

Universidad de las Ciencias Informáticas

Facultad I



Título:

Personalización del Open Harvester Systems.

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

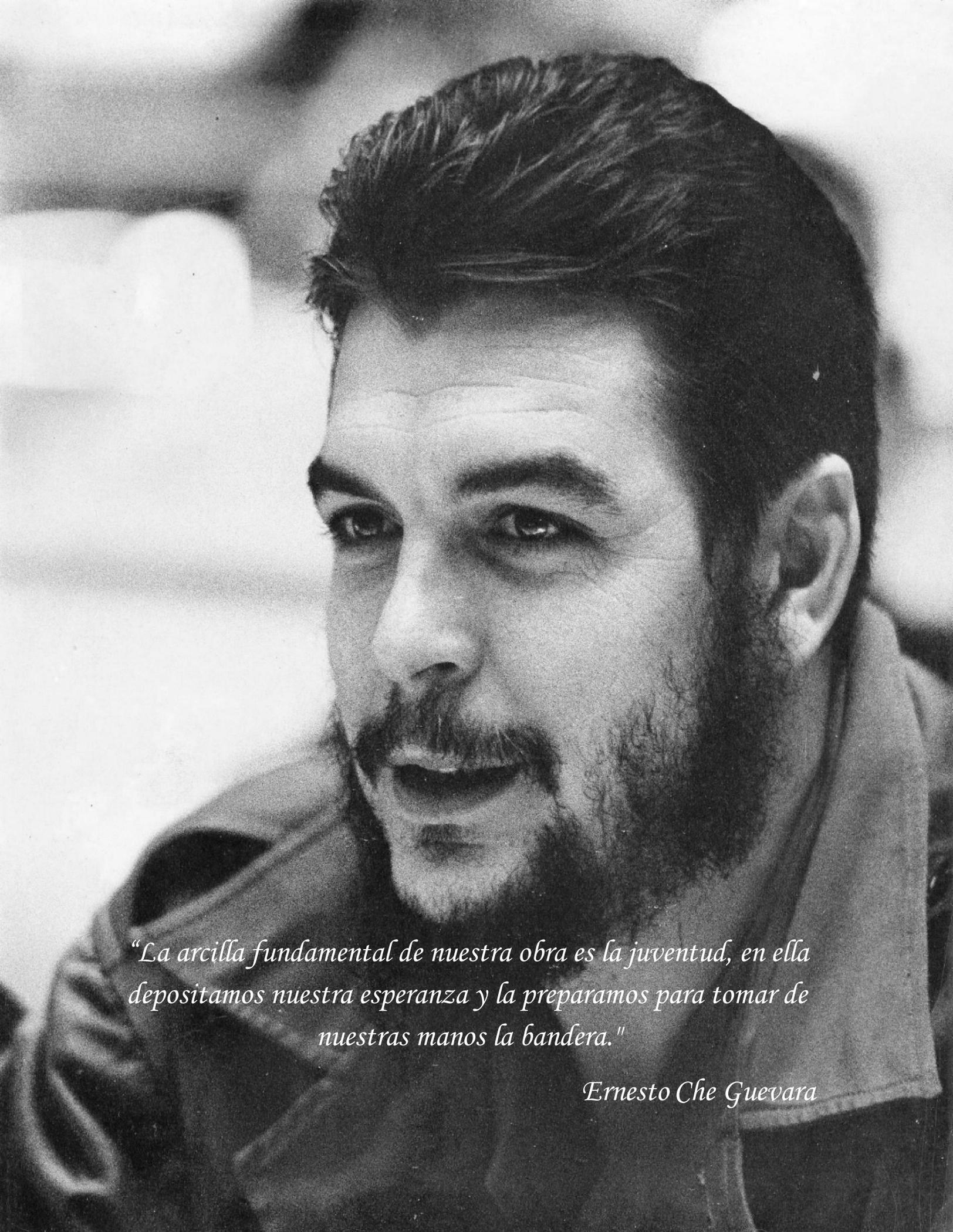
Autor: Yoevis Diaz Dominguez.

Tutores: Ing. Marilys Valiente González.

Ing. Luis Dominguez Cruz.

La Habana, 2013.

“Año 55 de la Revolución”.

A black and white close-up portrait of Ernesto Che Guevara. He is looking slightly to the left of the camera with a serious expression. He has dark, wavy hair and a full beard and mustache. He is wearing a dark jacket. The background is out of focus, showing some light-colored shapes.

“La arcilla fundamental de nuestra obra es la juventud, en ella depositamos nuestra esperanza y la preparamos para tomar de nuestras manos la bandera.”

Ernesto Che Guevara

***D*Declaración de autoría**

Declaración de autoría

Declaro ser el autor del presente Trabajo de Diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo el presente a los _____ días del mes de _____ del año _____.

Yoevis Diaz Dominguez

Autor

Ing. Marilys Valiente González

Tutora

Ing. Luis Dominguez Cruz

Tutor

Agradecimientos

Agradecimientos

Agradezco a todas las personas que de una forma u otra han aportado a mi formación personal, especialmente a las que han puesto todo su empeño en darme fuerzas y seguridad cuando más lo he necesitado.

Gracias a mi familia por su apoyo incondicional en todo momento, por guiarme y ayudarme a ser un mejor hombre cada día. A mi mamá, por ser la mejor madre del mundo, por su grandísimo apoyo, su amor, por ser tan fuerte y por ser una gran amiga. A mi papá, por darme su apoyo, por ser un ejemplo de hombre trabajador y enseñarme a luchar por lo que se quiere. A mi padrastro Felo, por su generosidad y sus grandes valores. A mi hermana Yuleidys, por ser como una madre para mí y enseñarme a ser una mejor persona cada día, y por darme unos lindos sobrinos que tanto quiero y amo. A mi tía Baude, por siempre estar pendiente de mí y dándome ánimos para seguir adelante, y todo su cariño. A mis abuelitas, por entregarme todas sus experiencias y virtudes. A mis primos, Herminio, Víctor y Norita, por darme aliento y fuerzas, los quiero.

A mis viejos amigos, Miguel, Rolando, Yoyi, Rafelito y Nelo; por estar siempre a mi lado cuando más los necesito, por su gran amistad, por su disposición de ayudar y por contribuir en mi formación. A mis mejores amigos, Freddy y Héctor. Por ser los mejores jóvenes que conozco, por ser ejemplos a seguir, por su comprensión, amabilidad, fortaleza y por demostrar ser verdaderos amigos. Mis amigos de la UCI: Herbert, Nilberto, Daniel, Roberto y José. Por compartir cada granito de arroz conmigo y por ser muy buenas personas. A los muchachos y muchachas de mi aula, que me han ayudado a seguir adelante, especialmente a Yoinel, Yenni y Lisandra.

A mi novia Ady, por estar cerca de mí en los momentos estresantes y por compartir su cariño conmigo. A los tutores, Marilys y Luis por su paciencia y por su gran ayuda en los momentos que más necesité de ellos.

De todo corazón muchas gracias a todos.

Dedicatoria

Dedico el presente trabajo a mis padres, a mi padrastro, mi tía que tanto empeño han puesto para ver este resultado y todo su esfuerzo en darme un futuro mejor; especialmente a mi mamá por ser todo para mí.

A mis abuelitas, sobrinos, primos y a mi hermana por ser mi gran compañera de la vida.

A todos mis amigos que tanto me han apoyado y a mi novia por su cariño.

En general, a todos mis familiares y amigos.

Resumen

El presente trabajo tiene como objetivo desarrollar una personalización del sistema de indexación de metadatos Open Harvester Systems (OHS) para mejorar la representación de la información científico-académica, a través de la integración de nuevas funcionalidades. La personalización tiene como propósito mejorar la interacción de los usuarios con el sistema y la realización de búsquedas de información científico-académica.

Es una herramienta basada en el protocolo de recolección de metadatos de la iniciativa abierta de archivos (OAI-PMH); el cual permite realizar búsquedas conjuntas sobre los metadatos de los repositorios asociados al sistema. Además, este sistema permite la posibilidad de indexar, recolectar y redistribuir metadatos, así como su cosecha en distintos esquemas.

Se realizó una investigación de todos aquellos requisitos que se deben tener en cuenta para lograr la implementación de la personalización del OHS. Se utilizaron tecnologías y herramientas como: Visual Paradigm, MySQL y JavaScript.

El sistema está concebido para funcionar en un ambiente de integración de sistemas informáticos, brindando y consumiendo Servicios Web.

Palabras clave:

Personalización, metadatos, interacción, indexación.

Índice de contenidos

Introducción	1
Capítulo #1 Estado actual de las tecnologías para la recuperación y presentación de información científico-académica.....	5
1.1 Recuperación de la información científico-académica.	5
1.1.1 Herramientas para la creación de repositorios científicos.	5
1.1.1.1 Green Stone	6
1.1.1.2 E-Print	6
1.1.1.3 Dspace.....	7
1.1.2 Herramientas para la recuperación y presentación de la información científico-académica.	8
1.1.2.1 DOAJ - Directory of Open Access Journals (Directorio de Revistas de Acceso Abierto)	8
1.1.2.2 SciELO - Scientific Electronic Library Online (Biblioteca Científica Electrónica en Línea)	9
1.1.2.3 OHS	9
1.1.2.4 Open Journal Systems (OJS).....	11
1.2 Sistemas de presentación de la información científico-académica.	11
1.2.1. Nacional	11
1.2.2. Revistas nacionales:	12
1.2.2.1 Revista Cubana de Ciencias Informáticas	12
1.2.2.2 La Serie Científica de la UCI	12
1.2.3. Internacional	13
1.2.4. Revistas internacionales:	13
1.2.4.1 Medicina Abierta	13
1.3 Protocolos para la transmisión de metadatos.	13
1.3.1 Protocolo de Internet (IP).....	13
1.3.2 Protocolo TCP/IP.....	14
1.3.3 Protocolo HTTP.....	14
1.3.4 Protocolo z39.50.....	14
1.3.4 Protocolo OAI-PMH.....	15
1.4 Formatos de representación de metadatos.	16
1.4.1 Dublin Core.....	17
1.4.2 MODS.....	17
1.5 Tecnologías y Herramientas utilizadas en el desarrollo de aplicaciones web.	18
1.5.1 HTML.....	18
1.5.2 XML.....	18
1.5.3 CSS	19
1.5.4 JavaScript.....	19

Índice de contenidos

1.5.5 PHP	19
1.5.6 Sistema Gestor de Base de Datos	20
1.6.6.1 MySQL	20
1.5.7 NetBeans IDE	21
1.5.8 Servidor web	21
1.5.8.1 Apache	21
1.5.9 Lenguaje de Modelado	21
1.5.10 Herramienta CASE	22
1.5.10.1 Visual Paradigm	22
1.5.11 Metodología de desarrollo	22
1.5.11.1 Proceso de Mejora CMMI nivel II	23
Capítulo #2 Características del sistema.....	25
2.1 Propuesta de Solución.....	25
2.2 Modelo de dominio	25
2.2.1 Diagrama de modelo de dominio	26
2.2.2 Descripciones de las clases representadas en el modelo de dominio.....	26
2.3 Levantamiento de requisitos.....	26
2.3.1 Requisitos funcionales.....	26
2.3.2 Requisitos no funcionales.....	27
2.4 Plan de Entrega.....	28
2.5 Historias de usuario.....	29
Capítulo #3 Construcción de la Solución Propuesta.	39
3.1 Crear un Block Plugin (Plugin Bloque)	39
3.2 Arquitectura del OHS	41
3.3 Patrones GRASP - Patrones de Principios Generales para Asignar Responsabilidades	43
3.4 Diseño del sistema	44
3.5 Diagrama de Clases Persistentes.....	45
3.6 Modelo de datos	45
3.7 Diagrama de Componentes.....	48
3.8 Diagrama de Despliegue.....	50
3.8.1 Descripción de los nodos.....	50
3.9 Pruebas.....	51
3.9.1 Pruebas de Caja negra.....	51
3.9.2 Pruebas de Caja blanca.....	56

Índice de contenidos

Conclusiones.....	61
Recomendaciones.....	62
Bibliografía consultada.....	63
Bibliografía referenciada.....	66
Glosario de términos.....	68
Anexos	69

Índice de ilustraciones

Ilustración 1: Diagrama de dominio.	26
Ilustración 2: Modelo Vista Controlador.	42
Ilustración 3: Arquitectura del OHS.	42
Ilustración 4: Diagrama de paquetes.	45
Ilustración 5: Diagrama de clases persistentes.	45
Ilustración 6: Modelo de datos.	46
Ilustración 7: Diagrama de componentes.	49
Ilustración 8: Diagrama de despliegue.	50
Ilustración 9: Resultados.	56
Ilustración 10: Código de la funcionalidad enviarColeccion.	57
Ilustración 11: Grafo del flujo asociado a la funcionalidad enviarColeccion.	58

Índice de tablas

Tabla 1: Plan de entrega.	29
Tabla 2: HU Modificar idioma del sistema.	30
Tabla 3: HU Mostrar colecciones personales.	31
Tabla 4: HU Eliminar colección personal.	32
Tabla 5: HU Cambiar nombre a colección personal.	33
Tabla 6: HU Mostrar registros de colección personal.	34
Tabla 7: HU Ver registro de colección personal.	35
Tabla 8: Ver registro original de colección personal.	36
Tabla 9: HU Realizar integración a gestores bibliográficos.	37
Tabla 10: Descripción de la tabla user del modelo de datos.	46
Tabla 11: Descripción de la tabla user_collection del modelo de datos.	47
Tabla 12: Descripción de la tabla registro del modelo de datos.	48
Tabla 13: Lista de chequeo de requisitos funcionales.	51
Tabla 14: HU1_CP1 Modificar idioma del sistema.	52
Tabla 15: HU5_CP5 Mostrar colecciones personales.	52
Tabla 16: HU6_CP6 Eliminar colección personal.	52
Tabla 17: HU7_CP7 Cambiar nombre a colección personal.	52
Tabla 18HU9_CP9 Mostrar registros de colección personal.	54
Tabla 19: HU10_CP10 Ver registro de colección personal.	54
Tabla 20: HU11_CP11 Mostrar registros de colección personal.	54
Tabla 21: HU13_CP13 Realizar integración a gestores bibliográficos.	55
Tabla 22: No conformidades encontradas y resueltas por iteración.	56
Tabla 23: Caminos básicos de enviarColeccion.	58
Tabla 24: Caso de prueba del camino #1	59
Tabla 25: Caso de prueba del camino #2	59
Tabla 26: Caso de prueba del camino #3	59

Introducción

La información ha ido ocupando un espacio predominante en la sociedad hasta llegar a ser vital en la actividad diaria del hombre. Su uso y generación ha crecido a un ritmo acelerado, dando paso a la denominada “explosión digital”, originada por la rápida evolución de la tecnología y el crecimiento de Internet con sus múltiples posibilidades de acceso y socialización de la información (GARCÍA y CABALLERO, 2010).

En el contexto global y cambiante que ofrece Internet, los espacios de registro y almacenamiento de la información científico-académica, como las bibliotecas digitales y los repositorios institucionales, presentan el desafío de integrar herramientas a los objetos digitales, como los esquemas de metadatos, con el objetivo de hacerlos visibles y dinámicos.

A partir de los avances de la tecnología digital, surge la Iniciativa de Archivos Abiertos (OAI) por sus siglas en inglés (Open Archives Initiative), como una alternativa para promover la interoperabilidad de los sistemas de información, con la finalidad de colocar al alcance de toda la sociedad los contenidos científicos-académicos, que son gestionados por repositorios y bases de datos referenciadas de acceso abierto.

Enmarcado en este paradigma, Cuba ya cuenta con referentes que han alcanzado logros admirables dando impulso al desarrollo estandarizado de las bibliotecas digitales y los repositorios institucionales. La visibilidad de los recursos preservados se va incrementando gradualmente porque crece también su capacidad de interoperar con otros sistemas de información digital. Muchas de las instituciones han incursionado en el desarrollo de sistemas con este fin, tal es el caso de la Universidad de las Ciencias Informáticas (UCI), que como centro de altos estudios, cuenta con una biblioteca digital y con un repositorio institucional de información. Esto ha generado oportunidades de ser protagonistas de la organización documental, creando y adaptando estos sistemas para un tratamiento más eficiente de la información en el contexto digital.

Al Centro de Informatización Universitaria (CENIA) de la UCI, se le ha encomendado como tarea realizar una personalización del sistema recolector o cosechador de metadatos OHS, debido a que no cuenta con una buena representación de la información, su diseño dificulta el uso por parte de los usuarios menos experimentados. Otra de las necesidades de personalizar el sistema, es que no facilita la gestión de colecciones personales, donde los usuarios puedan tener

un espacio dentro del sistema que les garantice poder guardar la información de los elementos consultados. Este sistema no tiene una integración con redes sociales, aspecto fundamental para conocer e intercambiar ideas con otras personas en el mundo. El OHS tampoco cuenta con la integración a gestores de referencias bibliográficas, los cuales facilitan el almacenamiento, organización y control de las referencias bibliográficas, permiten la inserción automática de citas y la aplicación de diferentes formatos bibliográficos para generar bibliografías. Además, no permite el cambio de idiomas.

A partir de lo anteriormente planteado, se define el siguiente **problema de la investigación**: ¿Cómo mejorar la interacción del usuario con el sistema de recuperación de la información científico-académica OHS?

El diseño de la investigación permite identificar como **objeto de estudio**: La representación de la información en los sistemas de recuperación de la información. Definiéndose el **campo de acción**: La representación de la información en el sistema de recuperación OHS.

En el contexto investigativo, se determina como **objetivo general de la investigación**: Personalizar el sistema de indexación de metadatos OHS para mejorar la representación de la información científico-académica, a través de la integración de nuevas funcionalidades.

Para dar cumplimiento al objetivo general, se han desglosado los siguientes **objetivos específicos**:

- Describir los fundamentos teóricos sobre las tendencias actuales de los sistemas de recuperación y presentación de la información científico-académica.
- Definir una propuesta de solución para la personalización del OHS.
- Describir las funcionalidades que tendrá la personalización del OHS.
- Diseñar la personalización del OHS.
- Implementar y probar la personalización del OHS.

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

- Documentación de los procesos de recuperación y representación de la información científico-académica.

- Descripción de los principales procesos del OHS.
- Obtención de la lista de reserva del producto.
- Descripción del proceso de personalización del OHS.
- Confección del modelo de dominio.
- Realización del modelo de diseño según la metodología de desarrollo de software seleccionada.
- Elaboración del modelo de datos.
- Realización del diagrama de despliegue.
- Elaboración del diagrama de componentes.
- Implementación de nuevas funcionales para el OHS.
- Realización de los casos de pruebas.

Para la realización de este trabajo se utilizaron diversos métodos de investigación, tanto **teóricos** como **empíricos**, los cuales se exponen a continuación:

Métodos teóricos:

Analítico - sintético: con la utilización del método se han buscado, investigado y analizado documentos para extraer los elementos de información más importantes que pudieran estar relacionados con el objeto de estudio; realizando consultas a diversas fuentes bibliográficas.

Histórico - lógico: su uso permitió investigar la evolución y desarrollo del OHS. Esencia del problema y la necesidad de mejorar la presentación de la información del sistema.

Métodos empíricos:

Observación: se utilizó con el objetivo de observar cómo funciona el OHS, se analizaron cuáles son sus principales desventajas y funcionalidades implementadas.

Posibles resultados:

La personalización del OHS garantizará una mejor representación de la información, con una serie de funcionalidades que le proporcionarán a los usuarios tener un espacio dentro del sistema, así como el almacenamiento y recuperación de la información de diversas fuentes,

además permitirá interactuar con gestores bibliográficos. El sistema prestará servicios de la Web 2.0, permitiéndoles a los usuarios guardar elementos consultados y otras clasificaciones.

Estructura de los capítulos

El contenido del trabajo se distribuirá en 3 capítulos, las conclusiones generales, recomendaciones, referencias bibliográficas, bibliografía empleada, los anexos y un glosario de términos, como parte del trabajo para facilitar la comprensión del mismo.

Capítulo I. Estado actual de las tecnologías para la recuperación y presentación de información científico-académica.

Tiene como objetivo establecer las bases teóricas generales que son fundamentales para la inicialización del trabajo. Recoge conceptos y características que se hacen necesarios dominar, así como una descripción de las tecnologías a emplear para su desarrollo. Se define el lenguaje de programación a utilizar, así como el Sistema Gestor de Base de Datos (SGBD). Por último, se realiza un análisis detallado del estado del arte relacionado con los sistemas de indexación de metadatos, tanto en el ámbito nacional como internacional.

Capítulo II. Características del Sistema: Brinda la modelación detallada y la construcción de la estructura de la aplicación para garantizar una mayor comprensión del problema existente. Se modela el problema a través de un modelo de dominio. Se obtiene la lista de reserva del producto y se describen las historias de usuarios.

Capítulo III. Construcción de la Solución Propuesta: Tiene como objetivo detallar todos los elementos presentes en el diseño como son: el modelo de diseño, se realiza una breve descripción de los patrones de diseño utilizados, así como el diagrama de clases persistentes y el modelo de datos. Por último, se representará el modelo de implementación a través de los diagramas de componentes y de despliegue.

Capítulo #1 Estado actual de las tecnologías para la recuperación y presentación de información científico-académica.

En el presente capítulo se realiza una breve descripción del proceso de recuperación de información científico-académica, a través de conceptos. Se realiza un estudio del estado actual de las tecnologías para la recuperación y presentación de información científico-académica para lograr arribar a una sólida conclusión, con el fin de obtener amplios conocimientos de los sistemas de presentación y recuperación de la información científico-académica, de forma tal, que permita posteriormente la selección del sistema para proceder a la personalización. Se hace una breve descripción de las tecnologías empleadas para el desarrollo de la personalización del sistema, lenguaje de programación, herramientas de apoyo y el sistema gestor de base de datos a emplear.

1.1 Recuperación de la información científico-académica.

La información es un conjunto de datos acerca de algún suceso, hecho, fenómeno o situación, que organizados en un contexto determinado tienen su significado, cuyo propósito puede ser el de reducir o incrementar el conocimiento acerca de algún avance científico-técnico.

Según algunos investigadores, definen la Recuperación de la Información (RI) como:

“el conjunto de tareas mediante las cuales el usuario localiza y accede a los recursos de información que son pertinentes para la resolución del problema planteado. En estas tareas desempeñan un papel fundamental los lenguajes documentales, las técnicas de resumen y la descripción del objeto documental” (CROFT, 1987).

“Encontrar y clasificar documentos relevantes que satisfagan la necesidad de información de un usuario, expresada en un determinado lenguaje de consulta” (TOLOSA y BORDIGNON, 2008).

A partir de los conceptos antes expuestos la RI es el proceso de comunicación que hay entre los usuarios y los sistemas, en el que mediante un término se logra recuperar un conjunto de elementos evaluados por los interesados en la búsqueda, con el fin de lograr satisfacer las necesidades de información.

1.1.1 Herramientas para la creación de repositorios científicos.

1.1.1.1 Green Stone

Es una aplicación para la construcción y administración de bibliotecas digitales accesible desde la web, proporciona el acceso a documentos textuales como: imágenes, vídeos y sonidos. Su organización permite distribuir los documentos en colecciones que se pueden crear atendiendo a las características que posee. Por otra parte, favorece la importación y asignación de metadatos, de manera manual o automática. Permite configurar el tipo de indexación a utilizar, modalidades de búsqueda y obtener listados de resultados en función de la relevancia con los términos consultados (CYRANEK, 2010).

La descripción de los documentos se puede realizar de manera individual, documento a documento, o describir las distintas agrupaciones en conjunto. De esta última manera, se le asigna a todos los recursos incluidos en cada agrupación la misma descripción de manera automática, desarrollando así un práctico sistema multinivel (SOLA, 2008).

Green Stone soporta la utilización de esquemas de metadatos. El usuario es quien define el esquema a utilizar a través de la interfaz de Bibliotecario. Estos conjuntos de metadatos ya vienen predefinidos y pueden seleccionarse para ser utilizados en las colecciones: Dublin Core (esquema básico), NZGLS (New Zealand Government Locator Service) y AGLS (Australian Government Locator Service) (CYRANEK, 2010).

Ventajas de Green Stone:

- Integración de todos los procesos necesarios para gestionar miles o millones de recursos digitales.
- Potente búsqueda a texto completo y múltiples índices de búsqueda basados en metadatos.
- Posibilidad de compartir experiencias sin restricciones dentro de la comunidad internacional dado su carácter de software de fuente abierta (ARIAS, 2008).
- Posibilidad de importar directamente bases de datos CDS/ISIS (Green Stone, E-Print, Dspace).

1.1.1.2 E-Print

Es un software muy usado para la gestión de repositorios, se distribuye bajo la licencia GNU, lo cual significa que el código fuente es accesible y modificable por cualquier programador, con la condición que las modificaciones se hagan también accesibles públicamente.

Provee posibilidades de organizar los documentos del repositorio en colecciones. Este sistema puede almacenar cualquier tipo de ficheros que el administrador decida aceptar, incluyendo HTML (por sus siglas en inglés Hypertext Markup Language, Lenguaje de Marca de Hipertexto). Cada documento puede ser almacenado en distintos formatos, no impone límite alguno sobre el tamaño de los archivos que se pueden subir, determinado únicamente por el sistema de archivos subyacentes o el servidor web (BARRUECO, 2010).

Un repositorio creado con esta herramienta, ofrece un conjunto de servicios a todos los miembros de una institución, proporcionando acceso abierto a los datos. Asegura la preservación, enseñanza y colaboración de las publicaciones e investigaciones. Permite subir actualizaciones de un documento con el mismo formato de metadatos y se crea un árbol de versiones del documento posibilitando el acceso a cualquiera de ellos.

Ventajas que facilita el E-Print:

- Posibilita ingresar nuevos módulos que se desarrollen.
- La interfaz es modificable a las características de la Unidad de Información.
- El proceso de descripción de un libro en el sistema es muy fácil e intuitivo de usar.
- Posibilita indicar los tipos de documentos que podrá cargar el sistema.
- Realiza búsquedas a partir de las realizadas.
- Permite limpiar el formulario para hacer nuevas búsquedas.

1.1.1.3 Dspace

Software diseñado para crear y distribuir colecciones digitales, proporcionando así una nueva forma de organizar y publicar la información a través de Internet. Es un sistema de biblioteca digital diseñado para capturar, almacenar y publicar información en formatos digitales.

Provee una forma de gestionar materiales y publicaciones resultantes, tanto de la actividad de investigación como de educación, almacenadas en un repositorio que ofrece a los usuarios accesibilidad en todo momento. Creado en colaboración entre la empresa Hewlett-Packard y las

bibliotecas del Instituto de Tecnología de Massachusetts, el programa en sus inicios satisfizo las necesidades de estos centros de información, pero luego se colocó a disposición de la comunidad internacional como una herramienta de código abierto y gratuito.

Soporta todo tipo de formatos de documentos, recomienda a los administradores de cada sistema decidir qué formatos aceptar, diferenciando tres categorías de formatos: soportados, conocidos y no soportados. Provee por defecto un esquema de metadatos Dublin Core. Permite configurar múltiples esquemas y seleccionar campos de metadatos de una combinación de esquemas configurados para describir un ítem. Cualquier colección de DSpace puede ser importada en Green Stone (MENDOZA, 2009).

Ventajas de DSpace:

- Multiplataforma.
- Basado en tecnología Web.
- Emplea estándares como Dublin Core.
- Código abierto, bajo la Licencia BSD (Berkeley Software Distribution).
- Sistema adaptable.

1.1.2 Herramientas para la recuperación y presentación de la información científico-académica.

Los Sistemas de Recuperación de Información (SRI) son herramientas que posibilitan localizar la información digital presente en internet o en una subred determinada. Un SRI permite la RI, previamente almacenada, por medio de la realización de una serie de consultas a los documentos contenidos en la base de datos.

Para la presentación de la información, debe distinguirse entre dato e información. Dato es cualquier número, imagen, sonido, símbolo, nombre que no permite tomar ninguna decisión, porque no tienen un significado concreto, los datos deben procesarse para dotarlos de significado y convertirlos en información. Eso es lo que ocurre cuando trabajamos con un sistema informático, los resultados que nos muestra el ordenador son para nosotros información que nos ayuda en alguna tarea.

1.1.2.1 DOAJ - Directory of Open Access Journals (Directorio de Revistas de Acceso Abierto)

Directorio multitemático que provee acceso a revistas científicas y académicas de acceso abierto, que se someten a un específico y exigente sistema de calidad. Proporciona búsquedas por temáticas, por títulos de revistas y artículos dentro del Directorio. El objetivo del Directorio tiene como objetivo incrementar la visibilidad y la facilidad de uso de las revistas científicas y académicas de acceso abierto, para fomentar el aumento de su uso e impacto pretendiendo ser global y abarcar todas las revistas científicas y académicas de acceso abierto (DOAJ, 2007).

1.1.2.2 SciELO - Scientific Electronic Library Online (Biblioteca Científica Electrónica en Línea)

Es un modelo para la publicación electrónica cooperativa de revistas científicas en Internet. Especialmente desarrollado para responder a las necesidades de la comunicación científica en los países en desarrollo y particularmente de América Latina y el Caribe, el modelo proporciona una solución eficiente para asegurar la visibilidad y el acceso universal a su literatura científica.

1.1.2.3 OHS

Desarrollado y mantenido por Public Knowledge Project (PKP), es un software gratuito de código abierto, implementado en Hypertext Preprocessor (PHP) y distribuido bajo la licencia GPL (General Public Licence), que ofrece la posibilidad de indexación, recolección y redistribución de metadatos. Su interfaz flexible, permite la búsqueda simple y avanzada, utilizando campos crosswalk¹ de todos los archivos cosechados. Al crear campos pasos peatonales para la búsqueda, los administradores pueden definir los elementos de tipo texto, fecha o HTML (WILLINSKY, 1998).

Para buscar material almacenado en OHS, depende del tipo de plugin² de búsqueda utilizado por el sistema. Hay dos opciones de búsquedas a disposición de los usuarios:

- Simple: en todo el material indexado, disponible en la barra lateral izquierda (para buscar, simplemente hay que introducir los criterios de búsqueda en el campo y presionar el botón de búsqueda).

¹ Pasos peatonales, permiten búsquedas con el objetivo de abarcar múltiples formatos de metadatos.

² Complemento.

- Avanzada: disponible en la barra de navegación superior. Al hacer clic en la opción Buscar llevará al usuario a la página de búsqueda avanzada, donde múltiples tipos de campos están disponibles. Los cuales son gestionados por el administrador del sitio.

Cuenta con herramientas de lectura, las cuales están diseñadas para ayudar a los lectores experimentados y novatos, proporcionando un buen contexto de los materiales relacionados, desde una amplia variedad de fuentes de acceso abierto. Además, utilizan palabras claves para buscar automáticamente en la base de datos.

Al administrador le brinda la posibilidad de poder configurar el sistema, añadir, eliminar, administrar archivos, configurar los pasos peatonales, criterios de ordenación e introducción de la información básica sobre el sitio, incluyendo la descripción e información de contacto.

El administrador del sitio controla la visualización del diseño de los elementos de la barra lateral, también conocida como bloques. A nivel de bloques, los componentes de interfaz de usuario, como plugins instalados y administración desde la página de plugins, pueden moverse de izquierda a derecha, desaparecer a través de la página de diseño. Además, puede ver y configurar los plugins que se encuentran en varias categorías generales de plugins como: genéricos, bloques, harvester (cosechador), preprocesador, esquemas y temas.

El OHS posibilita la creación de interfaces con CSS (por sus siglas en inglés Cascading Style Sheets, Hojas de Estilo en Cascada) y basado en plantillas HTML. Posee una base de datos relacional como medio de almacenamiento de metadatos. Brinda soporte para los manejadores de bases de datos MySQL y PostgreSQL. Posibilita cosechar metadatos con OAI (Iniciativa Abierta de Archivos) en distintos esquemas (Dublin Core, MODS (por sus siglas en inglés Metadata Object Description Schema, Esquema para la Descripción de Objetos de Metadatos) o MARC (por sus siglas en inglés Machine Readable Cataloging, Catalogación Legible por Máquina). Posee un motor de búsqueda que permite realizar búsquedas avanzadas sobre los metadatos recolectados. Garantiza la recolección granular (sólo partes específicas del repositorio), también es un proveedor de metadatos (WILLINSKY, 1998).

Gestión de permisos en el OHS:

Se realiza a través de la asignación de roles, donde el administrador del sistema tiene acceso a todas las funcionalidades, además se encarga de asignar un rol específico a cada usuario. A los

usuarios se les puede asignar el rol de administrador de archivos, que es el encargado de crear archivos en el sistema, o ningún otro rol. El administrador del sistema se encargará de dar los permisos y el acceso a las funcionalidades del sistema a los usuarios, según el rol asignado.

El OHS no brinda servicios como: la integración con redes sociales, gestores de referencias bibliográficas, espacio personal para coleccionar registros de interés del usuario y no permiten el cambio de idiomas.

1.1.2.4 Open Journal Systems (OJS)

Es un sistema de gestión y publicación de revistas que ha sido desarrollado por PKP. OJS ayuda con todas las etapas del proceso de publicaciones, a través de los sistemas de gestión, buscando mejorar la calidad académica y pública de la investigación. Es un software de código abierto, de libre acceso a las revistas de todo el mundo (PKP, 1998).

Estos sistemas no tienen una representación de la información adecuada. No brinda la posibilidad de compartir documentos en las redes sociales, ya que no poseen integración con las mismas. Además, que no cuenta con espacios personales donde los usuarios puedan guardar sus elementos de información (documentos). No posibilita la integración con gestores de referencias bibliográficas como: Bibtex, lo cual impide a los usuarios poder exportar sus registros.

1.2 Sistemas de presentación de la información científico-académica.

Con el avance de las Tecnologías de la Información y las Comunicaciones (TIC), se han generado grandes volúmenes de información en las comunidades científicas. La diversidad de formatos portadores de contenidos valiosos, así como las dificultades para el almacenamiento y recuperación de la información; llevaron a que investigadores, especialistas, técnicos, en el ámbito de la computación y las telecomunicaciones representaran en el medio electrónico los sistemas de almacenamiento y recuperación de metadatos, con el propósito de satisfacer las necesidades de información de los usuarios. A raíz de estas necesidades, surgió la creación de múltiples sistemas que se utilizan en las diferentes esferas de la sociedad.

1.2.1. Nacional

A nivel nacional, los sistemas de información en su mayoría usan OJS. El desarrollo de sistemas de información en la UCI se ha incrementado cada día; la misma cuenta con el acceso desde la biblioteca digital a un Repositorio Institucional para la realización de búsquedas de información, que de cierta manera, es un recolector de información, montado sobre Dspace. A continuación se muestran algunos ejemplos de revistas:

1.2.2. Revistas nacionales:

- Revista de Ingeniería Energética.
- Revista Cubana de Ciencias Informáticas.
- Revista de Ingeniería Electrónica, Automática y Comunicaciones.
- La Serie Científica de la UCI (SC-UCI).
- Revista Cubana de Ingeniería.

1.2.2.1 Revista Cubana de Ciencias Informáticas

La Revista Cubana de Ciencias Informáticas (RCCI), cuenta con un sistema de arbitraje por pares académicos, especializada en Ciencias de la Computación e Informática y usa OJS v.2.3.8.0, permite las búsquedas de los contenidos de la revista a través de todas las categorías o por una categoría específica como: palabras claves, título del trabajo, autor, etc. Posibilita registrarse, hacer envíos en línea; además posee notificaciones de inscripción y un mapa del sitio para una mejor guía del trabajo de los usuarios. Permite la navegación en el sitio por: número, autor y título; brinda la posibilidad de información a los lectores, autores y bibliotecarios (RCCI, 2013).

1.2.2.2 La Serie Científica de la UCI

Revista de publicación periódica, diseñada para divulgar y promover los resultados de la investigación desarrollados en la UCI. Se publica mensualmente de forma electrónica y todos los procesos editoriales se gestionan sobre la plataforma OJS v. 2.3.8.0, permite que los contenidos estén libremente disponibles mediante autenticación desde la Intranet universitaria, facultades regionales y centros de desarrollo. Los usuarios pueden realizar búsquedas de los contenidos de la revista a través de todas las categorías o por una categoría específica como: palabras claves, título del trabajo, autor, etc. Muestra una página de notificaciones con las actualizaciones más

importantes asociadas a esta revista, como nuevos números o anuncios como guía para el usuario. La revista cuenta con un mapa del sitio (SERIE_CIENTÍFICA, 2013).

1.2.3. Internacional

En cuanto al acontecer internacional se reflejan ejemplos de revistas, las cuales facilitan el acceso y disponibilidad de la información. Con el fin de obtener más conocimientos acerca de los sistemas que almacenan y recuperan información.

1.2.4. Revistas internacionales:

- Open Medicine (Medicina Abierta).
- Portal de Portales Latindex.
- The Australasian Journal of Information Systems (AJIS).
- Health and Human Rights (Salud y Derechos Humanos).

1.2.4.1 Medicina Abierta

Es una revista independiente, de libre acceso, usa OJS v.2.2.3.0. Facilita la difusión equitativa y global de la investigación de la salud de alta calidad, promover información del diálogo y la colaboración internacional en temas de salud, tanto en Canadá como a nivel mundial. Los usuarios pueden acceder a dicha revista a través de Internet y ordenadores, permitiéndoles acceder a un registro de inscripción (KENDALL y PALEPU, 2012).

1.3 Protocolos para la transmisión de metadatos.

Los protocolos son un conjunto de procedimientos utilizados en las relaciones interpersonales. Cuando es usado bajo el contexto de redes de comunicación, el término protocolo tiene un significado similar pero a un nivel más específico. Un protocolo de red es un conjunto de reglas, secuencias, formatos de mensajes y procedimientos bien detallados que posibilita la transferencia de datos entre dos o más sistemas de computación, es el término utilizado para describir cómo los sistemas de computación se comunican con otros a nivel de bit y de byte (CARRERAS y ÁLVAREZ, 2012).

1.3.1 Protocolo de Internet (IP)

El sistema de indexación de metadatos OHS, el cual emplea protocolos de Internet como: Protocolo de Control de Transmisión (TCP), IP y Protocolo de Transferencia de Hipertexto (HTTP) que garantizarán la transmisión de información.

1.3.2 Protocolo TCP/IP

Conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. Se le denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: TCP e IP, que fueron los dos primeros en definirse, y que son los más utilizados de la familia, proporcionan la base para el Internet global, una gran red de redes operacional que interconecta universidades, corporaciones y dependencias gubernamentales en muchos países alrededor del mundo (*Protocolo TCP/IP.*, 2010).

El conjunto TCP/IP está diseñado para enviar paquetes de información y tiene un grado muy elevado de fiabilidad, es adecuado para redes grandes y medianas como: una gran red de redes operacional que interconecta universidades, corporaciones y dependencias gubernamentales en muchos países alrededor del mundo, siendo la base para el Internet global. Es compatible con las herramientas estándar para analizar el funcionamiento de la red.

1.3.3 Protocolo HTTP

Protocolo usado en cada transacción de la World Wide Web (www). Mediante HTTP los clientes y los servidores determinan de forma dinámica el formato de los documentos, lo cual permite que utilicen formato de datos no estándar para el intercambio de datos. Si el receptor no tiene un modo de ver o acceder a los datos puede descargar un complemento que le permita recibir el contenido.

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la siguiente forma: un proceso servidor escucha en un puerto de comunicaciones TCP, por defecto el 80, y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores (*Protocolo HTTP*, 2010).

1.3.4 Protocolo z39.50

Protocolo de recuperación de información basado en la estructura cliente-servidor que facilita la interconexión de sistemas informáticos. Le permite al usuario realizar búsquedas en bases de datos que cuenten con un servidor Z39.50, sin tener que conocer para ello las sintaxis de búsqueda que utilizan dichos sistemas.

Especifica un conjunto de reglas para gestionar las formas y procedimientos de interconexión remota de computadoras, con el propósito de buscar y recuperar información, aunque su aplicación actual es más amplia pues incluye la consulta y el intercambio de datos bibliográficos, la intercomunicación de índices y resúmenes de: información geoespacial, documentos oficiales, objetos digitales o de metadatos que describen los documentos de las bibliotecas electrónicas y digitales (SALINAS, 2010).

1.3.4 Protocolo OAI-PMH

El Protocolo de recolección de metadatos de la Iniciativa Abierta de Archivos (OAI-PMH) es una herramienta de interoperabilidad que posibilita el intercambio de metadatos sobre cualquier material almacenado en soporte electrónico. Básicamente utiliza transacciones HTTP para emitir preguntas y obtener respuestas entre un servidor o archivo y un cliente o servicio recolector de metadatos. El segundo puede pedir al primero que le envíe metadatos, según determinados criterios como la fecha de creación de los datos. En respuesta, el primero devuelve un conjunto de registros en formato XML (por sus siglas en inglés Extensible Markup Language, Lenguaje de Marcas Extensible), incluyendo identificadores de los objetos descritos en cada registro.

Las peticiones se emiten utilizando los métodos GET o POST del protocolo HTTP y constan de una lista de opciones con la forma de pares del tipo: clave=valor. Existen seis peticiones que un cliente puede realizar a un servidor:

- **GetRecord:** Utilizado para recuperar un registro concreto. Necesita dos argumentos, identificador del registro pedido y especificación del formato bibliográfico en que se debe devolver.
- **Identify:** Utilizado para recuperar información sobre el servidor: nombre, versión del protocolo que utiliza y dirección del administrador.

- **ListIdentifiers:** Recupera los encabezamientos de los registros, en lugar de los registros completos. Permite argumentos como el rango de fechas entre los que queremos recuperar los datos.
- **ListRecords:** Igual que el anterior pero recupera los registros completos.
- **ListSets:** Recupera un conjunto de registros. Estos conjuntos son creados opcionalmente por el servidor para facilitar una recuperación selectiva de los registros.
- **ListMetadataFormats:** Devuelve la lista de formatos bibliográficos que utiliza el servidor.

Soporta múltiples formatos para expresar los metadatos, no obstante requiere que todos los servidores ofrezcan los registros utilizando Dublin Core codificado en XML. Además de este formato, cada servidor es libre de ofrecer los registros en otros formatos adicionales como MARC. Un cliente puede pedir que los registros se le sirvan en cualquiera de los formatos soportados por el servidor.

La arquitectura de este protocolo se basa en proveedores de datos y proveedores de servicios; los primeros son los archivos que proporcionan la información y los segundos son los recolectores o servicios que toman los datos, y los presentan a los usuarios finales.

La aplicación de OAI-PMH permite realizar el intercambio de información para que desde puntos centralizados, proveedores de servicios puedan realizar búsquedas conjuntas sobre los metadatos de todos los repositorios asociados, específicamente los Accesos Abierto (Open Access) que promueve eliminar las barreras económicas, legales y tecnológicas, y trata de obtener a cambio, como beneficios, una mayor accesibilidad para los documentos y una mayor visibilidad para los autores. Los documentos que están disponibles en Acceso Abierto son más consultados y tienen más posibilidades de ser citados (BARRUECO, 2010).

1.4 Formatos de representación de metadatos.

Como resultado del aumento de grandes cantidades de datos digitales disponibles en la red aparece la sobrecarga de información, situación por la cual se comienzan a usar los metadatos para facilitar la recuperación y catalogación de la misma. Existen diferentes definiciones del término metadato, comúnmente son llamados datos sobre datos, otros conceptos se muestran a continuación:

Los metadatos son datos que se pueden guardar, intercambiar y procesar por medio del ordenador y que están estructurados de forma tal que permiten ayudar a la identificación, descripción, clasificación y localización del contenido de un documento o recurso web (LAPUENTE, 2011).

El término de metadatos, se refiere a aquellos datos que describen las características de un recurso y son sumamente importantes cuando se quiere tener un conocimiento avanzado y total de las características de los recursos (DEMPSEY y HEERY, 1997) .

De acuerdo a las definiciones planteadas anteriormente, se puede definir metadatos como: datos que son utilizados para estructurar información de tal manera que permitan mostrar las características y todos los datos que la integran.

1.4.1 Dublin Core

Es un modelo de metadatos elaborado y auspiciado por Dublin Core Metadata Initiative (DCMI), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos y a promover el desarrollo de los vocabularios especializados de metadatos. Está diseñado para facilitar la recuperación de recursos en las redes de una forma similar a un catálogo de biblioteca, pero con una estructura mucho más sencilla. Está formado por 15 elementos de datos y su punto más fuerte es que el diseño es tan intuitivo que los propios proveedores de información pueden codificar los documentos al mismo tiempo que los crean (REPISO, 1999).

Permite describir los elementos básicos de identificación de cualquier tipo de documento. Tiene como objetivo principal crear un conjunto de elementos de datos que describan los documentos electrónicos de las redes con el fin de facilitar su búsqueda y recuperación.

Características de Dublin Core:

- Formato de descripción simple de un gran número de fuentes.
- Brinda una base semántica de metadatos analizados posteriormente.
- Define una base para la interoperabilidad semántica entre dominios de metadatos.

1.4.2 MODS

Es un esquema de metadatos descriptivo derivado del MARC para la descripción de recursos originales o seleccionados de registros existentes en MARC. MODS se puede emplear como un formato SRU específico. También como un esquema de ampliación a METS (Metadata Encoding and Transmission Standard), para representar los metadatos a recolectar y la descripción de recursos originales en sintaxis XML. Un registro de esquema MARC simplificado en XML y los metadatos en XML pueden ser ofrecidos junto con el recurso electrónico.

Selección del formato:

Se seleccionó Dublin Core como estándar de la representación de metadatos, en la recuperación de los recursos de información necesarios para facilitar el funcionamiento de la personalización.

1.5 Tecnologías y Herramientas utilizadas en el desarrollo de aplicaciones web.

1.5.1 HTML

Es un lenguaje utilizado comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes. Los archivos desarrollados con HTML, usan la extensión .htm o .html. Funciona por medio de “**etiquetas**” que describen la apariencia o función del texto enmarcado. Puede llegar a incluir un script o código que tenga incidencia en el comportamiento del navegador web de elección (DEFINICIÓNABC, 2013).

El propio World Wide Web Consortium (W3C) define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas en línea y banca electrónica (PÉREZ, 2008).

Se utilizó en la personalización del OHS HTML en su versión 4.0 y algunas características de HTML 5.0.

1.5.2 XML

Metalenguaje extensible de etiquetas desarrollado por el W3C. Es una simplificación y adaptación del SGML (Standard Generalized Markup Language) y permite definir la gramática de lenguajes.

Conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes.

Ventajas de XML:

- Separa radicalmente la información o el contenido de su presentación o formato.
- Diseñado para ser utilizado en cualquier lenguaje o alfabeto.
- Su análisis sintáctico es fácil debido a las estrictas reglas que rigen la composición de un documento.
- Estructura Jerárquica.
- El número de marcas es ilimitado.

Su objetivo es separar el contenido de la presentación del mismo, característica que le proporciona gran flexibilidad en el procesamiento y la visualización. La estructuración de datos usando XML permite acceder a datos a través de múltiples plataformas y aplicaciones, realizar búsquedas eficientes y para la visualización de datos mediante diversas interfaces. Se utilizó la versión de XML 1.0, en la personalización del OHS.

1.5.3 CSS

Son hojas de estilos en cascada, con un mecanismo muy simple que describe cómo es que el documento va a ser presentado en la pantalla, o como es que se va a imprimir. Ofrece a los implementadores un control total sobre el estilo y formato de sus propios documentos. Permite controlar el estilo y formato de múltiples páginas Web al mismo tiempo (STYLE SHEET, 2008).

Se utilizaron algunas características de CSS 3.0, para los estilos de las páginas y los formularios.

1.5.4 JavaScript

JavaScript es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. Es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. La versión que se utilizó en la personalización fue JavaScript 1.8.5.

1.5.5 PHP

Es un lenguaje interpretado de alto nivel, incrustado en páginas HTML y ejecutado en el servidor. Este permite hacer cualquier cosa que se pueda hacer con un script CGI (por sus siglas en inglés Common Gateway Interface, Interfaz Común de Compuerta) como procesar la información de los formularios, generar páginas con contenido dinámico, mandar o recibir cookies (MAGNUSSON, 2013).

PHP cuenta con muchas ventajas, entre las que se encuentran las siguientes:

- Alto rendimiento.
- Interfaces para los gestores de bases de dato: MySQL, PostgreSQL, Oracle, entre otros.
- Bibliotecas incorporadas para muchas tareas Web habituales.
- Facilidad de aprendizaje y uso.
- Portabilidad.
- Acceso al código abierto.

Se usó en el desarrollo de la personalización del OHS, la versión PHP 5.3.

1.5.6 Sistema Gestor de Base de Datos

Es una colección de programas cuyo objetivo es servir de interfaz entre las bases de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

1.6.6.1 MySQL

Es un sistema para la administración de bases de datos relacional (RDBMS) rápido y sólido. Las bases de datos permiten almacenar, buscar, ordenar y recuperar datos de forma eficiente. El servidor de MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, para proporcionar acceso a dichos datos y para asegurarse de que sólo tienen acceso a los mismos los usuarios con autorización. Por lo tanto, MySQL es un servidor multiusuario y de subprocesamiento múltiple (THOMSON, 2007).

MySQL cuenta con muchas ventajas, entre las que se encuentran las siguientes:

- Alto rendimiento.
- Facilidad de configuración y aprendizaje.
- Portabilidad.
- Accesibilidad a código fuente.

Como sistema gestor de bases de datos, se seleccionó MySQL 5.5.29 usando phpMyAdmin para la personalización del OHS, debido a que permite realizar el almacenamiento, búsqueda, ordenación y recuperación de los datos de manera eficiente.

1.5.7 NetBeans IDE

Es un entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que le permite a los desarrolladores crear rápidamente web, empresa, escritorio y aplicaciones móviles utilizando la plataforma Java, así como PHP y JavaScript (NETBEANS, 2012).

Se seleccionó el IDE NetBeans 7.2, el cual ofrece un rendimiento significativamente mejorado y la experiencia de codificación.

1.5.8 Servidor web

Un servidor web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles a los usuarios los recursos que solicitan mediante el protocolo HTTP a través de una conexión TCP/IP.

1.5.8.1 Apache

Se decidió usar como servidor web Apache, ya que es un servidor de código libre, robusto, cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. Está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Debido a su diseño, le garantiza a los administradores de sitios web elegir las características que deben ser incluidas en el servidor. Dadas estas características, se hace una selección para el servidor web utilizando Apache 2.2 para personalizar el OHS.

1.5.9 Lenguaje de Modelado

UML (por sus siglas en inglés Unified Modeling Language, Lenguaje Unificado de Modelado). Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura las decisiones y los conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre tales sistemas. UML incluye conceptos semánticos, notación, y principios generales.

1.5.10 Herramienta CASE

CASE (por sus siglas en inglés Computer Aided Software Engineering, Ingeniería de Software Asistido por Computadora). Una herramienta que ayuda en todos los aspectos del ciclo de vida de desarrollo del software en tareas como: el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras. Es la aplicación de la tecnología informática en las actividades, las técnicas y las metodologías propias de desarrollo. Tiene como objetivo acelerar el proceso para el que han sido diseñadas.

1.5.10.1 Visual Paradigm

Como herramienta CASE se seleccionó Visual Paradigm 8.0, debido a que ofrece un entorno de creación de diagramas para UML. La misma permite diseñar todos los tipos de diagramas de clases, código inverso, generar documentación y código desde los diagramas. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Características de Visual Paradigm:

- Generación de documentación: Brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.
- Disponibilidad en múltiples plataformas: Microsoft Windows, Linux, Mac OS X, Solaris o Java.
- Generación de código e ingeniería inversa: Brinda la posibilidad de generar código a partir de los diagramas para plataformas como: Net, Java y PHP.

1.5.11 Metodología de desarrollo

En cada uno de los procesos llevados a cabo para desarrollar un software se debe tener en cuenta que para obtener mejores resultados se tiene que aplicar el uso de una metodología, facilitando una mejor organización y satisfacción por parte de los desarrolladores y de los clientes que esperan el producto.

Una vez definida la necesidad del uso de una metodología es muy difícil definir cuáles de las existentes para el desarrollo de software se utilizará para el trabajo. Hay que centrarse en las necesidades y solicitudes del cliente para de esta forma analizar los artefactos que se deben producir, las notaciones que se usarán, las herramientas y especialmente el control del proceso en el que se participa. Se ha definido utilizar las buenas prácticas de las metodologías SCRUM y XP con un proceso de desarrollo de software guiado por el proceso de mejora CMMI nivel II.

SCRUM

- Define un marco para la gestión de proyectos.
- Indicada para proyectos con un rápido cambio de requisitos.
- El desarrollo de software se realiza mediante sprint.
- Reuniones a lo largo del proyecto.

XP

- Centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software.
- Basada en continua realimentación entre el cliente y el equipo de desarrollo.
- Simplicidad en las soluciones implementadas.
- Adecuada para proyectos con requisitos imprecisos y muy cambiantes.

1.5.11.1 Proceso de Mejora CMMI nivel II

Nivel II de CMMI:

Lo que se pretende con el nivel II de CMM-CMMI es conseguir que en los proyectos de la organización haya una gestión de los requisitos y que los procesos estén planeados, ejecutados, medidos y controlados.

Una organización con CMMI nivel II asegura que sus procesos son planeados, documentados,

realizados, monitorizados y controlados a nivel de proyectos. Los requerimientos, estándares y objetivos para los procesos, los productos de trabajo y servicios son definidos y documentados. El estado de los productos es visible para la administración en puntos de control preestablecidos (MONTALVÁN y AGÜERO, 2010).

Áreas de procesos del Nivel II de CMMI:

- Planeación de Proyectos.
- Monitoreo y control del proyecto.
- Administración de acuerdo con proveedores.
- Administración de requerimientos.
- Medición y análisis.
- Aseguramiento de la calidad del proceso y el producto.
- Administración de la configuración.

Conclusiones

Se realizó un análisis del estado actual de las tecnologías para la recuperación y presentación de la información científico-académica tanto en el ámbito nacional como internacional, en el que no se encontraron los requisitos necesarios que deben poseer estos tipos sistemas para brindar una mejor ayuda a los usuarios. Los mismos no brindan servicios como: la integración con redes sociales, gestores de referencias bibliográficas, espacio personal para coleccionar registros de interés. Además, no permite el cambio de idiomas. También se efectuó un estudio sobre las tendencias actuales de los sistemas de recuperación y presentación de información científico-académica, herramientas y metodologías que permitirán el desarrollo de la personalización del OHS. Seleccionándose el IDE Netbeans 7.2, como lenguajes: PHP, HTML, XML, CSS, entre otras y como gestor de bases de datos MySQL 5.5.29. La metodología para el desarrollo de dicho software: Proceso de Mejora CMMI nivel II con las buenas prácticas de XP y SCRUM, y como Herramienta CASE Visual Paradigm.

Capítulo #2 Características del sistema.

En el presente capítulo se exponen las características que presenta el sistema de indexación de metadatos OHS. Se realizará una descripción del modelo de dominio. Se especificarán los requisitos funcionales y no funcionales del sistema, se describen las Historias de Usuarios (HU) correspondientes para el desarrollo de la personalización del sistema OHS.

2.1 Propuesta de Solución

La Personalización del OHS tiene como objetivo mejorar la interacción de los usuarios con el sistema, a través de la integración de nuevas funcionalidades.

Su diseño será modificado, creando así una interfaz fácil para el control, por parte de los usuarios menos experimentados, y búsquedas de contenidos en el sistema de registros ya indexados. El sistema contará con la gestión de colecciones personales, donde los usuarios tendrán un espacio dentro del sistema donde puedan guardar los elementos de información (documentos) consultados.

Permitirá la exportación a gestores de referencias bibliográficas, brindando el almacenamiento, organización y control de las referencias bibliográficas pertenecientes a los usuarios. Se le brindará al usuario la opción de cambiar el idioma del sistema a: inglés o español, de acuerdo a su dominio de idioma, y tendrá una integración a las redes sociales.

2.2 Modelo de dominio

El modelo de dominio es una representación visual y estática del entorno real objeto del proyecto y se centra en una parte del negocio, se relaciona con el ámbito del proyecto. Es una representación de los conceptos significativos en el dominio del problema e incluye clases de objetos, asociaciones entre clases de objetos y atributos de las clases de objetos (SÁNCHEZ, 2006).

Se hace modelo de dominio, ya que no existe un entorno de negocio definido para ser analizado, por lo que la solución va a estar centrada en la mejora de un software, y en este caso, solo se muestra la representación y relación entre las clases existentes del problema a resolver.

2.2.1 Diagrama de modelo de dominio

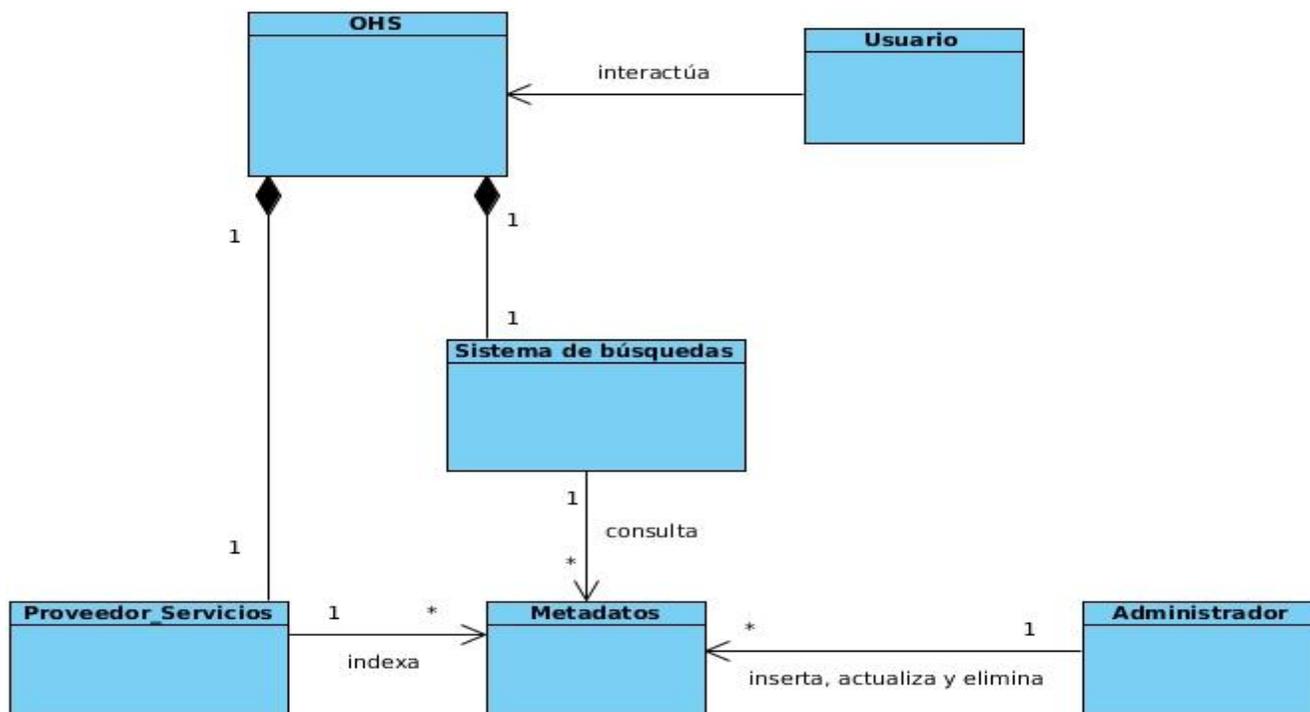


ILUSTRACIÓN 1: DIAGRAMA DE DOMINIO.

2.2.2 Descripciones de las clases representadas en el modelo de dominio.

User: interactúa con el sistema.

Proveedor_Servicios: se encarga de indexar los registros recolectados, obteniendo los metadatos.

Sistema de búsquedas: simplifica y agiliza las búsquedas y recuperación de los datos, a través de las consultas a los metadatos.

Metadatos: almacena información correspondiente a los registros.

Administrador: encargado de insertar, actualizar y eliminar los metadatos.

2.3 Levantamiento de requisitos

2.3.1 Requisitos funcionales

Los Requisitos funcionales (RF), son condiciones que el sistema debe ser capaz de cumplir. Declaraciones de los servicios que debe proporcionar el sistema, de la manera en que debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares.

Lista de reserva del producto

Prioridad

RF-1: Modificar idioma del sistema.	Media
RF-2: Realizar la integración con redes sociales.	Media
RF-3: Crear colección personal simple.	Alta
RF-4: Crear colección personal avanzada.	Alta
RF-5: Mostrar colecciones personales.	Media
RF-6: Eliminar colección personal.	Media
RF-7: Cambiar nombre a colección personal.	Media
RF-8: Agregar registro a colección personal.	Alta
RF-9: Mostrar registros de colección personal.	Media
RF-10: Ver registro de colección personal.	Baja
RF-11: Ver registro original de colección personal.	Baja
RF-12: Eliminar registro de colección personal.	Media
RF-13: Realizar integración a gestores bibliográficos.	Alta

2.3.2 Requisitos no funcionales

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

1.- Requisito de Hardware:

Hardware mínimo para el cliente:

- PC Pentium 3 o superior.
- CPU 2.3 MHZ o superior.
- 256 RAM o superior.

Hardware mínimo para el servidor Web:

- PC Intel Pentium 4 o superior.
- CPU 2.4GHZ o superior.
- 512 MB RAM o superior.
- 80 GB de disco duro o superior.

2.- Requisito de Software:

- Servidor Apache 2.2 o superior.
- MySQL 5.5 o superior.

- PHP 5.3 o superior.

3.- Confiabilidad

3.1- En caso de que el sistema presente alguna falla, los errores se deben mostrar sin detalles de información que pueda comprometer la seguridad e integridad del mismo.

4.- Usabilidad

4.1- El sistema podrá ser usado solo en ambiente web por personas con pocos conocimientos de informática.

5.- Soporte

- El sistema debe permitir la posibilidad de ser mejorado, así como de incorporarle nuevos servicios, en caso de ser necesarios.
- Es necesaria la documentación para el uso o desarrollo de nuevas funcionalidades del sistema.

6.- Interfaz

6.1- Interfaces de usuario:

- Deberá visualizarse de manera apropiada en los navegadores, como: Internet Explorer versión 9, Mozilla Firefox versión 15.

6.2- Interfaces de Comunicación:

- La comunicación entre el cliente y el servidor Web será realizada a través del protocolo HTTP.

7.- Seguridad

- El acceso al sistema será controlado mediante el nombre de usuario y contraseña.
- Los usuarios tendrán acceso solo a las funcionalidades que les sean permitidas según el rol.

2.4 Plan de Entrega

Definidas y agrupadas las funcionalidades por historias de usuario se define el plan de entrega donde se muestran las iteraciones para realizar las entregas intermedias y final de cada funcionalidad. Teniendo en cuenta la prioridad definida por funcionalidad y la entrega del OHS personalizado se ha realizado la siguiente planificación:

TABLA 1: PLAN DE ENTREGA.

Entrega	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	Implementar las funcionalidades: Modificar idioma del sistema y Realizar la integración con redes sociales.	1,2	4
2	Implementar las funcionalidades: Crear colección personal simple y avanzada, y Mostrar colecciones personales.	3,4,5	3
3	Implementar las funcionalidades: Eliminar colección personal, Cambiar nombre a colección personal, Agregar registro a colección personal y registros de colección personal.	6,7,8,9	4
4	Implementar las funcionalidades: Ver registro de colección personal, Ver registro original de colección personal y Eliminar registro de colección personal.	10,11,12	2
5	Implementar la funcionalidad: Realizar integración a gestores bibliográficos.	13	3

2.5 Historias de usuario

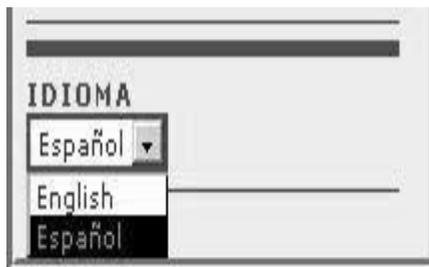
Las HU son la técnica utilizada en XP para especificar los requisitos del software, lo que equivale a los casos de uso en el proceso unificado. Son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Son escritas en lenguaje natural, no excediendo el tamaño de unas pocas líneas

de texto. Guían la construcción de las pruebas de aceptación, deben generarse una o más pruebas para verificar que la HU ha sido correctamente implementada y son utilizadas para estimar tiempos de desarrollo.

2.5.1 RF: Modificar idioma del sistema.

- HU: Modificar idioma del sistema.
- Descripción de la HU Modificar idioma del sistema.

TABLA 2: HU MODIFICAR IDIOMA DEL SISTEMA.

Historia de Usuario	
Código: 1	Nombre Historia de Usuario: Modificar idioma del sistema.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Media	Puntos Estimados: 4
Riesgo en Desarrollo: Alto	Puntos Reales: 3
<p>Descripción: La HU inicia cuando el usuario decide realizar una modificación del idioma del sistema. El usuario accede desde la barra lateral izquierda al bloque de idiomas. El sistema le muestra en ese bloque una interfaz para que el usuario seleccione el lenguaje de acuerdo a su dominio de idiomas (inglés o español).</p> <ul style="list-style-type: none"> • Idioma (Opcional, lista desplegable) <p>Una vez seleccionado el idioma en que el usuario desea realizar su navegación, el sistema inmediatamente traduce todo el contenido al idioma seleccionado por el usuario, finalizando así la HU.</p>	
Observaciones:	
<p>Prototipo de interfaz:</p> 	

2.5.2 RF: Mostrar colecciones personales.

- HU: Mostrar colecciones personales.
- Descripción de la HU Mostrar colecciones personales.

TABLA 3: HU MOSTRAR COLECCIONES PERSONALES.

Historia de Usuario	
Código: 5	Nombre Historia de Usuario: Mostrar colecciones personales.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Media	Puntos Estimados: 3
Riesgo en Desarrollo: Medio	Puntos Reales: 3
<p>Descripción: La HU inicia cuando el usuario desea que se visualicen todas sus colecciones personales en el sistema. El usuario accede desde la barra lateral izquierda a la opción “Administrar colecciones”. El sistema le muestra en una página todas sus Colecciones Personales, finalizando así la HU.</p>	
Observaciones:	
<p>Prototipo de interfaz:</p> 	

2.5.3 RF: Eliminar colección personal.

- HU: Eliminar colección personal.
- Descripción de la HU Eliminar colección personal.

TABLA 4: HU ELIMINAR COLECCIÓN PERSONAL.

Historia de Usuario	
Código: 6	Nombre Historia de Usuario: Eliminar colección personal.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Media	Puntos Estimados: 4
Riesgo en Desarrollo: Alto	Puntos Reales: 3
<p>Descripción: La HU inicia una vez que el usuario decide eliminar una colección personal en el sistema. El usuario accede desde la barra lateral izquierda al enlace Administrar colecciones. El sistema muestra en una página todas sus Colecciones Personales, donde el usuario puede eliminar la colección que desee. El usuario selecciona el enlace “Eliminar”, el sistema muestra un mensaje de confirmación de la acción y el nombre de la colección que desea eliminar.</p> <p>El usuario al decidir realizar dicha operación presiona el botón “Aceptar”. El sistema elimina la colección de sus Colecciones Personales, finalizando así la HU.</p>	
<p>Observaciones:</p> <p>1.- Debe existir al menos una colección personal en el sistema.</p>	
<p>Prototipo de interfaz:</p> 	

2.5.4 RF: Cambiar nombre a colección personal.

- HU: Cambiar nombre a colección personal.
- Descripción de la HU Cambiar nombre a colección personal.

TABLA 5: HU CAMBIAR NOMBRE A COLECCIÓN PERSONAL.

Historia de Usuario	
Código: 7	Nombre Historia de Usuario: Cambiar nombre a colección personal.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Media	Puntos Estimados: 4
Riesgo en Desarrollo: Alto	Puntos Reales: 2
<p>Descripción: La HU inicia cuando el usuario decide cambiarle el nombre a una colección personal. El usuario accede desde Inicio -> Administrar colecciones -> Cambiar nombre. El sistema le muestra en una página todas sus Colecciones Personales, donde el usuario puede cambiarle el nombre a la colección. El sistema le muestra una interfaz con el nombre de la colección que desea modificar y el siguiente metadato asociado:</p> <p>Nuevo nombre (Obligatorio, campo texto)</p> <p>El usuario completa los datos correspondientes y presiona el botón “Aceptar”. El sistema guarda la modificación realizada, finalizando así la HU.</p>	
<p>Observaciones:</p> <p>1.- El sistema verifica que el campo de texto obligatorio no se quede vacío, de lo contrario emite un mensaje de error.</p>	
<p>Prototipo de interfaz:</p>	

2.5.5 RF: Mostrar registros de colección personal.

- HU: Mostrar registros de colección personal.
- Descripción de la HU Mostrar registros de colección personal.

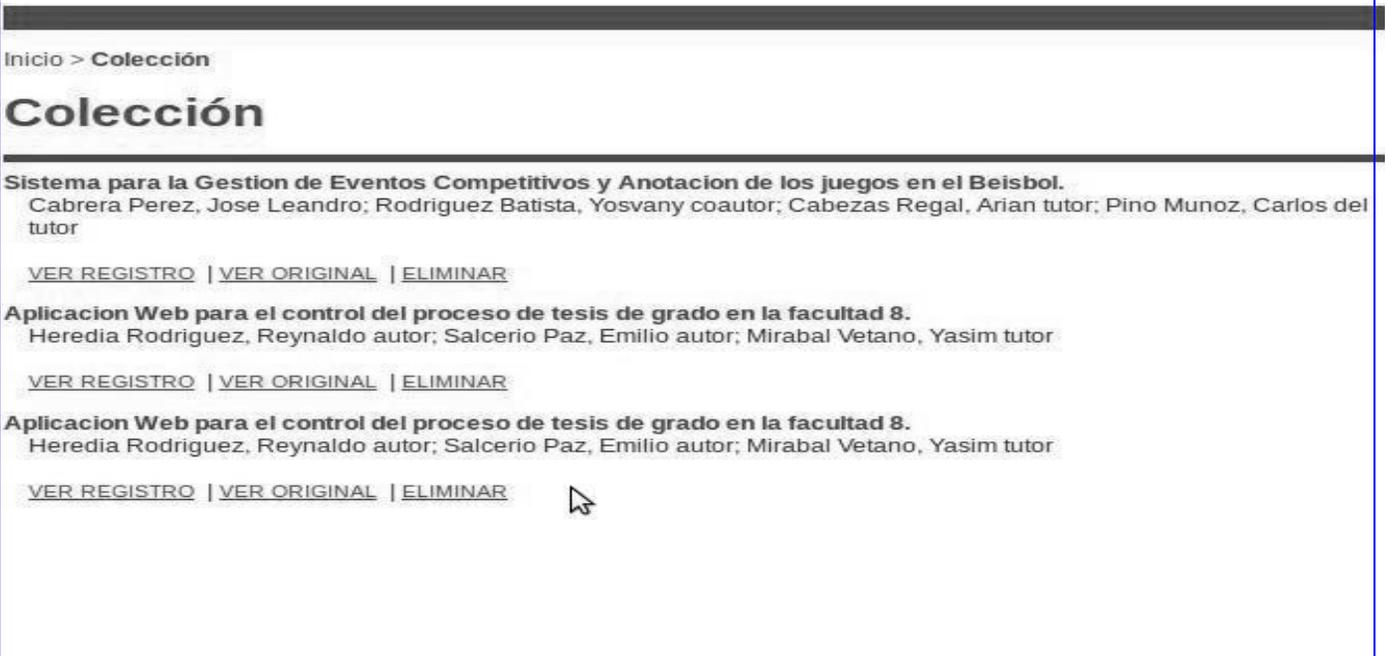
TABLA 6: HU MOSTRAR REGISTROS DE COLECCIÓN PERSONAL.

Historia de Usuario	
Código: 9	Nombre Historia de Usuario: Mostrar registros de colección personal.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Media	Puntos Estimados: 4
Riesgo en Desarrollo: Alto	Puntos Reales: 2
<p>Descripción: La HU inicia cuando un usuario decide ver los registros que tiene una colección personal en el sistema. El usuario accede desde la barra lateral izquierda desplegando Mis Colecciones, selecciona la colección que desea que se visualicen sus registros. El sistema le muestra en una página todos los registros que se le han agregado a la colección seleccionada, finalizando así la HU.</p>	
Observaciones:	
Prototipo de interfaz:	

2.5.6 RF: Ver registro de colección personal.

- HU: Ver registro de colección personal.
- Descripción de la HU Ver registro de colección personal.

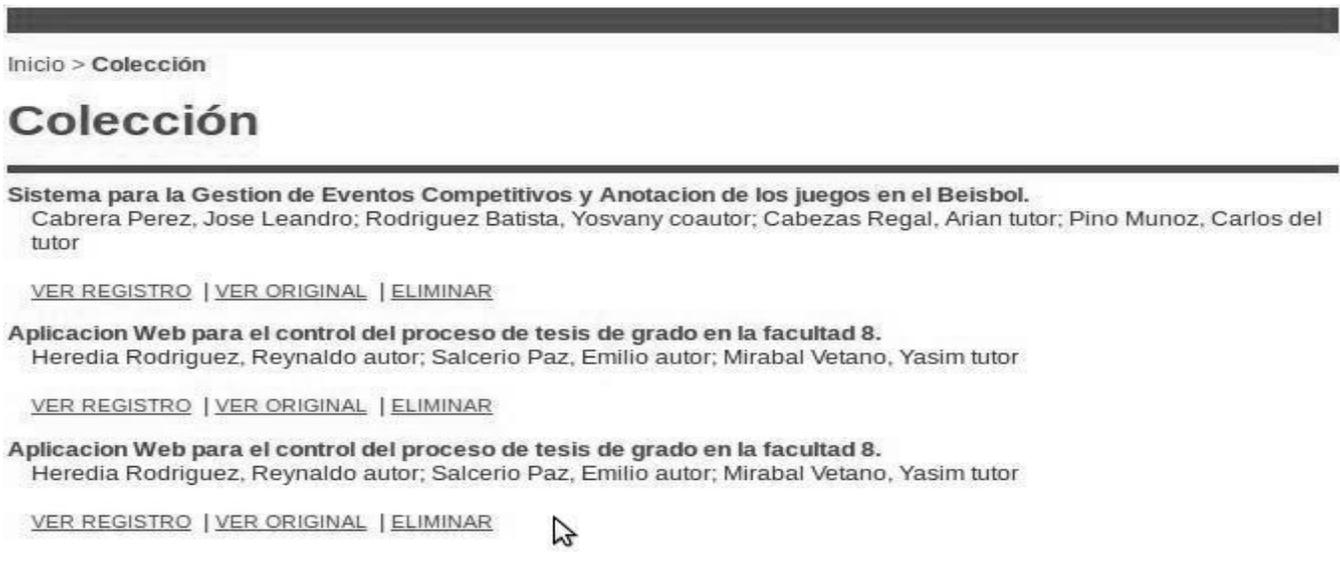
TABLA 7: HU VER REGISTRO DE COLECCIÓN PERSONAL.

Historia de Usuario	
Código: 10	Nombre Historia de Usuario: Ver registro de colección personal.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Baja	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 1
<p>Descripción: La HU inicia cuando un usuario decide ver contenido de un registro que tiene la colección personal en el sistema. El usuario accede desde la barra lateral izquierda desplegando Mis Colecciones, selecciona la colección que desea que se visualicen sus registros. El sistema le muestra en una página todos los registros que se le han agregado a la colección seleccionada. El usuario para ver el contenido del registro presiona el enlace “Ver registro”, finalizando así la HU.</p>	
Observaciones:	
Prototipo de interfaz:	
 <p>Inicio > Colección</p> <h2>Colección</h2> <hr/> <p>Sistema para la Gestion de Eventos Competitivos y Anotacion de los juegos en el Beisbol. Cabrera Perez, Jose Leandro; Rodriguez Batista, Yosvany coautor; Cabezas Regal, Arian tutor; Pino Munoz, Carlos del tutor</p> <p>VER REGISTRO VER ORIGINAL ELIMINAR</p> <p>Aplicacion Web para el control del proceso de tesis de grado en la facultad 8. Heredia Rodriguez, Reynaldo autor; Salcerio Paz, Emilio autor; Mirabal Vetano, Yasim tutor</p> <p>VER REGISTRO VER ORIGINAL ELIMINAR</p> <p>Aplicacion Web para el control del proceso de tesis de grado en la facultad 8. Heredia Rodriguez, Reynaldo autor; Salcerio Paz, Emilio autor; Mirabal Vetano, Yasim tutor</p> <p>VER REGISTRO VER ORIGINAL ELIMINAR</p>	

2.5.7 RF: Ver registro original de colección personal.

- HU: Ver registro original de colección personal.
- Descripción de la HU Ver registro original de colección personal.

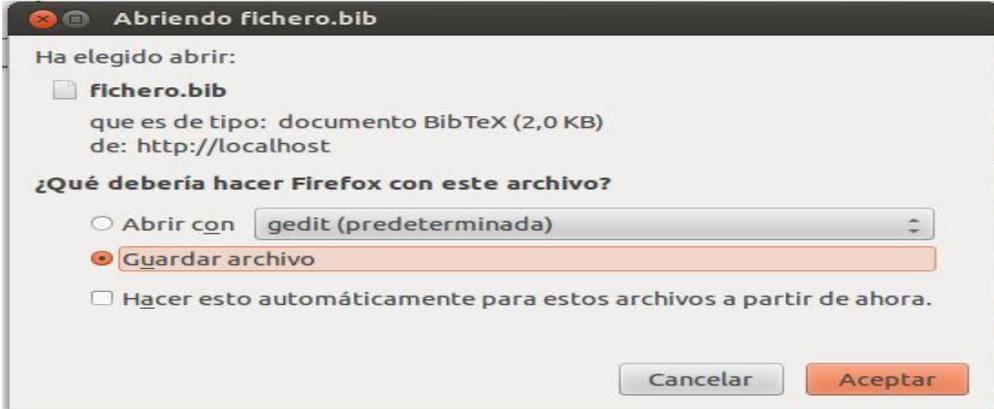
TABLA 8: VER REGISTRO ORIGINAL DE COLECCIÓN PERSONAL.

Historia de Usuario	
Código: 11	Nombre Historia de Usuario: Ver registro original de colección personal.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Baja	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 1
<p>Descripción: La HU inicia cuando un usuario decide ver el contenido original de un registro que tiene la colección personal en el sistema. El usuario accede desde la barra lateral izquierda desplegando Mis Colecciones, selecciona la colección que desea que se visualicen sus registros. El sistema le muestra en una página todos los registros que se le han agregado a la colección seleccionada. El usuario para ver el contenido del registro presiona el enlace “Ver original”, finalizando así la HU.</p>	
Observaciones:	
Prototipo de interfaz:	
 <p>The screenshot shows a web interface for a collection. At the top, there is a breadcrumb 'Inicio > Colección' and a large heading 'Colección'. Below this, there is a subtitle 'Sistema para la Gestion de Eventos Competitivos y Anotacion de los juegos en el Beisbol.' followed by author information: 'Cabrera Perez, Jose Leandro; Rodriguez Batista, Yosvany coautor; Cabezas Regal, Arian tutor; Pino Munoz, Carlos del tutor'. There are three records listed, each with a title, author information, and three action links: 'VER REGISTRO', 'VER ORIGINAL', and 'ELIMINAR'. A mouse cursor is pointing at the 'VER ORIGINAL' link of the third record.</p>	

2.5.8 RF: Realizar integración a gestores bibliográficos.

- HU: Realizar integración a gestores bibliográficos.
- Descripción de la HU Realizar integración a gestores bibliográficos.

TABLA 9: HU REALIZAR INTEGRACIÓN A GESTORES BIBLIOGRÁFICOS.

Historia de Usuario	
Código: 13	Nombre Historia de Usuario: Realizar integración a gestores bibliográficos.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3
<p>Descripción: La HU inicia cuando un usuario desea exportar registros a gestores de referencias bibliográficas en el sistema. El usuario accede a la opción: “Buscar”, que se encuentra ubicada en la barra lateral izquierda. El sistema le muestra una interfaz con el siguiente campo:</p> <ul style="list-style-type: none"> • Buscar (Opcional, campo de texto) <p>El usuario introduce el criterio de búsqueda y presiona el botón “Buscar”. El sistema realiza la búsqueda, según la temática, introducida y muestra los resultados. El usuario selecciona el enlace “Ver Registro”, muestra el contenido del registro que desee integrar a gestores bibliográficos. El sistema le muestra en el pie de página de los registros un vínculo “Exportar a Bibtex”. El usuario al dar clic en este vínculo, el sistema le muestra una ventana para que pueda realizar la exportación al gestor bibliográfico. El usuario presiona el botón “Aceptar” para realizar la exportación, finalizando así la HU.</p>	
Observaciones:	
<p>Prototipo de interfaz:</p> 	

Conclusiones

En este capítulo se describió la propuesta de solución, para lograr un mejor entendimiento de la personalización del OHS. Se realizó el modelo de dominio con el objetivo de identificar y relacionar las clases del OHS, garantizando una mejor comprensión del problema en cuestión. Se obtuvo la lista de reserva del producto, a través de la cual se pudieron conocer las principales funcionalidades que tendrá la personalización del sistema. Se describieron los requisitos funcionales, a través de las HU para realizar la futura implementación.

Capítulo #3 Construcción de la Solución Propuesta.

Se describen en el presente capítulo los elementos necesarios para la personalización del OHS. Una descripción para crear un plugin de bloque en el sistema. Muestra las clases usadas, a través del diagrama de clases persistentes y las tablas utilizadas mediante la representación del modelo de datos, y una distribución del sistema mediante nodos en el diagrama de despliegue. Se realiza una organización de los componentes con sus respectivas relaciones lógicas entre ellos, mediante el diagrama de componentes. También se refleja el diagrama de paquete y se abordan los métodos de pruebas utilizados en la personalización del sistema, con el interés de que cumpla con los requerimientos definidos y calidad requerida.

3.1 Crear un Block Plugin (Plugin Bloque)

Plugin bloque básico:

Crear plugin block (plugins bloque) o duplicar un plugins creado en: plugins/blocks/.

A continuación, realice las siguientes modificaciones en caso de querer duplicar algún plugins de bloque:

Nota: Plugin myPlugin y vamos a modificar (bloque) cambiándole el nombre por (myPlugin), dentro de esta carpeta realizar los siguientes cambios.

1.- Primeramente cambiar el nombre a CreadoByBlockPlugin.inc.php por MyPluginBlockPlugin.inc.php. (Todo esto dentro del directorio plugins /blocks/ su carpeta ejemplo: myPlugin).

2.- Se abre MyPluginBlockPlugin.inc.php y hacer los ajustes siguientes:

- Se edita en la parte superior del archivo los comentarios para reflejar el nombre del nuevo plugin, por ejemplo:

```
@ File MyPluginBlockPlugin.inc.php
@ Class MyPluginBlockPlugin
@ Plugins_blocks_myPlugin
@ Class breve plugin "myPlugin" bloque
```

- Cambiar el nombre de la clase de CreadoByBlockPlugin por MyPluginBlockPlugin.

- Cambiar el valor de retorno de la función getName () desde CreadoByBlockPlugin por MyPluginBlockPlugin.
- Cambiar el valor de retorno de las funciones GetDisplayName () y getDescription () desde plugins.block.creadoBy.x por plugins.block.myPlugin.x.

3.- Modificar en el directorio plugins / blocks / myPlugin / index.php

- Cambiar el nombre de la clase de CreadoByBlockPlugin por MyPluginBlockPlugin.
- Cambiar el valor de retorno de la función getName () desde CreadoByBlockPlugin por MyPluginBlockPlugin.
- Cambiar el valor de retorno de las funciones GetDisplayName () y getDescription () desde plugins.block.creadoBy.x por plugins.block.myPlugin.x.

4.- Cambiar los archivos de configuración regional para el plugin. Suponiendo que lo se desea en inglés, plugins/ block / myPlugin / locale / en_US / locale.xml.

- Cambiar la configuración regional de plugins.block.developedBy.x por plugins.block.myPlugin.x.
- Realizar el proceso para los otros idiomas para traducir. Si no se puede traducir el plugin a un idioma determinado, lo mejor es borrar el directorio de ese idioma para que los usuarios de esa lengua no vean la información incorrecta.

5.- settings.xml, que debería estar bien como está.

6. - Abrir plugins / block / myPlugin / block.tpl

- Esta es la plantilla de Smarty que mostrará el contenido de su plugin. Usted puede escribir directamente código HTML, pero también se puede aprovechar el poder de Smarty (<http://www.smarty.net>). Para cualquier contenido dinámico, tendrá que hacer uso de las funciones y variables de plantilla smarty, y probablemente también algo de PHP.

7. - Abrir "plugins / blocks / myPlugin / version.xml". Este archivo contiene información de versión y descripción API, que debe ser alimentado en la base de datos de OHS. Usted puede para cambiar lo siguiente:

- cambiar el <application>...</ application> valor de "creadoBy" a "myPlugin".
- cambiar el <fecha>... </ date> valor de ejemplo. 2012/07/07 a la fecha actual.

- (en el formato AAAA-MM-DD).
- cambiar el `<clase>...</ class>` valor de `creadoByBlockPlugin` por `MyPluginBlockPlugin`.

8.- Dar permiso a la carpeta del nuevo bloque creado.

9.- A continuación, registrar el plugin en la base de datos de OHS. Realizar mediante la ejecución del siguiente directorio raíz de OHS en su servidor:

- Abrir la consola y poner esto:

```
(php /var/www/nombre_de_la_carpeta_de_su_proyecto/tools/upgrade.php upgrade)
```

3.2 Arquitectura del OHS

La arquitectura del OHS se basa en un patrón Modelo Vista Controlador (MVC), en el que el controlador recibe las peticiones del usuario. La vista muestra al usuario los contenidos de una forma legible, y el modelo almacena los contenidos en una base de datos. De esta manera, el usuario puede consultar los contenidos y los administradores pueden insertar, actualizar y eliminar datos (CAMILO y SÁNCHEZ, 2012).

- El usuario interactúa con la interfaz.
- El controlador recibe la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo.
- La interfaz espera nuevas acciones por parte del usuario, para comenzar nuevamente el ciclo.

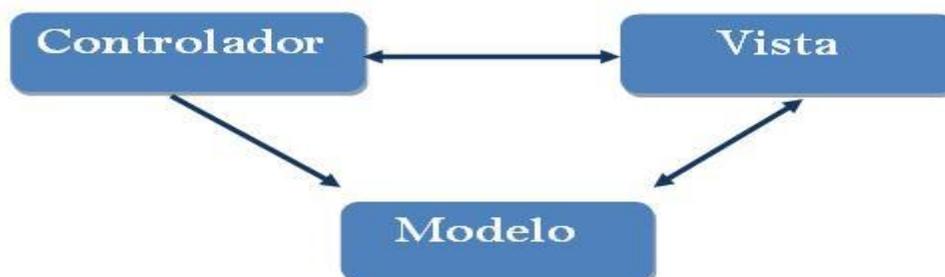


ILUSTRACIÓN 2: MODELO VISTA CONTROLADOR.

La arquitectura propia del OHS, define las siguientes capas:



ILUSTRACIÓN 3: ARQUITECTURA DEL OHS.

- *Servicio HTTP* (Apache HTTP versión 2.2.x. o superior). Servicio que permitirá ofrecer una interfaz gráfica de usuario basada en el uso de hipertexto y hojas de estilo en cascada.
- Intérprete PHP (versión 5.x. o superior). Instalado en paralelo al servicio HTTP, este servicio se encarga de la ejecución de métodos y funciones para interpretar las peticiones del usuario.
- Aplicación de correo electrónico (Sendmail versión 8.x. o superior). Servicio para la comunicación de los administradores a través de correo electrónico.
- Base de Datos MySQL (versión 5.x. o superior). Es el medio con el que se almacenarán, gestionarán, editarán o eliminarán los datos recolectados y las referencias a los objetos digitales.
- Sistema Operativo Linux.
- Contenedor de servlets: son objetos que corren dentro del contexto de un servidor de aplicaciones.

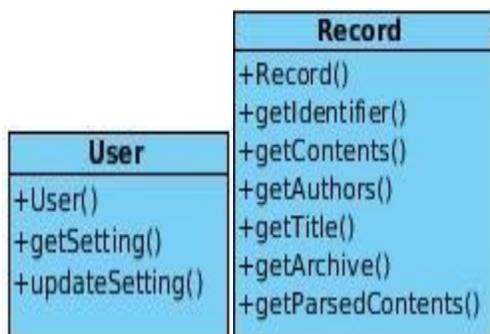
3.3 Patrones GRASP - Patrones de Principios Generales para Asignar Responsabilidades

Estos patrones describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones.

Experto: Se aplica para la asignación de responsabilidades a las clases, de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción específica. El uso de este patrón permitirá a los objetos valerse de su propia información para hacer lo que se les pide, favorece la existencia de mínimas relaciones entre las clases, lo que permite contar con un sistema robusto y fácil de mantener. A continuación se muestra la clase RecordDAO, como experta en el dominio de la información.

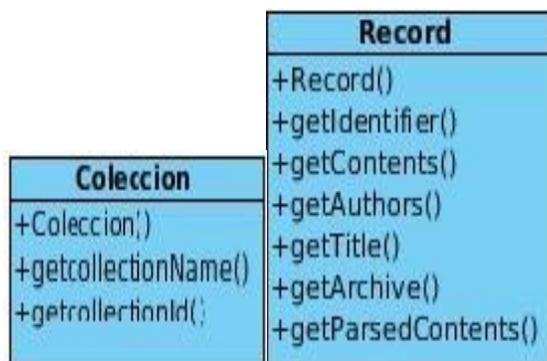


Bajo Acoplamiento: El acoplamiento evalúa la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencias con otras clases. A continuación, se muestran las clases User y Record, las cuales tienen una débil dependencia, y cualquier cambio en una de ellas no afectaría a la otra.



Alta cohesión: Se aplica para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podrían haber delegado a otros objetos o que tengan responsabilidades muy complejas. Se diseñaron las clases de forma tal que contengan

las mínimas responsabilidades necesarias y colaboren con otras para llevar a cabo una tarea. A continuación, se muestran las clases Colección y Record, ya que para la creación de colecciones personales se necesitan los registros.



Controlador: Delega la responsabilidad de controlar el flujo de eventos del sistema en varias clases con las que tiene una alta cohesión. Por lo que no existe una sola clase controladora que realice todas las actividades, garantizando de esta forma mejorar las validaciones y seguridad en el sistema. A continuación, se muestra como ejemplo la clase RecordHandler, la cual posee responsabilidades específicas.



3.4 Diseño del sistema

Para el diseño que propone la metodología de desarrollo de software XP, se confeccionó un diagrama de paquetes. En el cual se muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

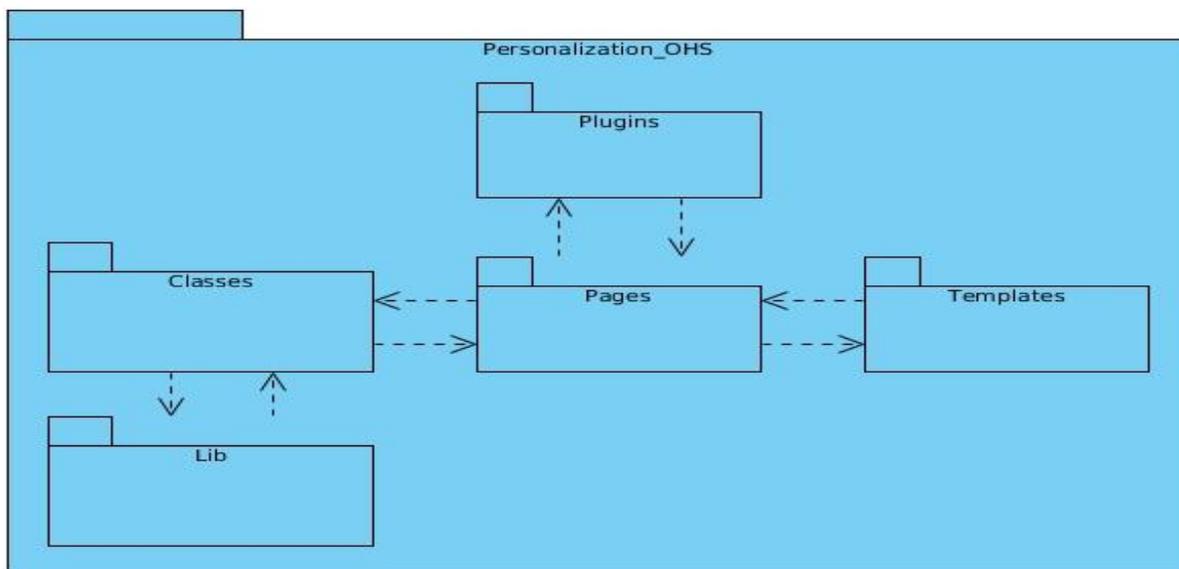


ILUSTRACIÓN 4: DIAGRAMA DE PAQUETES.

3.5 Diagrama de Clases Persistentes

A través de este diagrama se puede representar, en forma de clases toda aquella información que debe persistir como información duradera dentro de la base de datos. A continuación se muestra la representación de dicho diagrama:

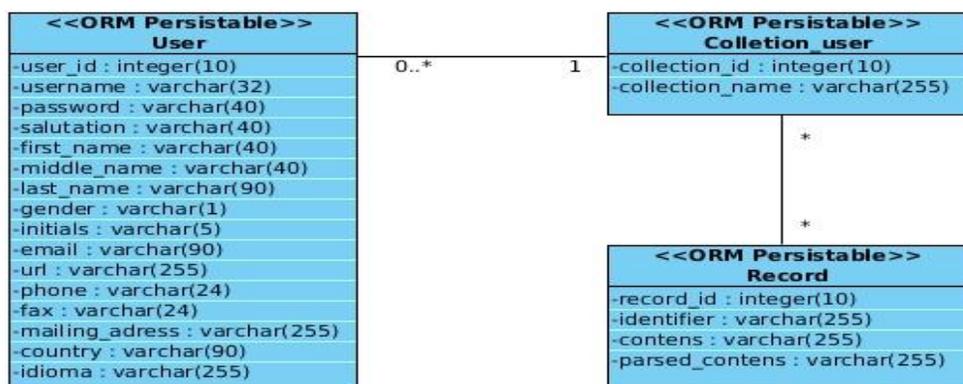


ILUSTRACIÓN 5: DIAGRAMA DE CLASES PERSISTENTES.

3.6 Modelo de datos

Un modelo de datos describe la representación lógica y física de los datos persistentes de la base de datos. Básicamente consiste en una descripción de un contenedor de datos donde se

guarda la información, así como los métodos a almacenar y recuperar información de dichos contenedores. Consiste en objetos, atributos y relaciones.

Se puede establecer de una manera más específica las relaciones entre las clases que se han definido en el diagrama de clases persistentes. A continuación se muestra la representación de dicho modelo:

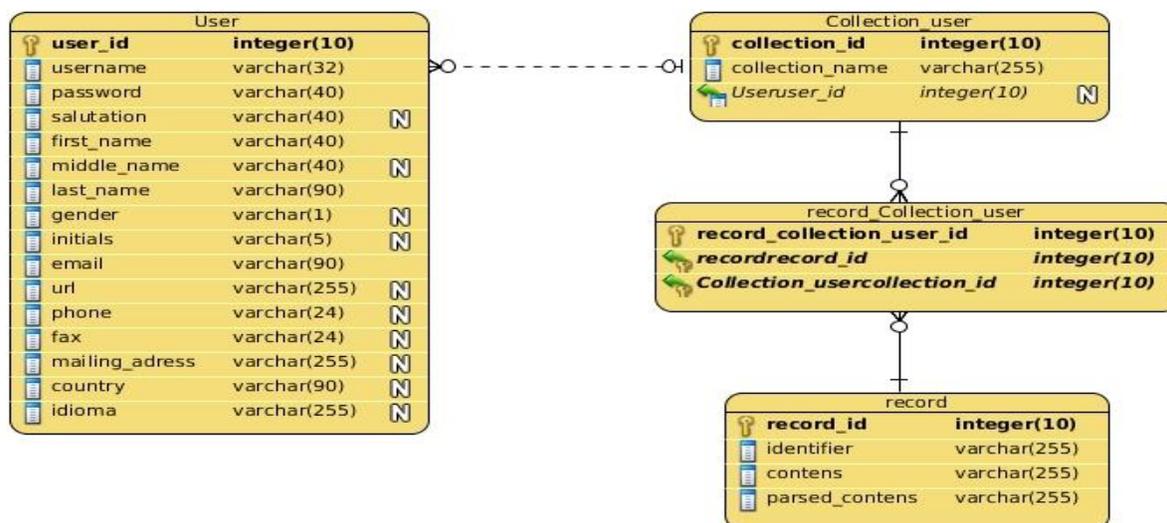


ILUSTRACIÓN 6: MODELO DE DATOS.

3.6.1 Descripción de las tablas del modelo de datos

Tabla User

TABLA 10: DESCRIPCIÓN DE LA TABLA USER DEL MODELO DE DATOS.

Nombre: User		
Descripción: En esta tabla se recoge toda la información referente a los usuarios que intervienen con la aplicación.		
Atributo	Tipo	Descripción
user_id	integer	Este atributo es el identificador que representa el usuario.
username	varchar	Este atributo especifica el nombre de usuario que tendrá el usuario.
password	varchar	Este atributo es la contraseña que tendrá el usuario.

salutation	varchar	Atributo que representa el tratamiento del usuario.
first_name	varchar	Representa el primer nombre del usuario.
middle_name	varchar	Este atributo que representa el segundo nombre del usuario.
last_name	varchar	Este atributo que representa el apellido del usuario.
gender	varchar	Este atributo especifica el género del usuario.
initials	varchar	Este atributo especifica las iniciales del usuario, ejemplo: YDD.
email	varchar	Representa el correo del usuario.
url	varchar	Este atributo representa la url del usuario.
phone	varchar	Representa el teléfono del usuario.
fax	varchar	Este atributo especifica el fax del usuario.
mailing_adress	varchar	Este atributo especifica la dirección de correo.
country	varchar	Representa el país del usuario.
idioma	varchar	Representa el idioma que domina el usuario.

Tabla Collection_user

TABLA 11: DESCRIPCIÓN DE LA TABLA USER_COLLECTION DEL MODELO DE DATOS.

Nombre: Collection_user		
Descripción: En esta tabla se tiene toda la información referente a las colecciones de los usuarios.		
Atributo	Tipo	Descripción
collection_id	integer	Este atributo es el identificador que representa la colección del usuario.
collection_name	varchar	Este atributo representa el nombre de la colección.

Tabla Registro

TABLA 12: DESCRIPCIÓN DE LA TABLA REGISTRO DEL MODELO DE DATOS.

Nombre: Registro		
Descripción: En esta tabla se tiene toda la información referente al registro.		
Atributo	Tipo	Descripción
record_id	integer	Este atributo es el identificador que representa al registro.
identifier	varchar	Este atributo representa el identificador del registro.
contens	varchar	Este atributo representa el contenido del registro.
parsed_contens	varchar	Guarda todos los metadatos de los registros.

3.7 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

Model Classes: implementan objetos de PHP que representan el sistema de diversas entidades, tales como los registros.

DAOs: generalmente ofrecen actualizar, crear y eliminar funciones para sus clases de modelos asociadas. Son responsables de toda la interacción con la base de datos.

Page Classes: reciben peticiones de los navegadores web de los usuarios, delegan cualquier procesamiento necesario para otras clases, y llamar a la plantilla adecuada para generar una respuesta.

Templates: plantillas responsables de mostrar a los usuarios las interfaces de páginas construidas en HTML.

El siguiente diagrama de componentes, se agrupan los elementos físicos en paquetes evidenciando de esta forma el uso del patrón de arquitectura MVC.

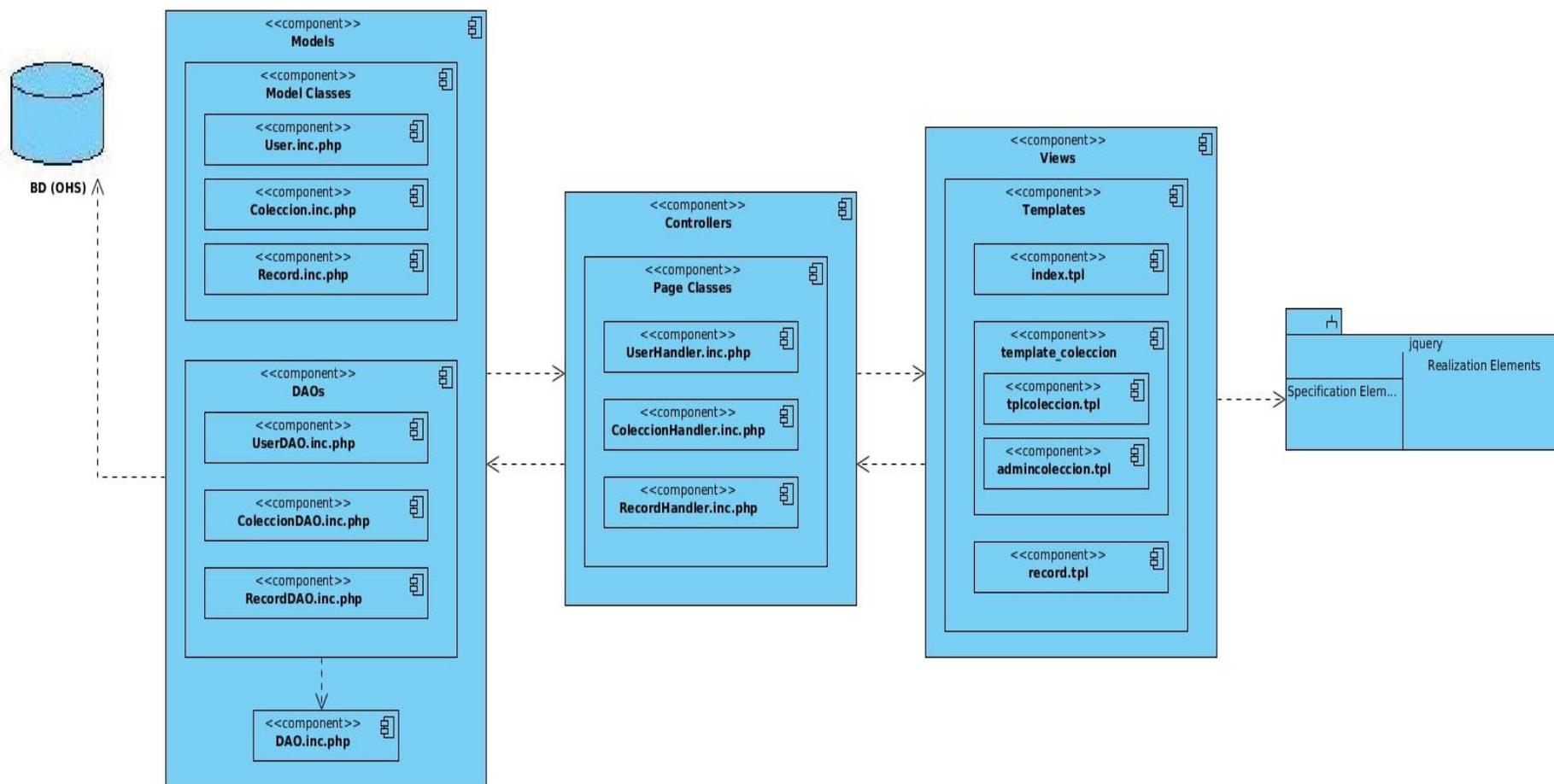


ILUSTRACIÓN 7: DIAGRAMA DE COMPONENTES.

3.8 Diagrama de Despliegue

El diagrama de despliegue describe cómo una aplicación se despliega a través de una infraestructura y muestra la configuración ó relaciones físicas de los nodos que participan en la ejecución de los componentes de hardware y software que residen en ellos. Contiene nodos como servidores o procesadores y dispositivos, además de relaciones de dependencia y asociación.

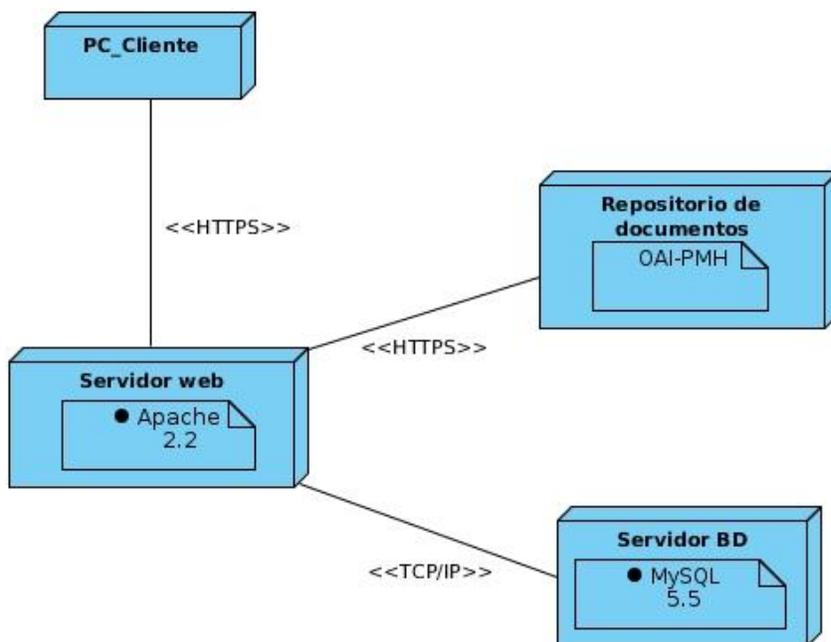


ILUSTRACIÓN 8: DIAGRAMA DE DESPLIEGUE.

3.8.1 Descripción de los nodos

PC Cliente

Se puede acceder desde cualquier PC vía Web a la aplicación. Los requerimientos de hardware necesarios son mínimos debido a que el grueso del procesamiento es llevado a cabo en el servidor, se propone: Microprocesador Intel Pentium III o superior, 450 MHz y Memoria RAM 512 MB.

Servidor de Aplicaciones Web

El Servidor Web debe tener alta disponibilidad y un rendimiento adecuado, se propone: Microprocesador Intel Pentium III o superior, 800 MHz y Memoria RAM 512 MB o superior. Capacidad del disco duro: 80 GB o superior.

Servidor de BD

El servidor de almacenamiento de datos debe tener buena capacidad, para que pueda responder a grandes volúmenes de información. Se propone: Microprocesador Intel Pentium III o superior, 800 MHz y Memoria RAM 512 MB o superior. Capacidad del disco duro 80 GB o superior.

Validación de requisitos funcionales.

Las validaciones a los requisitos funcionales se realizaron mediante la técnica de lista de chequeo. Donde se midieron una serie de parámetros que deben cumplir estos requisitos.

TABLA 13: LISTA DE CHEQUEO DE REQUISITOS FUNCIONALES.

Lista de chequeo de requisitos funcionales	Si	No	NP
1. ¿Los requisitos están escritos en un lenguaje no técnico y comprensible para el usuario?	x		
2. ¿Los requisitos pueden ser entendidos, implementados y verificados por un grupo independiente?	x		
3. ¿Están incluidos todos los requisitos relacionados con la funcionalidad?	x		
4. ¿Para cada requisito hay un proceso que puede ser ejecutado por un humano o una máquina para verificar los requisitos?	x		
6. ¿Todas las funciones del software están especificadas?	x		
7. ¿Hay alguna redundancia en los requisitos?		x	
8. ¿Hay algún requisito que pueda tener más de una interpretación?		x	
9. ¿Existen requisitos contradictorios?		x	

3.9 Pruebas

3.9.1 Pruebas de Caja negra

Son pruebas funcionales dedicadas a mirar en el exterior de lo que se está probando. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos en base a las pruebas.

Casos de Pruebas de Aceptación

TABLA 14: HU1_CP1 MODIFICAR IDIOMA DEL SISTEMA.

Escenario	Descripción	Idioma	Respuesta del sistema	Flujo central
EC 1.1 Modificar idioma del sistema correctamente.	Permite cambiar el idioma del sistema.	V	El sistema muestra el contenido del sitio en dependencia de la selección del idioma.	1.- El usuario accede a la barra lateral izquierda del sistema. 2.- Selecciona el lenguaje en que desea ver el contenido del sistema.
		Inglés		
		V		
		Español		

TABLA 15: HU5_CP5 MOSTRAR COLECCIONES PERSONALES.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar colecciones personales correctamente.	Permite mostrar las colecciones personales del usuario en una página.	El sistema le muestra todas las colecciones personales que han sido creadas.	1.- El usuario accede a la parte superior de la barra lateral izquierda del sistema. 2.- Presiona el enlace “Administrar colecciones” .

TABLA 16: HU6_CP6 ELIMINAR COLECCIÓN PERSONAL.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar colección personal correctamente.	Permite eliminar una colección personal de un usuario.	El sistema le muestra todas las colecciones y le permite eliminar la colección que desee, se muestra el siguiente mensaje de confirmación de la acción: “Está seguro que desea eliminar la colección: Seguridad” .	1.- El usuario accede a la parte superior de la barra lateral izquierda del sistema. 2.- Presiona el enlace “Administrar colecciones” .

TABLA 17: HU7_CP7 CAMBIAR NOMBRE A COLECCIÓN PERSONAL.

Escenario	Descripción	Nuevo nombre	Respuesta del sistema	Flujo central
EC 1.1 Cambiar nombre a colección personal correctamente.	Permite cambiarle el nombre a una Colección personal.	V Nombre	El sistema le muestra los cambios realizados en “Administrar colecciones” , donde aparece la colección con su nuevo nombre.	1.- El usuario accede a la parte superior de la barra lateral izquierda del sistema. 2.- Presiona el enlace “Administrar colecciones” . Se muestran todas las colecciones.
EC 1.2 Cambiar nombre a colección personal incorrectamente.	No permite cambiarle el nombre a la colección personal.	I Vacío	El sistema no realiza los cambios, debido a que se dejó el campo de texto en blanco y le muestra al usuario un mensaje de error: “Por favor, rellene este campo” .	3.- El usuario puede cambiar el nombre de la colección que desee. Al presionar el enlace Cambiar Nombre . Se muestra un formulario para introducir el dato correspondiente. 4.- El usuario presiona el botón Aceptar , para que se realicen los cambios.

TABLA 18HU9_CP9 MOSTRAR REGISTROS DE COLECCIÓN PERSONAL.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar registros de colección personal correctamente.	Permite mostrar los registros de una colección personal.	El sistema muestra todos los registros que le han sido añadidos a una colección.	<p>1.- El usuario accede a la parte superior de la barra lateral izquierda del sistema.</p> <p>2.- Presiona el enlace “Mis colecciones”. Se muestran todas las colecciones.</p> <p>3.- Al desplegar “Mis colecciones”, se selecciona la colección que desee ver sus registros agregados.</p>

TABLA 19: HU10_CP10 VER REGISTRO DE COLECCIÓN PERSONAL.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Ver registro de colección personal correctamente.	Se muestra el contenido del registro.	El sistema muestra el contenido del registro.	<p>1.- El usuario accede a la opción: “Buscar”, que se encuentra ubicada en la barra lateral izquierda.</p> <p>2.- Se introduce el criterio de búsqueda en el campo de texto y se presiona el botón “Buscar”.</p> <p>3.- Presiona el enlace “Ver registro”.</p>

TABLA 20: HU11_CP11 MOSTRAR REGISTROS DE COLECCIÓN PERSONAL.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Ver registro	Permite visualizar el	El sistema	1.- El usuario accede a la opción:

original de colección personal correctamente.	registro original.	muestra el contenido del registro original.	<p>“Buscar”, que se encuentra ubicada en la barra lateral izquierda.</p> <p>2.- Se introduce el criterio de búsqueda en el campo de texto y presiona el botón “Buscar”.</p> <p>3.- Presiona el enlace “Ver original”.</p>
---	--------------------	---	--

TABLA 21: HU13_CP13 REALIZAR INTEGRACIÓN A GESTORES BIBLIOGRÁFICOS.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Realizar integración a gestores bibliográficos correctamente.	Permite la integración con gestores de referencias bibliográficas.	El sistema genera un fichero (.bib) con los metadatos del registro.	<p>1.- El usuario accede a la opción: “Buscar”, que se encuentra ubicada en la barra lateral izquierda.</p> <p>2.- El usuario introduce el criterio de búsqueda en el campo de texto y presiona el botón “Buscar”.</p> <p>3.- Presiona el enlace “Ver registro”.</p> <p>4.- El usuario selecciona el enlace “Exportar a Bibtex”, ubicado en el pie de página del registro.</p>

Resultados de las pruebas

Se realizaron 13 casos de pruebas de aceptación, donde se muestra a continuación el análisis realizado en cada una de las iteraciones:

TABLA 22: NO CONFORMIDADES ENCONTRADAS Y RESUELTAS POR ITERACIÓN.

Iteraciones	No conformidades	
	Encontradas	Resueltas
Iteración_1	3	3
Iteración_2	2	2
Iteración_3	ninguna	ninguna

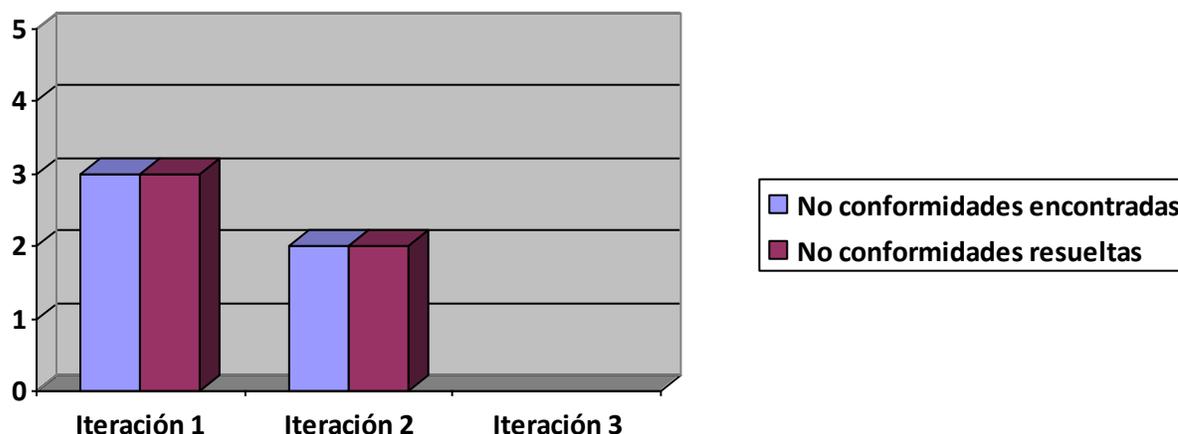


ILUSTRACIÓN 9: RESULTADOS.

Como resultado de las pruebas realizadas fueron resueltas todas las no conformidades encontradas en cada una de las iteraciones, por lo que el resultado de las pruebas fue satisfactorio pudiéndose obtener un software con calidad que satisface los requisitos establecidos.

3.9.2 Pruebas de Caja blanca

Se realizaron pruebas de Caja blanca, hechas al código del sistema, a través del método del camino básico. El cual permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecute por lo menos una vez cada sentencia del programa (PRESSMAN, 2002).

El cálculo de la complejidad ciclomática se realiza de tres formas diferentes:

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.

- La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como: $V(G) = A - N + 2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G también se define como: $V(G) = P + 1$, donde P es el número de nodos predicaado (son los nodos de los cuales parten dos o más aristas) que tiene contenido el grafo de flujo G .

Ejemplo de código de la funcionalidad enviarColeccion:

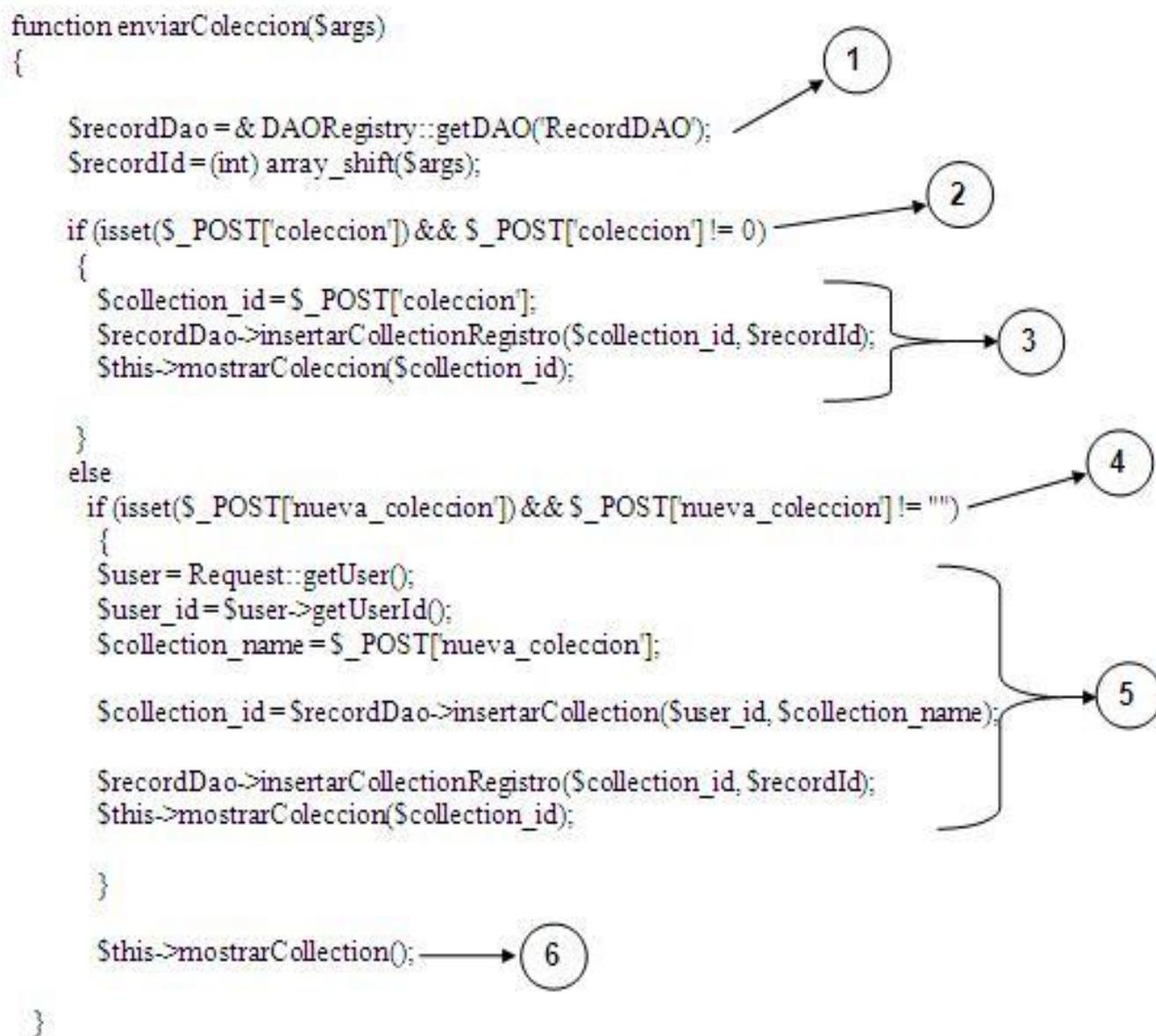


ILUSTRACIÓN 10: CÓDIGO DE LA FUNCIONALIDAD ENVIAR COLECCION.

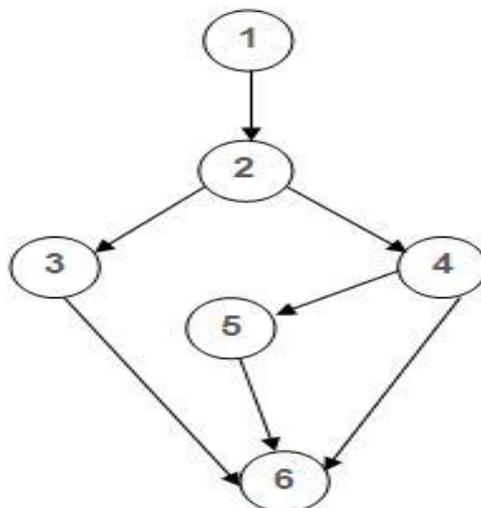


ILUSTRACIÓN 11: GRAFO DEL FLUJO ASOCIADO A LA FUNCIONALIDAD ENVIAR COLECCION.

Resultado del cálculo de la complejidad ciclomática:

- El grafo de flujo tiene 3 regiones.
- $V(G) = A - N + 2 = 7 - 6 + 2 = 3$
- $V(G) = P + 1 = 2 + 1 = 3$

A partir del resultado obtenido, se determina que la funcionalidad presenta una complejidad ciclomática de 3, lo que deriva que presenta a lo sumo 3 caminos lógicos por donde ejecutarse dicha funcionalidad. En la siguiente tabla se presentan los caminos básicos:

TABLA 23: CAMINOS BÁSICOS DE ENVIAR COLECCION.

Nro.	Camino básico
1	1,2,3,6
2	1,2,4,5,6
3	1,2,4,6

Se procede a realizar los casos de prueba para la funcionalidad en cuestión. Se debe elaborar al menos un caso de prueba por cada camino básico.

Casos de pruebas

TABLA 24: CASO DE PRUEBA DEL CAMINO #1

Camino	1,2,3,6
Descripción	Enviar registro a colección personal.
Condición de ejecución	Debe existir una colección personal cargada en la lista desplegable de colecciones a seleccionar para agregar el registro.
Entrada	Seleccionar la colección personal a la que se enviará el registro.
Resultados esperados	El registro ha sido enviado a la colección seleccionada.

TABLA 25: CASO DE PRUEBA DEL CAMINO #2

Camino	1,2,4,5,6
Descripción	Agregar nueva colección personal.
Condición de ejecución	No debe existir una colección personal con el nombre de la nueva que se desea agregar.
Entrada	Introducir el nombre de la nueva colección que se desea añadir.
Resultados esperados	Colección agregada correctamente.

TABLA 26: CASO DE PRUEBA DEL CAMINO #3

Camino	1,2,4,6
Descripción	Enviar colección personal o registro a una colección creada.
Condición de ejecución	No debe existir la colección personal, no debe estar agregado el registro.
Entrada	-
Resultados esperados	-

Los casos de pruebas permitieron garantizar que:

- Se ejecuten por lo menos una vez todos los caminos.
- Todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Las estructuras internas de datos para asegurar su validez.

Conclusiones

En el capítulo se realizó una descripción de cómo crear un plugin de bloque básico al sistema, para un mayor conocimiento por parte del usuario. Se representó la arquitectura del OHS, definida por el patrón de arquitectura MVC. Quedó representada una descripción de los patrones GRASP y ejemplos de clases donde se reflejan los mismos. Se mostró la forma en que se

encuentra dividido en agrupaciones el sistema, y la representación en forma de clases de la información que debe persistir dentro de la base de datos, a través del diagrama de Clases persistentes. Se elaboró el diseño del modelo de datos y la descripción de las tablas de la base de datos. Mediante el diagrama de componentes, se muestra como un componente utiliza los servicios ofrecidos por otros. Además, se construyeron las estrategias de pruebas, utilizándose la técnica de pruebas de Caja negra y Caja blanca, para constatar que las funcionalidades implementadas cumplen con los requisitos establecidos.

Conclusiones

Con el desarrollo de este trabajo se ha podido demostrar cómo se han cumplido los objetivos principales definidos para el desarrollo de una personalización del OHS, que permita una mejor interacción del usuario con el sistema. Luego de analizado todo el contenido anteriormente expuesto, se arriban las siguientes conclusiones:

- ❖ La investigación realizada a las tecnologías relacionadas con la presentación la información en los sistemas de recuperación de información, facilitó una correcta elección de las herramientas y metodología para el desarrollo de la personalización del OHS.
- ❖ La utilización de las tecnologías y herramientas para el desarrollo de la personalización de sistema, facilitó mejorar la representación de la información y la interacción por parte de los usuarios.
- ❖ La confección de los diagramas como: Modelo de dominio, despliegue, paquetes, componentes y el uso del patrón arquitectónico MVC; permitieron un entendimiento para el desarrollo de la propuesta de solución.
- ❖ La implementación de nuevas funcionalidades al OHS, facilita una correspondiente representación de la información y que los usuarios tengan una mejor interacción con el mismo.
- ❖ El diseño de la personalización del OHS, permite que se mejore la presentación de la información y que los usuarios puedan hacer búsquedas de material almacenado en el sistema de acuerdo a sus necesidades de información.
- ❖ Las pruebas, permitieron constatar que las funcionalidades implementadas cumplen con los requisitos establecidos.

Recomendaciones

Al finalizar la investigación, se plantean la siguiente recomendación:

- ❖ Implementar la funcionalidad, que permita garantizar la integración con otros gestores de referencias bibliográficas como: Endnote y Zotero.

Bibliografía consultada

1. KUANG, H. Finding the needle: controlled vocabularies, resource discovery, and Dublin Core. Library collections, acquisitions, and technical services. 2000.
2. REPISO, O. J. Nuevas perspectivas para la Catalogación: Metadatos versus MARC. [Consultado el: 20/11/2012 Disponible en: <http://redc.revistas.csic.es>.
3. SOLA, M. Greenstone: distribución de colecciones digitales [Consultado el: 25/01/2013 Disponible en: <http://www.ub.edu/bid/21/sola2.htm>.
4. CYRANEK, G. Greenstone: Un software libre de código abierto para la construcción de bibliotecas digitales [Consultado el: 14/03/2013 Disponible en: <http://www.unesco.org/new/en/>.
5. KENNEDY, B. y MUSCIANO, C. HTML Guía Completa. 2 ed. México: 1995. ISBN 1-56592-235.
6. MENDOZA, T. Análisis de DSpace como repositorio institucional [Consultado el: 13/01/2013 Disponible en: <http://es.scribd.com/doc/48250416/Analisis-de-DSpace-como-repositorio-institucional>.
7. SÁNCHEZ, M. Definición del modelo del negocio y del dominio utilizando Razonamiento Basado en Casos. [Consultado el: 02/04/2013 Disponible en: <http://www.inf.udec.cl/~revista/ediciones/edicion8/Rbc.pdf>.
8. PÉREZ, M. y L.F. Silva. 2007. "Cómo funciona el protocolo OAI – PMH en la recuperación de información".
9. LANCASTER, F. W. y MOLINA, M. P. Procesamiento de la información científica España: 2001. ISBN 84-7635-485-1.
10. Dspace-general Info Page. [Consultado el: 10/04/2013 Disponible en: <http://mailman.mit.edu/mailman/listinfo/dspace-general>.
11. Acceso Abierto a la literatura científica y académica. DOAJ -Directory of Open Access Journals. [Consultado el: 20/04/2013 Disponible en: <http://abierto.blogspot.com/2007/03/doaj-directory-of-open-access-journals.html>.
12. Taringa. Información. 2009. [Consultado el: 10/02/2013 Disponible en: <http://www.taringa.net/posts/info/2689547/Que-es-la-informacion.html>.
13. MODS metadata object description schema [Consultado el: 18/05/2013 Disponible en: <http://www.loc.gov/standards/mods/>.

Bibliografía consultada

14. Recuperación de información. 2010 [Consultado el: 09/05/2013 Disponible en: <http://sistemasrecuperaciondeinformacion.blogspot.com/2010/09/recuperacion-de-informacion-en-la-web.html>].
15. Ecured. Pruebas de Caja Negra, 2010. [Consultado el: 15/04/2013 Disponible en: http://www.ecured.cu/index.php/Pruebas_de_caja_negra].
16. Globe Testing. Pruebas de Caja Negra. [Consultado el: 19/04/2013 Disponible en: <http://www.globetesting.com/2012/08/pruebas-de-caja-negra/>].
17. Site Informática. Introducción a Java Servlets con Netbeans. [Consultado el: 05/04/2013 Disponible en: <http://site-informatica.blogspot.com/2011/02/introduccion-java-servlets-con-netbeans.html>].
18. Análisis de DSpace como repositorio institucional. Disponible en: <http://es.scribd.com/doc/48250416/Analisis-de-DSpace-como-repositorio-institucional>.
19. Serie Científica de la Universidad de las Ciencias Informáticas. 2013, vol. 6, nº 3, Disponible en: <https://publicaciones.uci.cu/index.php/SC>.
20. Revista Cubana de Ciencias Informáticas (RCCI). 2013, vol. 7, nº Disponible en: <http://rcci.uci.cu/index.php/rcci>. ISSN 2227-1899.
21. Metadata Object Description Schema. [Consultado el: 02/04/2013 Disponible en: <http://www.loc.gov/standards/mods/>].
22. Plataforma Netbeans. [Consultado el: 26/02/2013 Disponible en: <http://netbeans.org/>].
23. MATOS, N. L. y RIOPEDRE, Y. D. L. A. C. Sistema de indexación de documentos para la Biblioteca Digital "Alma Mater". 2011.
24. HERNÁNDEZ, F. L. Mejores prácticas de codificación con XP. 2008, Disponible en: <http://www.caudalweb.com/blog/mejores-practicas-de-codificacion-con-xp>.
25. GONZÁLEZ, M. V. y MARTINEZ, D. R. PGST: "Módulo para la gestión de configuración del equipamiento tecnológico en la Universidad de las Ciencias Informáticas". 2010.
26. PKP. Open Journals Systems Disponible en: <http://pkp.sfu.ca/?q=ojs>.
27. FERNÁNDEZ, L. C. A. y INERARTE, E. V. Implantación de un Repositorio Institucional en la Universidad de las Ciencias Informáticas. 2010.
28. UML. Diagramas de Componentes. [Consultado el: 02/04/2013 Disponible en: https://uvirtual.unet.edu.ve/pluginfile.php/7632/mod_resource/content/0/UML_-_Diagramas_de_Componentes.htm].

Bibliografía consultada

29. Taringa. Información. [Consultado el: 10/02/2013 Disponible en: <http://www.taringa.net/posts/info/2689547/Que-es-la-informacion.html>.
30. INGENIEROSOFTWARE. CMMI Nivel II. 2005, nº [Consultado el: 28/04/2013]. Disponible en: www.ingenierosoftware.com/calidad/cmm-cmmi.php.
31. Yi, J. M. A. Núcleo de la red social universitaria de la Universidad de las Ciencias Informáticas. 2012.
32. EMMANUEL, A. Z. D.; LUIS, E. H. J., et al. CMMI & MOPROSFOT. 2013.
33. Entorno Virtual de Postgrado de la Universidad de las Ciencias Informáticas. nº Disponible en: <http://evapostgrado.uci.cu/>.
34. MONTALVO, M. M. XML. El Nuevo Lenguaje Universal. 2008, nº [Consultado el: 10/02/2013].
35. SLIDESHARED. Herramienta Case [Consultado el: 12/03/2013 Disponible en: <http://www.slideshare.net/Giorlysole/herramientas-case-5004699>.
36. UBEROI, D. y BRUYN, M. D. Health and Human Rights. 2013, Disponible en: <http://www.hhrjournal.org/index.php/hhr>.
37. KENDALL, C. y PALEPU, A. Open Medicine. 2012, vol. 6, nº p. pp.59-61. [Consultado el: 02/03/2013]. Disponible en: <http://www.openmedicine.ca/>. ISSN 1911-2092
38. THOMSON, L. W. Y. L. Programación: Desarrollo web con PHP y Mysql. 2007.

Bibliografía referenciada

Bibliografía referenciada

- ARIAS, J. Análisis para la creación y desarrollo de la biblioteca digital de Colombia [Consultado el: 01/12/2012 Disponible en: <http://www.tesisenred.net/bitstream/handle/10803/10913/AriasOrdonez.pdf?sequence=1>].
- BARRUECO, J. M. OAI-PMH: Protocolo para la transmisión de contenidos en Internet. 2010.
- CAMILO, M. A. O. y SÁNCHEZ, G. C. Un portal de acceso abierto a la literatura científica en iberoamérica. 2012, vol. 13, nº 10, [Consultado el: 28/03/2013]. Disponible en: <http://www.revista.unam.mx/vol.13/num10/art104/index.html>. ISSN 1067-6079.
- CARRERAS, Y. D. L. Á. y ÁLVAREZ, J. M. R. Protocolos para el intercambio de información entre bibliotecas digitales. 2012, vol. 15, nº 11, Disponible en: <https://publicaciones.uci.cu/index.php/SC>. ISSN 2343.
- CROFT, W. B. Approaches to intelligent information retrieval. 1987, vol. 23, 249-254 p.
- CYRANEK, G. Greenstone: Un software libre de código abierto para la construcción de bibliotecas digitales [Consultado el: 14/03/2013 Disponible en: <http://www.unesco.org/new/en/>].
- DEFINICIÓNABC. Definición de HTML. [Consultado el: 15/02/2013 Disponible en: <http://www.definicionabc.com/tecnologia/html.php>].
- DEMPSEY, L. y HEERY, R. DESIRE - Development of a European Service for Information on Research and Education [Consultado el: 15/12/2012 Disponible en: <http://www.ukoln.ac.uk/metadata/desire/overview//overview.pdf>].
- DOAJ. Acceso Abierto a la literatura científica y académica. DOAJ -Directory of Open Access Journals. [Consultado el: 20/04/2013 Disponible en: <http://a-abierto.blogspot.com/2007/03/doaj-directory-of-open-access-journals.html>].
- GARCÍA, N. E. y CABALLERO, S. Metadatos: necesidad e importancia de integrar estándares. 2010,
- KENDALL, C. y PALEPU, A. Open Medicine. 2012, vol. 6, nº p. pp.59-61. [Consultado el: 02/03/2013]. Disponible en: <http://www.openmedicine.ca/>. ISSN 1911-2092
- LAPUENTE, M. J. L. Metadatos. Hipertexto: El nuevo concepto de documento en la cultura de la imagen [Consultado el: 21/01/2013 Disponible en: <http://www.hipertexto.info/documentos/metadatos.htm>].
- MAGNUSSON, H. Manual de PHP. 2013.
- MENDOZA, T. Análisis de DSpace como repositorio institucional [Consultado el: 13/01/2013 Disponible en: <http://es.scribd.com/doc/48250416/Analisis-de-DSpace-como-repositorio-institucional>].
- MONTALVÁN, D. P. y AGÜERO, D. N. Expediente de proyecto programa de mejora. 2010, [Consultado el: 12/05/2013].
- NETBEANS. Plataforma Netbeans [Consultado el: 26/02/2013 Disponible en: <http://netbeans.org/>].

Bibliografía referenciada

PÉREZ, J. E. Introducción a XHTML. España: 2008.

PKP. Open Journals Systems Disponible en: <http://pkp.sfu.ca/?q=ojs>.

PRESSMAN, R. S. Ingeniería del Software: Un Enfoque Práctico. Quinta Edición ed. McGraw-Hill Companies ed. 2002. ISBN 8448132149.

Protocolo HTTP. [Consultado el: 12 /10/2012 Disponible en: <http://www.ecured.cu/index.php/HTTP>.

Protocolo TCP/IP. [Consultado el: 06/12/2012 Disponible en: http://www.ecured.cu/index.php/Protocolo_TCP/IP.

RCCI. *Revista Cubana de Ciencias Informáticas*. 2013, vol. 7, nº Disponible en: <http://rcci.uci.cu/index.php/rcci>. ISSN 2227-1899.

REPISO, O. J. Nuevas perspectivas para la Catalogación: Metadatos versus MARC. [Consultado el: 20/11/2012 Disponible en: <http://redc.revistas.csic.es>.

SALINAS, A. G. *Z39.50 y OAI-PMH: Protocolos de Transferencia y Recuperación de Información*. 2010,

SÁNCHEZ, M. Definición del modelo del negocio y del dominio utilizando Razonamiento Basado en Casos. [Consultado el: 02/04/2013 Disponible en: <http://www.inf.udec.cl/~revista/ediciones/edicion8/Rbc.pdf>.

SERIE_CIENTÍFICA. Serie Científica de la Universidad de las Ciencias Informáticas. 2013, vol. 6, nº 3, Disponible en: <https://publicaciones.uci.cu/index.php/SC>.

SOLA, M. Greenstone: distribución de colecciones digitales [Consultado el: 25/01/2013 Disponible en: <http://www.ub.edu/bid/21/sola2.htm>.

STYLESHEET, C. Hojas de estilos en cascadas. [Consultado el: 05/04/2013 Disponible en: <http://asistencia.foroactivo.com/t19264-css-stylesheet>.

THOMSON, L. W. Y. L. Programación: Desarrollo web con PHP y Mysql. 2007.

TOLOSA, G. H. y BORDIGNON, F. R. A. Introducción a la Recuperación de Información. Conceptos, modelos y algoritmos básicos. 2008.

WILLINSKY, J. John. PKP|Public Knowledge Project [Consultado el: 05/10/2012 Disponible en: <http://pkp.sfu.ca/?q=harvester>.

Glosario de términos

Glosario de términos

Crosswalked: Se utilizan para proporcionar búsqueda y clasificación por campos equivalentes en esquemas múltiples, como los campos Título en Dublin Core.

CSS: Hojas de estilo en cascada.

HTML: Lenguaje de Marca de Hipertexto.

Indexación: Es una función central de las herramientas de Gestión Documental que consiste principalmente en adjuntar metadatos a los documentos.

Dublin Core: Forma abreviada para el Dublin Metadata Core Element Set (Conjunto Básico de Elementos de Metadatos).

Protocolo: Conjunto de reglas, secuencias, formatos de mensajes y procedimientos bien detallados que posibilitan la transferencia de datos entre dos o más sistemas de computación.

W3C: Comunidad internacional que desarrolla estándares Web.

MARC: Un conjunto de estándares para manipular datos bibliográficos legibles por ordenador.

CGI: Interfaz Común de Compuerta, medio de comunicación que emplea un servidor Web para enviar información.

Anexos

Historias de usuario

RF: Realizar la integración con redes sociales.

- HU: Realizar la integración con redes sociales.
- Descripción de la HU realizar la integración con redes sociales.

Historia de Usuario	
Código: 2	Nombre Historia de Usuario: Realizar la integración con redes sociales.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Media	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3
<p>Descripción: La HU inicia cuando el usuario decide acceder a las redes sociales desde el sistema. El usuario accede al bloque lateral izquierdo y selecciona la opción: Inicio -> Buscar. El sistema muestra una interfaz para realizar la búsqueda con el siguiente campo:</p> <ul style="list-style-type: none"> • Buscar (Opcional, Campo de texto) <p>El usuario introduce el criterio de búsqueda y presiona el botón "Buscar". El sistema le muestra una vista con los elementos encontrados acerca de la temática seleccionada. Para visualizar su contenido y acceder a las redes sociales el usuario debe seleccionar el vínculo Ver registro -> Compartir, que se encuentra ubicado al pie de la página. El usuario se posiciona sobre el vínculo Compartir y al pasar el cursor o al dar clic sobre dicha opción, el sistema le muestra una ventana con las redes sociales como: Facebook, Twitter, para que pueda realizar el acceso a las mismas. El sistema lo envía a la red social seleccionada, finalizando así la HU.</p>	
Observaciones:	
Prototipo de interfaz:	



RF: Crear colección personal simple.

- HU: Crear colección personal simple.
- Descripción de la HU Crear colección personal simple.

Historia de Usuario	
Código: 3	Nombre Historia de Usuario: Crear colección personal simple.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Díaz Dominguez	Iteración Asignada: 1
Prioridad: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3

Descripción: La HU inicia cuando un usuario decide crear una colección simple en el sistema. El usuario accede a la opción: “**Buscar**”, que se encuentra ubicada en la barra lateral izquierda. El sistema le muestra una interfaz con el siguiente campo:

- Buscar (Opcional, campo de texto)

El usuario introduce el criterio de búsqueda y presiona el botón “**Buscar**”. El sistema realiza la búsqueda según la temática introducida y muestra los resultados.

1.- Para crear una colección personal el usuario debe dar clic en el vínculo “**Ver registro**”, luego el sistema le muestra el contenido del registro. El usuario debe seleccionar el enlace “**Agregar a mis colecciones**”, ubicado en el pie de página. Al presionar dicho enlace el sistema muestra un formulario **Agregar a mi Colección Personal** con los siguientes campos:

- Colección (Opcional, lista desplegable).
- Nueva colección (Opcional, campo de texto).

El usuario introduce en el campo de texto el nombre de la nueva colección, luego presiona el botón “**Agregar**”. El sistema agrega la nueva colección creada a sus colecciones personales y la muestra en la barra lateral izquierda al desplegar **Mis colecciones**. En caso de querer deshacer la operación debe presionar el botón “**Cancelar**”, y el sistema cierra el formulario, finalizando así la HU.

Observaciones:

1.- El sistema debe verificar que para crear una colección se debe llenar el campo de texto “**Nueva colección**”. En caso contrario el sistema no permite que el usuario presione el botón “**Agregar**”.

Prototipo de interfaz:

The screenshot displays a document record with the following fields:

- Identificador:** http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2697_09
- Descripción:** Desarrollo de una aplicación web que permite optimizar el trabajo y la documentación referente al Proceso de Tesis de Grado en la Facultad 2, brinda la posibilidad de llevar un control durante el transcurso de cada periodo sobre cada una de las tesis y otros asuntos que afectan a los componentes del proceso. Esta aplicación es independiente del entorno de hardware mínimo, teniendo así que un gran
- Asunto:**
- Titulo:**

Overlaid on this is a modal window titled "Agregar a mi Colección Personal" with the following elements:

- Colección:** A dropdown menu currently showing "--Seleccione--". Below it is the instruction: "Seleccione una colección a la que enviará este documento".
- Nueva Colección:** A text input field containing the word "Nueva". Below it is the instruction: "Defina una nueva colección a la que enviará este documento".
- Buttons:** Two buttons labeled "Agregar" and "Cancelar".

At the bottom of the page, there are three navigation links: "»Agregar a mis colecciones", "»Exportar a Endnote", and "»Exportar a Bibtext".

RF: Crear colección personal avanzada.

- HU: Crear colección personal avanzada.
- Descripción de la HU Crear colección personal avanzada.

Historia de Usuario	
Código: 4	Nombre Historia de Usuario: Crear colección personal avanzada.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Díaz Dominguez	Iteración Asignada: 1
Prioridad: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3
<p>Descripción: La HU inicia cuando un usuario decide crear una colección avanzada en el sistema. El usuario accede a la opción: “Buscar”, que se encuentra ubicada en la barra superior. El sistema le muestra una interfaz con los siguientes campos:</p> <ul style="list-style-type: none"> • Todos (Opcional, campo de texto) • Archivos (Opcional, campo de texto) • Resumen (Opcional, campo de texto) • Contribuidor (Opcional, campo de texto) • Cobertura (Opcional, campo de texto) • Creador (Opcional, campo de texto) • Fecha (Opcional, lista desplegable) • Hasta (Opcional, lista desplegable) • Descripción (Opcional, campo de texto) • Formato (Opcional, campo de texto) • Identificador (Opcional, campo de texto) • Lenguaje (Opcional, campo de texto) • Editor (Opcional, campo de texto) • Relación (Opcional, campo de texto) • Derechos (Opcional, campo de texto) • Recurso (Opcional, campo de texto) • Asunto (Opcional, campo de texto) 	

- Título (Opcional, campo de texto)
- Tipo (Opcional, campo de texto)

El usuario introduce la información y presiona el botón **“Buscar”**. El sistema realiza la búsqueda, según los datos introducidos, y muestra los resultados. Para crear una colección personal el usuario da clic en el vínculo **“Ver registro”** que desee visualizar, el sistema le muestra el contenido del registro. El usuario selecciona el enlace **“Agregar a mis colecciones”**, ubicado en el pie de página del registro. Al presionar sobre el mismo el sistema le muestra al usuario un formulario **Agregar a mi Colección Personal** con los siguientes campos:

- Colección (Opcional, lista desplegable)
- Nueva colección (Opcional, campo de texto)

El usuario introduce en el campo de texto el nombre de la nueva colección, luego presiona el botón **“Agregar”**. El sistema agrega la nueva colección creada a sus colecciones personales y la muestra en la barra lateral izquierda al desplegar **Mis colecciones**. En caso de caso de querer cancelar debe presionar el botón **“Cancelar”**, se cierra el formulario, finalizando así la HU.

Observaciones:

1.- El sistema debe verificar que para crear una colección se debe llenar el campo de texto “Nueva colección”. En caso contrario el sistema no permite que el usuario presione el botón **“Agregar”**.

Prototipo de interfaz:

The screenshot displays a web interface with a record detail and a modal form. The record detail includes fields for Identificador, Descripción, Asunto, and Título. The modal form, titled "Agregar a mi Colección Personal", contains a dropdown menu for "Colección" (set to "--Seleccione--"), a text input for "Nueva Colección" (containing "Nueva"), and two buttons: "Agregar" and "Cancelar". Below the modal, there are links for "»Agregar a mis colecciones", "»Exportar a Endnote", and "»Exportar a Bibtex".

RF: Agregar registro a colección personal.

- HU: Agregar registro a colección personal.
- Descripción de la HU Agregar registro a una colección.

Historia de Usuario	
Código: 8	Nombre Historia de Usuario: Agregar registro a colección personal.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Alta	Puntos Estimados: 4
Riesgo en Desarrollo: Alto	Puntos Reales: 3
<p>Descripción: La HU inicia cuando un usuario decide agregar un registro a una colección personal en el sistema. El usuario accede a la opción: “Buscar”, que se encuentra ubicada en la barra lateral izquierda. El sistema le muestra una interfaz con el siguiente campo:</p> <ul style="list-style-type: none"> • Buscar (Opcional, campo de texto) <p>El usuario introduce el criterio de búsqueda y presiona el botón “Buscar”. El sistema realiza la búsqueda según la temática introducida y muestra los resultados. El usuario selecciona el enlace “Ver Registro”, del elemento que desee adicionar a la colección personal. Se selecciona el vínculo Agregar a mis colecciones, ubicado en el pie de página del registro. El sistema le muestra una vista con el formulario Agregar a mi Colección Personal, donde el mismo permite agregar un registro a una colección. El sistema le muestra una interfaz para que el usuario introduzca el nombre de una colección nueva a la cual se enviará el registro o selecciona una de las que han sido creadas para agregar el registro; a través de los siguientes metadatos asociados:</p> <ul style="list-style-type: none"> • Colección (Opcional, lista desplegable) • Nueva Colección (Opcional, campo texto) <p>El usuario completa los datos correspondientes y presiona el botón “Agregar”. El sistema almacena la información, finalizando así la HU.</p>	
<p>Observaciones:</p> <p>1.- El sistema debe verificar que para agregar registro a una colección personal se debe llenar el</p>	

campo de texto “**Nueva colección**” para crear una nueva colección a la que se envíe el registro o seleccionar una de las colecciones creadas. En caso de que el usuario elija crear una nueva colección para enviar el registro y se deja en blanco dicho campo, el sistema no le permite presionar el botón “**Agregar**”.

Prototipo de interfaz:

Descripción	El sistema brinda una amplia información sobre todo el tema referente a las tesis de grado, garantiza la centralización de los documentos generados entorno a este proceso, permite publicar y asignar los temas de tesis propuestos a los estudiantes de quinto año y mantener un seguimiento de la evaluación y evolución de las mismas. Esta aplicación presenta un requerimiento de hardware mínimo. Cuenta con una interfaz web amigable y fácil de usar que permite que un gran número de usuarios puedan disfrutar de sus servicios. Para el desarrollo de la aplicación se usa como metodología de desarrollo de software Rational Unified Process (RUP), lenguaje de modelado UML 2.0, herramienta Case Enterprise Architect 7.0, marco de trabajo Django, lenguaje de programación Python 2.6, Sistema Entero Integrado de Desarrollo Wing IDE.
Asunto	<div style="border: 1px solid gray; padding: 5px;"> <p style="text-align: center;">Agregar a mi Colección Personal</p> <p>Colección: <input type="text" value="--Selecione--"/></p> <p>Seleccione una colección a la que enviará este documento</p> <p>Nueva Colección:</p> <input type="text"/> Defina una nueva colección a la que enviará este documento. <p style="text-align: center;"><input type="button" value="Agregar"/> <input type="button" value="Cancelar"/></p> </div>
Título	al de gestion administrativa
<p>»Agregar a mis colecciones »Exportar a Endnote »Exportar a Bibtex</p>	

RF: Eliminar registro de colección personal.

- HU: Eliminar registro de colección personal.
- Descripción de la HU Eliminar registro de colección personal.

Historia de Usuario	
Código: 12	Nombre Historia de Usuario: Eliminar registro de colección personal.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yoevis Diaz Dominguez	Iteración Asignada: 1
Prioridad: Media	Puntos Estimados: 2

Riesgo en Desarrollo: Alto

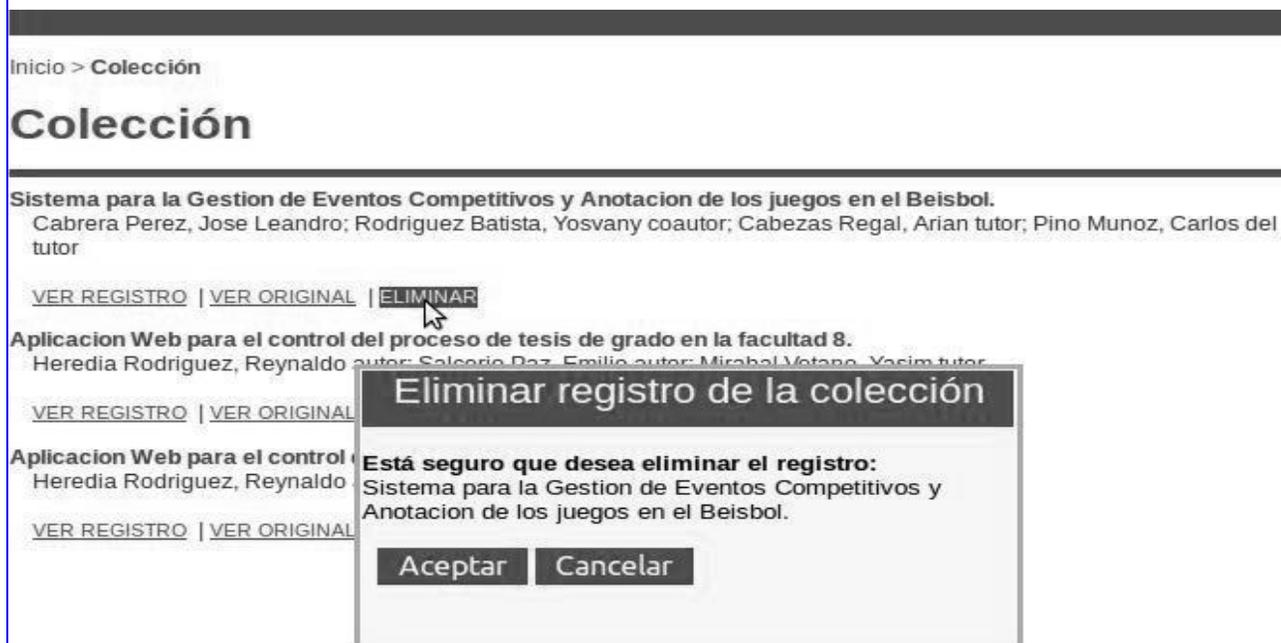
Puntos Reales: 2

Descripción: La HU inicia cuando un usuario decide eliminar un registro de una colección personal en el sistema. El usuario accede desde la barra lateral izquierda desplegando **Mis Colecciones**, seleccionando la colección que desea que se muestren sus registros. El sistema muestra en una página todos los registros que se le han agregado a dicha colección seleccionada, donde puede eliminar el registro. El sistema le muestra un mensaje de confirmación si desea realmente realizar esta operación. Al usuario presiona el botón **“Aceptar”**, se elimina el registro del sistema. En caso de deshacer dicha operación, se presiona el botón **“Cancelar”** y el sistema cierra el formulario, finalizando así la HU.

Observaciones:

1.- Debe existir al menos un registro en la colección personal.

Prototipo de interfaz:



Casos de pruebas de aceptación.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Realizar la integración con redes sociales correctamente.	Permite la integración con redes sociales.	El sistema envía al usuario a la página de la red social seleccionada.	<p>1.- El usuario accede a la opción: “Buscar”, que se encuentra ubicada en la barra lateral izquierda.</p> <p>2.- El usuario introduce el criterio de búsqueda en el campo de texto y presiona el botón “Buscar”.</p> <p>3.- Presiona el enlace “Ver registro”.</p> <p>4.- El usuario pasa el cursor de mouse por encima o da clic en el enlace “Compartir”; ubicado en el pie de página del registro, se muestra una ventana con las redes sociales como: Facebook, Twitter.</p>

HU2_CP2 Prueba de funcionalidad para realizar la integración con redes sociales.

Escenario	Descripción	Nueva Colección	Respuesta del sistema	Flujo central
EC 1.1 Crear colección personal simple correctamente.	Permite crear una colección personal simple.	V Nombre	El sistema le muestra en la parte superior de la barra lateral izquierda en “Mis colecciones” , la nueva colección creada.	<p>1.- El usuario accede a la opción: “Buscar”, que se encuentra ubicada en la barra lateral izquierda.</p> <p>2.- Se introduce el criterio de búsqueda en el campo de texto y presiona el botón “Buscar”.</p> <p>3.- Presiona el enlace “Ver</p>

EC 1.2 Crear colección personal simple incorrectamente.	No permite crear una colección personal simple de forma incorrecta.	I	El sistema no crea la colección personal simple.	registro”. 4.- Se selecciona el enlace “Agrega a mis colecciones”, ubicado en el pie de página del registro. Mostrándose un formulario para introducir en un campo de texto el nombre de la nueva colección personal; luego debe presionar en el botón “Agrega”, para que se cree la colección.
		Vacío		

HU3_CP3 Prueba de funcionalidad para crear colección personal simple.

Escenario	Descripción	Nueva Colección	Respuesta del sistema	Flujo central
EC 1.1 Crear colección personal avanzada correctamente.	Permite crear la colección personal avanzada.	V	El sistema le muestra en la parte superior de la barra lateral izquierda en “Mis colecciones”, la nueva colección creada.	1.- El usuario accede a la opción: “Buscar”, que se encuentra ubicada en la parte superior derecha. 2.- Se introducen los datos correspondientes de cada metadato solicitado para hacer la búsqueda, luego presiona el botón “Buscar”.
		Nombre		
EC 1.2 Crear colección personal avanzada incorrectamente.	Se muestra un formulario donde el usuario debe introducir datos.	I	Al dejarse el campo de texto vacío, el sistema no le permite al usuario presionar el botón “Agrega”, para	3.- Al mostrarse los resultados, el usuario presiona el enlace “Ver registro”.
		Vacío		

			enviar información.	4.- El usuario selecciona dando clic en el enlace “ Agregar a mis colecciones ”, ubicado en el pie de página del registro. Mostrándose un formulario para introducir en un campo de texto el nombre de la nueva colección personal; luego debe presionar en el botón “ Agregar ”, para que se cree la colección.
--	--	--	---------------------	--

HU4_CP4 Prueba de funcionalidad para crear colección personal avanzada.

Escenario	Descripción	Nuevo nombre	Respuesta del sistema	Flujo central
EC 1.1 Agregar registro a colección personal correctamente.	Permite agregar un registro a una de las colecciones creadas o crear una nueva donde enviar el registro.	V Nombre	El sistema agregar el registro a la colección personal seleccionada o enviar el registro a una nueva que se cree.	1.- El usuario accede a la opción: “ Buscar ”, que se encuentra ubicada en la barra lateral izquierda. 2.- El usuario introduce el criterio de búsqueda en el campo de texto y presiona el botón “ Buscar ”. 3.- Presiona el enlace “ Ver registro ”.
EC 1.2 Agregar registro a colección personal incorrectament	No permite agregar un registro a la colección personal.	I Vacío	El sistema no le permite al usuario presionar el botón Agregar , por lo que no se	4.- El usuario selecciona el enlace “ Agregar a mis colecciones ”, ubicado en el pie de página del registro. 5.- Al dar clic en el enlace, se

e.			agrega el registro.	<p>muestra un formulario.</p> <p>6.- El usuario debe seleccionar una de las colecciones creadas o crear una nueva colección a la que desee enviar el registro.</p> <p>7.- Presiona el botón "Agregar", para que se agregue el registro.</p>
----	--	--	---------------------	--

HU8_CP8 Prueba de funcionalidad para agregar registro a colección personal.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar registro de colección personal correctamente.	Permite eliminar un registro de una colección personal.	El sistema elimina el registro de la colección personal seleccionada.	<p>1.- El usuario accede a la parte superior de la barra lateral izquierda del sistema al enlace "Mis colecciones".</p> <p>2.- Al desplegar "Mis colecciones", se selecciona la colección para visualizar sus registros.</p> <p>3.- Se presiona el enlace "Eliminar".</p> <p>4.- Se muestra una interfaz para eliminar el registro. El usuario presiona el botón "Aceptar".</p>
EC 1.2 Cancelar operación.	Permite cancelar la operación.	El sistema cierra el formulario y regresa a	1.- El usuario accede a la parte superior de la barra lateral izquierda del sistema

		la interfaz anterior.	al enlace “ Mis colecciones ”. 2.- Al desplegar “ Mis colecciones ”, se selecciona la colección para visualizar sus registros. 3.- Se presiona el enlace “ Eliminar ”. 4.- Se muestra una interfaz para eliminar el registro. El usuario presiona el botón “ Cancelar ”.
--	--	-----------------------	---

HU12_CP12 Prueba de funcionalidad para eliminar registro de colección personal.