http://www.sld.cu>



usuarios de la REDUNIV hacia internet en busca de información en los populares buscadores internacionales (LUPA, 2016).

A pesar de la complejidad de los procesos internos del motor de búsqueda que aseguran el rendimiento de la aplicación, la utilización por un usuario final se reduce a la interacción con la interfaz web, con lo cual se han puesto esfuerzos en lograr un diseño limpio y amigable, que intenta mostrar de forma intuitiva las principales funcionalidades del sistema y que además es compatible con dispositivos móviles para darle soporte al creciente número de usuarios que acceden a las redes cubanas desde este tipo de dispositivos. No tiene implementado sistemas de recomendación, ni perfiles de usuario.

Algunas de las funcionalidades notables son (LUPA, 2016):

- **Documentos Similares:** Los documentos similares a otro documento son aquellos que comparten el mismo tema. El buen funcionamiento de esta característica depende de lo bien definido que esté el tema en el documento seleccionado como punto de partida.
- **Filtrado de Documentos por tipo de archivo:** Lupa es capaz de organizar por tipo de archivo los resultados obtenidos en una consulta.
- **Búsquedas relacionadas:** Las búsquedas relacionadas están formadas por una nube de frases o términos representativos del conjunto de documentos devueltos en una consulta. En Lupa se utilizan para ayudar al usuario a direccionar su búsqueda hacia los resultados que desea, esto se logra haciendo *clic* en uno o varios términos a los que se desea elevar la relevancia o importancia de dichos términos en la consulta.

Orión

El motor de búsqueda Orión¹¹ es un sistema de recuperación de información que permite realizar búsquedas sobre los contenidos publicados en la web cubana mediante una interfaz amigable e intuitiva. Esta herramienta está compuesta por tres componentes (mecanismo de rastreo, mecanismo de indexación, interfaz web) que en su conjunto brindan las funcionalidades de búsqueda general y avanzada de documentos publicados en la web. Orión está desarrollado con Nutch en su versión 1.5 para el rastreo en la red, Solr versión 3.6 como mecanismo de indexación, Symfony en su versión 2.0 para la interfaz web

¹¹ <http://www.orion.uci.cu>

y la interacción con el mecanismo de indexación, PostgreSQL versión 9.5 como gestor de base de datos relacional para almacenar estadísticas, Java versión 7.0 para el manejo de *plugins* en Nutch y Solr y HTML 5 y CSS 3 para la interfaz web ((Leyva, 2016), (Hernández, 2013)).

Actualmente el buscador Orión, aunque aún se encuentra en proceso de desarrollo, ha sido desplegado en determinadas instituciones con el objetivo de realizarle pruebas funcionales. De esta forma, Orión se ha convertido en la principal herramienta cubana que permite la recuperación de información existente en la intranet nacional (Utrera, 2015). Además, constituye el motor de búsqueda de la red c.u.b.a (Contenidos Unificados para Búsqueda Avanzada), pero no incluye sistemas de recomendación o perfiles de usuarios que ajusten los resultados de las búsquedas a las preferencias del usuario.

Valoración sobre los sistemas de recuperación de información

Los sistemas de recuperación de información utilizan entre sus herramientas el perfil de usuario, mediante el cual garantizan la personalización y el uso sostenido del sistema. En el perfil las personas y entidades se agrupan atendiendo a diferentes aspectos éticos, culturales, de intercambio, de negociaciones, entre otros. El mismo se construye a partir de las características que distinguen a un usuario de otro y de los factores de influencia que lo circundan (Simmonds, 2012). Entre los elementos habituales que componen un perfil de usuario se encuentran los siguientes:

- Información básica (nombre, edad, sexo, fecha de nacimiento, país, etc.)
- Información académica (curso, año de matrícula, grupo, facultad, etc.)
- Imagen del perfil.
- Intereses del usuario (cultura, deporte, política, arte, historia, economía, etc.)
- Nivel de acceso a la información (privado, amigos, usuarios autenticados, todos).
- Recursos de información utilizados o requeridos (actividad de navegación reciente).
- Comportamiento en la búsqueda de información.
- Actualización sobre temas de interés común, propiciando la participación en eventos, actos, conferencias, etc.

Entre las herramientas homólogas estudiadas existen algunas privativas, cuya utilización se dificulta por el limitado acceso a internet que posee Cuba debido al bloqueo económico. Además, el buscador cubano Orión no brinda la posibilidad de realizar una búsqueda de información a partir de su perfil de usuario,

posibilitando obtener resultados adecuados a sus preferencias. Se evidencia la necesidad de contar con la creación de un perfil de usuario que permita identificar las preferencias del usuario mediante su navegación web, brindando una serie de criterios de búsqueda que ayuden a obtener resultados más exactos.

Para el desarrollo del subsistema de gestión de perfil de usuario se necesita la utilización de un conjunto de tecnologías, lenguajes y herramientas. Estas serán seleccionadas teniendo en cuenta el alcance de sus prestaciones y las características de la solución a desarrollar.

1.3 Lenguajes de programación

Los lenguajes de programación están formados por un conjunto de reglas sintácticas y semánticas, que les permiten definir su estructura y el significado de sus expresiones. Permiten, además, especificar los datos que se deben procesar, almacenar o transmitir, y las acciones que se deben realizar bajo determinadas circunstancias. A continuación, se exponen las características de cada uno de los lenguajes estudiados y empleados en la implementación de la solución propuesta.

PHP 5.6

PHP (*Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web, que puede ser incrustado en HTML, pero solo puede ejecutarse en el lado del servidor (Bakken, 2013). Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el *script*, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP (PHP, 2016).

De manera general, las principales características que distinguen a PHP son: su rapidez, su facilidad de aprendizaje, su soporte multiplataforma como servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta. Es un lenguaje orientado a objetos y posee variedad de funciones ya implementadas de utilidad para los desarrolladores (Triana, 2012).

JavaScript

JavaScript es un lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. En ese contexto podríamos decir que nació como un "*lenguaje de scripting*" del

lado del cliente, sin embargo, hoy es mucho más. Las necesidades de las aplicaciones web modernas y el HTML5 han provocado que el uso de JavaScript que encontramos hoy haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel (Java, 2014).

JavaScript permite:

- Máxima interactividad entre el usuario y la página.
- Verificación de los datos introducidos por el usuario, antes de enviar el formulario al servidor.
- Ejecución de carritos de compra en el navegador.
- Ejecución de pequeñas cantidades de información al igual que en una base de datos.
- Manejo de *applets* y *plugins* dentro de múltiples marcos de HTML.
- Pre-procesado de información antes de enviarla al servidor.
- Es un lenguaje muy potente de alto nivel, es dinámico y responde a eventos en tiempo real.

HTML 5

HTML 5 es un lenguaje de marcado para Hipertextos (en inglés, *Hyper Text Markup Language*) usado para estructurar y presentar el contenido para la Web, está especificado como un estándar por el W3C (*World Wide Web Consortium*) que permite la creación de páginas web. HTML5 está relacionado también con la entrada en decadencia del viejo estándar HTML 4, que se combinaba con otros lenguajes para producir los sitios que podemos ver hoy en día. Con HTML5, entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación (Bitelia, 2015).

La aparición del lenguaje influyó notablemente en el crecimiento de Internet, donde la información era distribuida mediante colecciones fragmentadas de textos, imágenes y sonidos. HTML es independiente de la plataforma utilizada y se basa fundamentalmente en el uso de etiquetas estructurales y semánticas, adecuadas para la creación de documentos relativamente simples que permiten simplificar su estructura (Consortium, 2014).

CSS3

Mientras que HTML permite definir la estructura de una página web, las hojas de estilo en cascada (*Cascading Style Sheets* o CSS) ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web (CSS3, 2016). Las hojas de estilo permiten definir de

manera eficiente la representación de páginas y es uno de los conocimientos fundamentales que todo diseñador web debe manejar a la perfección para realizar su trabajo.

A partir del año 2005 se comenzó a definir el sucesor de esta versión, al cual se le conoce como CSS3 o *Cascading Style Sheets Level 3*. Actualmente en definición, esta versión ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, CSS3 es el estándar que dominará la web por los siguientes años (CSS3, 2016).

1.4 Marcos de trabajo para PHP

Otro de los componentes de un buscador es la interfaz web. Para su desarrollo existen herramientas que poseen una estructura personalizable e intercambiable llamadas marcos de trabajo o *framework*. Estos constituyen estructuras conceptuales y tecnológicas de soporte definido, componen la base con la cual otro proyecto de software puede ser fácilmente organizado y desarrollado (Soap, 2015). A continuación, se describen los *frameworks* que serán utilizados para la implementación de la propuesta de solución.

Symfony 2.7.9

Symfony es un proyecto PHP de *software* libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional, es un *framework* PHP de tipo *full-stack* construido con varios componentes independientes. Está diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador (Potencier, 2014).

Ventajas del uso de Symfony como marco de trabajo (Guiluz, 2013):

- Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en las bases de datos y una mejor experiencia para el usuario.
- El manejo de cache reduce el uso de banda ancha y la carga del servidor.
- La presentación usa *templates* y *layouts* que pueden ser construidos por diseñadores de HTML que no posean conocimientos del *framework*.
- Los *plugins* proveen un alto nivel de extensibilidad.
- Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.

- El *framework* de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas.
- La barra de depuración web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.
- La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.
- Es posible realizar cambios "en caliente" de la configuración (sin necesidad de reiniciar el servidor).
- El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

Luego del análisis realizado, teniendo en cuenta las características y beneficios expuestos, se decide emplear en el desarrollo del sistema propuesto Symfony2. Además de ser sencillo de usar y lo suficientemente flexible como para adaptarse a los casos más complejos, puede integrarse al buscador cubano Orión que contiene una interfaz web desarrollada con este popular *framework* (Potencier, 2011).

Bootstrap 3.3.1

Bootstrap es un *framework* que simplifica el proceso de creación de diseños web es originalmente creado por Twitter, permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, tableta u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “*responsive design*” o diseño adaptativo. Los diseños creados con Bootstrap son simples, limpios e intuitivos, esto le proporciona agilidad a la hora de cargar y al adaptarse a otros dispositivos. El *framework* trae varios elementos con estilos predefinidos fáciles de configurar: botones, menús desplegables, formularios incluyendo todos sus elementos e integración con jQuery para ofrecer ventanas y *tooltips*¹² dinámicos (Otto, 2013).

Características principales (Genvetadev, 2012):

- Bootstrap ofrece una serie de plantillas CSS y ficheros JavaScript que permiten integrar el *framework* de forma sencilla y potente en proyectos webs.

¹² El complemento *tooltips* es un pequeño cuadro emergente que aparece cuando el usuario mueve el puntero del ratón sobre un elemento.

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo, *jQuery*.
- Ofrece un diseño sólido usando LESS¹³ y estándares como CSS3/HTML5.
- Funciona con todos los navegadores, incluido Internet Explorer usando HTML Shim para que reconozca las etiquetas HTML5.
- Dispone de distintos *layout* predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

1.5 Herramienta CASE

Las herramientas de ingeniería de software asistida por computación, también conocidas por sus siglas en inglés CASE (*Computer Aided Software Engineering*) facilitan el trabajo de ingeniería de un proyecto, aportando funcionalidades para la organización, el modelado y el seguimiento de un *software*.

Visual Paradigm 8.0

Visual Paradigm es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Además, es una herramienta de diseño multiplataforma. Ofrece un conjunto completo de herramientas que los equipos de desarrollo necesitan para la captura de requisitos, planificación de software, planificación de pruebas, modelamiento de clases, modelamiento de datos y otras actividades (Paradigm, 2016).

Entre sus principales características se encuentran:

- Entorno de creación de diagramas para UML.
- Diseño centrado en casos de uso y enfocado al negocio que genera un *software* de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo. Disponibilidad de múltiples versiones, para cada necesidad.

¹³ Es un dinámico lenguaje de hojas de estilo que puede ser compilado en Hojas de estilo en cascada (CSS) y ejecutarse en el lado del cliente o en el lado del servidor.

- Generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos (Sommerville, 2007).

Se selecciona *Visual Paradigm 8.0* como herramienta CASE. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas.

1.6 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Como entorno de desarrollo integrado se utilizará NetBeans en su versión 8.0.

NetBeans 8.0

Liberado bajo el licenciamiento dual de CDDL¹⁴ y GPL¹⁵ (versión 2), NetBeans es un potente IDE para programadores que proporciona una plataforma ideal para escribir, compilar, depurar y ejecutar programas informáticos (Oracle, 2017). Aunque inicialmente fue ideado para Java, puede ser empleado para la codificación de aplicaciones en múltiples lenguajes de programación. Este, además de ser gratuito y sin restricciones de uso, posee versiones para los distintos sistemas operativos del mercado, convirtiéndolo en una alternativa con grandes ventajas para los desarrolladores. La estructura modular de NetBeans le proporciona estabilidad y grandes posibilidades de ser extendido gradualmente por desarrollos comunitarios, permitiendo agregar continuamente nuevas funcionalidades. Su versatilidad lo ha convertido en el IDE por excelencia entre miles de programadores alrededor del mundo (Oracle, 2017).

¹⁴ *Common Development and Distribution License* (Licencia Común de Desarrollo y Distribución)

¹⁵ *General Public License* (Licencia Pública General)

1.7 Sistema de gestión de base de datos

Los sistemas de gestión de bases de datos son un tipo de software dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de estos sistemas es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización (Martínez, 2012).

PostgreSQL 9.5

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD¹⁶ y con el código fuente más potente del mercado disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (PostgreSQL, 2017).

Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Implementa el uso de *rollback's*, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel (PostgreSQL, 2017).

Para el desarrollo de la solución de la presente investigación se selecciona PostgreSQL como sistema gestor de base de dato por su desempeño trabajando con grandes volúmenes de información. Es un sistema estable, robusto y fácil de administrar para un usuario con conocimientos básicos. Actualmente es asimilado como tecnología del proyecto Orión (Leyva, 2016).

1.8 Servidor de aplicación web

NGINX es un servidor web de código abierto de alto rendimiento. Contiene un servidor proxy inverso para protocolos IMAP/POP3/SMTP, ofrece bajo consumo de recursos del sistema, gestiona la cache y el balanceo de carga. Una de las características más importantes de NGINX es que se trata de un software que es asíncrono y esto trae consigo la escalabilidad provocando que se gestionen las peticiones en muy

¹⁶ Berkeley Software Distribution

pocos hilos, reduciendo las posibilidades de sobrecarga en el servidor (Velasco, 2015). La versión de NGINX que se utiliza para el desarrollo del subsistema de gestión de perfil de usuario es 1.10.1.

Otras características que ofrece el servidor NGINX son:

- Capaz de manejar más de 10.000 conexiones simultáneas con un uso bajo de memoria.
- Balanceo de carga, distribuye la carga entre los servidores que formen parte de la estructura, redirigiendo cada vez la petición hacia aquella máquina que tenga una menor carga.
- Alta tolerancia a fallos.
- Compatible con el nuevo estándar de direcciones IPv6.

1.9 Lenguaje Unificado de Modelado (UML 8.0)

Es el lenguaje de modelado de sistemas (*UML-Unified Modeling Language*) de software más conocido y utilizado en la actualidad; está respaldado por el Grupo de Gestión de Objeto (OMG, siglas del inglés *Object Management Group*). Se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas (Larman, 2003). Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios.

El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código, así como generadores de informes. La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos (Cornejo, 2008) .

1.10 Herramientas de validación

Apache JMeter 2.8.2

JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones Web. JMeter puede

ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, servicios Web, JMS, HTTP y conexiones TCP genéricas (Foundation-Jmeter, 2016).

Acunetix Web Vulnerability Scanner 9.5

Acunetix comprueba los sistemas en busca de múltiples vulnerabilidades que un atacante podría aprovechar para obtener acceso a los sistemas y datos. Puede utilizarse para realizar escaneos de 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 servidores web o de las aplicaciones que se analizan (Acunetix, 2016).

Incluye un analizador automático para pruebas de seguridad para *AJAX*¹⁷, desarrollado con el lenguaje de programación JavaScript. Además, permite realizar pruebas para ataques de inyección de código, secuencia de comandos en sitios cruzados (XSS por sus siglas en inglés) y falsificación de peticiones. Posee un componente que facilita la realización de pruebas a formularios y a áreas protegidas por contraseña. Permite realizar varias peticiones simultáneamente, siendo capaz de explorar cientos de páginas sin interrupciones (Acunetix, 2016).

1.11 Metodología de desarrollo

Las metodologías son un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayudan a los desarrolladores a realizar software. Estas brindan soporte a la toma de decisiones en un equipo de trabajo; es decir, permiten conocer qué persona hace una determinada actividad, cuándo y cómo la debe hacer. Además, se pueden clasificar en dos tipos: pesadas y ágiles (Pressman, 2006).

AUP-UCI. Desarrollo para la Actividad Productiva en la UCI

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado Relacional (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP (Sánchez, 2015).

¹⁷ Es una técnica de desarrollo web que permite realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

AUP-UCI se basa en una variación de la metodología “Proceso Unificado Ágil” (AUP por sus siglas en inglés) en unión con el modelo CMMI-DEV v 1.3. Además, es aplicable a todos los proyectos de desarrollo de software que se desarrollen en la Universidad.

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, que es Ejecución y se agrega la fase de Cierre.

1.12 Conclusiones parciales

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

- Las relaciones existentes entre los principales conceptos asociados al dominio de la presente investigación permitieron una mayor comprensión de la propuesta de solución.
- Las deficiencias encontradas en los motores de búsqueda nacionales, el limitado acceso a los internacionales y la manipulación de los resultados presentes en ellos hacen necesario la creación de un subsistema de perfil de usuario que permita identificar sus preferencias en la red cubana.
- La selección de la metodología, herramientas y tecnologías con soporte multiplataforma y basadas en software libre, posibilitó obtener una base tecnológica enfocada en los componentes que utilizan los sistemas de recuperación de información.

2

CAPÍTULO 2. DISEÑO DEL SUBSISTEMA DE GESTIÓN DE PERFIL DE USUARIO PARA EL MOTOR DE BÚSQUEDA ORIÓN

En el presente capítulo se describen los aspectos fundamentales relacionados con el diseño del subsistema de gestión de perfil de usuario para el motor de búsqueda Orión. Se identifican las clases del dominio y la relación que existe entre ellas. Se identifican los requisitos funcionales y no funcionales con los que debe cumplir el sistema como vía para definir las futuras funcionalidades de la aplicación. Además, se elaboran los principales artefactos propuestos por la metodología, se modelan los diagramas, se definen los patrones de diseño y se presenta el modelo de despliegue.

2.1 Descripción de la propuesta de solución

El subsistema de gestión de perfil de usuario para el buscador Orión debe permitir crear un perfil de usuario ajustado a sus preferencias implícitas y explícitas. Las trazas de navegación de los usuarios se utilizarán como indicador para determinar sus preferencias y determinar patrones de uso frecuente. La aplicación contará con dos roles de usuarios: el usuario anónimo y el usuario autenticado. El usuario anónimo solo tendrá acceso a las consultas del buscador con la asignación de un perfil estándar que se modificará continuamente mediante el aprendizaje de sus preferencias en la navegación web que realice. El usuario autenticado poseerá un perfil confeccionado en primera instancia por la introducción manual de datos personales y preferencias; este perfil se actualizará teniendo en cuenta la navegación web que realice recopilando datos de interés, permitirá consultar las trazas de navegación y definir los niveles de preferencia por temas tratados.

2.2 Modelo del dominio

El modelo del dominio es una representación de clases conceptuales significativas en un determinado problema del mundo real, no de componentes de software (Larman, 2004). Su objetivo es comprender y describir las clases más importantes del sistema y ayudar a los usuarios, clientes, desarrolladores y otros interesados a utilizar un vocabulario común mediante la asociación de las entidades o conceptos que se manejan en el entorno en el que trabaja el sistema mediante un diagrama de clases UML (Rumbaugh, 2000).

Con el fin de contribuir a la comprensión del contexto actual del problema, en la Figura 3, se muestra el procedimiento de gestión de perfil de usuario. Se puede observar que el buscador presenta los resultados obtenidos de la consulta al usuario. Mediante su navegación el usuario genera información personalizada con sus preferencias y datos representativos.

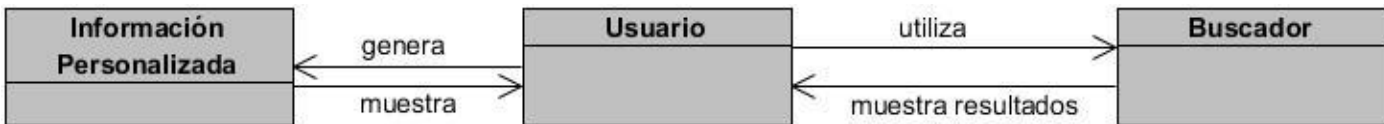


Figura 3. Diagrama de clases del modelo del dominio (Elaboración propia)

Descripción de clases del modelo del dominio

En el modelo del dominio se visualizan las clases o conceptos y sus relaciones más importantes dentro del contexto del problema. A continuación, se presenta la descripción de los conceptos identificados, posibilitando una mayor comprensión sobre el funcionamiento del sistema.

- **Usuario:** es la persona que se encuentra utilizando las facilidades que brinda el buscador Orión.
- **Información Personalizada:** conjunto de características identificativas, preferencias y datos del usuario.
- **Buscador:** constituye una herramienta de recuperación de información en la web.

2.3 Especificación de los requisitos del software

La ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactúan los usuarios finales con el software (Pressman, 2002). Teniendo en cuenta sus características, los requisitos se clasifican en: funcionales y no funcionales (Rumbaugh, 2000).

Requisitos funcionales

Un requisito funcional especifica una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas. Es una condición o capacidad que tiene que ser alcanzada o poseída, para satisfacer un contrato, estándar u otro documento impuesto formalmente (Sommerville, 2007). Seguidamente se enumeran los requisitos funcionales del subsistema objeto de esta investigación:

Tabla 2. Requisitos funcionales (Elaboración propia)

No.	Requisito funcional	Prioridad
RF1	Insertar datos personales de usuario autenticado	Alta
RF2	Modificar datos personales de usuario autenticado	Alta
RF3	Asignar perfil de usuario anónimo	Medio
RF4	Mostrar preferencias	Alta
RF5	Calcular índice de preferencias según consultas realizadas	Alta
RF6	Mostrar consultas realizadas	Alta
RF7	Mostrar datos relacionados a las consultas	Medio
RF8	Filtrar consultas por dato (hora, palabra clave, categoría)	Medio

Requisitos no funcionales

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad (Rumbaugh, 2000). A continuación, se presentan los requisitos no funcionales de la herramienta de gestión de perfil de usuario para el buscador Orión.

✓ Requerimientos de software

RNF 1. Se requiere del sistema operativo Ubuntu 14.04.

RNF 2. Se requiere la instalación 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 1.10.1 y PHP 5.5 o superior para poder visualizar la interfaz web.

✓ Requerimientos de hardware

RNF 5. CPU con 2 núcleos de procesamiento a velocidad de 2.0 GHz.

RNF 6. 2 GB de RAM.

RNF 7. 160 GB de espacio libre en disco.

✓ **Requerimientos de diseño e implementación**

RNF 8. Como lenguaje de programación para la interfaz web se deberá utilizar PHP en su versión 5.5 o mayor.

RNF 10. Para el desarrollo de la aplicación web se deberá utilizar Symfony 2.7 LTS o superior como marco de trabajo.

RNF 11. La aplicación se desarrollará sobre el sistema operativo Ubuntu 14.04 LTS.

✓ **Requerimientos de apariencia o interfaz de usuario**

RNF 12. Todos los iconos serán acompañados de un texto descriptivo en las resoluciones mayores o igual a 992px de ancho.

✓ **Requerimientos de usabilidad**

RNF 14. Debe contar con la portabilidad necesaria para poder ser transferido de un ambiente a otro o reemplazado por nuevas versiones.

RNF 15. Los dispositivos clientes que utilizarán la herramienta deben contar con navegadores web que soporten HTML5 y CSS3.

RNF 16. Se requiere el uso de herramientas y recursos de software libre, las cuales se podrán usar, modificar y distribuir libremente.

✓ **Requerimientos de eficiencia**

RNF 17. La respuesta del sistema debe encontrarse entre los 3 y 5 segundos aproximadamente.

✓ **Requerimientos de seguridad**

RNF 18. Integridad: el sistema debe permitir la realización de salvadas periódicas de la información

RNF 19. Disponibilidad: el sistema debe permitir una instalación distribuida contribuyendo al balanceo de carga y la redundancia.

RNF 20. Los campos de entrada deben ser validados y escapados.

2.4 Modelo de casos de uso del sistema

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un Caso de Uso (en lo adelante CU) representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema (Sparks, 2013). El diagrama de casos de uso del sistema define las relaciones e interacciones entre actores del sistema y casos de uso, además permite guiar el proceso de desarrollo.

Definición de los actores del sistema

Los actores representan personas, otros sistemas o hardware externo que interactúan con el sistema. Un usuario físico puede actuar como uno o varios actores, y varios usuarios pueden actuar como diferentes ocurrencias del mismo actor (Rumbaugh, 2000). A continuación, se definen los actores del sistema y la jerarquía que estos presentan en la aplicación por el rol que juegan en la misma:

- **Usuario autenticado:** Es la persona (usuario) que una vez autenticado en la aplicación puede crear y modificar su perfil de usuario. Este tipo de usuario posee la máxima jerarquía que otorga el subsistema ya que su perfil se actualiza en cada ocasión que utiliza el buscador.
- **Usuario anónimo:** Se le asigna a cualquier persona (usuario) que accede a la aplicación a través de la web para realizar una búsqueda. Posee la jerarquía mínima en el componente. Una vez que abandona el buscador sus datos no se registran.

Diagrama de casos de uso del sistema

Un diagrama de casos de uso describe parte del modelo de casos de uso y muestra la asociación entre cada par actor/CU que interactúan (Rumbaugh, 2000). Para favorecer la organización y el entendimiento de los casos de uso, se muestran cada uno de estos, inicializados por el actor correspondiente, en diagramas separados (Ver Figura 4 y 5):

- **CU 1 Asignar perfil de usuario anónimo**
- **CU 2 Gestionar perfil de usuario autenticado**
- **CU 3 Mostrar preferencias**
- **CU 4 Calcular índice de preferencias**
- **CU 5 Mostrar consultas realizadas**
- **CU 6 Filtrar por datos**

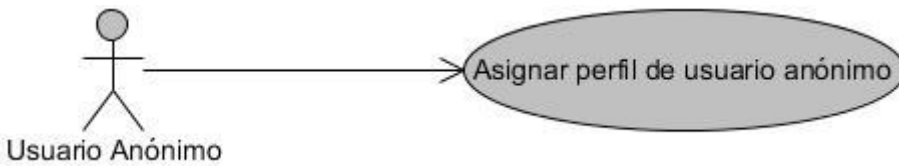


Figura 4. Diagrama de casos de uso del sistema inicializados por el Usuario Anónimo (Elaboración propia)

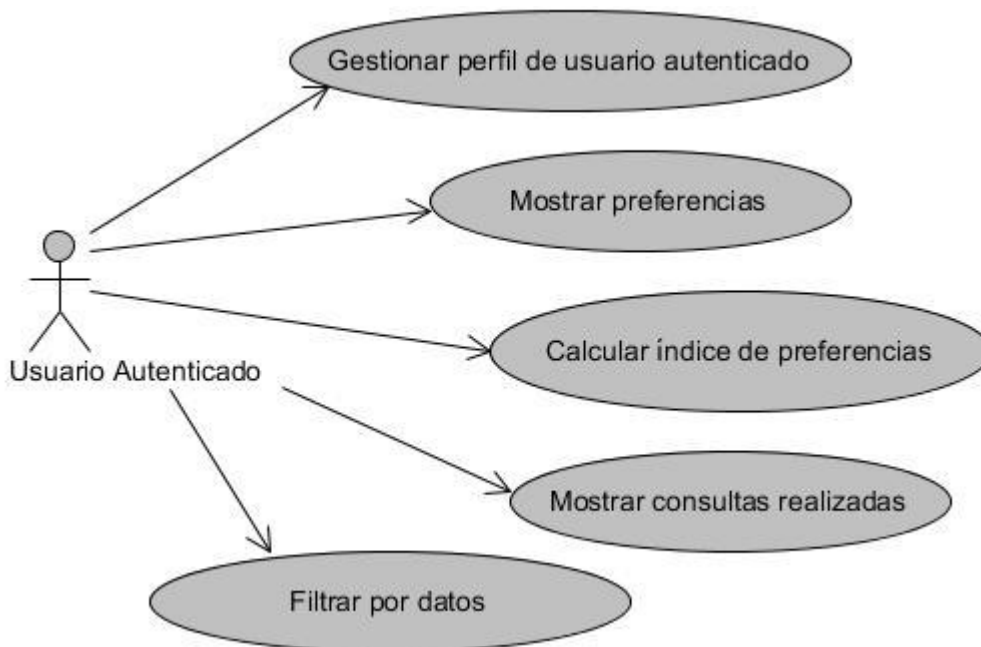


Figura 5. Diagrama de casos de uso del sistema inicializados por el Usuario Autenticado (Elaboración propia)

Descripción de los casos de uso del sistema

El diagrama de casos de uso es insuficiente para comprender las funcionalidades asociadas a cada uno de ellos. Por tal motivo se debe realizar una descripción detallada de los mismos para que se tenga una noción sobre el funcionamiento del software que se desea implementar. A continuación, se realiza la descripción de tres de los CU más críticos del sistema, el resto de ellas se encuentran en el Anexo 2.

Tabla 3. Descripción del CU “Gestionar perfil de usuario autenticado” (Elaboración propia)

Objetivo	Gestionar perfil de usuario en el buscador Orión.	
Actores	Usuario Autenticado (Inicia).	
Resumen	El CU se inicia cuando el usuario desea crear o modificar un perfil de usuario en el buscador Orión. El CU termina cuando se finaliza alguna de estas operaciones.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado.	
Post-condiciones	Se crea o modifica el perfil de usuario en el buscador.	
Flujo de eventos		
Flujo básico “Gestionar perfil de usuario autenticado”		
	Actor	Sistema
1.	Selecciona en la interfaz de perfil de usuario la opción Información Personal .	
2.		<p>Muestra la interfaz de Información Personal que brinda la posibilidad de introducir los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre y Apellidos • Nombre de usuario • Contraseña • Repetir Contraseña • Correo • Género • Celular • Fecha de nacimiento • Categorías <p>Permite:</p> <ul style="list-style-type: none"> • Guardar los datos.

		<ul style="list-style-type: none"> • Cancelar la operación en cualquier momento.
3.	Selecciona la opción de guardar los datos.	
4.		Valida los datos.
6.		Muestra los datos del usuario (Ver sección 1).
7.		Termina caso de uso.
Flujos alternos		
1a. El actor cancela la acción		
	Actor	Sistema
1.	Elige cancelar la acción.	
2.		Elimina los datos creados.
3.		Regresa a la vista principal de la sección en la que se encuentra.
4.		Termina caso de uso.
1b. Existen datos obligatorios o incompletos		
	Actor	Sistema
1.		Muestra un indicador sobre los campos vacíos o incorrectos y muestra los mensajes de información: “Este campo es requerido” o “El valor no es válido”, respectivamente.
2.		Regresa al paso 2 del Flujo básico.
Sección 1: “Modificar datos del usuario”		
Flujo básico		
	Actor	Sistema
1.	.	<p>Muestra la interfaz de Información Personal que brinda la posibilidad de introducir los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre y Apellidos • Nombre de usuario

		<ul style="list-style-type: none"> • Contraseña • Repetir Contraseña • Correo • Género • Celular • Fecha de nacimiento • Categorías
2.	Modifica los datos guardados.	
3.	Selecciona la opción de guardar los datos.	
4.		Valida los datos.
5.		Termina caso de uso

Tabla 4. Descripción del CU “Mostrar preferencias” (Elaboración propia)

Objetivo	Mostrar las preferencias del usuario	
Actores	Usuario Autenticado	
Resumen	El CU se inicia cuando el usuario desea ver sus preferencias de navegación en el buscador Orión. El CU termina cuando el usuario desee abandonar la interfaz.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	El usuario debe tener creado un perfil.	
Post-condiciones	Se muestran las preferencias del usuario en el buscador.	
Flujo de eventos		
Flujo básico “Mostrar preferencias”		
	Actor	Sistema
1.	Selecciona en la interfaz de perfil de usuario la opción Mis Preferencias .	

2.		Muestra la gráfica porcentual correspondiente a cada una de las preferencias del usuario según su navegación en la web.
3.	Examina las consultas realizadas.	
4.		Termina caso de uso.

Tabla 5. Descripción del CU “Mostrar consultas realizadas” (Elaboración propia)

Objetivo	Mostrar historial de todas las consultas realizadas.	
Actores	Usuario Autenticado (Inicia).	
Resumen	El usuario en la interfaz principal del perfil selecciona la opción “ Mi actividad ” donde se le mostrará todas las consultas realizadas en el buscador. El CU termina cuando el usuario desee abandonar la interfaz.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	El usuario debe tener creado un perfil.	
Post-condiciones	Se muestra el historial de consultas realizadas por el usuario.	
Flujo de eventos		
Flujo básico “Mostrar consultas realizadas”		
	Actor	Sistema
1.	Selecciona en la interfaz de perfil de usuario la opción Mi actividad .	
2.		Muestra la interfaz de Mi Actividad que brinda la posibilidad de ver todas las consultas realizadas diariamente por el usuario y sus especificaciones.
3.	Selecciona las especificaciones de las consultas realizadas.	

4.		Ver especificaciones de las consultas realizadas (Ver sección 1).
5.	Examina las consultas realizadas.	
6.		Termina caso de uso.
Sección 1: “Ver especificaciones de la consulta realizada”		
Flujo básico		
	Actor	Sistema
1.	.	<p>Muestra la interfaz de Mi Actividad donde se brinda la posibilidad de consultar los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre de la consulta • URL visitada • Fecha • Hora • Palabra clave • Categoría
2.		Termina caso de uso.

2.5 Arquitectura de la propuesta de solución

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

Para el desarrollo de la solución propuesta se decide utilizar el estilo arquitectónico Modelo-Vista-Controlador, en lo adelante MVC, dado que el *framework* *Symfony2* basa su funcionamiento y estructura en el mismo. No obstante, según su creador Fabien Potencier (2014): "*Symfony* no es un *framework* MVC. *Symfony* sólo proporciona herramientas para la parte del Controlador y de la Vista. La parte del Modelo es responsabilidad del usuario. En la Figura 5 se muestra la arquitectura del sistema desarrollado:

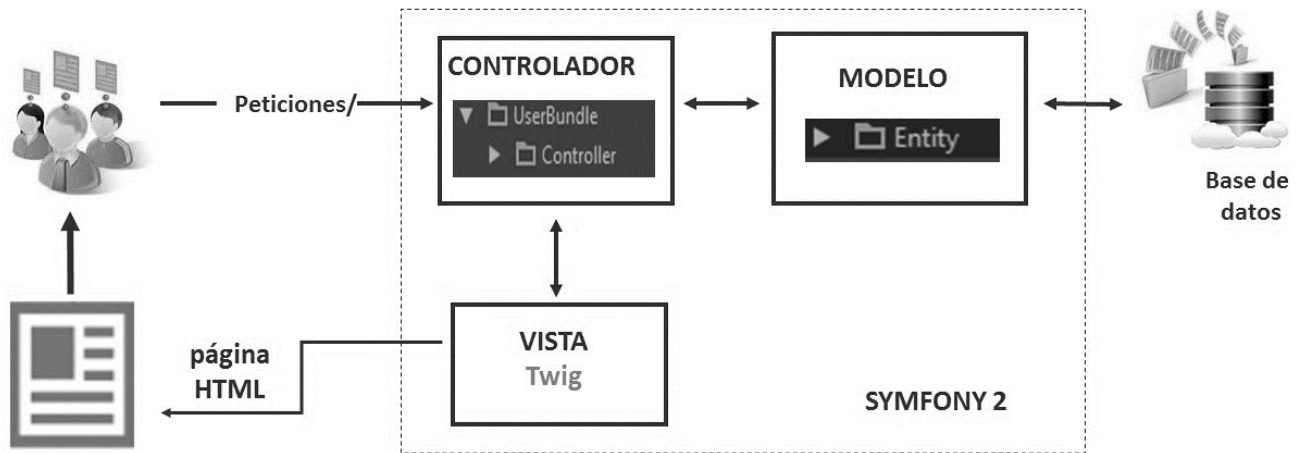


Figura 5. Arquitectura del sistema (Elaboración propia)

En el modelo se encuentran las clases `OrionProfileUser.php`, `OrionUrl.php`, `OrionPreference.php` que permiten la representación de los datos del módulo, evitando mezclar los accesos a bases de datos con el código que implementa la lógica de negocio actuando como capa de abstracción entre el usuario y los datos. El controlador es el centro del módulo donde se implementa la lógica de negocio; responde a las peticiones que hace el usuario y se encarga de realizar las solicitudes de datos al modelo, en el sistema estas clases se encuentran en la carpeta *Controller* del *bundle UserBundle*. La vista es la capa de presentación de los datos de la propuesta de solución, se limita a representar de manera gráfica los datos y todos los elementos que el usuario visualiza dentro la aplicación mediante las clases *.twig* implementadas dentro de *UserBundle/Resources/view*.

2.6 Patrones de diseño

Un patrón es un par problema/solución que se construye basado en la experiencia previa y reiterada con la ocurrencia de alguna situación problemática y que se soluciona de igual manera (Larman, 2004). El *framework Symfony* utiliza en su implementación un conjunto de patrones para dar solución a problemas específicos del diseño orientado a objetos durante el flujo de ejecución de una petición.

En el diseño del subsistema de gestión de perfil de usuario se tuvieron en cuenta los Patrones de *Software* para la Asignación General de Responsabilidad o GRASP (*General Responsibility Assignment Software Patterns*) y Banda de los Cuatro o GOF (*Gang of Four*). A continuación, se muestra una selección de estos patrones:

- **Alta cohesión:** En el sistema se utiliza en la implementación de las clases entidades OrionUrl.php, OrionPreference.php y OrionUser.php. Las clases modelan toda la información y funcionalidades asociadas al objeto url, preferencia y usuario respectivamente, lo cual significa que hay poca dependencia de otras clases externas para su correcto funcionamiento asignando responsabilidades con una alta cohesión.
- **Bajo acoplamiento:** Asigna una responsabilidad a una clase para mantener bajo acoplamiento, o sea, disminuir la dependencia entre clases, evitando que una modificación en alguna de ellas repercuta en gran medida en el resto, posibilitando además una mayor reutilización (Larman, 2004). En la solución la clase *UrlController.php* hereda de *Controller.php*, que es una clase estable en cuanto a su implementación por lo que se obtiene un grado bajo de acoplamiento entre las clases. Además, al no asociar las clases del modelo con las de la vista o el controlador, la dependencia entre las clases, en este caso, se mantiene baja.
- **Patrón creador:** El patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento (Larman, 2004). En *Symfony 2* en la clase *Controller.php* se definen y ejecutan las acciones. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase *Controller.php* es creador de dichas entidades.
- **Patrón controlador:** Asigna la responsabilidad del manejo de mensajes de los eventos del sistema a una o varias clases. Sirve como intermediario entre una determinada interfaz y la acción que debe ejecutarse, es decir, recibe los datos del usuario y los envía a las distintas clases según el método llamado (Larman, 2004). En *Symfony 2* el patrón es utilizado en las clases de la capa del controlador del patrón MVC, como son: *app.php*, *app_dev.php*.
- **Patrón fachada:** La “fachada” satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas (Universidad Politécnica de Madrid, 2005). En *Symfony 2* se puede acceder al manejador de entidades de las formas siguientes: *\$this->getDoctrine()->getEntityManager()* y *\$this->container->get("doctrine")->getEntityManager()* lo que evidencia el uso del patrón. Proporciona una interfaz unificada de alto nivel que, representando a todo un subsistema, facilita su uso.
- **Patrón decorador:** Añade responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de

añadir funcionalidades (Universidad Politécnica de Madrid, 2005). Se puede observar el uso de este patrón en la propuesta de solución en todas las plantillas, ya que todas heredan de una plantilla padre determinada, estableciendo una herencia de tres niveles. En el presente segmento de código fuente se ejemplifica lo antes descrito:

```
index.html.twig
{ % extends 'UIBundle:Layout:layout_subsystems.html.twig' % }
```

2.7 Modelo de diseño

El modelo de diseño es aquel que se encarga de describir la realización de los casos de uso del sistema, y se utiliza como medio de abstracción del modelo de implementación y el código fuente del software. Su objetivo fundamental es transmitir, a través de la representación mediante diagramas, una comprensión en profundidad de los aspectos relacionados con los requerimientos no funcionales y restricciones concernientes a los lenguajes de programación (Larman, 2004).

Diagrama de clases del diseño

Un diagrama de clases del diseño es un tipo de diagrama estático que describe la estructura de un sistema; mostrando sus clases, atributos, métodos y las relaciones entre ellos. Las clases de diseño se especifican utilizando la sintaxis del lenguaje de programación elegido. A continuación, se relaciona el diagrama de clases del diseño correspondiente al CU “Mostrar preferencias”.

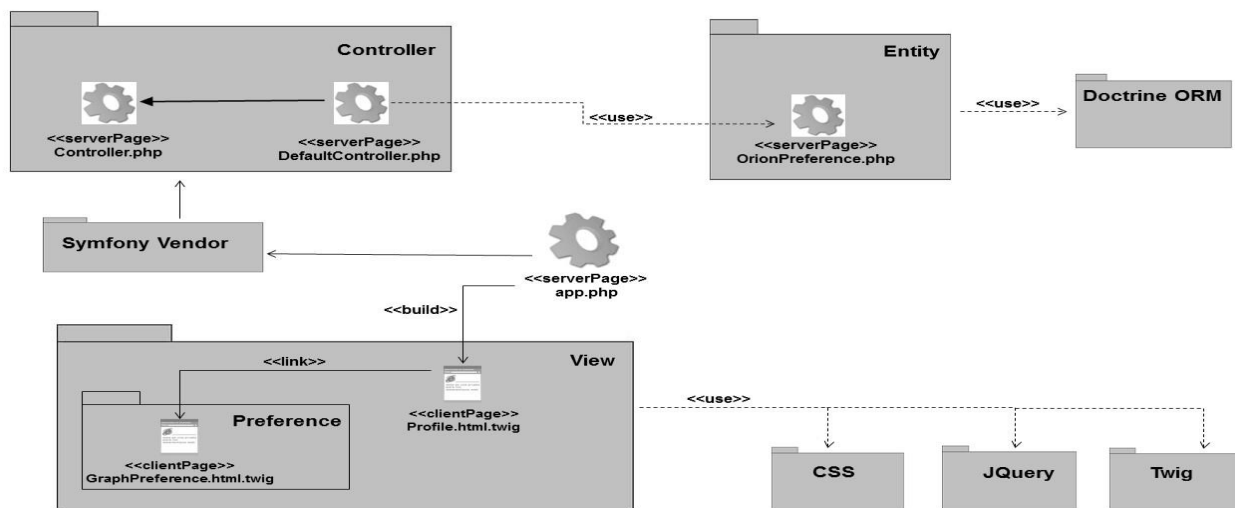


Figura 6. Diagrama de clases de diseño del CU “Mostrar preferencias” (Elaboración propia)

2.8 Diseño de la base de datos

El modelo de datos contiene una representación física de los datos del componente que deben persistir en la base de datos. Como ya se ha definido anteriormente, el gestor encargado de almacenar la información capturada por la aplicación es PostgreSQL, una base de datos documental escalable y de alto rendimiento. En la Figura 7 se muestra el modelo Entidad-Relación de la base de datos del sistema desarrollado.

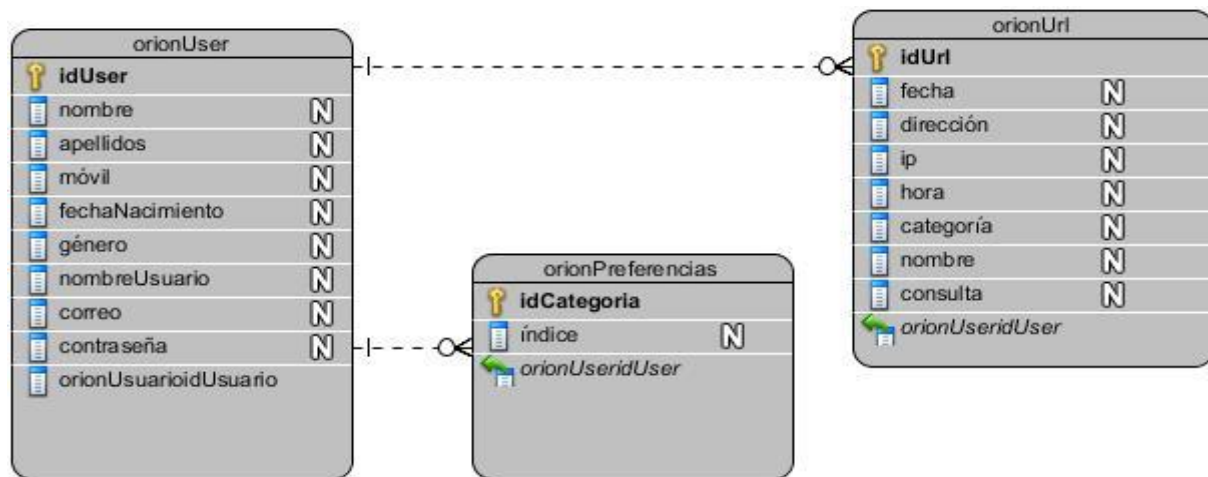


Figura 7. Diagrama del modelo Entidad-Relación de la base de datos (Elaboración propia)

A continuación, se muestra la descripción de las tablas, que conforman el modelo de datos, empleadas por el sistema para su funcionamiento:

- **orionUsuario:** Es la entidad encargada de almacenar los datos del usuario una vez que se autentica en el buscador, además de almacenar los datos personales correspondientes al usuario. Está asociada a muchas preferencias de usuario y a muchas url.
- **orionPreferencias:** Registra las preferencias del usuario. Está asociada a un usuario.
- **orionUrl:** Entidad encargada de registrar todas las url visitadas por el usuario, así como el conjunto de atributos que genera estas consultas en el buscador. Está asociado a un usuario.

2.9 Modelo de despliegue

Los elementos del diseño al nivel del despliegue indican cómo se ubicarán la funcionalidad y los subsistemas dentro del entorno computacional físico que soportará al software (Pressman, 2002). En el modelo de despliegue se define la organización física del sistema en términos de nodos de cómputo y se verifica que los casos de uso puedan implementarse como componentes que se ejecutan en esos nodos (Rumbaugh, 2000).

El siguiente diagrama de despliegue describe la arquitectura física del sistema de gestión de perfil de usuario durante la ejecución, en términos de procesadores, dispositivos y componentes de software (Ver figura 8). Permite una mejor comprensión entre la correspondencia de la arquitectura de software y la arquitectura de hardware.



Figura 8. Diagrama de despliegue (Elaboración propia)

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

Servidor de aplicaciones web NGINX: Aquí se encuentra el código fuente de la aplicación, brinda a los usuarios las interfaces necesarias durante todos los procesos del negocio. Esta estación se comunica con el servidor de base de datos PostgreSQL donde se almacenan los datos de los perfiles de usuarios, realizando la comunicación a través del protocolo TCP.

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

2.10 Conclusiones parciales

En este capítulo se abordaron una serie de aspectos correspondientes al análisis y diseño del subsistema de gestión de perfil de usuario llegándose a la siguiente conclusión:

- La representación y descripción de los artefactos generados garantizaron un mejor entendimiento de los flujos de trabajos presentes en el proceso de gestión del perfil de usuario.
- La especificación de los requisitos funcionales y no funcionales del sistema, dieron paso a una mejor comprensión de los resultados que se pretenden obtener de una manera precisa y sirvieron de guía para la implementación del sistema.
- La arquitectura MVC seleccionada permitió definir la estructura del software y su interrelación entre los diferentes componentes.
- El diagrama de despliegue propuesto permitió exponer cómo se relacionan los elementos de hardware del sistema.

3

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SUBSISTEMA DE PERFIL DE USUARIO PARA EL MOTOR DE BÚSQUEDA ORIÓN

La implementación del sistema es una de las fases imprescindibles dentro del proceso de desarrollo de software. 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).

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

3.1 Modelo de componentes que integran el sistema

El modelo de componentes representa la forma en que es estructurado un sistema informático atendiendo a las diferentes partes o componentes físicos de un sistema, así como las relaciones entre ellos (IEEE, 2004). El modelo enunciado permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces (Microsoft, 2016).

Diagrama de componentes

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

A continuación, se describen los principales paquetes que componen el diagrama de componentes correspondiente al subsistema de gestión de perfil de usuario:

- **UserBundle:** Agrupa en su interior todos los componentes de la interfaz web, estableciendo una estructura organizativa acorde al patrón de arquitectura MVC.

- **Controller:** Contiene la clase controladora encargada de procesar las peticiones del usuario.
- **Form:** Alberga las clases para construir los formularios utilizados para el envío de información por parte de los usuarios.
- **Config:** Contiene en su mayoría los archivos donde se definen las rutas de la aplicación.
- **Views:** Agrupa las páginas referentes a las vistas de la aplicación, así como las plantillas bases.
- **Document:** En este paquete residen las clases documentos que se corresponden con las colecciones de la base de datos en PostgreSQL.
- **Resources:** Contiene los paquetes relacionados con las vistas y las configuraciones del módulo.

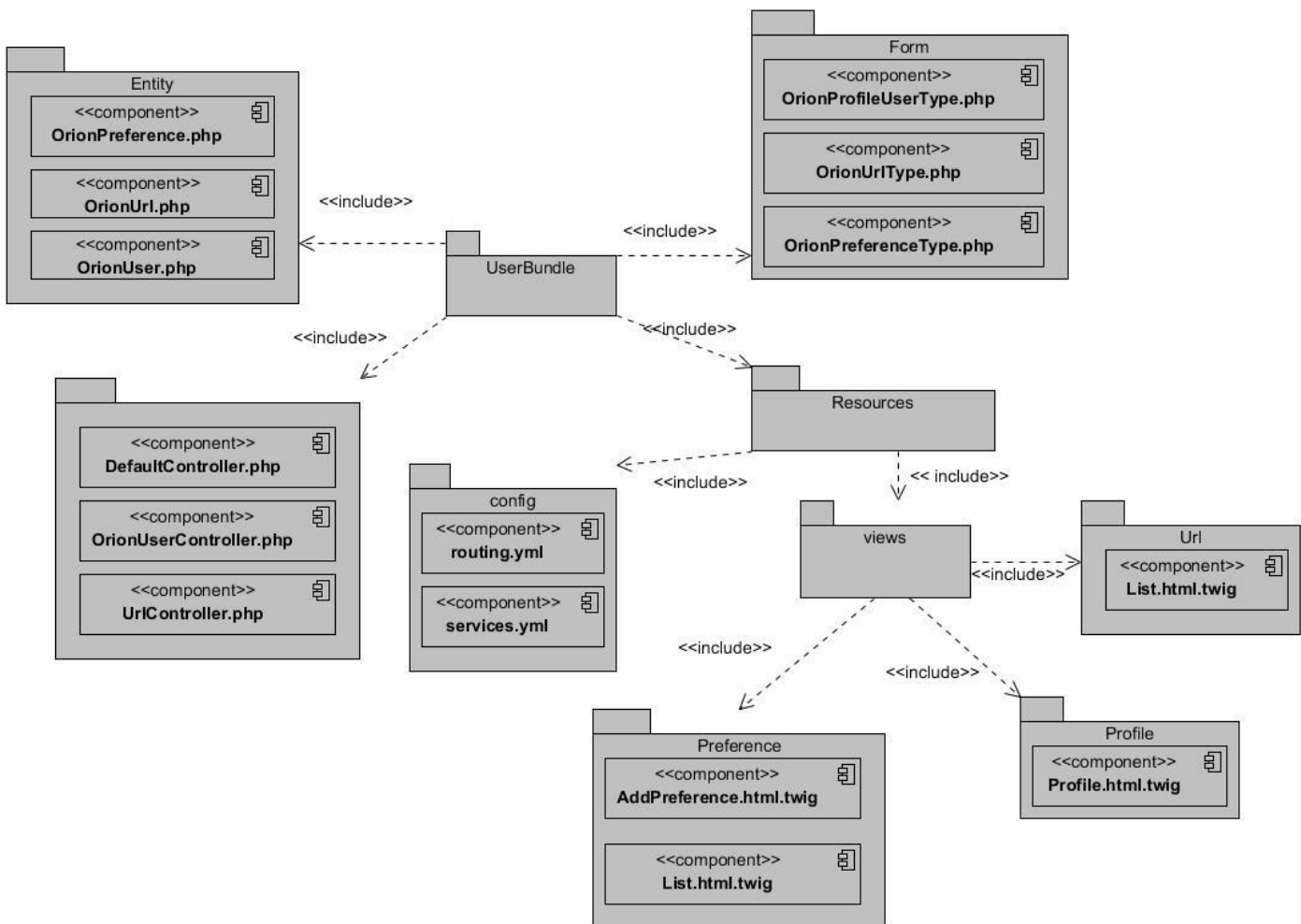


Figura 9. Diagrama de componentes correspondiente a la interfaz web del subsistema de gestión de perfil de usuario para el buscador Orión (Elaboración propia)

3.2 Estándares de codificación

Los estándares de codificación son especificaciones o estilos que establecen la forma de generar el código de las aplicaciones informáticas. En ocasiones, los sistemas de cómputo son implementados por varios programadores, la adopción inicial de un único estilo de codificación constituye uno de los factores de mayor peso en la calidad, rendimiento, legibilidad y capacidad de mantenimiento del producto final (Studios.NET, 2016).

El estándar de codificación utilizado en el lenguaje PHP es el que establece el marco de trabajo Symfony2 que sigue los estándares definidos en los documentos PSR-1, PSR-2, PSR-3, PSR-4 ((SENSIOLABSNETWORK, 2014); (PHP FRAMEWORK INTEROP GROUP (PSR-1), 2016); (PHP FRAMEWORK INTEROP GROUP (PSR-2), 2016); (PHP FRAMEWORK INTEROP GROUP (PSR-4), 2016)). Entre sus elementos más relevantes y comunes se encuentran:

- Añadir un espacio después de cada delimitador coma ', '.

```
->add('email', null, array('label' => 'Correo'))
->add('username' , null, array('label' => 'Usuario'))
```

- Añadir un único espacio a ambos lados de un operador: =, ==, && etc.

```
$em = $this->get('doctrine.odm.mongodb.document_manager');
$user = $em->getRepository('OrionUser')->findOneById($id);
```

- En los *array* multilínea, añade una coma al final de cada elemento.

```
->add('password', 'repeated', array(
    'type' => 'password',
    'invalid_message' => 'Las contraseñas deben coincidir',
    'first_options' => array('label' => 'Contraseña', 'attr' => array('class' => 'form-control')),
    'second_options' => array('label' => 'Confirmar contraseña', 'attr' => array('class' => 'form-control'))
))
```

- Añade un salto de línea antes de una sentencia *return*, a menos que el *return* se encuentre solo en un bloque de sentencias, y un salto después de cada llave cierre de sentencia, excepto después de la llave de cierre de clase.

```

foreach ($resultset as $document) {
    array_push($results, $document['textsuggest']);
}

return $results;
}

```

- Usa notación camelCase sin guiones bajos en variables, funciones, métodos y argumentos.

```

public function newPreferenciaAction()
{
    $entity = new OrionPreferencia();
    $form = $this->createForm(new PreferenciasType(), $entity);
    return $this->render('ORIONUser:Preferencia:addPreferencia', array(
        'form' => $form->createView()
    ));
}

```

- El nombre de las clases se realiza en UpperCamelCase, es decir, que comienza por mayúscula.

```
class OrionProfileUserType extends AbstractType
```

- Declarar explícitamente la visibilidad de las clases, métodos y propiedades (el uso de var está prohibido).

```

public function getName() {
    return 'orion_profiletype';
}

```

- Utiliza constantes de tipo PHP nativas en minúsculas: *false*, *true* y *null*. Lo mismo aplica para los *array()*.

```

->add('gender', 'choice', ['label'=> 'Género',
    'expanded' => true,
    'choices' => array('masculino'=> 'Masculino', 'femenino'=> 'Femenino'),
    'required' => false,
    'empty_value' => false
]);

```

- Utilizar los *namespace* para todas las clases.

```
namespace ORION\Bundle\UserBundle\Form;
```

3.3 Validación del sistema

La validación del sistema incluye un conjunto de actividades para asegurar que el software desarrollado se corresponde con los requisitos del cliente. Dentro de estas actividades se encuentran las pruebas de validación, las cuales tienen como objetivo evaluar la calidad y de manera más pragmática, descubrir errores en el sistema (Pressman, 2006). A continuación, se describen las pruebas realizadas al subsistema de gestión de perfil de usuario para el buscador cubano Orión, así como los resultados obtenidos.

Pruebas funcionales

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

Diseño de los casos de prueba basados en casos de uso

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (Pressman, 2002).

A continuación, se plantean los casos de pruebas más significativos para algunos requisitos con prioridad alta en el subsistema de gestión de perfil de usuario para el motor de búsqueda Orión. El resto de los diseños de casos de prueba se pueden encontrar en el Anexo 3.

Tabla 6. Variables empleadas en el diseño del caso de prueba basado en el CU “Gestionar perfil de usuario autenticado”
(Elaboración propia)

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1.	Nombre	Campo de texto	No	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños, que no excedan un total de 60 caracteres.
2.	Apellidos	Campo de texto	No	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños, que no excedan un total de 60 caracteres.
3.	Fecha de nacimiento	Campo de fecha	No	Se debe seleccionar una fecha de las preestablecidas.
4.	Género	Campo de texto	No	Se debe seleccionar una sola opción de las mostradas.
5.	Usuario	Campo de texto	No	Se deben escribir una palabra comenzando por minúscula y sin caracteres extraños, que no excedan un total de 60 caracteres.
6.	Celular		No	Se debe insertar un número que no exceda de los 8 caracteres y siempre comience con el número 5.
7.	Correo	Campo de texto	No	Se debe insertar una cadena de texto, que coincida con la estructura válida de un correo.
8.	Contraseña	Campo de texto	No	Se debe insertar una cadena de texto que no exceda de 60 caracteres.
9.	Verificar Contraseña	Campo de texto	No	Se debe insertar una cadena de texto que no exceda de 60 caracteres, idéntica a la del campo “Contraseña”.
10	Temas de búsqueda	Campo de texto	No	Se debe seleccionar uno o más de los temas de búsquedas que se muestran.

Las celdas de la tabla contienen V (indica válido), I (indica inválido), o N/A (no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante).

Tabla 7. Caso de prueba correspondiente al CU “Gestionar perfil de usuario autenticado” (Elaboración propia)

Escenario	Descripción	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	Respuesta del sistema	Flujo Central
EC 1.1	El usuario autenticado selecciona dentro de las opciones de su perfil ver Información Personal	N/A	N/A	N/A	N/A	V	N/A	V	V	N/A	N/A	Muestra la interfaz de Datos Personales que brinda la posibilidad de introducir los siguientes datos: <ul style="list-style-type: none"> -Nombre y Apellidos -Nombre de usuario -Contraseña -Repetir Contraseña -Correo -Género -Celular -Fecha de nacimiento -Categorías Permite: Guardar los datos	El usuario selecciona en la página principal la opción Mi Perfil y dentro de ella selecciona a opción de Información

												Cancelar la operación en cualquier momento	maci ón pers onal
EC 1.2	El usuario introduce de manera correcta los datos en el sistema	V	V	V	V	V	V	V	V	V	V	El sistema registra al usuario.	El usuario rellena los campos de forma correcta y da clic en el botón Aceptar.
		I	V	V	I	V	V	V	V	V	V		

EC 1.3 Campos vacíos	El usuario deja uno o más campos vacíos.	vacío	González Rodríguez	8-enero-1992	vacío	aglez	54405731	aglez@estudiantes.ucu	ojala	ojala	Cultura	El sistema muestra un mensaje: "Rellene este campo".	El usuario llenando los campos deja uno o más vacíos y da clic en el botón Aceptar.
		V	V	V	I	V	I	V	V	I	I		
		Arlenis	González Rodríguez	8-enero-1992	vacío	aglez	vacío	aglez@estudiantes.ucu	ojala	vacío	vacío		
		I	I	I	I	V	V	V	V	V	I		
		vacío	vacío	vacío	vacío	aglez	54405731	aglez@estudiantes.ucu	ojala	ojala	vacío		

								i.c u					
		V	I	I	I	V	I	V	V	V	V		
		Arl eni s	vacío	va cío	vací o	ag le z	vací o	agl ez @ est udi an tes .uc i.c u	oja la	oja la	Cul tur a		
EC 1.4	El usuari o introd uce uno o más valore s incore ctos	V	I	V	V	V	V	V	V	V	V	El sistema	El
Introdu cir datos incore ctos	usuari o introd uce uno o más valore s incore ctos	Arl eni s	892	8- en ero - 19 92	fem enin o	ag le z	5440 5731	agl ez @ est udi an tes .uc i.c u	oja la	oja la	Cul tur a De por te	muestra un mensaje: "Ajústese al formato solicitado".	usuar io introd uce uno o más datos incore ctos y da clic en el botón
		V	V	V	V	V	I	V	V	I	V		
		Arl eni s	Gonz ález Rodr íguez	8- en ero -	fem enin o	ag le z	544	agl ez @ est udi	oja la	Oj ala	Cul tur a		

			19 92					an tes .uc i.c u			De por te		Accept ar.
		I	I	V	V	V	I	V	V	V	V		
		23 33	Gon 4lez	8- en ero - 19 92	fem enin o	ag le z	5440	agl ez @ est udi an tes .uc i.c u	oja la	oja la	Cul tur a De por te		

Resultados de las pruebas funcionales

Con el objetivo de probar el correcto funcionamiento de las funcionalidades del sistema se realizaron cuatro iteraciones de pruebas al sistema. En la Tabla 8 se muestran los resultados obtenidos en cada iteración de prueba al subsistema de gestión de perfil de usuario para el motor de búsqueda Orión, así como la corrección de cada uno de los errores.

Tabla 8. Cantidad de no conformidades por cada iteración de las pruebas funcionales (Elaboración propia)

No Conformidades	1era iteración	2da iteración	3era iteración	4ta iteración
Detectadas	10	7	3	0
Resueltas	6	8	6	0
Pendientes	4	3	0	0

Durante su desarrollo se detectaron un total de 20 no conformidades que fueron corregidas satisfactoriamente. Los principales errores detectados fueron ortográficos, estructurales dados por la inexistencia de botones para deshacer acciones y validaciones incorrectas. Para apreciar de forma ilustrativa el comportamiento de las no conformidades, se muestra en la Figura 10 los resultados obtenidos en las cuatro iteraciones de pruebas funcionales realizadas al sistema.

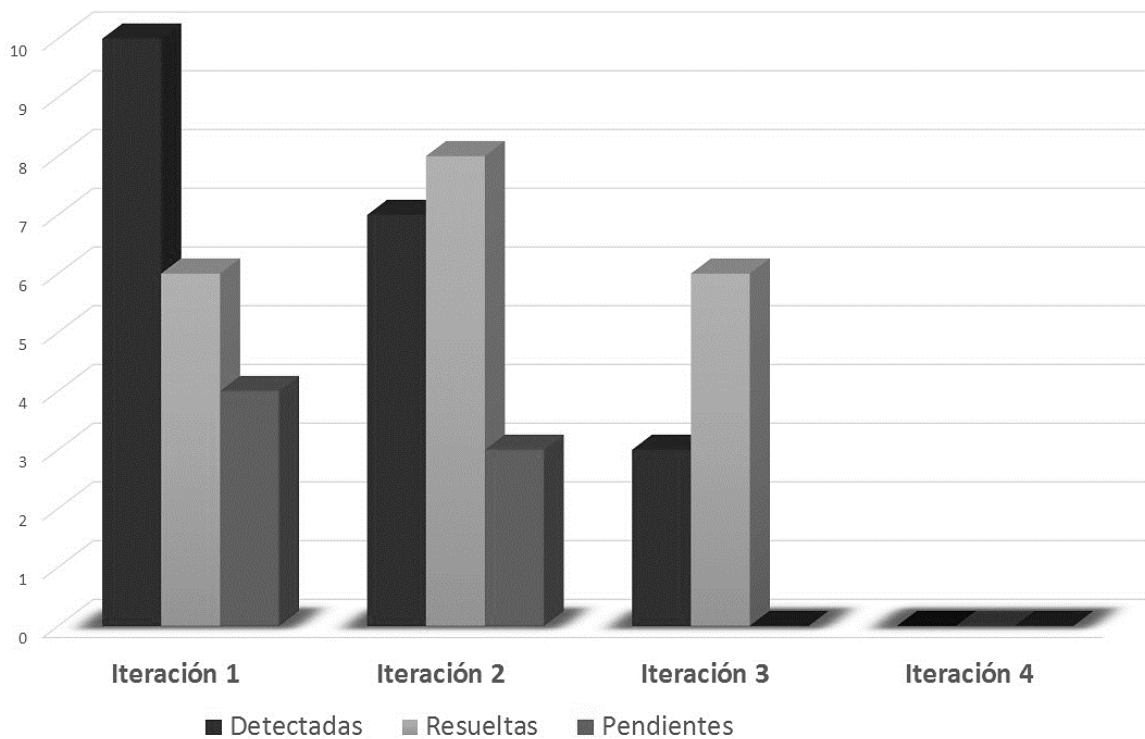


Figura 10. Comportamiento de las no conformidades por cada iteración de las pruebas funcionales (Elaboración propia)

Pruebas de integración

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático. Las mismas validan que estos componentes realmente funcionan juntos, son llamados correctamente y además, transfieren los datos adecuados en el tiempo preciso y por las vías de comunicación establecidas (Sommerville, 2007).

En el caso de la solución desarrollada, para la verificación de una correcta interoperabilidad entre el sistema desarrollado y el buscador Orión, se llevaron a cabo las acciones siguientes:

- Integración entre el sistema de gestión de perfil de usuario con el motor de búsqueda Orión.
- Verificación de la conexión entre el sistema y Solr.

Se muestra, en la Tabla 9, los resultados de la ejecución de las dos iteraciones de las pruebas de integración realizadas al sistema:

Tabla 9. Cantidad de no conformidades por cada iteración de las pruebas funcionales (Elaboración propia)

No Conformidades	1era iteración	2da iteración
Detectadas	0	0
Resueltas	0	0
Pendientes	0	0

Mediante la ejecución de las pruebas de integración se logró verificar la operación conjunta de cada uno de los componentes del sistema de gestión de perfil de usuario del buscador Orión, junto a los componentes que conforman el *bundle* de usuarios de Orión (ORIONUserBundle) y *layout* del *bundle* en que se definen las vistas principales de Orión (UIBundle), haciendo énfasis en la interacción entre estos y comprobando el correcto funcionamiento de los mismos. No se encontró ninguna no conformidad, llegándose a la conclusión que existe una correcta integración entre los componentes internos del sistema.

Pruebas de rendimiento

Mediante las pruebas de rendimiento es posible hallar tendencias y comportamientos para los elementos de una aplicación. Estas pruebas permiten identificar cuellos de botella, capacidad de concurrencia de usuarios, tiempos de respuesta de operaciones del negocio a nivel de sistema, establecer un marco de referencia para pruebas futuras, determinar el cumplimiento de los objetivos de rendimiento y requerimientos no funcionales, entre otros (V&V Quality, 2013).

Dependiendo del objetivo perseguido se puede diseñar distintos tipos de pruebas, entre ellas se puede encontrar las pruebas de carga y las pruebas de estrés (QATécnico, 2012). Una prueba de carga se realiza generalmente para determinar y validar la respuesta de la aplicación cuando es sometida a una carga de usuarios y/o transacciones que se espera en el ambiente de producción (Rincon, 2015). Esta carga puede ser el número esperado de usuarios concurrentes, que realizan un número específico de transacciones durante el tiempo que dura la carga, utilizando la aplicación.

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

Resultados de las pruebas de rendimiento

Para las pruebas de carga y estrés realizadas al subsistema de gestión de perfil de usuario para el buscador Orión fue utilizada la herramienta *JMeter* v2.8.2. Durante la realización de las pruebas se simuló el entorno donde debe interactuar el sistema para obtener una correcta información de comportamiento y resultados en general. El escenario elegido posee las siguientes características.

Hardware:

- Tipo de procesador: corei5 2450M a 2.50 GHz
- Memoria: 2GB RAM.
- Tipo de Red: Ethernet 10/100Mbps.

Software:

- Tipo de Servidor Web: NGINX 1.10.1.
- Máximo de hilos concurrentes (simulación de usuarios): 100.
- Plataforma: sistema operativo Ubuntu 14.04 de 32 bits.
- Servidor de BD: PostgreSQL 9.5.

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

- **Usuarios:** total de usuarios.
- **Número de muestras:** El número de peticiones.
- **Media:** El tiempo medio transcurrido para un conjunto de resultados.
- **Mín.:** El mínimo tiempo transcurrido en milisegundos para las muestras de la URL dada.
- **Máx.:** El máximo tiempo transcurrido en un milisegundo para las muestras de la URL dada.

- **% Error:** Porcentaje de las peticiones con errores.
- **Rendimiento:** Rendimiento medido en base a peticiones por segundo/minuto/hora.

Tabla 10. Resultados de pruebas de carga y estrés (Elaboración propia)

Usuarios	Número de muestras	Media	Mín.	Máx.	% Error	Rendimiento (peticiones/segundos)
100	2105	2056	2	65022	0	12.5

El sistema desarrollado, para un total de 100 usuarios conectados de forma concurrente respondió 2105 peticiones al servidor en un promedio de 2.056 segundos, lo que equivale a 12.5 peticiones por segundo. Atendiendo a la cantidad de peticiones por cada segundo que se enviaron, las prestaciones del hardware donde se realizaron las pruebas y teniendo en cuenta lo establecido en el RNF 17, se considera que, de manera general, es un resultado satisfactorio para la presente investigación.

Pruebas de seguridad

Se considera un producto (de software) seguro aquel que protege la confidencialidad¹⁸, integridad¹⁹ y disponibilidad²⁰ de los recursos y el servicio, bajo el control del propietario o administrador del sistema (Leblanc, y otros, 2006).

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

- Su comportamiento es predecible y seguro.
- No expone ninguna vulnerabilidad o debilidades.
- Las rutinas que se ocupan de los errores y excepciones le permiten que mantenga un estado seguro frente a modelos de ataque o fallas intencionales.
- Satisface todos los requisitos no funcionales de seguridad especificados e implícitos.

¹⁸ Sólo los "usuarios" debidamente autorizados deben tener acceso a los datos y/o los recursos.

¹⁹ Sólo los usuarios debidamente autorizados pueden modificar los datos y/o los recursos de modo apropiado y controlado.

²⁰ Los recursos deben poder ser accedidos cuando son necesarios y funcionar a un nivel aceptable.

- No viola ninguna restricción de seguridad especificada.
- El código ha sido disimulado para evitar la ingeniería inversa.

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

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

Se encontraron seis alertas en total, de ellas ninguna de clasificación alta, una de clasificación media, cuatro de clasificación baja y una informativa. De manera general los resultados obtenidos se presentan en la tabla 11 y 12.

Vulnerabilidades encontradas

Tabla 11. Vulnerabilidades del entorno (Elaboración propia)

Tipo	Cantidad	Descripción	Recomendaciones
Software	1	La versión de PHP instalada contiene vulnerabilidades que han sido solucionadas en versiones más recientes.	Actualizar a la versión 5.6.4 o superior.
Configuración	4	La configuración del servidor permite la captación de información sensible por el atacante.	Esta vulnerabilidad aunque no permite al atacante tomar el control acerca de la información a obtener puede proporcionar al atacante información sensible que puede utilizar para realizar ataques más específicos, pueden solucionarse limitando el acceso a los archivos del servidor y los elementos de la configuración ajustando el "expose_php" en "off" en el archivo php.ini.

Tabla 12. Vulnerabilidades del sistema (Elaboración propia)

Tipo	Cantidad	Descripción	Recomendaciones
Falso positivo	1	El sistema permite la inyección de código.	Esta vulnerabilidad no puede ser utilizada por el atacante para alterar el sistema ni obtener información por lo que se denomina falso positivo.

Todas las vulnerabilidades fueron resueltas en el servidor y se recomienda que se tenga en cuenta la nueva configuración en el servidor para un futuro despliegue del sistema, llegando a la conclusión que el sistema desarrollado es seguro y está en condiciones de ser usado por el cliente con el propósito de generar perfiles de usuario en el motor de búsqueda Orión.

Validación de la hipótesis de la investigación

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

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

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

Selección de expertos

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

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

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

Tabla 13. Expertos utilizados en la validación de la investigación (Elaboración propia)

No.	Experto	Entidad	Años de experiencia
1	Ing. Miguel A. Chávez Alfonso	CIDI	7
2	Ing. Paúl Rodríguez Leyva	CIDI	8
3	Ing. Eric Bárbaro Utrera	CIDI	3
4	Ing. Tamara Betancourt Santana	CIDI	7
5	Ing. José Gabriel Espinosa Ramírez	CIDI	7
6	Ing. Michel Lázaro Frómata Burey	CIDI	3
7	Ing. Leiny Amel Pons Flores	CIDI	8

Realización de la consulta a expertos

Se somete a consideración de los expertos seleccionados un instrumento evaluativo, disponible en el Anexo 4, compuesto de cuatro sentencias para la validación del subsistema de gestión de perfil de usuario. La variable independiente se evaluó mediante las sentencias de interfaz intuitiva, visualización de consultas realizadas y recopilación de datos del usuario durante la interacción con el buscador, mientras que, la evaluación de la variable dependiente midió el nivel de actualización de preferencias del usuario. Para conocer la opinión de los expertos, se emplearon cinco categorías evaluativas: muy adecuado (MA), bastante adecuado (BA), adecuado (A), poco adecuado (PA) e inadecuado (I).

Análisis de la concordancia de las valoraciones de los expertos

En un segundo momento se calcula el coeficiente de Kendall que permite analizar la concordancia en las valoraciones realizadas por los expertos (Sampieri, 2014). En este caso el coeficiente concordancia (W) será un índice de la divergencia del acuerdo efectivo entre los expertos. El coeficiente de concordancia de Kendall se obtiene de la expresión: $W = 12S/K^2(N^3 - N)$.

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

Cálculo del coeficiente de concordancia de Kendall: $W = 12S/K^2(N^3 - N)$

$$W = 12 * 76.08/7^2(7^3 - 7)$$

$$W = 0.951$$

El valor de W oscila entre 0 y 1. El valor de 1 significa una concordancia de acuerdo total y el valor de 0 un desacuerdo total. Como **W = 0.951** existe concordancia de acuerdos entre los expertos.

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

Cálculo del Chi cuadrado real: $\chi^2 = K(N - 1)W$

$$\chi^2 = 11.4$$

El Chi cuadrado calculado se compara con el tabulado en la tabla del percentil de la distribución Chi cuadrado. Para tener un 95% de confianza se utilizará $\alpha = 0.05$. Si se cumple que $\chi^2_{calculada} < \chi^2(\alpha, N - 1)$, entonces se infiere que existe concordancia de criterios entre los expertos al considerar válida la hipótesis alternativa H1. Como $11,4 < 33,924$ existe concordancia entre los expertos.

Procesamiento estadístico

Los criterios aportados por los expertos se someten a una prueba estadística no paramétrica que permite concluir qué valoración final tiene cada uno de los aspectos a evaluar. Para los datos anteriores se debe confeccionar una distribución de frecuencia a partir de los datos primarios para cada uno de los aspectos sometidos a consulta (Castro, 2014).

Tabla 14. Distribución de frecuencia para los datos primarios obtenidos (Elaboración propia)

Categorías evaluativas	Frecuencia Absoluta	Frecuencia Relativa
Muy Adecuado	15	0,75
Bastante Adecuado	5	0,125
Adecuado	5	0,125
Poco adecuado	0	0,0
Inadecuado	0	0,0

Los resultados obtenidos en la validación pueden observarse en la figura siguiente:

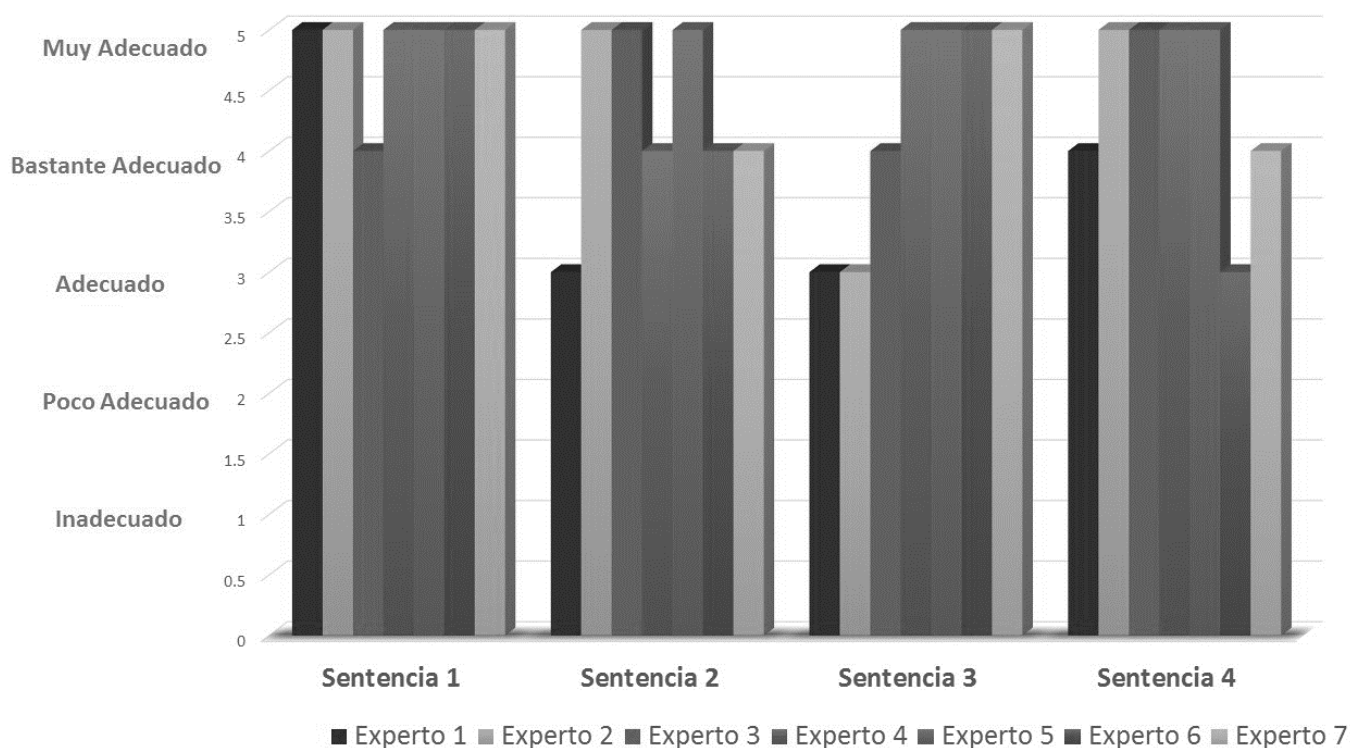


Figura 11. Comportamiento de la valoración de los expertos por categorías evaluadas (Elaboración propia)

De acuerdo con los datos el 75 % de los aspectos analizados fueron valorados de muy adecuado, el 12.5 % de bastante adecuado y el 12.5 % adecuado. El análisis de los resultados obtenidos de la consulta de expertos permitió identificar que existe una coincidencia en las valoraciones realizadas sobre el alto valor



que presenta el subsistema de gestión de perfil de usuario, evidenciando satisfactoriamente la calidad de la propuesta presentada.

3.4 Conclusiones parciales

En este capítulo se abordaron una serie de aspectos correspondientes a la implementación y validación del subsistema de gestión de perfil de usuario para el buscador Orión llegándose a las siguientes conclusiones:

- La representación y descripción del diagrama de componentes permitió visualizar con más facilidad la estructura general de la herramienta.
- La utilización de estándares de codificación de código permitió adoptar una estructura homogénea que facilita la comunicación, asegura la calidad, y el fácil mantenimiento del sistema.
- La ejecución de pruebas a la herramienta permitió detectar las deficiencias presentes, subsanarlas en el menor tiempo posible y ofrecer una aplicación con mayor calidad, seguridad y usabilidad.

Conclusiones

Una vez completada la presente investigación, se puede concluir que:

- El estudio del estado del arte de los sistemas de recuperación de información y el análisis de las diferentes herramientas y tendencias para la gestión de perfiles de usuarios permitieron determinar las características que constituyen la base para el diseño de las funcionalidades que se definen en la propuesta de solución.
- La elaboración de los artefactos propuestos por la metodología de desarrollo y el levantamiento de requisitos permitieron una mayor comprensión, identificación de los procesos y características del subsistema desarrollado.
- El subsistema implementado permitió solucionar los problemas existentes planteados en la problemática de la presente investigación relacionados con la gestión de perfil de usuario, así como, la identificación de sus preferencias en el buscador Orión.
- Las pruebas de software aplicadas al subsistema de gestión de perfil de usuario, demostraron que es una solución funcional, segura, con un rendimiento adecuado y que se integra sin dificultad al buscador Orión.
- La validación de la hipótesis mediante el criterio de expertos demostró que el subsistema de gestión de perfil de usuario contribuye identificar sus preferencias en el buscador Orión.



Recomendaciones

Durante la realización del subsistema y como resultado de la investigación realizada, surgieron algunas ideas que pueden ampliar las prestaciones del software creado. Se recomienda profundizar en:

- La aplicación de algoritmos de minería de datos que permitan mejorar la recuperación de información publicada en la web a través de las preferencias del usuario.
- Agrupar sesiones de usuario, mediante un algoritmo de *clustering* difuso c-medias, para obtener diferentes grupos de usuarios con características similares que permitan identificar tendencias de comportamiento en la web para generar recomendaciones personalizadas.

Referencias

- Acunetix. 2016.** Audit your website security with Acunetix Web Vulnerability Scanner. [Online] 2016. [Cited: diciembre 5, 2016.] <http://www.acunetix.com/>.
- Baeza-Yates, R. y Ribeiro-Neto, B. 1999.** *Modern Information Retrieval*. Addison Wesley : ACM Press, 1999.
- Baeza-Yates, R. 2005.** "Excavando en la web". s.l. : El profesional de la información, 2005.
- Bakken, S. S. 2013.** *Manual de PHP*. s.l. : The PHP Documentation Group, 2013.
- Betancourt, D. ; Moreno, J. y Ovalle, D. 2009.** *Modelo para la recomendación y recuperación de objetos de aprendizaje en entornos virtuales de enseñanza/aprendizaje*. Revista Avances en Sistemas e Informática Vol 6 No 1., 2009.
- Bitelia. 2015.** [Online] 2015. [Cited: diciembre 9, 2016.] <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>.
- Blázquez, O. 2013.** *Manual Técnicas avanzadas de recuperación de información: procesos, técnicas y métodos*. Madrid. ISBN 978-84-695-8030-1, 2013 : ISBN 978-84-695-8030-1, 2013.
- Bonett, M. 2001.** "Personalization of Web Services: Opportunities and Challenges". [Online] 2001. [Cited: noviembre 17, 2016.] <http://www.ariadne.ac.uk/issue28/personalization/>.
- Camiño, R. 2003.** Motores de búsqueda sobre salud en Internet. [Online] 2003. [Cited: noviembre 16, 2016.] scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-.
- Carrillo, G. y Ramírez, Y. 2012.** Colocándonos en la web. [Online] Instituto Internacional de Periodismo "José Martí". Diplomados, Cursos, Talleres para periodistas de Cuba y A. Latina, 2012. <https://periodismojosemarti.wordpress.com/2012/11/28/colocandonos-en-la-web/>.
- Castro, L. 2014.** *Guía de gestión del riesgo tecnológico para el tratamiento de la seguridad durante el proceso de desarrollo de software*. 2014.
- Consortium, W. 2014.** World Wide Web Consortium(W3C). [Online] 2014. <http://www.w3.org/Style/CSS/current-work..>
- Cornejo, J. E. 2008.** ¿Qué es UML? [Online] 2008. [Cited: noviembre 17, 2016.] <http://www.docirs.cl/uml.htm>.
- Corso, C. 2011.** *Minería de uso Web: Creación de perfiles, para la recomendación y optimización del uso en plataformas virtuales educativas*. Córdoba : Universidad Tecnológica Nacional, 2011.
- CSS3. 2016.** CSS3 HTML5. [Online] 2016. [http://html5.dwebapps.com/que-es-css3/..](http://html5.dwebapps.com/que-es-css3/)
- Cubanic. 2017.** Cubanic. *Cubanic*. [Online] 2017. [Cited: abril 15, 2017.] www.nic.cu/estadisticas.php.
- Eguiluz, J. 2011.** *Desarrollo web ágil con Symfony2*. 2011.
- Ellis, D. 1996.** Progress and problems in information retrieval. Los Angeles : London, 1996.

- Escobar, J. 2007.** *Tesis doctoral: Minería Web de uso y perfiles de usuario: aplicaciones con lógica difusa.* Granada : Editorial de la Universidad de Granada, 2007.
- Espinoza, M. 2014.** *Extracción de Preferencias Televisivas desde los Perfiles de Redes Sociales.* Ecuador : Revista EPN Universidad de Cuenca, 2014.
- Facebook. 2016.** Facebook Inc. [Online] 2016. <http://www.facebook.com>.
- Ferrell, C. O. y Geoffrey, Hirt. 2010.** *Introducción a los Negocios en un Mundo Cambiante.* s.l. : McGraw-Hill Interamericana, 2010.
- Foundation-Jmeter, The Apache software. 2016.** Apache JMeter. [Online] 2016. <http://jmeter.apache.org/>.
- Gamma, E. 2013.** *Design Patterns: Elements of Reusable Object-Oriented Software.* s.l. : Education Pearson, 2013.
- Genvetadev. 2012.** Genvetadev. [Online] 2012. [Cited: diciembre 5, 2016.] <http://www.genbetadev.com/frameworks/bootstrap>.
- Gonzalo, C. ; Codina, L. y Cristofol, R. 2015.** 5, España : Perspectivas sobre la Documentación Informativa, 2015.
- Google. 2016.** [Online] 2016. <http://www.google.com>.
- Guiluz, J. 2013.** *Desarrollo web ágil con Symfony2.* 2013.
- Hagen, P. 1999.** *Smart personalization.* s.l. : Forrester Report, 1999.
- Hernández, M., C. 2013.** *Manual de Usuario del Motor de Búsqueda Cubano.* La Habana : Universidad de las Ciencias Informáticas, 2013.
- Hidalgo, Y. 2010.** *Orión, un motor de búsquedas para la web de la UCI.* La Habana : s.n., 2010.
- Holte, R. C. y Yan, J. 1996.** *Inferring what a user is not interested in.* Springer : Advances in Artificial Intelligence, 1996. Vol. I.
- IEEE. 2004.** *Guide to the Software Engineering Body of Knowledge.* 2004. 0-7695-2330-7.
- Internet Live Stats. 2016.** [Online] octubre 20, 2016. <http://www.internetlivestats.com..>
- ISBN, IEEE. Guide to the Software Engineering Body of Knowledge. 2004. 2004.** *Guide to the Software Engineering Body of Knowledge.* 2004.
- Jaimés, L. G y Vega, F. 2005.** *Modelos clásicos de recuperación de la información.* 2005.
- Jalón, J. 2000.** *Aprenda Java como si estuviera en primero.* Navarra.2000.
- Java, T. 2014.** Conozca más sobre la tecnología Java. [Online] 2014. <http://www.java.com/es/about/>.
- JetBrains. 2016.** JetBrains. [Online] 2016. www.jetbrains.com.
- Justicia, C. 2004.** *Formas intermediasde representación en minería de texto.* Granada : Universidad de Granada, 2004.

- Larman, C. 2003.** *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. s.l. : Pearson Educación, 2003.
- Larman, C. 2004.** *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative developmen*. New Jersey : Prentice Hall PTR Upper Saddle River, 2004.
- Leblanc, M. y Howard, D. 2006.** *Writing Secure Code*. 2006. ISBN-13: 978-0735617223.
- Leyva, P ; Viltres, H. y Pons, L. A. 2016.** *Componentes y funcionalidades de un sistema de recuperación de la información*. La Habana : Revista Cubana de Ciencias Informáticas, 2016.
- LUPA. 2016.** Lupa. [Online] 2016. <http://www.lupa.upr.edu.cu>.
- Martínez, Rafael. 2012.** Postgresq. [Online] 2012. [Cited: noviembre 17, 2016.] http://www.postgresql.org.es/sobre_postgresql.
- Martínez, W. B. 2013.** Impactos del bloqueo Yanqui en la UCI. [En línea]. Cubadebate contra el terrorismo. *Cubadebate*. [Online] 2013. [Cited: noviembre 15, 2016.] cubaxdentro.wordpress.com/2013/10/25/impactos-del-bloqueo-yanqui-en-la-uci/.
- Méndez, F. 2002.** Propuesta y desarrollo de un modelo para la evaluación de la recuperación de información en internet. España : Tesis Doctoral, Universidad de Murcia, Facultad de Ciencias de la Documentación, 2002.
- Mexidor, F.D. 2011.** Ciberguerra contra Cuba: Mentiras en la red. . *Cubadebate*. [Online] 2011. [Cited: noviembre 15, 2016.] www.cubadebate.cu/opinion/2011/03/22/ciberguerra-contra-cuba-mentiras-en-la-red.
- Microsoft. 2016.** Microsoft. [Online] 2016. <http://www.microsoft.com>.
- Mobasher, B. 2005.** *Web usage mining and personalization*. 2005.
- Nasraoui, O. 2003.** *Mining involving user profiles in noisy web clickstream data whit a scalable immune system clustering algorithm*. Washintong DC. 2003.
- NetMarketShare. 2017.** Market Share Statistics for Internet Technologies. [En línea]. Market share for mobile, browsers, operating systems and search engines | NetMarketShare, 2017. [Citado el: 8 de marzo de 2017.] . Disponible en: [www.netmarketshare.com].
- Núñez, M. 2012.** Criterios para la evaluación de la calidad de las fuentes de información sobre salud en internet. *Scielo*. [Online] 2012. [Cited: febrero 25, 2017.] http://scielo.sld.cu/scielo.php?pid=S1024-94352002000500005&script=sci_arttext.
- Olivas, J. 2011.** *Búsqueda Eficaz de información en la web*. Buenos Aires : EDUNLP, 2011.
- Oracle. 2017.** NetBeans IDE Features. [Online] 2017. <https://netbeans.org/features/index.htm>.
- Otto, M. y Thornton, J. 2013.** *Bootstrap3 Manual Oficial*. 2013.

- Paradigm, Visual. 2016.** Visual Paradigm for UML- Software design tools for agile software development. . [Online] 2016. [Cited: noviembre 5, 2016.] <http://www.visualparadigm.com/product/vpuml/>.
- Partido Comunista de Cuba. 2016.** *Plan Nacional de Desarrollo Económico y Social hasta 2030: Propuesta de Visión de la Nación, Ejes y Sectores Estratégicos.* La Habana. 2016.
- Pazzani, M. J. y Muramatsu, J. 1996.** *Syskill & Webert: Identifying interesting web sites.* IAAI, 1996. Vol. 1.
- PHP. 2016.** PHP FRAMEWORK INTEROP GROUP (PSR-1). *Codificación estándar básica.* [Online] 2016. <http://www.php-fig.org/psr/psr-1/es>.
- PHP FRAMEWORK INTEROP GROUP (PSR-1). 2016.** *Codificación estándar básica.* 2016.
- PHP FRAMEWORK INTEROP GROUP (PSR-2). 2016.** PHP FRAMEWORK INTEROP GROUP (PSR-2). [Online] 2016. [Cited: marzo 30, 2017.] [<http://www.php-fig.org/psr/psr-2/es/>].
- PHP FRAMEWORK INTEROP GROUP (PSR-4). 2016.** [Online] 2016. [Cited: marzo 30, 2017.] <http://www.php-fig.org/psr/psr-4/>.
- PostgreSQL. 2017.** PostgreSQL. *PostgreSQL.* [Online] abril 15, 2017. <https://www.postgresql.org/>.
- Potencier, F. 2014.** Web PHP Framework. [Online] 2014. [Cited: noviembre 28, 2016.] <http://www.symfony-project.org/>.
- Pressman, R S. 2006.** *Ingeniería del software. Un enfoque práctico.* 2006. Vols. ISBN 970-10-5473-3.
- Pressman, Roger. 2002.** *Ingeniería de software. Un enfoque práctico.* 6ta. 2002.
- QATécnico. 2012.** QATécnico. [Online] marzo 3, 2012. [Cited: marzo 16, 2017.] <http://qatecnico.blogspot.com/2012/03/pruebas-de-rendimiento-tipos-y.html>.
- 2016.** Real Academia Española. [Online] 2016. <http://www.rae.com>.
- Remedios, M. J. 2014.** *Procedimiento para la recomendación de documentos de archivo.* La Habana : RCCI, 2014.
- Reyes, SC Ruiz Lobaina M. 2007.** *Minería Web: un recurso insoslayable para el profesional de la información.* s.l. : Acimed, 2007.
- Rincon, M. 2015.** Prezi. [Online] abril 30, 2015. [Cited: marzo 16, 2017.] <https://prezi.com/hnxdes2szvdp/pruebas-de-carga-y-estres/>.
- Rossini, D. 2003.** *Los archivos y la nuevas tecnologías de la información.* 2003.
- Rumbaugh, J. ; Jacobson, I. y Booch, G. 2000.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison-Wesley, 2000.
- Russell, A. 2011.** *Mining the Social Web Analyzing Data from Facebook, Twitter, LinkedIn and Other Social Media Sites.* s.l. : O'REILLY, 2011.

- Salazar, P. H. 1993.** El perfil del usuario de información. [Online] 1993. [Cited: noviembre 16, 2016.] <http://www.ejournal.unam.mx/ibi/vol07-15/IBI000701502.pdf>.
- Salomón, P. 2014.** Los planes de Google y el acceso a Internet en Cuba. *Cubadebate*. [Online] 2014. [Cited: noviembre 15, 2016.] www.cubadebate.cu/opinion/2014/07/04/los-planes-de-google-y-el-acceso-a-internet-en-cuba.
- Sampieri, R. 2014.** *Metodología de la investigación*. McGraw-Hill Education, 2014.
- Sánchez, S. 2015.** *Estrategia de soporte técnico para el proceso de migración a código abierto en los Organismos de la Administración Central del Estado*. La Habana : Universidad de las Ciencias Informáticas, 2015.
- Sánchez, T. 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI*. La Lisa, Habana : Universidad de las Ciencias Informáticas, 2015.
- SENSIOLABSNETWORK. 2014.** SensioLabs Product. [Online] 2014. [Cited: marzo 30, 2017.] <http://symfony.com/doc/current/contributing/code/standards.html>.
- Simmonds, T. 2012.** Top 10 Social Networking Websites . [Online] 2012. [Cited: diciembre 5, 2016.] <http://www.podline.co.uk/info/top-10-social-networking-websites/>.
- Sommerville, I. 2007.** *INGENIERÍA DE SOFTWARE*. Madrid : Pearson Educación S.A, 2007.
- Sparks, G. 2013.** *Introducción al modelado de sistemas de software usando el Lenguaje Unificado de Modelado (UML)*. Australia. 2013.
- Visual Studios.NET. 2016.** Visual Studios.NET. *Revisiones de código y estándares de codificación*. [Online] Microsoft, 2016. [Cited: abril 15, 2017.] <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
- Tolosa, G. 2013.** *Information retrieval: A Growing Area of research*. Argentina : Revista Electrónica de Estudios Telemáticos, 2013.
- Triana, D. y Fernández, Y. 2012.** *Análisis, Diseño e Implementación del Módulo Resultados de la Multisaber, en su versión multiplataforma*. 2012.
- Twitter. 2016.** Twitter Inc. [Online] 2016. <http://www.twitter.com>.
- Twitter, About Investors. 2016.** Investors. *Twitter About*. [Online] 2016. <https://investor.twitterinc.com/>.
- Universidad Politécnica de Madrid, Facultad de informática. 2005.** *Patrones del "Gang of Four"*. Madrid : s.n., 2005.
- Utrera, E y Daniel, W. 2015.** *Herramienta informática para la recuperación de imágenes digitales publicadas en la web*. La Habana : Universidad de las Ciencias Informáticas, 2015.
- V&V Quality. 2013.** V&V Quality. [Online] 2013. [Cited: marzo 16, 2017.] http://www.vyvquality.com/w1/index.php?option=com_content&view=article&id=91&Itemid=162.
- Vega, R Y. 2012.** *Guía metodológica para implementar la seguridad durante el desarrollo de aplicaciones informáticas*. La Habana. 2012.



Velasco, R. 2015. Nginx, el servidor de páginas web comparado con Apache HTTP Server. [Online] 2015. [Cited: abril 15, 2017.] <http://www.redeszone.net/2015/03/29/nginx-el-servidor-de-paginas-web-comparado-conapache-http-server>.