



Universidad de las Ciencias Informáticas

Facultad 1

Personalización de la interfaz web del buscador Orión

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

Autora:

Iguelit Bárbara Morán Hernández

Tutores:

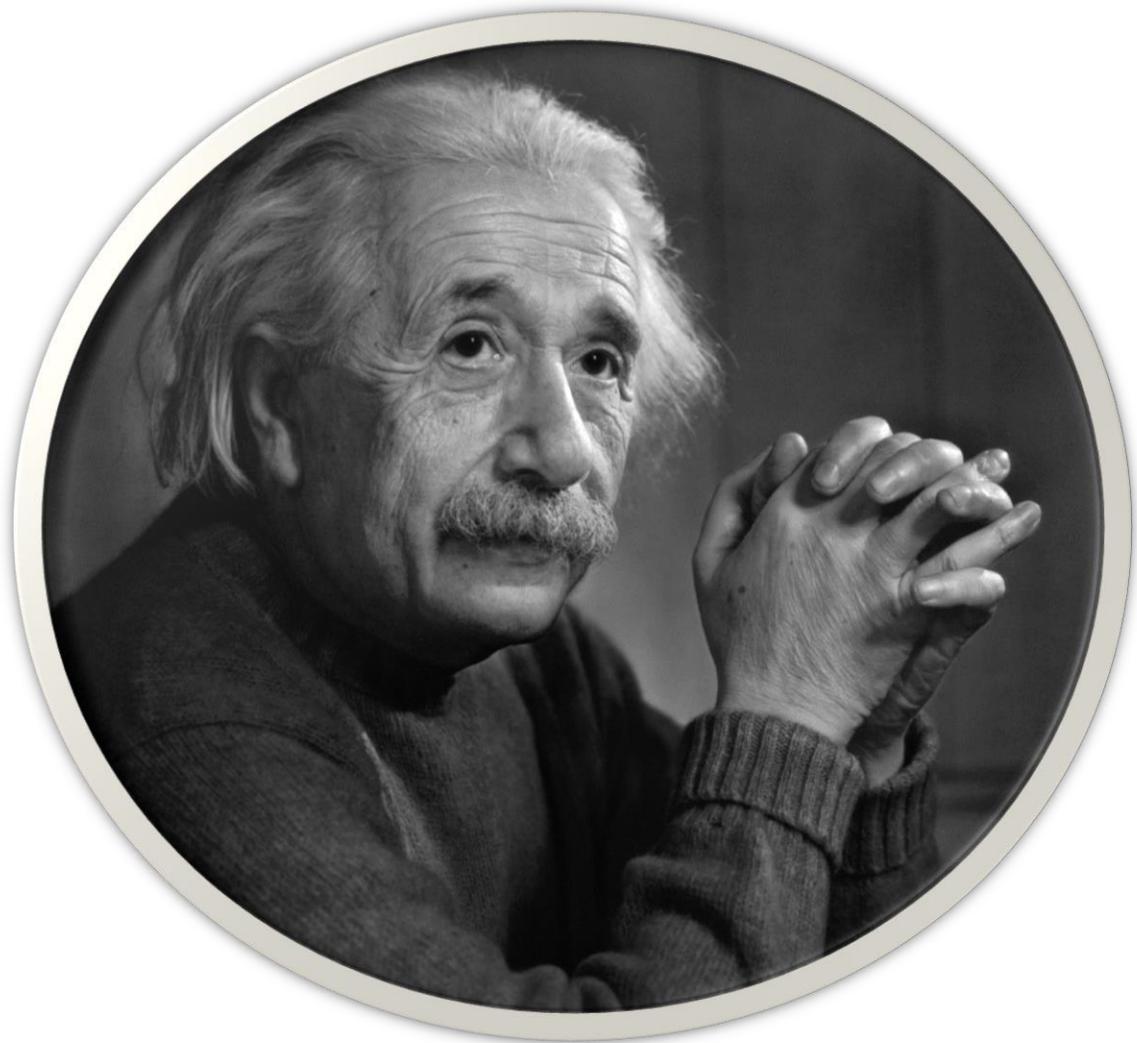
Ing. Yaima Oval Riverón

Ing. Eric Utrera Sust

La Habana, Junio de 2016

“Año 58 de la Revolución”

Frase



*“Hay una fuerza motriz más poderosa que el vapor,
la electricidad y la energía atómica: la voluntad”.*

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro ser la autora de la presente tesis y autorizo a la Facultad 1 de la Universidad de Ciencias Informáticas a hacer uso de la misma en su beneficio.

Para que así conste se firma la presente declaración jurada de autoría en La Habana a los ___ días del mes de ___ del año ___.

Autora:

Iguelit B. Morán Hernández

Tutores:

Ing. Yaima Oval Riverón

Ing. Eric Utrera Sust

Dedicatoria

A mis padres Sonia y Miguel Ángel por ser mi razón de existir.

A mi hermano Miguelito por ser mi guía y consejero en los momentos que más lo necesité.

A mis abuelos María Elena y Fidel por ser personas tan especiales para mí.

A mi sobrina Adlielys por ser la personita inocente que más quiero en este mundo.

*A mis primitos Baby, Dany, Nesty y los que están por venir porque quiero que estudien y se
esfuercen para que sean grandes en la vida.*

A mi familia.

Agradecimientos

Doy gracias a mami y a papi por todo el apoyo, por las enseñanzas y consejos de toda una vida, por estar siempre pendientes de mí, por apoyarme en los momentos más difíciles, por cuidarme, por quererme, por todos sus sacrificios, porque

GRACIAS A ELLOS SOY QUIEN SOY.

A mi tata Miguelito por dar razones para que yo viniera a este mundo, por ser mi ídolo y el impulsor para que cursara esta carrera y por traer al mundo a mi sobrina adorada, junto a Yuneisy, a quien también agradezco por ser mi amiga, mi confidente y mi consejera.

A mis abuelos, tíos, familia en general, vecinos y amigos por estar pendientes de mí y ayudarme y apoyarme toda la vida y durante estos cinco años de la carrera.

A mi amigo Yosel que ha sido la familia que me faltaba en la universidad, y a su familia que han sido incondicionales estos cinco cursos escolares.

A mis amigos, compañeros de aula y profesores, que me han acompañado en el transcurso de la carrera, por hacer que la estancia en la escuela fuera más agradable.

A mis tutores Eric y Yaima, por la dedicación prestada durante la realización de este trabajo de diploma, especialmente a Eric por guiarme con tanta paciencia.

A la familia de mi novio por ayudarnos durante estos años.

A las tías Magalis y Migdalia por abrirnos tantas puertas y siempre con una sonrisa.

Y por último y no menos importante a mi novio Carlos, quien ha sido mi amigo, mi compañero de aula, mi compañía, mi fortaleza en momentos de debilidad, por acompañarme en el paso por esta carrera en los momentos buenos y malos, por su paciencia y comprensión, por ayudarme a ser mejor persona, GRACIAS.

**A TODOS AQUELLOS QUE DE UNA FORMA U OTRA HAN PUESTO SU GRANITO DE
ARENA EN MI FORMACIÓN COMO PERSONA, SER HUMANO Y PROFESIONAL...
GRACIAS DESDE LO MÁS PROFUNDO DE MI CORAZÓN.**

RESUMEN

En la presente investigación se contribuye al progreso de la sociedad y los servicios del ciudadano cubano a través del desarrollo de la personalización de la interfaz web del buscador Orión. Se realizó un análisis de fuentes bibliográficas relacionadas con la recuperación de información, los sistemas encargados de realizar esta actividad en la red y las tendencias en la personalización de sistemas informáticos, especialmente de buscadores. Se llevó a cabo un estudio que permitió identificar la falta de personalización del buscador cubano Orión y como se efectúa esta acción en algunos buscadores más usados a nivel mundial. Para la implementación de la propuesta de solución se empleó la siguiente base tecnológica en correspondencia con la metodología AUP-UCI: el marco de trabajo *Symfony 2.7*, *Bootstrap* en su versión 3.3.6 para el diseño de la interfaz web, como sistema gestor de bases de datos *MongoDB* versión 3.0 y *Visual Paradigm 8.0* como herramienta para el modelado. Para lograr una personalización con la menor cantidad de errores se le realizaron pruebas funcionales, de integración, de seguridad y de usabilidad. La personalización de la interfaz web del buscador cubano posee un conjunto de funcionalidades mediante las cuales se hace más atractivo el sistema, los usuarios se identifican con él y logra que la interacción cliente-sistema sea más amigable, intuitiva y dinámica.

Palabras clave: buscadores, buscador Orión, interfaz web, personalización.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	5
Capítulo 1: Sistemas de recuperación de información y su personalización.....	10
1.1. Introducción	10
1.2. Definición del marco teórico conceptual de la investigación.....	10
1.3. Estudio de herramientas homólogas en el ámbito internacional.....	13
1.4. Estudio de herramientas homólogas en el ámbito nacional.....	15
1.5. Selección del entorno de desarrollo para la construcción de la solución	17
1.6. Conclusiones parciales	24
Capítulo 2: Análisis y diseño de la personalización de la interfaz web del buscador Orión	26
2.1. Introducción	26
2.2. Modelo del dominio.....	26
2.3. Especificación de los requisitos del software	28
2.4. Historias de usuario	31
2.5. Estilo arquitectónico.....	33
2.6. Patrones de diseño.....	34
2.7. Modelo de diseño	36
2.8. Modelo de despliegue.....	37
2.9. Conclusiones parciales	38
Capítulo 3: Implementación y validación de la personalización de la interfaz web del buscador Orión	39
3.1. Introducción	39
3.2. Modelo de implementación	39
3.3. Estándares de codificación	41
3.4. Pruebas de software	44
3.5. Conclusiones parciales	52
CONCLUSIONES GENERALES.....	53

RECOMENDACIONES	54
REFERENCIAS BIBLIOGRÁFICAS	55

ÍNDICE DE TABLAS

Tabla # 1: Comparación entre buscadores en cuanto a la personalización de su interfaz	16
Tabla # 2: Características de framework de PHP	20
Tabla # 3: Requisitos funcionales	28
Tabla # 4: Historia de usuario # 1	31
Tabla # 5: Historia de usuario # 2	32
Tabla # 6: Caso de prueba # 1	44
Tabla # 7: Variables empleadas en el diseño del caso de prueba # 1	45
Tabla # 8: Cantidad de no conformidades por cada iteración las pruebas	45
Tabla # 9: Resultados de prueba de usabilidad utilizando lista de chequeo	47
Tabla # 10: Cantidad de errores por cada iteración las pruebas de integración	51
Tabla # 11: Comportamiento de los errores detectados en las pruebas de integración	51

ÍNDICE DE FIGURAS

Figura # 1: Arquitectura de un bucador	12
Figura # 2: Modelo de dominio	27
Figura # 3: Arquitectura del sistema.....	33
Figura # 4: Diagrama de clases del diseño	36
Figura # 5: Diagrama de despliegue de la propuesta de solución	38
Figura # 6: Diagrama de componentes de la personalización de la interfaz web de Orión	40
Figura # 7: Fragmento de código fuente de la clase: "PersonalizationController"	41
Figura # 8: Fragmento de código fuente de la clase: "PersonalizationController"	42
Figura # 9: Fragmento de código fuente de la clase: "PersonalizationController"	42
Figura # 10: Fragmento de código fuente de la clase: "PersonalizationController"	42
Figura # 11: Fragmento de código fuente de la clase: "PersonalizationController"	43
Figura # 12: Fragmento de código fuente de la clase: "PersonalizationController"	43
Figura # 13: Fragmento de código fuente de la clase: "PersonalizationController"	43
Figura # 14: Comportamiento de las no conformidades por cada iteración	46
Figura # 15: Cumplimiento de indicadores de usabilidad	49
Figura # 16: Nivel de usabilidad después de corregidos indicadores de usabilidad.....	50

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) se han convertido en un instrumento poderoso que influye significativamente en el desarrollo de la sociedad (Soler, 2008). Las TIC han proporcionado a las personas fácil acceso a la información, comunicación inmediata sin importar la posición geográfica, mayores capacidades de almacenamiento de la información digital y nuevas formas de trabajos.

Internet constituye un componente importante de las TIC; la red de redes permite la interconexión descentralizada de computadoras, el acceso remoto a computadoras, el sistema de transferencia de archivos, el correo electrónico y las conversaciones en línea, entre otros servicios. El desarrollo de internet ha superado ampliamente cualquier previsión y constituye una verdadera revolución en la sociedad moderna. El mundo se transformó en un pilar de las comunicaciones, el entretenimiento y el comercio en todos los rincones del planeta (Definición, 2015).

Estos avances han propiciado que en la actualidad se disponga de un gran número de portales web, donde se publican contenidos de disímiles temáticas. Las dimensiones crecientes de la información en formato digital, originada principalmente por la facilidad de distribución, accesibilidad y bajos costes de producción, han determinado que los portales web se conviertan en portadores de información y de conocimiento. Los usuarios, debido a la necesidad de informarse emplean habitualmente la Web como medio para acceder y consultar aquella información que precisan. Ante semejante volumen de información se crearon los buscadores, herramientas que permiten discernir las páginas que contienen la información útil para el usuario frente a las millones de páginas restantes que resultan irrelevantes. Hoy existen muchas de estas herramientas a nivel mundial entre las que se encuentran Google, Baidu, Bing y Yahoo, (Netmarketshare, 2015), (Ver Anexo # 1).

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 llega a ser indexada por lo general está mal posicionada, provocando que aparezcan al final de los resultados (Carrillo y otros, 2012). Cuba está inmersa en un proceso de informatización de la sociedad, a partir del cual se han llevado a cabo una serie de acciones que han contribuido al avance y desarrollo tecnológico del país. En el año 2002, la apertura de la Universidad de la Ciencias Informáticas (UCI) con la tarea de formar y superar profesionales de la informática, así como contribuir a la informatización del país y a la producción de software. Entre los software desarrollados por la universidad que contribuyen al proceso de informatización

de la sociedad se encuentra el buscador Orión, el cual tiene una importante labor en mejorar el acceso a la información que se encuentra en la intranet nacional.

Aunque la calidad de un buscador reside en lo abundante, relevante y actualizada que sea su colección (Baeza-Yates y otros, 1999), también atender las preferencias, hábitos o gustos manifestados de forma expresada por el usuario en su búsqueda de información es también un aspecto importante a tener en cuenta si se desea impedir el abandono temprano del buscador y aumentar la fidelidad de un usuario (Iworld, 2015).

Atender las preferencias de los usuarios es un indicador que hoy el buscador Orión no tiene en cuenta. Este buscador es poco atractivo, debido a que no permite la personalización de la interfaz web de los contenidos que muestra como resultado de una búsqueda, y limita el trabajo del usuario al interactuar con él. Tampoco brinda la posibilidad al usuario de elegir imágenes de fondo para la interfaz web del buscador, ocultar información o registrar el resultado de las búsquedas; además no permite adaptarse a fechas y momentos históricos que identifiquen a la sociedad cubana. A partir de intercambios realizados con el jefe del proyecto Orión y especialistas del departamento de SENIT se llegó a la conclusión que la interfaz del buscador no permite que un usuario se sienta identificado con este, debido a que sus componentes no son atractivos y no pueden ser personalizados. Además, encuestas aplicadas a 80 estudiantes de la facultad # 1 de la universidad, por parte del jefe de departamento SINI del centro CIDI al cual pertenece el buscador, reveló que el 50 % de ellos no se sienten familiarizados e identificados con la interfaz web de Orión y no la consideran atractiva (Paúl, 2016).

Considerando la situación problemática anteriormente descrita, se plantea como **problema de investigación**: ¿Cómo visualizar la interfaz web del buscador Orión de manera que sus componentes se ajusten a los intereses de los usuarios?

Para la realización de la investigación se define como **objeto de estudio** la personalización de los sistemas de recuperación de información y el **campo de acción** se encuentra enmarcado en la personalización del buscador Orión.

Para dar solución al problema planteado, se define como **objetivo general** desarrollar una interfaz web personalizable, utilizando tecnologías de la web, que permita visualizar los componentes del buscador Orión de acuerdo a los intereses de los usuarios.

Con el propósito de cumplimentar gradualmente el objetivo general antes mencionado, el mismo se ha descompuesto en los siguientes **objetivos específicos**:

1. Caracterizar los fundamentos teóricos relacionados con la personalización de los contenidos.
2. Definir las tecnologías, las herramientas y la metodología para la implementación de la interfaz web personalizable del buscador Orión.
3. Diseñar la interfaz web personalizable del buscador Orión.
4. Implementar la interfaz web personalizable del buscador Orión.
5. Validar la interfaz web personalizable del buscador Orión.

Idea a defender:

La construcción de una interfaz web personalizable, utilizando tecnologías web, permitirá visualizar los componentes del buscador Orión de acuerdo a los intereses de los usuarios.

Para dar cumplimiento a los objetivos específicos se definieron las siguientes **tareas de investigación**:

1. Realización de un estudio sobre las tendencias de la personalización de sitios web.
2. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
3. Selección de la metodología de desarrollo de software.
4. Elaboración de los artefactos requeridos por la metodología de desarrollo seleccionada.
5. Implementación de la propuesta de solución.
6. Documentación de las pruebas realizadas.

Para el desarrollo de las tareas de investigación se han combinado diferentes métodos teóricos y empíricos de la investigación científica en la búsqueda y procesamiento de la información. Estos son:

Métodos teóricos:

- Histórico - Lógico: con el propósito de constatar teóricamente cómo ha evolucionado en el tiempo el proceso de personalización en sistemas de recuperación de información, así como las herramientas y tecnologías utilizadas en el desarrollo de aplicaciones de este tipo.
- Analítico-Sintético: empleado para el análisis de los elementos esenciales referentes a la información relacionada con los Sistemas de Recuperación de Información y su personalización.
- Inductivo-Deductivo: se emplea para arribar a razonamientos que puedan ser aplicables al problema a resolver luego de adquirir una serie de elementos referentes a la personalización en buscadores.

Métodos empíricos:

- Observación: permitió comprender como se desarrolla la personalización en buscadores a través del análisis de sistemas que lo implementan.
- Modelación: este método, empleado a través de los diagramas del lenguaje de modelado UML, se utilizó para la representación de las características y las relaciones entre los objetos de la solución.
- Entrevista: este método se empleó para obtener los requisitos funcionales de la solución propuesta. Además fue de gran utilidad para interactuar con el personal familiarizado con el buscador Orión.

El presente documento está estructurado en tres capítulos:

Capítulo 1 “Sistemas de recuperación de información y su personalización”

En este capítulo se abordan los principales conceptos asociados a la investigación, se analizan las tendencias actuales en el desarrollo de interfaces web en los buscadores que visualizan sus componentes de acuerdo a los intereses del usuario, se describen las principales herramientas empleadas. Se realiza un estudio de las metodologías y se selecciona el entorno donde se personalizará el sistema de recuperación de información.

Capítulo 2 “Análisis y diseño de la personalización de la interfaz web del buscador Orión”

En este capítulo se hace un levantamiento de los requisitos funcionales y no funcionales necesarios para lograr que la interfaz web del buscador Orión funcione correctamente. Se describe el análisis y diseño de

las clases del sistema. Se construyen los artefactos correspondientes a estas fases de desarrollo, modelando la personalización del buscador.

Capítulo 3 “Implementación y validación de la personalización de la interfaz web del buscador Orión”

En este capítulo se muestra el modelo de implementación como resultado del diseño anteriormente realizado. Se describen las pruebas a utilizar y los resultados obtenidos con dichas pruebas, con el objetivo de comprobar el correcto funcionamiento del buscador y su personalización. Al finalizar el capítulo se muestran las conclusiones generales, las recomendaciones, las referencias bibliográficas, la bibliografía consultada y los anexos.

Capítulo 1: Sistemas de recuperación de información y su personalización

1.1. Introducción

En el presente capítulo se abordan los principales elementos que justifican y soportan teóricamente la personalización en buscadores. Además se presentarán las tecnologías, herramientas y metodología que serán empleadas en la implementación de la propuesta de solución.

1.2. Definición del marco teórico conceptual de la investigación

Para lograr un mejor entendimiento del entorno de la investigación es necesario la identificación y el estudio de las principales bibliografías y así analizar algunos de los conceptos asociados al área de los sistemas de recuperación de información y su personalización.

Unos de los conceptos que necesitan ser estudiados en la era de la digitalización es el de recuperación de información. El surgimiento de la era digital abrió nuevos horizontes en las formas en que el hombre puede tratar con la información que produce, haciendo posible su almacenamiento y distribución masiva de manera rápida y simple a través de las redes de computadoras. La importancia de esta información ha propiciado que desde la década de 1950 se haya venido desarrollando el área de la Recuperación de Información (RI) (Tolosa-Gabriel y otros, 2008).

1.2.1. Recuperación de información

La Recuperación de Información (RI) trabaja en la forma en que se representa, almacena, organiza y se accede a la información (Baeza-Yates y otros, 1999). Otros autores plantean que la Recuperación de Información es dar respuesta a una pregunta formulada trayendo información relevante desde un gran archivo (Pérez-Carballo y otros, 2000).

La RI reside en establecer qué documentos son relevantes para satisfacer las necesidades de información de un usuario determinado, expresada en un lenguaje de consulta (Tolosa- Gabriel H. y otros, 2008).

Luego de las definiciones antes expuestas la autora considera como Recuperación de Información el proceso de obtener información que se encuentra almacenada y que es requerida por un usuario a través de técnicas, métodos, y herramientas de búsqueda.

Este proceso de recuperación de información debido a la cantidad y variedad de datos que se encuentran hoy en internet y que cada día crece significativamente, cuenta con herramientas que obtienen la información que resulta significativa, es decir, útil, relevante y confiable (Maglione-Carla y otros, 2012). Estas herramientas son conocidas como sistemas de recuperación de información.

1.2.2. Sistemas de recuperación de información

Los Sistemas de Recuperación de Información (SRI) tienen como objetivo proporcionar información ordenada de acuerdo al grado de relevancia que estos posean para las consultas de los usuarios dentro de una amplia colección (Baeza–Yates, 2005). Dichos sistemas pueden clasificarse en buscadores, metabuscadores, directorios dependiendo de la función que realizan, la forma de operar, el alcance que poseen, los tipos de documentos que recuperan, entre otros. Siendo los buscadores los más adecuados para implementar sistemas autónomos dedicados a la recopilación, procesamiento y recuperación de grandes volúmenes de información.

1.2.3. Buscador

Según la Real Academia Española, en términos informáticos, un buscador o motor de búsqueda es un programa que permite acceder a información en internet sobre un tema determinado (Real Academia Española, 2014).

Un concepto más amplio y asumido en la presente investigación es: un buscador es un sistema o aplicación informática que permite la búsqueda de todo tipo de términos y palabras claves a partir del desarrollo de índices de archivos almacenados en servidores web, que opera indexando archivos y datos en la web (DefiniciónABC, 2015).

En la figura # 1 (Seker, 2012) se muestra la arquitectura de un buscador el cual está compuesto por tres subsistemas. Un primer subsistema es el mecanismo de indexación que se encarga de almacenar todos los documentos que han sido recuperados. Un segundo subsistema que se encarga de rastrear la web en busca de información y un tercer y último subsistema que es la interfaz de usuario, mediante la cual se hacen las búsquedas y se muestran los resultados. Se muestra además las relaciones de comunicación y funcionamiento que existen entre estos. (Herrera, 2006).

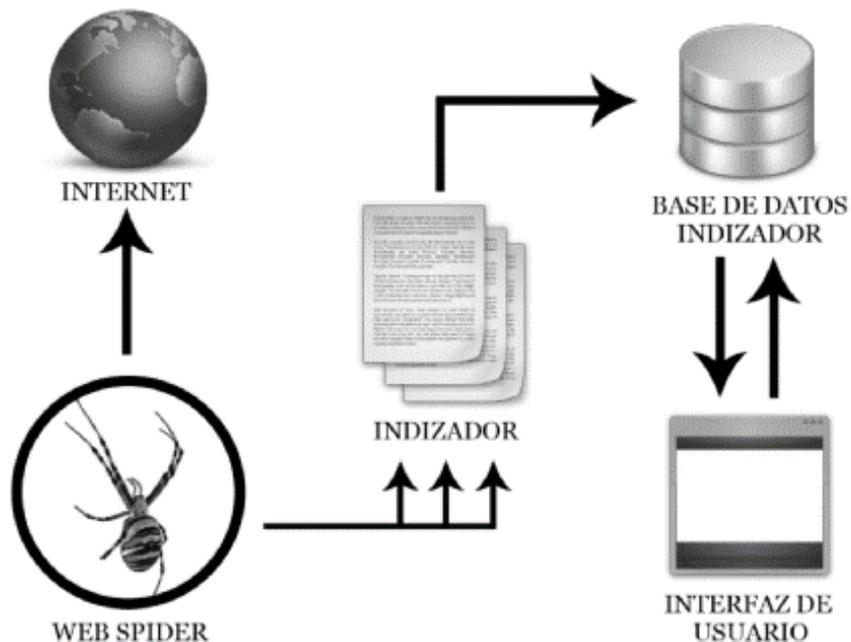


Figura # 1: Arquitectura de un buscador

(Fuente: Creación de la autora de esta investigación)

Las interfaces web aportan a las aplicaciones web elementos propios de presentación y organización de la información, que actualmente constituyen estándares del diseño web (Hernández y otros, 2008). También son consideradas una capa de presentación que funciona como intermediarias entre el usuario final y un software ejecutado en un servidor web remoto (Luján, 2002). Estas interfaces son un componente fundamental en un SRI (López, 2006); es donde el usuario introduce los criterios de búsqueda y obtiene un conjunto de resultados. Las interfaces web han captado la atención de los usuarios debido a su gran atractivo y facilidad de personalización (Aleman y otros, 2014).

1.2.4. Personalización

Es la acción de modificar algo de acuerdo a las preferencias personales. La personalización en los buscadores web es usada con el propósito de atraer más la atención de los clientes y así contribuir a su

fidelidad con el sitio. En estos programas, la capacidad de personalización por parte de cada usuario posibilita que los resultados sean lo más acorde posible a sus propios intereses (Stark, 2001).

La utilización de buscadores web se ha convertido en uno de los principales motivos del uso de internet, facilitando la obtención de información y el trabajo de índole investigativa pero también con fines sociales, recreativos y personales (DefiniciónABC, 2015). La tendencia, de que los usuarios exijan cada vez más productos y servicios personalizados, se ha extendido hasta esta área y ahora no solo se enfoca en las características y preferencias de la sociedad en general, sino también en las preferencias puntuales de cada consumidor.

1.3. Estudio de herramientas homólogas en el ámbito internacional

En la actualidad existe en el mundo diversidad de buscadores que permiten la personalización de su interfaz, algunas de las más usadas son Google, Baidu, Bing y Yahoo (Netmarketshare, 2015).

1.3.1. Google

El motor de búsqueda Google es propiedad de la empresa Google Inc. y fue fundado en septiembre de 1998 por Larry Page y Sergey Brin; estudiantes de la Universidad de Stanford. Es actualmente el buscador más usado para localizar información en la web a nivel mundial (StatCounter, 2011).

Google brinda la posibilidad de personalizar su interfaz dándole al usuario las siguientes opciones:

- **Menú de preferencias:** una vez que el usuario accede a la aplicación se le brinda un botón preferencias ubicado en la esquina superior derecha, que despliega un menú con la opción de configuración de búsqueda.
- **Variar la cantidad de resultados por páginas:** una vez que el usuario ha accedido a la configuración de búsqueda tiene la posibilidad de escoger, entre 10, 20, 30, 40, 50 y 100, los resultados que desea ver en cada página de respuesta a su búsqueda.
- **Dónde abrir el resultado de la búsqueda:** en la configuración de la búsqueda el usuario tiene la opción de escoger si desea ver los resultados de su búsqueda en la pestaña que se encuentra o en una pestaña nueva.

1.3.2. Bing

Bing es el buscador web perteneciente a la empresa Microsoft antes conocido como *Live Search*, *Windows Live Search* y *MSN Search*. Este SRI, además de implementar reglas e instrucciones para estrechar y filtrar los mejores resultados, también se ha centrado en la apariencia de su interfaz para lograr competitividad respecto a los demás buscadores (Bing, 2015).

Dentro de las opciones de personalización que ofrece Bing se pueden destacar:

- **Cambiar la imagen de fondo:** esta opción brinda a los usuarios la posibilidad de escoger la imagen de fondo dentro de una colección de siete imágenes haciendo uso de botones de hacia adelante y hacia atrás que se encuentran situados en la esquina inferior derecha de la interfaz principal del buscador.
- **Ocultar la Barra de Menú:** al pulsar el botón situado en la esquina inferior derecha de la interfaz de inicio para seleccionar la opción de pantalla completa se oculta la barra de menú, dejando solamente en la vista de la interfaz la barra de búsqueda, el nombre y el logo del sitio y la imagen de fondo.
- **Menú de preferencias:** una vez que el usuario se encuentra en la interfaz de resultados de búsqueda se le brinda un botón preferencias ubicado en la esquina superior derecha, que despliega un menú con las opciones de Historial de búsqueda y Configuración.
- **Redimensionar la imagen de fondo:** en la opción configuración del menú de preferencias el usuario tiene la posibilidad de escoger si desea ver la imagen de la página principal grande o más pequeña.
- **Variar la cantidad de resultados por páginas:** una vez que el usuario ha accedido a la configuración se le brinda la posibilidad de escoger la cantidad de resultados que desea ver en cada página de respuesta a su búsqueda que pueden ser automático, 10, 15, 30 ó 50.
- **Dónde abrir el resultado de la búsqueda:** en la opción de configuración el usuario tiene la posibilidad de escoger si desea ver los resultados de su búsqueda en la pestaña que se encuentra o en una pestaña nueva.
- **Regresar a la configuración por defecto:** una vez que el usuario realiza un cambio en algún parámetro de configuración se le da la opción a través de un botón de regresar a la configuración por defecto de la aplicación.

1.3.3. Yahoo

Yahoo! comenzó como un entretenimiento de los estudiantes de Ingeniería Eléctrica de la Universidad de Stanford David Filoy y Jerry Yang en el año 1994, y se convirtió en un buscador que permite encontrar y acceder a la información en la Web (Yahoo, 2005).

Yahoo presenta una interfaz que permite la personalización, dentro de las opciones que posee se encuentran:

- **Menú de preferencias:** una vez que el usuario se encuentra en la interfaz de resultados de búsqueda se le brinda un botón ubicado en la esquina superior derecha, que despliega un menú con las opciones de Mi cuenta, Preferencias, Búsqueda avanzada e Historial de búsqueda.
- **Variar la cantidad de resultados por páginas:** una vez que el usuario ha accedido a la opción Preferencias del menú que se encuentra en el menú de preferencias se le brinda la posibilidad de escoger la cantidad de resultados que desea ver en cada página de respuesta a su búsqueda que pueden ser 10, 15, 20, 30, 40 ó 50.
- **Dónde abrir el resultado de la búsqueda:** en la opción preferencias el usuario puede escoger si desea ver los resultados de su búsqueda en la pestaña que se encuentra o en una pestaña nueva.

1.4. Estudio de herramientas homólogas en el ámbito nacional

1.4.1. Orión

Este sistema de recuperación de información permite realizar búsquedas sobre los contenidos publicados en la Web cubana, puede ser accedido desde la UCI y el Ministerio de Educación Superior (MES). Orión está desarrollado con *Nutch* en su versión 1.5 para el rastreo en la red, *Solr* versión 3.6 como mecanismo de indexación, PHP en su versión 5.3 para el trabajo con el marco de trabajo *Symfony* en su versión 2.0 para la interfaz web y la interacción con el mecanismo de indexación, *MongoDB* versión 2.0 como gestor de base de datos no 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 (Hernández, 2013).

El buscador Orión aunque aún se encuentra en proceso de desarrollo ha sido desplegado en determinados servidores con el objetivo de realizarle pruebas funcionales, por lo que en este momento es una herramienta

cubana, apta para la recuperación de la información existente en la intranet nacional. Orión no cuenta con opciones de personalización de su interfaz web, por lo que no satisface la necesidad existente hoy en día de que los usuarios tengan la posibilidad de ver las aplicaciones que utilizan según sus gustos y preferencias en cuanto al posicionamiento, el estilo, el formato de los elementos que se encuentran en su interfaz.

1.4.2. Red Cuba (CUBA)

CUBA es una plataforma de búsqueda que surge por la necesidad existente en el país de desplegar un sistema que pudiera ser accedido desde cualquier territorio cubano. Posee un buscador que utiliza el motor de búsqueda implementado por Orión, además de otros servicios. Este buscador tampoco cuenta con opciones de personalización.

A partir del estudio realizado se han podido distinguir las funcionalidades brindadas por algunos de los buscadores más utilizados en el mundo y en Cuba con respecto a la personalización de sus interfaces.

Todo esto da paso a desarrollar una interfaz que permita la personalización del buscador Orión y así lograr que además de tener un buscador que posea una colección de información abundante, relevante y actualizada, también atraiga a los usuarios y así aumente su fidelidad para con el buscador.

En la Tabla # 1 se pueden apreciar las principales opciones de personalización que brindan estos buscadores y su relación con cada uno de los mismos.

Tabla # 1: Comparación entre buscadores en cuanto a la personalización de su interfaz
(Fuente: Creación de la autora de la investigación)

Elementos de personalización	Buscadores			
	Google	Bing	Yahoo	Orión
Cambiar la imagen de fondo	no	si	no	no
Ocultar la Barra de Menú	no	si	no	no
Menú de preferencias	si	si	si	no
Redimensionar la imagen de fondo	no	si	no	no

Variar la cantidad de resultados por páginas	si	si	si	no
Donde abrir el resultado de la búsqueda	si	si	si	no
Regresar a la configuración por defecto	no	si	no	no
Cambiar idioma	si	si	si	si

Luego de realizar el análisis antes presentado se concluye que:

- Los buscadores existentes en el ámbito internacional son privativos y de difícil acceso para la población cubana debido a las limitaciones que tiene el país a la hora de acceder a internet.
- En ocasiones los buscadores internacionales no indexan la información presente en sitios web cubanos y por consecuencia no aparecen en los resultados de las búsquedas.
- La información que llega a ser indexada por lo general está mal posicionada.
- El motor de búsqueda cubano Orión no brinda opciones a los usuarios de personalizar su interfaz.

Teniendo en cuenta las conclusiones antes presentadas se evidencia la necesidad de incluir opciones de personalización al buscador cubano Orión.

1.5. Selección del entorno de desarrollo para la construcción de la solución

Para el desarrollo de la propuesta de solución se hace necesario el estudio de las herramientas y tecnologías que se emplearán en correspondencia con la metodología usada en el desarrollo del proyecto Orión.

1.5.1. Metodología de desarrollo de software

Una metodología de desarrollo de software impone un proceso de forma disciplinada sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente (Valdéz, 2015). Está orientada a la forma adecuada de construcción del software, que puede incluir modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos (Sommerville, 2015).

Un concepto más ampliado y que agrupa a los dos anteriores es el que ofrece (Zambrano, 2013); plantea que una metodología de desarrollo, en ingeniería de software, es un conjunto de herramientas, técnicas, procedimientos y soporte documental encaminados a estructurar, planificar y controlar el proceso de desarrollo de forma organizada y lógica, que tiene como objetivo apoyar a los desarrolladores en la creación de un nuevo software.

Para el desarrollo de la solución se usará la variación de la metodología de Proceso Unificado Ágil (AUP por sus siglas en inglés), por ser la empleada en el centro CIDI en el desarrollo de proyectos.

La metodología AUP contribuye de manera significativa al desarrollo del software que se espera obtener, ya que esta metodología además de ser ágil es también adaptable a cualquier circunstancia que pueda surgir en el proceso de desarrollo del proyecto. La UCI le ha realizado modificaciones con el fin de adaptarlo al ciclo de vida definido para la actividad productiva de dicha institución; de las 4 fases que encierra la metodología AUP se simplificaron a (Ambler, 2014):

- Inicio: esta etapa se mantiene de AUP pero se modifica el objetivo de la misma; realización de la planeación del proyecto en cuanto a un estudio de la organización del cliente, el alcance del proyecto, estimaciones de tiempo esfuerzo y costo que serán necesarios para decidir si se desarrolla el producto o no.
- Ejecución: en esta fase se recogen las actividades que desarrolla AUP de elaboración, construcción y transición. Se ejecutan las actividades requeridas para obtener el software y ajustar los planes del proyecto, realizando las tareas de modelado del negocio, obtención y análisis de los requisitos, elaboración de la arquitectura, diseño, implementación y liberación de producto.

- Cierre: en esta fase se analizan los resultados del proyecto, su ejecución y se llevan a cabo actividades formales de cierre del proyecto.

1.5.2. Marco de trabajo

Un *framework* o marco de trabajo es una estructura de software integrada por componentes personalizables e intercambiables para desarrollar un software; se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir elementos (Acosta y otros, 2012).

El uso de un marco de trabajo para desarrollar aplicaciones permite a los programadores un desarrollo ágil, con menor esfuerzo y además promover las buenas prácticas ya que les da la posibilidad de reutilizar códigos, seguir una estructura definida y con ello hacer uso de funcionalidades ya implementadas (Gutiérrez, 2014).

Algunos de los marcos de trabajo de desarrollo de aplicaciones web más usados son *CodeIgniter*, *Yii*, *Zend* y *Symfony*. A continuación se muestra un resumen de sus características.

- **CodeIgniter**

CodeIgniter es un potente marco de trabajo que usa PHP como lenguaje de programación del lado del servidor construida con el propósito de que los desarrolladores posean una herramienta con un diseño compacto que les permita crear aplicaciones web completamente equipadas haciendo uso de herramientas simples y elegantes (CodeIgniter, 2015).

- **Yii**

Yii es un marco de trabajo genérico de programación Web que puede ser utilizado para todo tipo de aplicaciones Web. Gracias a que es liviano de correr y está equipado con soluciones de cacheo sofisticadas, es adecuado para desarrollar aplicaciones de gran tráfico (Yii, 2015).

- **Zend**

El marco de trabajo de desarrollo *Zend* está basado en la simplicidad, en las mejores prácticas orientadas a objetos, en la creación de aplicaciones seguras y confiables y en el consumo de *APIs* (*Application Programming Interface*) que proveen algunos líderes como Google, Yahoo, Amazon (Zend, 2015).

- **Symfony**

Symfony posee la característica de que es fácil de instalar y puede ser usado tanto en Linux como en Windows. Es compatible con varios sistemas gestores de base de datos como son *PostgreSQL*, *MongoDB*. Este marco de trabajo tiene una baja curva de aprendizaje, es decir, que es fácil aprender a trabajar con él y permite construir aplicaciones robustas. Incluye herramientas adicionales que ayudan a probar, depurar y documentar el proyecto. Es totalmente gratuito (Symfony, 2015).

En la tabla # 2 que se muestra a continuación se pueden observar las características y ventajas más significativas que presentan los *frameworks* antes presentados.

Tabla # 2: Características de framework de PHP
(Fuente: Creación de la autora de la investigación)

Características	Framework de PHP			
	CodeIgniter	Yii	Zend	Symfony
Arquitectura Modelo-Vista-Controlador.	si	si	si	si
Validación asociada.	si	si	si	si
Mapeador de objetos.	-	si	si	si
Soporta diversas bases de datos.	si	si	si	si
Almacena objeto en memoria cache.	si	si	si	si
Comunidad de código abierto.	si	si	si	si
Abundante documentación.	si	si	si	si

Debido a las potencialidades que presentan los marcos de trabajo antes expuestos, teniendo en cuenta sus características principales, no se asegura una ventaja entre uno u otro. Teniendo en cuenta que la interfaz

web del buscador cubano Orión está desarrollada haciendo uso del marco de trabajo *Symfony*, el cual aporta beneficios a los usuarios por su facilidad de adaptación a los casos más complejos (Potencier, 2011) se selecciona este marco de trabajo para realizar la personalización de la interfaz de dicho buscador.

1.5.3. Lenguajes

Los lenguajes de programación son el idioma mediante el cual se comunican las máquinas entre ellas y con los seres humanos. A continuación se presentan los lenguajes que serán utilizados para la realización de la solución tanto del lado del cliente como del lado del servidor, seleccionados debido a que son los que emplea el marco de trabajo *symfony*.

Del lado del servidor

Las tecnologías en el lado del servidor se encargan básicamente de atender las solicitudes de los clientes. Es donde se realiza la mayor parte del procesamiento de la información solicitada, generando una respuesta que facilite la comprensión por el usuario.

- **PHP**

Hypertext preprocessor (PHP por sus siglas en inglés) es un lenguaje multiplataforma, de alto nivel e interpretado, compatible con la mayoría de los gestores de base de datos que se utilizan en la actualidad, es un lenguaje de programación del lado del servidor utilizado para generar páginas dinámicas. Se distribuye bajo licencia libre, por lo que se presenta como una alternativa de fácil acceso para todos. No requiere definición de tipos de variables y cuenta con manejo de excepciones (García, 2010). La sintaxis de PHP es similar a la del lenguaje C lo cual permite su rápida ejecución y consumiendo poca memoria, además cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código (Mariño, 2008).

Por las características antes mencionadas y teniendo en cuenta que el marco de trabajo *Symfony* hace uso de PHP, se usará para realización de la propuesta de solución este lenguaje de programación del lado del servidor en su versión 5.6 LTS.

Del lado del cliente

Para el trabajo en la interfaz web se utilizarán HTML, JavaScript y CSS como lenguaje de programación del lado del cliente, integrados todos en el marco de trabajo *Bootstrap*.

- **Bootstrap**

Bootstrap, es un marco de trabajo creado por *Twitter*. Este framework 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, una Tablet 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 les da agilidad a la hora de cargar y de adaptarse a otros dispositivos. El Marco de trabajo trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegados, Formularios incluyendo todos sus elementos e integración con *jQuery* para ofrecer ventanas e información sobre herramientas dinámicas (Otto y otros, 2013).

- **HTML**

HTML, acrónimo de *HyperText Markup Language* (Lenguaje de marca de hipertexto), es un lenguaje de publicación especificado como un estándar por el W3C (World Wide Web Consortium) que permite la creación de páginas web (World Wide Web Consortium, 2014).

- **CCS**

CSS (*Cascading Style Sheets*, por sus siglas en inglés) es un mecanismo que permite aplicar de forma consistente diferentes estilos a los documentos creados, de manera que pudieran visualizarse de igual forma en cualquier navegador web (Lie y otros, 2005).

- **JavaScript**

JavaScript es un lenguaje de programación del lado del cliente encargado de realizar acciones dentro del ámbito de una página web. JavaScript permite crear efectos especiales sobre páginas web, para confeccionar contenidos dinámicos y posibilitar que los elementos de la página tengan movimiento, cambien de color o cualquier otro dinamismo; además permite ejecutar instrucciones como respuesta a las acciones del usuario, lo cual posibilita crear páginas interactivas (Mozilla Project, 2013).

1.5.4. Herramientas

A continuación se describen las herramientas que serán utilizadas para la implementación de la propuesta de solución.

- **Servidor web Apache**

Para procesar las peticiones de los clientes web y devolverles una respuesta con los datos procesados se utilizará como servidor web HTTP Apache 2.

Este servidor web de código abierto implementa el protocolo HTTP 1.1, caracterizado fundamentalmente por su alto nivel de configuración, modularidad, robustez y estabilidad. Está desarrollado bajo la licencia ASF por *The Apache Software Foundation* (NetCraft, 2013).

Se decide utilizar Apache 2 como servidor web, ya que además de poseer las características presentadas anteriormente es el utilizado en el desarrollo de la interfaz web del buscador Orión. Esto permitirá establecer la personalización a implementar en el mismo servidor donde se encuentre el motor de búsqueda, aprovechando los recursos disponibles.

- **Entorno de desarrollo Integrado NetBeans**

Como entorno de desarrollo integrado o también conocido como IDE por sus siglas en inglés (*Integrated Development Environment*) se utilizará *NetBeans*.

NetBeans es un potente IDE para programadores que le provee una plataforma ideal para escribir, compilar, depurar y ejecutar programas informáticos (Oracle, 2014), además proporciona estabilidad y grandes posibilidades de ser extendido gradualmente por desarrollos comunitarios, permitiendo agregar continuamente nuevas funcionalidades. 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 distintos sistemas operativos del mercado (NetBeans, 2015).

- **Herramienta CASE Visual Paradigm**

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

relaciones, ingeniería directa e inversa, la gestión de requisitos de software y la modelación de procesos del negocio (Visual Paradigm, 2014).

- **Lenguaje de modelado**

UML acrónimo de Lenguaje Unificado de Modelado es el lenguaje estándar especificado por el *Object Management Group* (OMG) para visualizar, especificar, construir y documentar los artefactos de un sistema, incluyendo su estructura y diseño. Utiliza un conjunto de símbolos y notaciones para representar gráficamente los diversos componentes que forman parte de la arquitectura de software. Permite el modelado de procesos de negocio y el modelado de requisitos apoyándose en el análisis orientado a objetos (Object Management Group, 2016).

- **Gestor de base de datos**

MongoDB es un sistema gestor de base de datos NoSQL, desarrollado bajo el concepto de código abierto. En lugar de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida (MongoDB, 2016).

- **Acunetix**

Acunetix comprueba los sistemas en busca de vulnerabilidades. 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).

1.6. Conclusiones parciales

La realización de este capítulo, en el cual se abordaron aspectos correspondientes a los fundamentos teóricos de propuesta de solución, permitió llegar a las siguientes conclusiones:

- El análisis de los conceptos teóricos abordados en el capítulo permitió un mejor entendimiento de la investigación que se realiza.

- En la evaluación de las tendencias de aplicaciones similares a la propuesta de solución presentada en este trabajo, se identificaron sistemas con un alto nivel de aceptación pero cada uno responde a las necesidades particulares; aun así fue posible reconocer a partir de ellos, nuevos elementos que contribuyen al perfeccionamiento de la propuesta de solución.
- La falta de personalización del buscador Orión, el limitado acceso a buscadores internacionales y a los resultados que muestran, hacen necesario el desarrollo de la personalización de dicho buscador.
- La selección de la metodología, herramientas y tecnologías con soporte multiplataforma y basadas en software libre, permitió obtener una base tecnológica adecuada que permitirá el desarrollo de la solución.

Capítulo 2: Análisis y diseño de la personalización de la interfaz web del buscador Orión

2.1. Introducción

En el presente capítulo se abordan los aspectos fundamentales relacionados con el diseño del sistema a desarrollar, definiendo el modelo de dominio según el objeto de estudio haciendo uso de la metodología *AUP*. 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. Se explican además los estilos, la arquitectura y el diseño del sistema a desarrollar para lograr buenas prácticas en el diseño y programación de la propuesta de personalización de la interfaz web del buscador Orión. A lo largo del capítulo se muestran los principales artefactos de ingeniería de software correspondientes a las funcionalidades más importantes.

2.2. Modelo del dominio

Un modelo de dominio, o modelo conceptual como también se lo conoce, es una representación visual en forma de diagrama de las clases conceptuales u objetos del mundo real que son significativos en un dominio de interés; no se trata de un conjunto de diagramas que describen clases u objetos de software con responsabilidades (Larman, 2004).

A continuación se presenta la figura 2, la cual muestra el diagrama de modelo de dominio que representa las relaciones que existen entre los principales conceptos asociados a la propuesta de personalización de la interfaz web del buscador Orión.

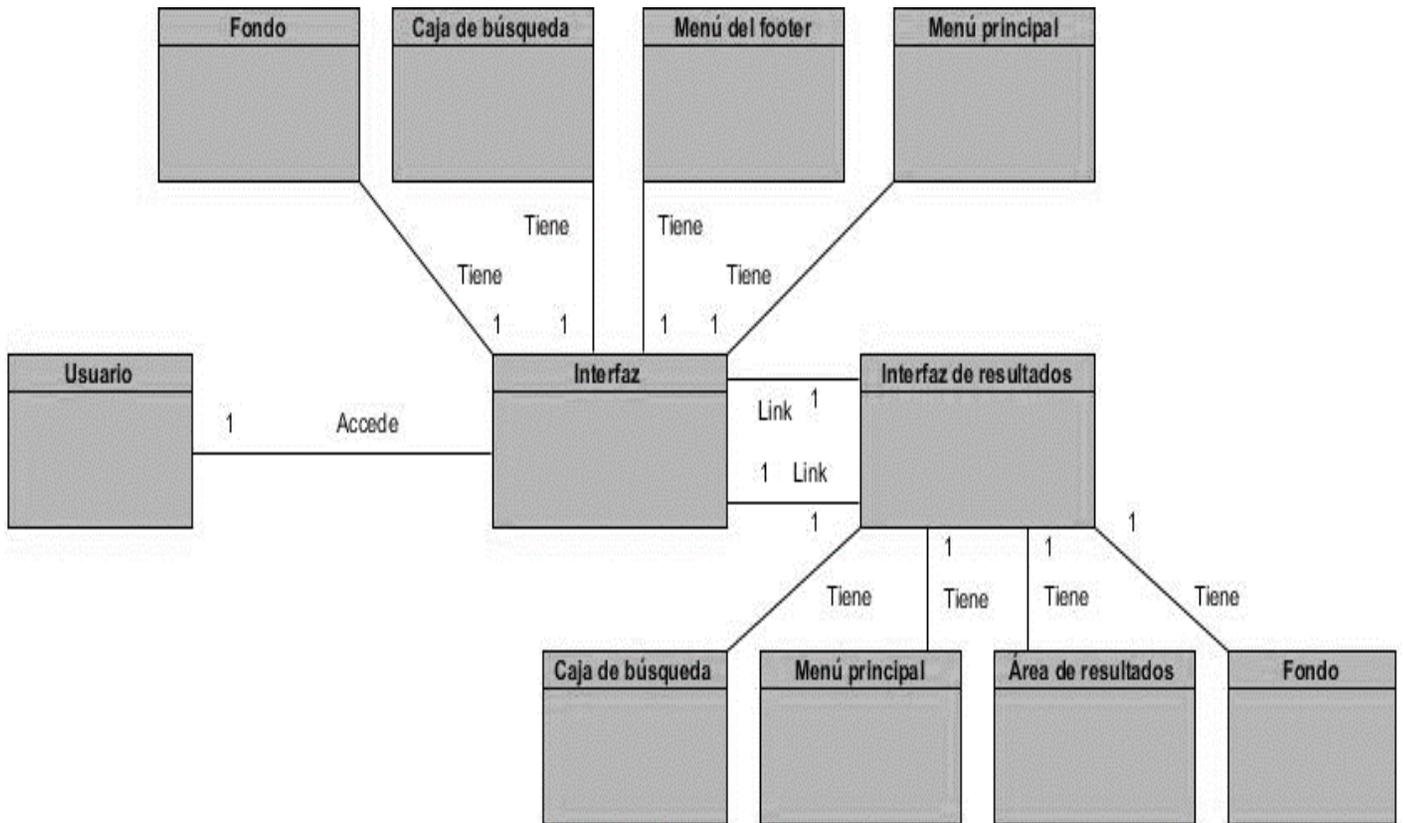


Figura # 2: Modelo de dominio

2.2.1. Descripción de los conceptos asociados al modelo del dominio

Usuario: es la persona que accede al sistema a través de la interfaz principal.

Interfaz: constituye la vista mediante la cual el usuario accede al sistema y realiza la búsqueda de información.

Interfaz de resultados: constituye la vista mediante la cual se le muestran los resultados de las búsquedas a los usuarios.

Menú principal: es un componente de la interfaz a la cual accede el usuario.

Caja de búsqueda: es un componente de la interfaz a la cual accede el usuario para introducir la búsqueda que desea realizar.

Menú del footer o menú del pie de página: es un componente de la interfaz principal a la cual accede el usuario.

Fondo: es un componente de la interfaz principal a la cual accede el usuario.

Área de resultados: es un componente de la interfaz de resultados mediante la cual se le muestra al usuario los resultados de la búsqueda q realizó.

2.3. Especificación de los requisitos del software

En la ingeniería del software, los requisitos se utilizan como datos de entrada en la etapa de diseño del producto y establecen qué debe hacer el sistema, pero no cómo hacerlo. Son una condición o capacidad que un usuario necesita para poder resolver un problema o lograr un objetivo. De manera general estos requisitos son lo que el sistema debe hacer o una cualidad que el sistema debe poseer (Somerville, 2007).

Una vez definidos los conceptos principales relacionados con el dominio, se muestran a continuación los requerimientos funcionales y no funcionales, de acuerdo con el objetivo planteado al inicio de este trabajo.

2.3.1 Requisitos Funcionales

Los requisitos funcionales (RF) definen el comportamiento interno de un software, son condiciones que el sistema ha de cumplir. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de software (Somerville, 2007).

Las entrevistas realizadas y el análisis previamente realizado de la personalización en buscadores, arrojaron un total de 23 requisitos funcionales, que a continuación se presentan.

Tabla # 3: Requisitos funcionales

RF 1:	Modificar color de fondo.
--------------	---------------------------

RF 2:	Cambiar imagen de fondo.
RF 3:	Redimensionar imagen de fondo.
RF 4:	Modificar color de letra.
RF 5:	Modificar tipo de letra.
RF 6:	Modificar cantidad de resultados por páginas.
RF 7:	Modificar posición del resultado de las búsquedas.
RF 8:	Cambiar el color del logo.
RF 9:	Cambiar el tamaño del logo.
RF 10:	Ocultar logo.
RF 11:	Ocultar barra de menú.
RF 12:	Modificar ubicación de la barra de menú.
RF 13:	Cambiar el color de la barra de menú.
RF 14:	Modificar ubicación del campo de búsqueda.
RF 15:	Cambiar color del campo de búsqueda.
RF 16:	Cambiar el color de letra del campo de búsqueda.
RF 17:	Modificar posición del menú de personalización.
RF 18:	Ajustar transparencia del menú de personalización.
RF 19:	Cambiar el color del menú de personalización.
RF 20:	Establecer temas por defecto.
RF 21:	Regresar a la configuración por defecto.
RF 22:	Registrar usuario.
RF 23:	Autenticar usuario.

2.3.2 Requisitos no Funcionales

Los requisitos no funcionales son propiedades que hacen al producto atractivo, usable, rápido o confiable. Se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar. Además se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software (Somerville, 2007).

Se identificaron un total de 12 requisitos no funcionales, agrupados en 6 categorías que a continuación se presentan.

Requisitos de software

- RNF1. Se requiere la instalación del servidor web Apache en su versión 2.4 y PHP 5.6 o superior para poder visualizar la interfaz web.

Requisitos de hardware

- RNF 2. Para la interfaz web: 4 GB RAM, CPU de 4 núcleos y al menos 40 GB de Disco Duro.

Requisitos de diseño e implementación

- RNF 3. Como lenguaje de programación para la interfaz web se deberá utilizar PHP en su versión 5.6 o mayor.
- RNF 4. Para el desarrollo de la aplicación web se deberá utilizar *Symfony 2.7 LTS* o superior como marco de trabajo.

Requisitos de apariencia o interfaz de usuario

- RNF 5. En la página de resultados siempre estará visible el menú de opciones, el formulario de búsqueda y el paginador, para resoluciones superiores a 800x600.

Requisitos de usabilidad

- RNF 6. Debe contar con la portabilidad necesaria para poder ser transferido de un ambiente a otro o reemplazado por nuevas versiones.

- RNF 7. Los dispositivos clientes que utilizarán la herramienta deben contar con navegadores web que soporten HTML5, CSS3 y Java Script.
- RNF 8. Se requiere el uso de herramientas y recursos de software libre, las cuales se podrán usar, modificar y distribuir libremente.

Requisitos de seguridad

- RNF 9. Integridad: el sistema debe permitir la realización de salvallas periódicas de la información.
- RNF 10. Disponibilidad: el sistema debe permitir una instalación distribuida contribuyendo al balanceo de carga y la redundancia.
- RNF 11. Los formularios deben ser protegidos por un *Token* de seguridad.
- RNF 12. Los campos de entrada deben ser validados.

2.4. Historias de usuario

Siguiendo las disciplinas propuestas por la metodología AUP-UCI y con el propósito de encapsular los requisitos funcionales, pertenecientes a la propuesta de personalización de la interfaz web del buscador cubano, se presentan en la Tabla #4 y #5 las principales Historias de Usuario.

Tabla # 4: Historia de usuario # 1

Historia de usuario	
Número: HU # 1	Nombre: Color del menú de personalización.
Prioridad en negocio: Baja	
Descripción: Permite cambiar el color del menú de personalización. <ul style="list-style-type: none"> • Modificar posición del menú de personalización. <p>La funcionalidad comienza cuando el usuario selecciona del menú de personalización, ubicado en la parte superior derecha de la interfaz, la opción personalización avanzada y de ahí selecciona el color con que</p>	

aparecerá el menú, de un panel de colores; al finalizar presiona el botón “guardar” para establecer los cambios en la interfaz.

Prototipo:

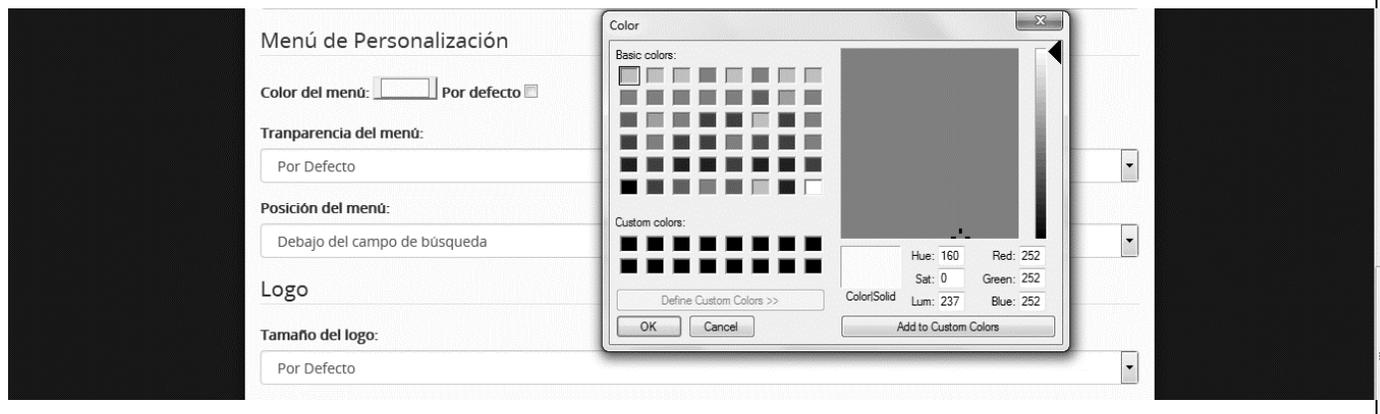


Tabla # 5: Historia de usuario # 2

Historia de usuario	
Número: HU # 2	Nombre: Ubicación del menú principal.
Prioridad en negocio: Baja	
<p>Descripción: Permite cambiar la ubicación del menú principal.</p> <ul style="list-style-type: none"> • Modificar ubicación de la barra de menú principal. <p>La funcionalidad comienza cuando el usuario selecciona del menú de personalización, ubicado en la parte superior derecha de la interfaz, la opción personalización avanzada y de ahí selecciona la posición donde aparecerá el menú de una lista con las opciones “por defecto”, “debajo del campo de búsqueda” o “lateral izquierdo”; al finalizar presiona el botón “guardar” para establecer los cambios en la interfaz.</p>	
Prototipo:	
<p>The image shows a screenshot of the 'Menú de Personalización' dialog box. The 'Color del menú:' section is visible. The 'Transparencia del menú:' dropdown is set to 'Por Defecto'. The 'Posición del menú:' dropdown is set to 'Debajo del campo de búsqueda'. The 'Logo' and 'Tamaño del logo:' sections are partially visible.</p>	

2.5. Estilo arquitectónico

Symfony basa su funcionamiento interno en la arquitectura Modelo - Vista - Controlador (MVC), utilizada por la mayoría de los marcos de trabajo web. No obstante, según su creador *Fabien Potencier*: “*Symfony* no es un marco de trabajo MVC. *Symfony* sólo proporciona herramientas para la parte del Controlador y de la Vista. La parte del Modelo es responsabilidad del usuario” (Potencier, 2011). La siguiente figura muestra la arquitectura del sistema a desarrollar.

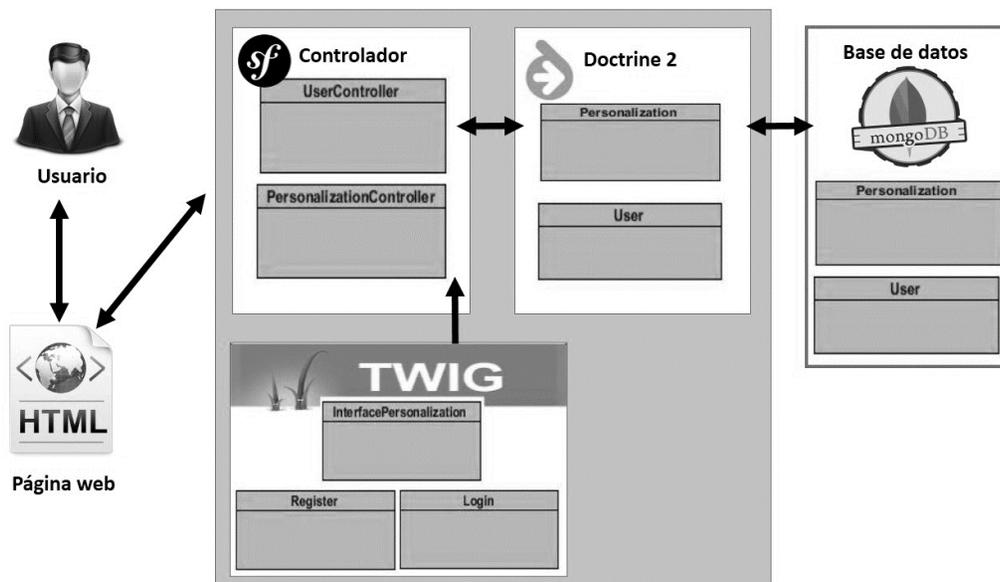


Figura # 3: Arquitectura del sistema

Como se observa en la figura # 3, a través del controlador *UserController* se registra, se le asigna una personalización y se autentica el usuario; y a través del controlador *PersonalizationController* son recibidas y atendidas todas las configuraciones de personalización realizadas por el usuario. Cuando el controlador recibe una petición del usuario, consulta las personalizaciones de cada usuario, almacenadas en la base de datos, para salvar y aplicar la personalización de dicho usuario. Esta arquitectura está basada en el patrón arquitectónico MVC, el manejo de las interfaces de usuario se realiza mediante la plantilla *InterfacePersonalization.html.twig* y la función del modelo la tiene la biblioteca *Doctrine* que mapea las

clases *Personalization* y *User* a las tablas *Personalización* y *Usuario* del sistema gestor de base de datos *MongoDB*.

2.6. Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se deben tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios) (Suarez, 2010).

Los Patrones Generales de Software para Asignar Responsabilidades (GRASP del inglés *General Responsibility Assignment Software Patterns*) tienen una importante utilidad en el diseño de una aplicación, al igual que los *Gang-of-Four* o Pandilla de los Cuatro (GoF por sus siglas en inglés). Los empleados en la solución de la problemática planteada, se exponen a continuación:

- Patrones GRASP:

Experto: Symphony2 utiliza el patrón experto con la inclusión de Doctrine para el mapeo de base de datos. Se utiliza específicamente para crear una capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases entidades con todas las funcionalidades comunes (GET, SET y el constructor de la entidad); las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla asociada (Zaninotto y otros, 2016).

En la solución implementada se pone de manifiesto este patrón en la clase *User*, cuando se añade un nuevo usuario, pues el sistema debe generar el código de seguimiento para dicho usuario y lo hace mediante esta clase la cual posee la información del mismo.

Creador: La instanciación de una clase es una de las actividades fundamentales en un sistema orientado a objetos. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, con lo que se logra menos dependencia y mayores oportunidades de reutilización de código (Zaninotto y otros, 2016).

En la implementación de la propuesta de solución se pone de manifiesto este patrón en la clase *UserController*, encargada de instanciar las clases *User* y *Personalitation*.

Bajo acoplamiento: El acoplamiento es una medida de la fuerza con que un elemento está conectado, tiene conocimiento o confía en otros elementos. El objetivo de este patrón consiste en mantener un bajo nivel de dependencia de otros elementos, por lo que constituye un principio que debe estar presente en todas las decisiones de diseño con lo que se reduce el impacto de los cambios (Zaninotto y otros, 2016).

En la implementación de la personalización de Orión se observa este patrón en: las interfaces visuales que acceden a la clase *IpersonalizationContoller*, encargada de declarar los métodos implementados en la clase *PersonalizationController*. Esto logra una independencia entre las interfaces y la clase *PersnoalizationController*.

Alta cohesión: En el diseño orientado a objetos, la cohesión es una medida de la fuerza con la que se relacionan dicho objetos y del grado de focalización de las responsabilidades de un elemento (clase o subsistema). Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que colaboran entre sí y con otros objetos para simplificar su trabajo. Una clase con alta cohesión es relativamente fácil de mantener, entender y reutilizar (Zaninotto y otros, 2016).

El patrón de alta cohesión se evidencia en las clases *UserController* y *PersonalizationController*, en las cuales cada una de ellas se hace responsable de los usuarios y la personalización respectivamente.

- Patrones GoF

Instancia única (Singleton): El objetivo de este patrón es asegurarse de que de una clase solo existe una instancia y que esta es accesible, es decir, que ofrecer un punto de acceso a ella (Zaninotto y otros, 2016).

El patrón de instancia única se evidencia cuando se realiza la llamada a la variable global *app.user*, que contiene los datos del usuario autenticado, para mostrarle los elementos de personalización. Esa instancia se usa en el fichero *layout.html.twig*.

Decorador (Decorator): Este patrón responde a la necesidad de añadir dinámicamente funcionalidad a un Objeto. Esto nos permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera (Zaninotto y otros, 2016).

En la implementación de la personalización de Orión se observa este patrón en el archivo *layout.html.twig*. Este archivo guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página.

Recuerdo (Memento): Este patrón tiene la finalidad de almacenar el estado de un objeto en un momento dado de manera que se pueda restaurar de manera sencilla. Mantiene almacenado el estado del objeto para un instante de tiempo en una clase independiente de aquella a la que pertenece el, de forma que permita que el objeto sea modificado y pueda volver a su estado anterior (Zaninotto y otros, 2016).

En la implementación de la propuesta de solución este patrón permite guardar el estado del objeto personalización, que puede ser modificado y regresado a su estado inicial.

2.7. Modelo de diseño

El modelo de diseño es aquel que se encarga de describir la relación entre las clases 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).

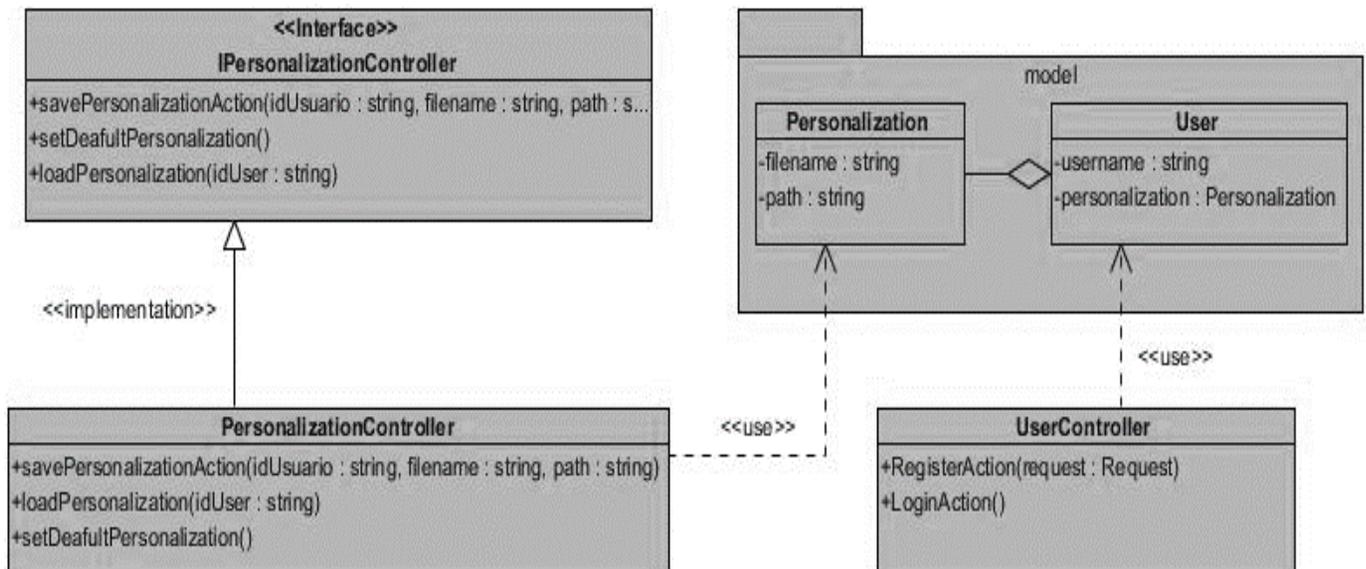


Figura # 4: Diagrama de clases del diseño

IPersonalizationController: es la clase encargada de declarar los métodos, para salvar, cargar y modificar, la personalización de la interfaz web del buscador Orión propia de cada usuario, que serán implementados en la clase PersonalizationController.

PersonalizationController: es la clase encargada de implementar los métodos, para salvar, cargar y modificar la personalización de la interfaz web del buscador Orión propia de cada usuario, declarados en la clase IPersonalizationController.

UserController: es la clase encargada de realizar la el registro y autenticación del usuario. Además asigna una personalización por defecto al usuario que se registra.

Personalization: es la clase que contiene los atributos de personalización de la interfaz web de Orión.

User: contiene los atributos del usuario.

2.8. Modelo de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se relacionan en esos nodos (SparxSystems, 2014).

En el diagrama que a continuación se presenta, el “Dispositivo Cliente” representa un dispositivo desde el cual se podrá acceder a la personalización de la interfaz web del buscador Orión, a través del protocolo *HTTPS*, haciendo uso de un navegador web.

El nodo “*Servidor Web Apache*” es el encargado de atender y ofrecer respuesta a cada una de las solicitudes del cliente.

La comunicación entre el servidor web y el servidor de base de datos se realizará a través de la familia de protocolos TCP-IP. Este último, representado por el nodo “Servidor de Base de Datos MongoDB”, es el encargado de administrar la información que se almacena de cada una de las personalizaciones que el usuario realice a la interfaz web, utilizando el sistema de gestión de base de datos *MongoDB*.

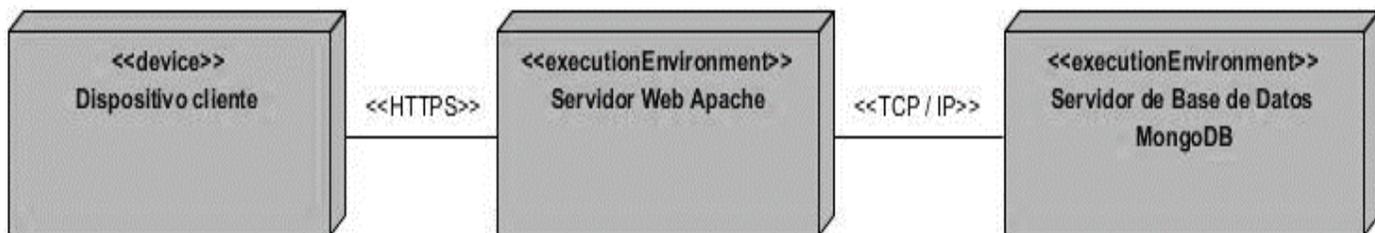


Figura # 5: Diagrama de despliegue de la propuesta de solución

2.9. Conclusiones parciales

En el capítulo se abordaron aspectos correspondientes al análisis y diseño de la propuesta de personalización de la interfaz web del buscador Orión lo que permitió arribar a las siguientes conclusiones:

- La representación y descripción de los artefactos generados garantizan un mejor entendimiento de los flujos de trabajos presentes en el proceso personalización de interfaces web.
- La especificación de los requisitos funcionales y no funcionales del sistema, dio 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 definición de la arquitectura y los patrones de diseño permitió establecer las bases para fomentar la reutilización y las buenas prácticas de programación durante la fase de implementación, así como disminuir el impacto de los cambios futuros en el código fuente.
- La elaboración del diagrama de despliegue permitió identificar la disposición física de los artefactos de la herramienta informática a desarrollarse.

Capítulo 3: Implementación y validación de la personalización de la interfaz web del buscador Orión

3.1. Introducción

En el presente capítulo se muestra el modelo de implementación como resultado del diseño anteriormente desarrollado. La fase de implementación comprende la materialización, en forma de código, de los artefactos, las descripciones y la arquitectura; con el objetivo de conformar el producto final requerido por el cliente (Larman, 2004). Además en este capítulo se presenta la validación del sistema, con el objetivo de corroborar la correspondencia entre el producto y los requisitos definidos anteriormente.

3.2. Modelo de implementación

El modelo de implementación brinda la posibilidad de probar y desarrollar componentes como unidades, que finalmente serán integrados como un sistema ejecutable. En este flujo de trabajo se organizan y realizan las pruebas unitarias y se integran los componentes implementados, basándose en las especificaciones de diseño.

3.2.1. Diagrama de componentes

El diagrama de componentes muestra las relaciones estructurales entre los componentes de un sistema. Los componentes se consideran, unidades autónomas encapsuladas dentro de un sistema o subsistema que proporcionan una o más interfaces. Los diagramas de componentes son generalmente dirigidos al personal de aplicación de un sistema, presenta una comprensión temprana del sistema global que se está construyendo (Bell, 2004).

Seguidamente se presenta el diagrama de componentes correspondiente a la personalización de la interfaz del buscador cubano Orión.

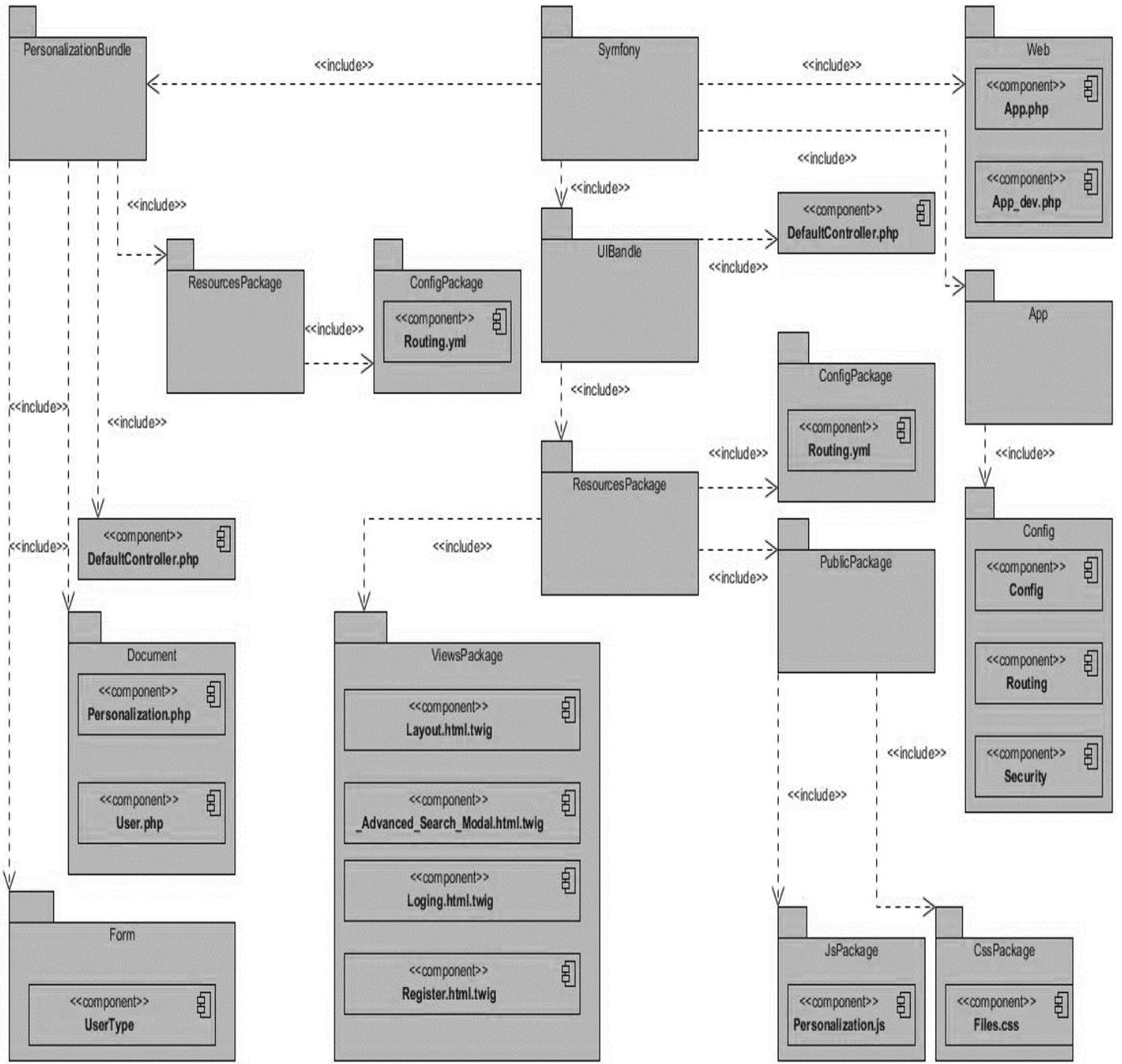


Figura # 6: Diagrama de componentes de la personalización de la interfaz web de Orion

3.3. Estándares de codificación

Un estándar de codificación comprende los aspectos de la generación de código. Este debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. El estándar de codificación debería establecer cómo operar con la base de código existente (Microsoft, 2016).

A continuación se especifican los estándares de codificación a utilizar en la construcción de la solución:

- Usar la notación *CamelCase* en la declaración de variables, funciones y métodos; donde los nombres son formados de varias palabras con la primera letra de cada una en mayúsculas.

```
class DefaultController extends Controller implements AbstractPersonalizationControll

public function defectoAction() {
    $var = $this->getRequest();
    $personalizacion = new Personalizacion();
    $formulario = $this->createForm(new PersonalizacionType(), $personalizacion);
    if ($var->getMethod() == 'POST') {
        $formulario->handleRequest($var);
        $dm = $this->get('doctrine_mongodb')->getManager();
        if ($formulario->isValid()) {

            $usuario = $this->get('security.context')->getToken()->getUser();
            $per = $usuario->getPersonalizacion();
            $dm->remove($per);
            $dm->flush();
        }
    }
}
```

Figura # 7: Fragmento de código fuente de la clase: "PersonalizationController"

- Usar *UpperCamelCase* en los nombres de las clases, declarando la primera letra en mayúscula.

```

class DefaultController extends Controller implements AbstractPersonalizationControll

public function defectoAction() {
    $var = $this->getRequest();
    $personalizacion = new Personalizacion();
    $formulario = $this->createForm(new PersonalizacionType(), $personalizacion);
    if ($var->getMethod() == 'POST') {
        $formulario->handleRequest($var);
        $dm = $this->get('doctrine_mongodb')->getManager();
        if ($formulario->isValid()) {

            $usuario = $this->get('security.context')->getToken()->getUser();
            $per = $usuario->getPersonalizacion();
            $dm->remove($per);
            $dm->flush();
        }
    }
}

```

Figura # 8: Fragmento de código fuente de la clase: "PersonalizationController"

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

```

public function defectoAction() {
    $var = $this->getRequest();
    $personalizacion = new Personalizacion();
    $formulario = $this->createForm(new PersonalizacionType(), $personalizacion);
    if ($var->getMethod() == 'POST') {
        $formulario->handleRequest($var);
        $dm = $this->get('doctrine_mongodb')->getManager();
        if ($formulario->isValid()) {

```

Figura # 9: Fragmento de código fuente de la clase: "PersonalizationController"

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

```

public function defectoAction() {
    $var = $this->getRequest();
    $personalizacion = new Personalizacion();
    $formulario = $this->createForm(new PersonalizacionType(), $personalizacion);
    if ($var->getMethod() == 'POST') {
        $formulario->handleRequest($var);
        $dm = $this->get('doctrine_mongodb')->getManager();
        if ($formulario->isValid()) {

```

Figura # 10: Fragmento de código fuente de la clase: "PersonalizationController"

- Añadir un salto de línea antes de una sentencia “return”.

```
public function reordenarAction() {
    $dm = $this->get('doctrine_mongodb')->getManager();

    return true;
}
```

Figura # 11: Fragmento de código fuente de la clase: "PersonalizationController"

- Utilizar la palabra reservada use una vez por cada declaración seguida de una línea en blanco.

```
<?php

namespace ORION\Bundle\StoredBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;

use ORION\Bundle\StoredBundle\Controller\AbstractPersonalizationController;

use ORION\Bundle\StoredBundle\Document\Usuario;

use ORION\Bundle\StoredBundle\Document\Orden;

use ORION\Bundle\StoredBundle\Document\Personalizacion;

use \ORION\Bundle\StoredBundle\Form\PersonalizacionType;

use Symfony\Component\HttpFoundation\Response;

use Symfony\Component\HttpFoundation\Request;

class DefaultController extends Controller implements AbstractPersonalizationControl:
```

Figura # 12: Fragmento de código fuente de la clase: "PersonalizationController"

- Declarar cada variable en una línea distinta.

```
public function defectoAction() {
    $var = $this->getRequest();
    $personalizacion = new Personalizacion();
    $formulario = $this->createForm(new PersonalizacionType(), $personalizacion);
```

Figura # 13: Fragmento de código fuente de la clase: "PersonalizationController"

3.4. Pruebas de software

El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las prueba para la detección de errores. Además son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2010).

En este subcapítulo se presentarán las pruebas que se aplican a la personalización de la interfaz web del buscador Orión, según las exigencias de la metodología AUP-UCI.

3.4.1. Pruebas funcionales

Las pruebas funcionales aplicadas al software tienen como objetivo probar que los sistemas desarrollados cumplan con las funciones específicas para las que fueron concebidos. Dichas pruebas están basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. El enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y los datos de salida (Oré, 2009).

A continuación se exponen, a partir de un fragmento del caso de prueba correspondiente a la historia de usuario **Cambiar color del menú de personalización**, los aspectos concernientes a las pruebas funcionales realizadas. Los casos de prueba restantes pueden ser consultados en el Anexo # 4.

Tabla # 6: Caso de prueba # 1

Escenario	Descripción	Color	Respuesta del sistema	Flujo central
EC 1.1 Cambiar el color del menú de personalización correctamente.	El usuario selecciona el color del menú de personalización.	Si	El sistema muestra el menú de personalización del color seleccionado.	El usuario accede al menú de personalización y modifica el color del menú de personalización. El sistema muestra como se ve el cambio efectuado. El usuario selecciona la opción "Guardar"
EC 1.2 Cambiar el color del menú de	El usuario selecciona el color del menú	Si	El sistema no muestra el menú de personalización	El usuario accede al menú de personalización y modifica el color del

personalización incorrectamente.	de personalización.		con el color seleccionado.	menú de personalización. El sistema no establece el color seleccionado como color del menú de personalización.
----------------------------------	---------------------	--	----------------------------	--

Tabla # 7: Variables empleadas en el diseño del caso de prueba # 1

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Color	Panel de colores	No	Permite seleccionar de un panel de colores, el color con que se verá el menú de personalización; o seleccionar el color por defecto del menú de personalización.

Para probar el correcto funcionamiento de las funcionalidades del sistema se realizaron tres iteraciones de pruebas a la herramienta. En la tabla que a continuación se presenta, se muestran los resultados obtenidos en cada iteración de pruebas a la personalización de la interfaz web del buscador cubano, así como la corrección de cada uno de los errores.

Tabla # 8: Cantidad de no conformidades por cada iteración las pruebas

No conformidades	Primera iteración	Segunda iteración	Tercera iteración
Detectadas	12	5	3
Resueltas	12	5	3
Pendientes	0	0	0

Para apreciar de forma más ilustrativa el comportamiento de las no conformidades, detectadas en cada iteración de las pruebas funcionales, se presenta la siguiente figura.

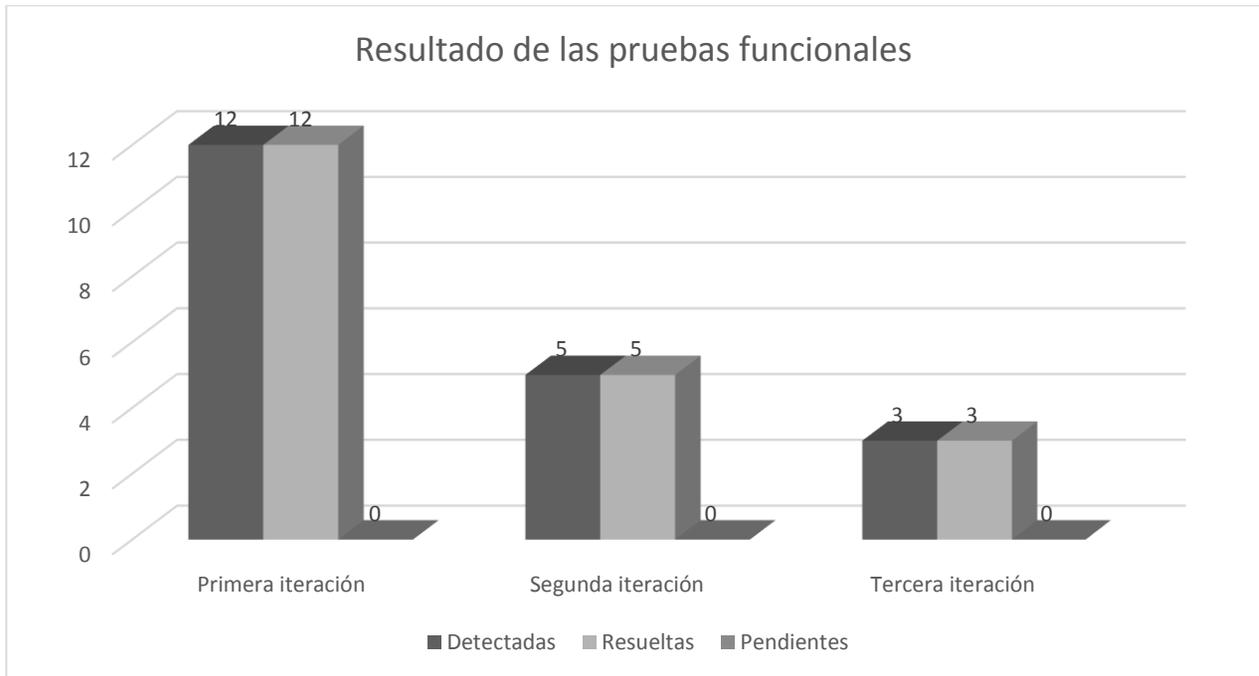


Figura # 14: Comportamiento de las no conformidades por cada iteración

Al finalizar el desarrollo de la personalización del buscador Orión se realizó una cuarta iteración y se comprobó que la corrección de los errores detectados, en las tres primeras iteraciones, no trajeron consigo más errores. Con esto se logró un sistema sin errores funcionales.

3.4.2. Pruebas de seguridad

Las pruebas de seguridad comprenden la puesta en práctica de un conjunto de medidas preventivas y reactivas en los sistemas, que permiten la protección de la información, persiguiendo como objetivo principal la integridad, confidencialidad y disponibilidad de la misma (Mifsud, 2012)

Para garantizar la seguridad de la personalización de la interfaz web del buscador cubano se le realizan una serie de pruebas donde se detectaron 9 alertas en total, de ellas ninguna de clasificación alta, 6 de clasificación media y 3 de clasificación baja.

Las pruebas realizadas se mencionan seguidamente.

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

- Detección de ficheros y directorios.

Después de analizar los resultados obtenidos en las pruebas de seguridad, se procedió a corregir las deficiencias encontradas y se realizó una nueva iteración en la que no se detectaron errores.

3.4.3. Pruebas de usabilidad

En el contexto del desarrollo de software, la usabilidad está considerada como uno de los factores de calidad de mayor importancia para el éxito de un proyecto. De manera general, el término usabilidad es empleado para referirse a la capacidad que posee un producto de ser utilizado por los usuarios de forma fácil, eficiente y con satisfacción, en un determinado contexto de uso (Perurena y otros, 2013).

Se le aplicó la lista de chequeo UCI a la personalización de la interfaz web del buscador Orión. Dicha lista está desarrollada por los especialistas del grupo de Seguridad del Departamento de Evaluación de Productos de Software (DEPSW), perteneciente al Centro Nacional de Calidad de Software (CALISOFT) radicado en la UCI.

A continuación se muestran los resultados de la aplicación de la lista de chequeo a la personalización de la interfaz web del buscador Orión.

Tabla # 9: Resultados de prueba de usabilidad utilizando lista de chequeo

Categorías de Indicadores	Indicadores	Correctos
Visibilidad del sistema	7	2
Adecuación del sistema	16	16
Control y libertad de usuarios	14	4
Consistencia y estándares	17	17
Prevención de errores	3	3
Reconocimiento	9	9
Flexibilidad y eficiencia de uso	7	7
Estética	21	10

Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores	3	1
Ayuda y documentación	3	0
Accesibilidad	35	35
Comprobaciones técnicas	9	9
TOTAL	144	113

En la tabla anterior se puede apreciar que la implementación de la personalización de la interfaz web del buscador cubano cumple con 113 indicadores de un total de 144 indicadores de usabilidad, lo que representa aproximadamente un 78,47 %. Para favorecer la comprensión de los resultados estadísticos obtenidos se han representado los mismos en la siguiente gráfica.

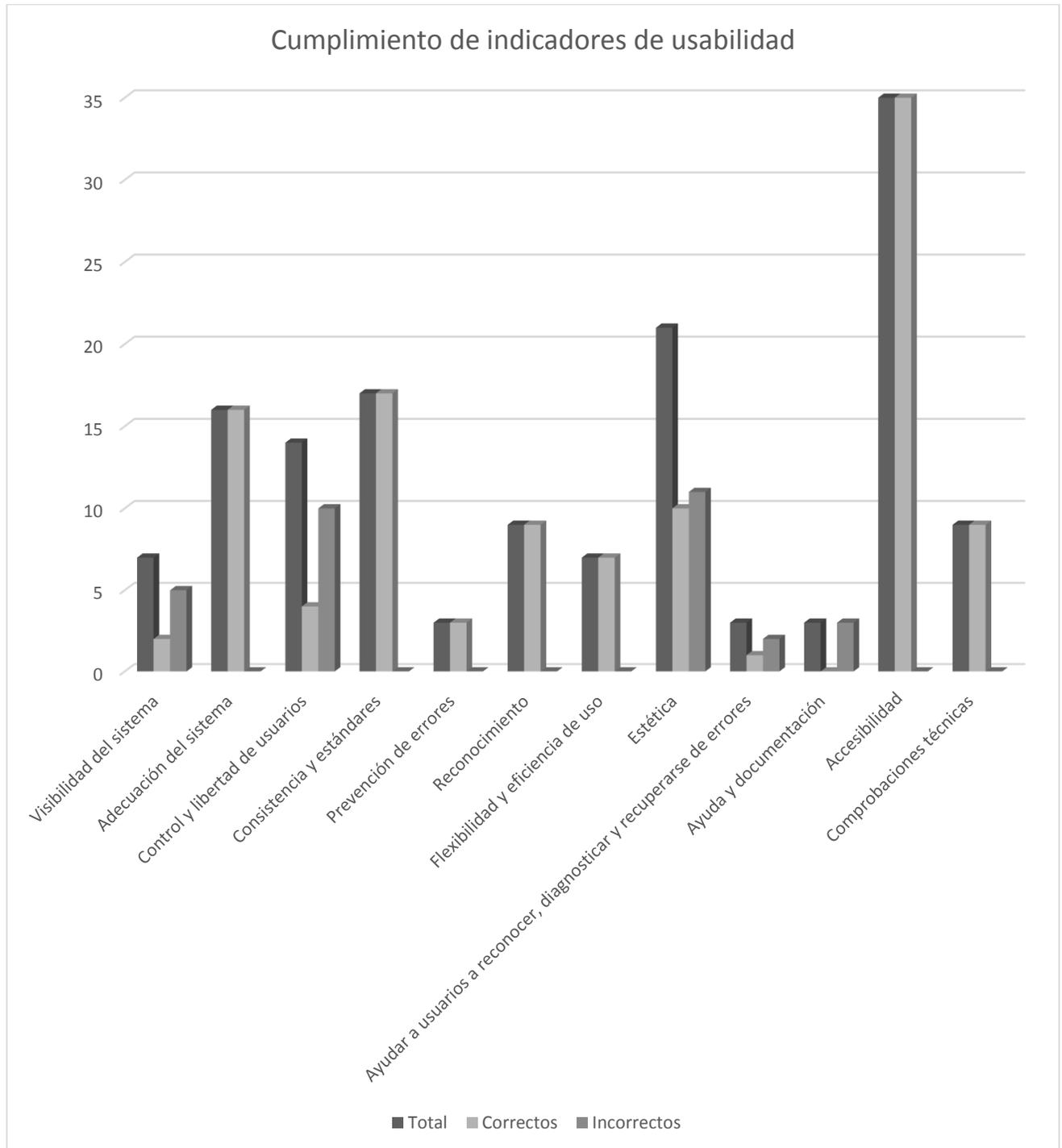


Figura # 15: Cumplimiento de indicadores de usabilidad

Luego de analizar los resultados obtenidos en las pruebas de usabilidad, se identificaron 20 indicadores con posibles mejoras de acuerdo al alcance del presente trabajo. Estos fueron corregidos en función de elevar el nivel de facilidad de uso e interacción entre el usuario y las operaciones que se realizan. Con esto se logró un incremento en el nivel de usabilidad hasta de un 13.89%. En la gráfica siguiente se representa el estado del nivel de usabilidad resultante.

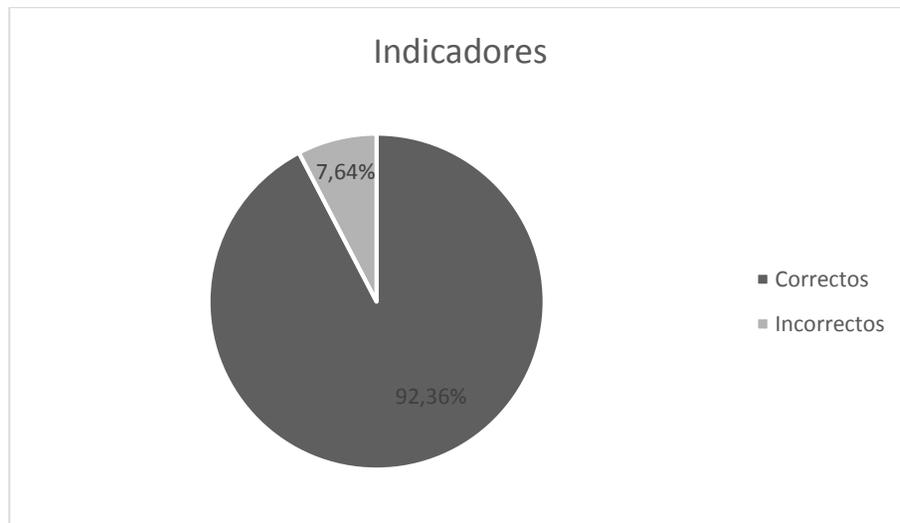


Figura # 16: Nivel de usabilidad después de corregidos indicadores de usabilidad

Con los resultados obtenidos se confirma que la implementación de la personalización de la interfaz web del buscador favorece su usabilidad. Esto constituye una ventaja en la tarea de captar la atención de los usuarios para fidelizarlos con el sitio.

3.4.1. Pruebas de integración

El proceso de integración del sistema implica construir este a partir de sus componentes y probar el sistema para detectar problemas que puedan surgir. Las pruebas de integración comprueban que estos componentes funcionen juntos, son llamados correctamente y transfieren los datos correctos en el tiempo preciso a través de sus interfaces (Sommerville, 2005).

Se realizaron pruebas de integración con la finalidad de validar la compatibilidad y el funcionamiento de las interfaces que comunican la personalización de la interfaz web del buscador Orión con el sistema. Para ello

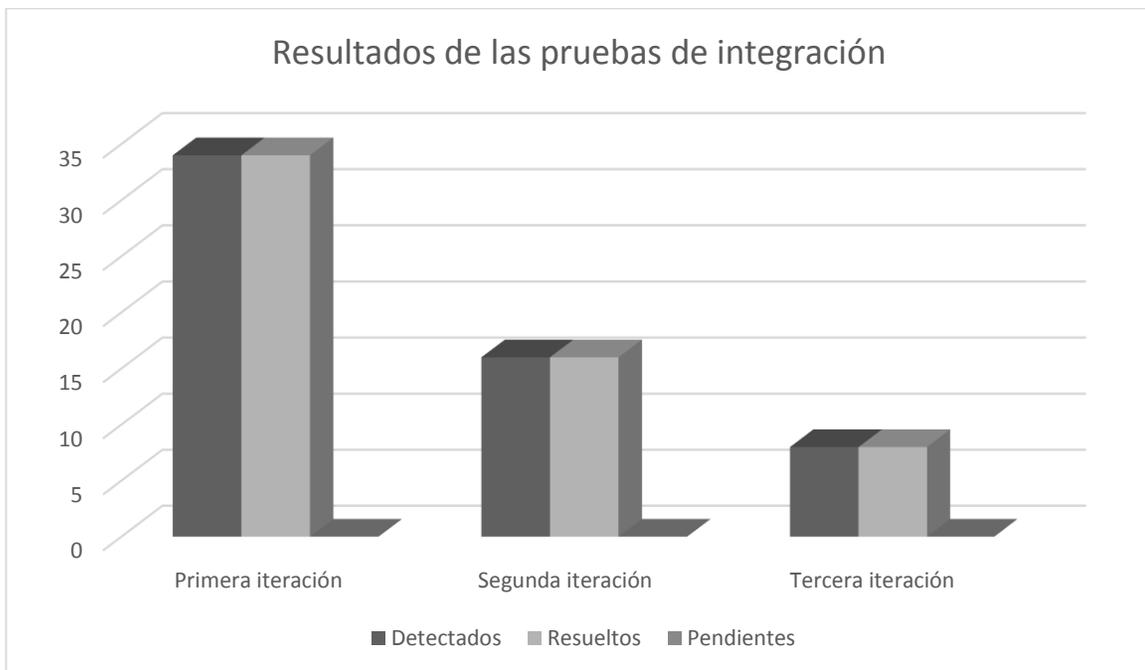
se integró el *StoreBundle*, que contiene la personalización implementada, al proyecto Orión; esta operación arrojó un total de 58 errores, a continuación se muestran los resultados obtenidos en cada iteración de pruebas, así como la corrección de cada uno de los errores.

Tabla # 10: Cantidad de errores por cada iteración las pruebas de integración

Errores	Primera iteración	Segunda iteración	Tercera iteración
Detectados	34	16	8
Resueltos	34	16	8
Pendientes	0	0	0

Para apreciar de forma más ilustrativa el comportamiento de los errores detectados en cada iteración de las pruebas de integración, se presenta la siguiente figura.

Tabla # 11: Comportamiento de los errores detectados en las pruebas de integración



Se realizó una cuarta iteración de las pruebas de integración de la personalización del buscador Orión y se comprobó que la corrección de los errores detectados, en las tres primeras iteraciones, no trajeron consigo más errores; con esto se afirma que la solución implementada se integra completamente al buscador Orión.

3.5. Conclusiones parciales

La realización de este capítulo, en el cual se abordaron una serie de aspectos correspondientes a la implementación y validación de la personalización de la interfaz web del buscador Orión, permitió llegar a las siguientes conclusiones:

- La realización del diagrama de componentes, así como la especificación de los estándares de codificación permitieron visualizar con facilidad la estructura general de la solución.
- Las pruebas realizadas a la personalización de la interfaz web del buscador permitieron detectar los errores presentes, corregirlos en el menor tiempo posible y entregar al cliente una aplicación con mayor calidad, seguridad y usabilidad.

CONCLUSIONES GENERALES

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

- El estudio de sistemas homólogos permitió determinar que existen buscadores con funcionalidades que implican una personalización de sus interfaces web, pero ninguno de licencia libre para su uso.
- La selección de la metodología AUP-UCI, las tecnologías y las herramientas para el desarrollo de la personalización de Orión, permitió concretar, en correspondencia con las especificaciones del cliente, las características que debía tener la herramienta a desarrollarse.
- La modelación de los artefactos facilitó el soporte a la implementación de los requisitos previamente expresados por el cliente y garantizó la estructura base para la organización lógica del código fuente.
- La implementación de la personalización de la interfaz web del buscador Orión permitió obtener una interfaz web del buscador Orión más atractiva, dinámica e intuitiva.
- La evaluación de las pruebas de software realizadas permitió erradicar las insuficiencias detectadas en la herramienta desarrollada y corregirlas en función de lograr un producto con calidad y que satisfaga las necesidades del cliente.

RECOMENDACIONES

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

- Implementar una funcionalidad que permita a los usuarios cargar, como imagen de fondo, cualquier imagen que tengan almacenada en el dispositivo por el cual acceden al buscador.
- Integrar a la plataforma de contenido unificado de búsqueda avanzada CUBA la personalización de su interfaz, tomando la realizada en esta investigación como base.

REFERENCIAS BIBLIOGRÁFICAS

ACOSTA, J.; GREINER, C., et al. *Medición de atributos POO en frameworks de desarrollo PHP*. [En línea] *Medición de atributos POO*, 2012. [Citado el: 10 de Mayo de 2016]. Disponible en: <http://docplayer.es/7051044-Medicion-de-atributos-poo-en-frameworks-de-desarrollo-php.html>

ACUNETIX. *Audit your website security with Acunetix Web Vulnerability Scanner*. [En línea] Acunetix, 2016. [Citado el: 3 de Mayo de 2016]. Disponible en: <http://www.acunetix.com/>

ALEMAN- JIMENEZ, Y; THOMAS-SOSA, Y. *Módulo de configuración para el mecanismo de rastreo del buscador Orión*. Trabajo de diploma de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana-Cuba, 2014.

ALPÍZAR, N.; ARGUELLO, O., et al. *Agile UP. El Proceso Unificado Ágil v1.1*. [En línea] Agile UP. 13 de mayo de 2006. [Citado el: 20 de Octubre de 2015]. [Citado el: 20 de Octubre de 2015]. Disponible en: <http://www.cc.una.ac.cr/AUP/>

AMBLER, S. *Ambyssoft*. [En línea] Ambyssoft, 2014. [Citado el: 10 de Noviembre de 2015] Disponible en: <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.

BAEZA, Y.; CASTILLO, C., et al. *Crawling a Country: Better Strategies than Breadth-First for Web Page Ordering*. [En línea] *Crawling a Country*, 2005. [Citado el: 18 de Noviembre de 2015]. Disponible en: <https://udesantiago.pure.elsevier.com/en/publications/crawling-a-country-better-strategies-than-breadth-first-for-web-p>

BAEZA, Y.; RIBEIRO, N. *Modern information retrieval*. [En línea] Modern information retrieval, 1999. [Citado el: 18 de Noviembre de 2015]. Disponible en: <http://people.ischool.berkeley.edu/~hearst/irbook/print/chap10.pdf>

BECK, K. *Extreme Programming Explained. Embrace Change, Pearson Education*. [En línea] Extreme Programming Explained, 2000. [Citado el: 12 de Noviembre de 2015]. Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

BELL, D. *UML basics: The component diagram*. [En línea] UML basics, 2004. [Citado el: 10 de marzo 2015]. Disponible en: <https://www.ibm.com/developerworks/rational/library/dec04/bell/>

CANOS J.; LETIER P., et al. *Metodologías ágiles en el desarrollo de software*. [En línea] Metodologías ágiles en el desarrollo de software, 2003. [Citado el: 20 de Noviembre de 2015] Disponible en: <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>

CARRILLO, G.; RAMÍREZ, Y. *Colocándonos en la web*. [En línea] Colocándonos en la web, 2012. [Citado el: 10 de Noviembre de 2014]. Disponible en: <https://periodismojosemarti.wordpress.com/2012/11/28/colocandonos-en-la-web/>

COCKBUN, A. *Agile Software Development*. [En línea] Agile Software Development, 2001. [Citado el: 20 de Noviembre de 2015]. Disponible en: <http://dl.acm.org/citation.cfm?id=502980>

CODEIGNITER. *Codelgniter Rocks*. [En línea] Codelgniter, 2015. [Citado el: 27 de Noviembre de 2015]. Disponible en: <https://www.codeigniter.com/>

DEFINICION ABC. *Definición de Buscador*. [En línea] Definición de un buscador, 2015. [Citado el: 4 de Noviembre de 2015]. Disponible en: <http://www.definicionabc.com/tecnologia/buscador.php>

DEFINICION.DE. *Definición de Internet*. [En línea] *Qué es internet, significado y concepto*, 2015. [Citado el: 4 de Noviembre de 2015]. Disponible en: <http://definicion.de/internet/>

GARCÍA, G. *PHP5 A través de ejemplos*. [En línea] PHP5 a través de ejemplos, 2010. [Citado el: 1 de Diciembre de 2015]. Disponible en: <http://somosprogramacion.blogspot.com/2015/02/php-5-traves-de-ejemplos-abraham-gutierrez-y-gines-bravo.html>

GROUSSARD, T. *Recursos Informáticos Java Enterprise*. Barcelona, ediciones software, 2010. 34 p.

GUTIÉRREZ, J. *¿Qué es un framework web?* [En línea] Framework web, 2014. [Citado el: 14 de Noviembre de 2015]. Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

HERNÁNDEZ-FERNÁNDEZ, L. *Manual de Usuario del Motor de Búsqueda Cubano*. Trabajo de diploma de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana-Cuba, 2013.

HERNÁNDEZ-CLARO; GREGUAS-NAVARRO. *Desarrollo de Interfaces Web para el SIGEP.* Trabajo de diploma de Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, La Habana-Cuba, 2008.

HERRERA, A. *Modelos de Sistemas de Recuperación de Información Lingüística Difusa.* [En línea] *Modelos de Sistemas de Recuperación de Información Lingüística Difusa*, 2006. [Citado el: 1 de Diciembre de 2015]. Disponible en: http://eprints.rclis.org/19103/1/10o-seminario-hispanomexicano_manuel-blazquez-ochando.pdf

HIDALGO-DELGADO, Y. *Orión, un motor de búsquedas para la web de la UCI.* Trabajo de diploma de Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, La Habana-Cuba, 2010.

INGENIERÍA DE SOFTWARE. *AUP Ingeniería de software.* [En línea] *AUP*, 2015. [Citado el: 14 de Noviembre de 2015]. Disponible en: http://ingenieriadesoftware.mex.tl/63758_AUP.html.

IWORLD. *La revista de Internet.* [En línea] *Técnicas y modelos de personalización de sitios web*, 2015. [Citado el: 4 de Noviembre de 2015]. Disponible en: <http://www.adm3.unlp.edu.ar/regular/material/tecnicasdepersonalizacion.pdf>

LARMAN, C. *UML y Patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado.* [En línea] *UML y Patrones*, 2004. [Citado el: 1 de Diciembre de 2015]. Disponible en: <http://www.fmonje.com/UTN/ADES%20-%202008/UML%20y%20Patrones%20%202da%20Edicion.pdf>

LETIER, P.; PENADES, C. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* [En línea] *Metodologías ágiles para el desarrollo de software*, 2006. [Citado el: 20 de Noviembre de 2015] Disponible en: www.cyta.com.ar/ta0502/v5n2a1.htm

LIE, H.; BOS, B. *Cascading Style Sheets – designing for the Web.* [En línea] *Cascading Style Sheets*, 2005. [Citado el: 1 de Diciembre de 2015]. Disponible en: <https://www.w3.org/Style/LieBos3e/>

LÓPEZ, H. *Modelos de Sistemas de Recuperación de Información Lingüística Difusa.* [En línea] *Modelos de Sistemas de Recuperación de Información Lingüística Difusa*, 2006. [Citado el: 18 de Noviembre de 2015]. Disponible en: <http://www.tdx.cat/handle/10803/15699>

LUJAN, S. M. *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Editorial Club Universitario, 2002. 321 p.

MAGLIONE, C.; VARLOTTA, N. *Investigación, gestión y búsqueda de información en Internet*. [En línea] Investigación, gestión y búsqueda de información en Internet, 2012. [Citado el: 18 de Noviembre de 2015]. Disponible en: <http://bibliotecadigital.educ.ar/uploads/contents/investigacion0.pdf>

MARINO, C. *Programación en PHP5. Nivel básico*. [En línea] Programación en PHP5, 2008. [Citado el: 1 de Diciembre de 2015]. Disponible en: http://administraciondesistemas.pbworks.com/f/Manual_PHP5_Basico.pdf

MICROSOFT. *Revisiones de código y estándares de codificación*. [En línea] Revisiones de código y estándares de codificación, 2016. [Citado el: 21 marzo 2016]. 2016. Disponible en: [https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx)

MIFSUD, E. *Introducción a la seguridad informática*. [En línea] Introducción a la seguridad informática, 2012. [Citado el: 22 de marzo 2016]. Disponible en: <http://recursostic.educacion.es/observatorio/web/en/software/software-general/1040-introduccion-a-la-seguridad-informatica?showall=1>

MONGODB. *Reinventando la gestión de datos*. [En línea] Reinventando la gestión de datos, 2016. [Citado el: 17 de mayo 2016]. Disponible en: <https://www.mongodb.com/es>

MOZILLA PROJECT. *JavaScript Overview*. [En línea] Mozilla Developer Network, 2013. [Citado el: 1 de Diciembre de 2015]. Disponible en: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Overview>

NETBEANS. *Bienvenido a NetBeans*. [En línea] NetBeans, 2015. [Citado el: 3 de Diciembre de 2015.] Disponible en: https://netbeans.org/index_es.html

NETCRAFT. *Servidores web más usados desde 1995*. [En línea] Sitio web de NetCraft, 2013. [Citado el: 25 de Diciembre de 2013]. Disponible en: <http://www.netcraft.com>

NETMARKETSHARE. *Market Share Statistics for Internet Technologies*. [En línea] Market share formobile, browsers, operating systems and search engines, 2015. [Citado el: 6 de Noviembre de 2015]. Disponible en: www.netmarketshare.com

NOVOA-PROENZA, M.; ALVAREZ-GÓMEZ, Y. *Capa de servicios web para el motor de búsqueda Orión*. Trabajo de diploma de Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas, La Habana-Cuba, 2012.

OBJECT MANAGEMENT GROUP. *Introduction to OMG's Unified Modeling Language*. [En línea] Introduction to OMG's Unified Modeling Language, 2015. Citado el: [3 de Diciembre de 2015]. Disponible en: http://www.omg.org/gettingstarted/what_is_uml.htm

ORACLE. *Características del IDE NetBeans*. [En línea] Sitio oficial del IDE NetBeans, 2014. Citado el: [3 de Diciembre de 2015]. Disponible en: <https://netbeans.org/features/index.html>

ORDENADORES Y PORTATILES. *Buscador Bing*. [En línea] Características del buscador Bing. 2015. [Citado el: 18 de Noviembre de 2015]. Disponible en: www.ordenadores-y-portatiles.com/buscador-bing.html

ORE, B. *Pruebas Funcionales*. [En línea] *Pruebas Funcionales*, 2009. [Citado el: 22 de marzo 2016]. Disponible en: http://www.calidadyssoftware.com/testing/pruebas_funcionales.php

PAUL, R., L.; *Personalización de Orión*. [Citado el: 13 de mayo 2016]. La Habana.

PERURENA, L.; MORAGUEZ, M. *Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación*. [En línea] *Usabilidad de los sitios Web*, 2013. [Citado el: 22 de marzo 2016]. Disponible en: <http://www.acimed.sld.cu/index.php/acimed/article/view/405/306>

POTENCIER, F. *What is Symfony2?* [En línea] SensioLabsNetwork, 2011. [Citado el: 27 de Noviembre de 2015]. Disponible en: <http://fabien.potencier.org/article/49/what-is-symfony2>

PRESSMAN, R. S. *Ingeniería del Software, un enfoque práctico*. México, McGraW-Hill Companies, 2010. 805 p.

REAL ACADEMIA ESPAÑOLA. *Diccionario de la lengua española*. [En línea] Real Academia Española, 2015. [Citado el: 4 de Noviembre de 2015]. Disponible en: <http://lema.rae.es/drae/?val=informaci%C3%B3n>

REYES, Y.; GONZÁLEZ, L., et al. *Estudio comparativo sobre las principales metodologías pesadas y orientadas a objeto en el desarrollo de software*. [En línea] Metodologías pesadas y orientadas a objeto, 2008. [Citado el: 4 de Octubre de 2015]. Disponible en <http://ccia.cujae.edu.cu/index.php/siia/siia2008/paper/viewFile/1240/298>

SCHWABER, K.; BEEDLE, M., et al. *Agile Software Development with SCRUM*. [En línea] Prentice Hall, 2001. [Citado el: 20 de Noviembre de 2015]. Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

SEKER, S. E. *Performance Evaluation of a Regular Expression Crawler and Indexer*. [En línea] 2012. [Citado el: 1 de Diciembre de 2015]. Disponible en: http://www.researchgate.net/publication/236622946_Performance_Evaluation_of_a_Regular_Expression_Crawler_and_Indexer/file/e0b4951877458dc6a2.pdf

SOLER, P. V. *El uso de las TIC (Tecnologías de la Información y la Comunicación) como herramienta didáctica en la escuela, en Contribuciones a las Ciencias Sociales* [En línea] El uso de las TIC, 2008. [Citado el: 6 de Noviembre de 2015] Disponible en: www.eumed.net/rev/cccss/02/vsp.htm

SOMERVILLE, I. *Software Engineering* [En línea] Software Engineering, 2007. [Citado el: 21 marzo 2016]. Disponible en: http://eva.uci.cu/mod/resource/view.php?id=9269&subdir=/Sommerville_8va_edicion

SPARXSYSTEMS. *Diagrama de Despliegue UML 2*. [En línea] Sparx Systems-Tutorial UML 2-Diagrama de Despliegue, 2014. [Citado el: 15 de Febrero de 2015] Disponible en: http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html

STARK, N. S. *Motores de búsqueda en internet* [En línea] 2001. [Citado el: 6 de Noviembre de 2015]. Disponible en: <http://www.unlu.edu.ar/~tyr/tyr/TYR-motor/stark-motor.pdf>

SUÁREZ, P. J. *Excelentes plugins para crear gráficas con jquery*. [En línea] Excelentes plugins para crear gráficas con jquery, 2010. [Citado el: 15 de Febrero de 2015]. Disponible en: <https://www.w3.org/Style/LieBos3e/>

SYMFONY. *Symfony*. [En línea] Symfony, 2015 [Citado el: 27 de Noviembre de 2015]. Disponible en: <https://www.symfony-project.com/>

TOLOSA, G.; BORDIGNON, F. *Introducción a la Recuperación de Información*. [En línea] *Introducción a la Recuperación de Información*, 2008. [Citado el: 20 de Octubre de 2015]. Disponible en: <http://admondeinformacion.blogspot.com/2011/09/introduccion-la-recuperacion-de.html>

VALDÉZ, J. L. *Enciclopedia y Biblioteca Virtual de las Ciencias Sociales, Económicas y Jurídicas*. [En línea] Enciclopedia y Biblioteca Virtual de las Ciencias Sociales, Económicas y Jurídicas, 1994. [Citado el: 22 de Octubre de 2015]. Disponible en: <http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>

VISUAL PARADIGM. *Visual Paradigm for UML - Software design tools for agile software development*. [En línea] Visual Paradigm for UML, 2014. Citado el: [3 de Diciembre de 2015]. Disponible en: <http://www.visual-paradigm.com/product/vpuml/>

WORLD WIDE WEB CONSORTIUM. *HTML/Specifications*. [En línea] HTML/Specifications, 2014. [Citado el: 1 de Diciembre de 2015]. Disponible en: <http://www.w3.org/community/webed/wiki/HTML/Specifications#HTM>

YAHOO. *The History of Yahoo! -How It All Started*. [En línea] The History of Yahoo! 2005. [Citado el: 16 de Noviembre de 2015]. Disponible en: <http://docs.yahoo.com/info/misc/history.html>

Yii. *Yii*. [En línea] *Yii*, 2015. [Citado el: 27 de Noviembre de 2015]. Disponible en: <https://www.yiiframework.com/>

ZAMBRANO, R. A. *Metodología de la investigación*. [En línea] Metodología de la investigación, 2013. [Citado el: 22 de Octubre de 2015]. Disponible en: <http://es.slideshare.net/Mal3J1Ta/resumen-capitulos-1234-del-libro>

ZANINOTTO, F. y POTENCIER, F. *Librosweb.es. Symfony 2.0, la guía definitiva*. [En línea] Symfony 2.0, la guía definitiva, 2016. [Citado el: 19 de mayo 2016]. Disponible en: http://librosweb.es/libro/symfony_2_0/

ZEND. *Zend*. [En línea] *Zend*, 2015. [Citado el: 27 de Noviembre de 2015]. Disponible en: <https://www.framework.zend.com/>

GLOSARIO DE TÉRMINOS

Personalización: acción de modificar algo de acuerdo a las preferencias personales.

Multiplataforma: se refiere a los programas informáticos que pueden funcionar en diversas plataformas.

Marco de trabajo: estructura de artefactos o módulos concretos con base en la que otro proyecto de software puede ser desarrollado.

Apache: servidor web de distribución libre y de código abierto.

Metodología: se refiere a una forma de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas informáticos.

ESPECIFICACIÓN DE ACRÓNIMOS

UCI: Universidad de las Ciencias Informáticas.

RI: Recuperación de información.

SRI: Sistemas de recuperación de información.

TIC: Tecnologías de la información y las comunicaciones.

AUP: Proceso unificado ágil.

SINI: Soluciones Informáticas para Internet.

CIDI: Centro de Ideoinformática.

CUBA: Contenidos unificados para la búsqueda avanzada.

MVC: Modelo Vista Controlador.