



Universidad de las Ciencias Informáticas
Facultad 2

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



*Componente para la Gestión de las Referencias Bibliográficas
en el sistema ABCD 3.0*



Autor: Maydelin Alvarez Morales

Tutores: Msc. Eliana Bárbara Ril Valentin

Ing. Gleidis Y. Rosabal Espinosa

La Habana, junio del 2018



"La única lucha que se pierde es la que se abandona"

Ernesto "Che" Guevara

Declaración de autoría

Declaro ser la única autora Maydelin Alvarez Morales de la presente tesis que tiene por título: "Componente para la Gestión de las Referencias Bibliográficas en el sistema ABCD 3.0" y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Maydelin Alvarez Morales

Firma del autor

Msc. Eliana Bárbara Ril Valentin

Firma del tutor

Ing. Gleidis Y. Rosabal Espinosa

Firma del tutor



Datos del contacto

Síntesis de los tutores

MSc. Eliana Bárbara Ril Valentin: graduada en el año 2008 como Ingeniera en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Máster en Gestión de proyecto informáticos. Se desempeña como profesora principal de 3er año de la facultad 2. Imparte clases de Ingeniería de *software* (ISW) en el Dpto de ISW en dicha facultad.

Correo electrónico: ebril@uci.cu

Ing. Gleidis Yuriannis Rosabal Espinosa: graduada en el año 2011 como Ingeniera en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Se desempeña como especialista B en ciencias Informáticas en el Departamento de Aplicaciones del Centro CIGED, ocupando el rol de analista del proyecto Repositorio Institucional.

Correo electrónico: gyrosabal@uci.cu



Dedicatoria

Le dedico esta tesis a una mujer que amo con la vida, por quien me levanto cada mañana. A esa mujer que creyó en mí siempre, que ha sido mi guía y maestra de la vida, mi amiga incondicional. A esa mujer que ha vivido este sueño cada segundo junto conmigo, y a quien admiro y respeto. Con todo el amor del mundo esto es para ti Mimi.

A Lachy por ser mi padre, a pesar de que la ciencia diga lo contrario para mi eres mi papá, no tengo palabras para agradecer lo que has hecho por mí. Gracias por luchar porque cada segundo de mi vida sea feliz y por caminar siempre a mi lado, por eso este logro es para ti papá.

A mi papá por tu apoyo, cariño incondicional y comprensión para cumplir mi sueño, porque a pesar de la distancia has estado siempre para mí, gracias padre.

A mis hermanos por darme fuerzas para seguir adelante. Para ellos es este triunfo y que les sirva de ejemplo para sus vidas.

A mi bisabuela por su cariño y mimos, aunque ahora está enferma y en ocasiones no se acuerde de mí, sé que le encantaría estar aquí conmigo disfrutando este éxito y estaría orgullosa de mí. Hoy te dedico mi tesis.

A mis abuelos por quererme más allá de los límites, por sus incansables sacrificios, por consentirme todos mis caprichos, por los valores que me inculcaron y han hecho de mí la mejor persona que puedo ser. Por eso esto es para ustedes.

A mi madrina por ser mi ejemplo de mujer luchadora e inteligente, por hacerme sentir especial, por su dedicación y paciencia, por enseñarme el valor de las cosas, por ser infinitamente mi modelo de persona a seguir. Por ser mi segunda madre. Este logro también es para ti.



Agradecimientos

Empiezo mis agradecimientos citando la frase del Che "La única lucha que se pierde es la que se abandona" pues me siento identificada con ella porque nunca pensé que llegaría tan lejos, me propuse hace 5 años que sería ingeniera y hoy lo he logrado por tal motivo quiero agradecer a todas las personas que de una forma u otra contribuyeron a que mis sueños se hicieran realidad.

A mis padres, no me alcanzan las palabras para expresarles cuánto les agradezco y estoy orgullosa de ellos. Por darme siempre las palabras de apoyo cuando las cosas parecen derrumbarse. A ellos, que me mostraron el camino a seguir y no vacilaron ni un segundo en reprenderme en los momentos en que flaqueo. A ellos le debo todo y cuanto soy hoy.

A mi hermana por ser la mejor hermana que podría tener, por ser mi amiga, aunque siempre nos peleamos te adoro y espero que estés orgullosa de mí.

A mi hermano mi bebé, mi hermanito menor a pesar de tu corta edad eres todo un hombrecito para mí.

A mi abuelo Rolando por confiar en mí desde el principio de la carrera siempre dejándome claro que sí lo lograría, por tus consejos y tu preocupación.

A mi Abuela Isa y mi abuelo Chachy por consentirme todos mis caprichos, los Adoro.

A mi Abuela Lucia por su cariño incondicional por sus consejos. Te quiero.

A mi tío Adiam más que mi tío eres mi hermano mayor, siempre cuidándome, consintiéndome y apoyándome en todo. Haz sido testigo de tantos años de sacrificios y por eso sé que estás orgulloso de tu sobrina Te quiero.

A Yadelkis eres muy especial para mí no solo eres la mujer de mi papa, eres mi amiga, mi familia, gracias por tu dedicación por tu apoyo en todos estos años, por ser una excelente madre, gracias.

A Laidé mi tía de tan corta edad, quien se robó mi cariño y aprecio por su forma de ser y ocurrencias. Te agradezco todo lo que haces por mi familia siempre preocupándote por todos, estoy muy orgullosa de que seas parte de mí y que lleves en tu vientre el mayor regalo que podría tener.



A mi tía Eliset no tengo palabras para agradecer todo lo que has hecho por mí, a pesar de la distancia siempre has estado presente en mi vida.

A mis tíos y primos, por sus consejos, por mantener siempre unida a la familia más bella del mundo. Los quiero mucho.

A Ainadis por ser mi amiga, mi hermana desde la primaria siempre has estado ahí para mí. Entre nosotras nada es tuyo, ni mío, todo es nuestro (sobre todo los problemas), por darme la oportunidad de ser madrina, gracias por todo.

A mis hermanitas Sheila Luaces y Sheila Laguardia las quiero muchísimo, gracias por su apoyo, su preocupación, por ser parte de todas mis aventuras, siempre hemos estado en las buenas y en las malas juntas y así seguiremos.

A la gran familia universitaria, mis amigos, compañeros y profesores por ser mi familia durante estos 5 años.

A Suri por ser mi compañera de cuarto, mi amiga incondicional por soportar todas mis malcriadeces y abrazarme en los momentos más difíciles siempre apoyándome en todo, por ser mi consejera, por ser como mi hermana. Agradezco tanto tu apoyo y dedicación. Espero siempre tenerte a mi lado.

A Osmani mi compañero de travesuras, mi amigo leal, el seguidor de todas mis locuras hoy nos hacemos ingenieros y empezara una nueva etapa de nuestras vidas, te agradezco mucho por toda tu paciencia y dedicación fue un placer compartir estos años contigo. Por complacerme en todo y estar siempre para mí. Muchas gracias mi hermano.

A Maikel por tu ayuda, tu comprensión durante todo este tiempo y mucho más en la recta final, que aun cuando él también estaba en el proceso de llegar a la meta tuvo tiempo para mí, gracias.

A Mildrey por ser una persona muy especial para mí, por ser la mujer que eres emprendedora que nunca te das por vencida, por confiar en mí siempre, por tus consejos y protección.

A Jose Ernesto por ser una maravillosa persona por ayudarme tanto, sin él no estaría donde estoy. Gracias.

A Yaniselis por ser una de las mejores profesoras en mi paso por la universidad, por ser una amiga, por tu ayuda y apoyo. Muchas gracias.



A mis tutoras gracias por su apoyo y ayuda en todo momento, por compartir conmigo sus conocimientos, por estar siempre que las necesite. Espero que estén orgullosas de mí.

A Gleidis por ser una maravillosa persona, por su paciencia por estar a mi lado desde el primer corte, por siempre tener tiempo para mí. Gracias.

A Baby eres una excelente profesora desde que fuiste mi profesora te cogí mucho cariño y siempre dije que serías mi tutora te agradezco mucho tu esfuerzo.

A Dailián muchas gracias por tu dedicación, siento mucho que no hallas podido estar aquí conmigo.

A Alberto que ha sido un tutor para mí, siempre ayudándome, gracias por tu paciencia y dedicación.

A Basilio mi oponente, más que un oponente has sido un tutor para mí, gracias por tu apoyo.

Al tribunal muchas gracias por sus consejos, siempre ayudándome en especial a Zoila te agradezco mucho tu tiempo, tus consejos. Gracias.

A Fidel, a la Revolución y a la UCI por darme la oportunidad de materializar mi sueño y formarme profesionalmente.

A todas las personas que de una forma u otra contribuyeron a mi formación profesional y a que este sueño se hiciera realidad.

A todos GRACIAS.



Resumen

Las referencias bibliográficas son el conjunto de datos y elementos que describen de forma detallada una publicación. La búsqueda y gestión de referencias bibliográficas es una parte fundamental de todo proceso investigativo. En este sentido, el Centro de Informatización de la Gestión Documental de la Universidad de las Ciencias Informáticas cuenta con el proyecto Automatización de Bibliotecas y Centros de Documentación que actualmente desarrolla el sistema de Automatización de Bibliotecas y Centros de Documentación en su versión 3.0. Este sistema brinda un conjunto de servicios de almacenamiento, gestión y distribución de materiales digitales disponibles para los miembros de una determinada institución.

Entre las limitantes que el sistema posee se encuentra que no permite gestionar las referencias bibliográficas de los documentos consultados, sin tener que hacer uso de gestores bibliográficos externos para gestionar correctamente la bibliografía, o realizarlo de forma manual. Esto provoca que el proceso de gestión de las referencias bibliográficas se pueda realizar con errores. Por lo antes mencionado se desarrolló un componente, el cual permite a los usuarios que consultan y realizan búsquedas en el catálogo en línea, extraer los datos de los registros bibliográficos para conformar la referencia bibliográfica; estas referencias estarán guardadas en un formato de fichero que podrá ser consultado mediante cualquier navegador web o editor de texto plano. Para el desarrollo de este componente se utilizó como metodología de desarrollo AUP_UCI, como lenguaje de modelado UML y la herramienta CASE Visual Paradigm, como lenguaje de desarrollo Java, Eclipse Kepler para desarrollar la propuesta de solución y como base de datos no relacional J-ISIS.

Palabras Claves: componente, estilo bibliográfico, gestor bibliográfico, referencia bibliográfica, registro bibliográfico.



Abstract

Bibliographic references are the set of data and elements that describe in detail a publication. The search and management of bibliographic references is a fundamental part of any investigative process. In this sense, the Computerization Center of Documentary Management of the University of Informatics Sciences has the project called Automation of Libraries and Documentation Centers that the system for library management ABCD 3.0 is currently developing. This system provides a set of storage services, management and distribution of digital materials which are available to members of a particular institution. The fact that it does not allow managing the bibliographic references of the documents consulted, without having to use external bibliographic managers to correctly manage the bibliography, or to do it manually is among the limitations the system has. This causes that the process of management of the bibliographic references can be carried out with errors. For the aforementioned, a component for the ABCD 3.0 System was developed, which allows users who consult and perform searches in the online catalog, extract the data from the bibliographic records to conform the bibliographic reference; these references will be stored in a file format that can be accessed through any web browser or plain text editor. For the development of this component, we used AUP_UCI as a development methodology, UML as a modeling language and the CASE Visual Paradigm tool, as a java development language, Eclipse Kepler to develop the solution proposal and J-ISIS as a non-relational database.

Keywords: bibliographic manager, bibliographic reference, bibliographic record, bibliographic style, component.



Índice de Contenido

| | |
|---|----|
| Capítulo 1: Fundamentación teórica sobre la gestión de referencias bibliográficas..... | 5 |
| 1.1. Conceptos asociados al tema..... | 5 |
| 1.2. Sistemas que utilizan un componente para la gestión de referencias bibliográficas..... | 12 |
| 1.3. Metodología, herramientas y lenguajes utilizados..... | 15 |
| 1.3.1. <i>Proceso Unificado Ágil (AUP)</i> | 15 |
| 1.3.2. <i>Lenguaje de programación</i> | 16 |
| 1.3.3. <i>Entorno de Desarrollo Integrado</i> | 17 |
| 1.3.4. <i>Lenguaje Unificado de Modelado (UML)</i> | 17 |
| 1.3.5. <i>Herramienta de modelado</i> | 18 |
| 1.3.6. <i>Sistema Gestor de Base de Datos</i> | 19 |
| 1.4. Conclusiones parciales..... | 19 |
| Capítulo 2: Descripción del componente Referencias Bibliográficas..... | 20 |
| 2.1. Descripción de la propuesta de solución..... | 20 |
| 2.2. Modelo conceptual..... | 21 |
| 2.2.1. <i>Conceptos del modelo conceptual</i> | 21 |
| 2.3. Requisitos del <i>software</i> | 22 |
| 2.3.1. <i>Requisitos funcionales</i> | 22 |
| 2.3.2. <i>Requisitos no funcionales</i> | 23 |
| 2.4. Definición de los actores y casos de uso..... | 24 |
| 2.4.1. <i>Actores del sistema</i> | 24 |
| 2.4.2. <i>Casos de usos del sistema</i> | 25 |
| 2.4.3. <i>Diagrama de caso de uso del sistema</i> | 25 |
| 2.4.4. <i>Descripción de los casos de uso</i> | 25 |
| 2.5. Modelo de diseño..... | 30 |



| | |
|---|----|
| 2.5.1. Diagrama de clases de diseño..... | 30 |
| 2.5.2. Diagrama de colaboración..... | 31 |
| 2.6. Arquitectura del componente | 33 |
| 2.7. Patrón de diseño..... | 35 |
| 2.8. Conclusiones parciales | 37 |
| Capítulo 3: Implementación y validación del componente Referencias Bibliográficas | 38 |
| 3.1. Modelo de implementación | 38 |
| 3.1.1. Estándares de codificación | 38 |
| 3.2. Diagrama de despliegue | 39 |
| 3.3. Pruebas del <i>software</i> | 40 |
| 3.4. Estrategias de pruebas | 40 |
| 3.5. Niveles de prueba..... | 41 |
| 3.6. Prueba de sistema..... | 41 |
| 3.6.1. Resultado de las pruebas de sistema del componente Referencia Bibliográfica..... | 48 |
| 3.7. Pruebas de unidad..... | 49 |
| 3.8. Pruebas de aceptación | 55 |
| 3.9. Conclusiones parciales | 56 |
| Conclusiones generales..... | 57 |
| Recomendaciones | 58 |
| Referencias bibliográficas | 59 |
| Bibliografía..... | 63 |
| Anexos..... | 66 |
| Anexos 1. Entrevista | 66 |
| Anexos 2. Prueba de aceptación..... | 67 |



Índice de tablas

| | |
|--|----|
| Tabla 1: Formatos que importan los gestores de referencias bibliográficas | 8 |
| Tabla 2: Compatibilidad de los estilos bibliográficos con los gestores de referencia | 12 |
| Tabla 3: Descripción del actor del sistema | 24 |
| Tabla 4: Descripción del caso de uso Crear Referencia Bibliográfica..... | 25 |
| Tabla 5: Descripción del caso de uso Exportar Referencia Bibliográfica | 27 |
| Tabla 6: Descripción del caso de uso Enviar Referencia Bibliográfica por correo electrónico | 29 |
| Tabla 7: Prueba de caja negra del caso de uso Crear referencias bibliográficas | 44 |
| Tabla 8: Prueba de caja negra del caso de uso Exportar referencias bibliográficas | 45 |
| Tabla 9: Prueba de caja negra del caso de uso Enviar referencias bibliográficas por correo electrónico ... | 46 |
| Tabla 10: Descripción de las variables..... | 47 |
| Tabla 11: Clasificación de las No conformidades del componente Referencias Bibliográficas | 49 |
| Tabla 12: Caso de prueba de caja blanca para la ruta 1 | 53 |
| Tabla 13: Caso de prueba de caja blanca para la ruta 2 | 53 |
| Tabla 14: Caso de prueba de caja blanca para la ruta 3 | 54 |
| Tabla 15: Caso de prueba de caja blanca para la ruta 4 | 54 |
| Tabla 16: Caso de prueba de caja blanca para la ruta 5 | 54 |



Índice de figuras

| | |
|---|----|
| Figura 1: Descripción de la propuesta de solución | 20 |
| Figura 2: Modelo conceptual | 21 |
| Figura 3: Diagrama de casos de uso del componente Referencias Bibliográficas..... | 25 |
| Figura 4: Diagrama de clases de diseño del caso de uso Crear Referencia..... | 31 |
| Figura 5: Diagrama de clases de diseño del caso de uso Exportar Referencia | 31 |
| Figura 6: Diagrama de clases de diseño del caso de uso Enviar Referencia por correo electrónico | 31 |
| Figura 7: Diagrama de colaboración del requisito funcional Crear referencias bibliográficas | 32 |
| Figura 8: Diagrama de colaboración del requisito funcional Exportar referencias bibliográficas | 32 |
| Figura 9 :Diagrama de colaboración del requisito funcional Enviar referencias bibliográficas por correo electrónico | 33 |
| Figura 10: Arquitectura en n Capas | 34 |
| Figura 11: Diagrama de despliegue del componente Referencias Bibliográficas | 39 |
| Figura 12: Pruebas funcionales..... | 42 |
| Figura 13: Método de prueba Caja negra..... | 43 |
| Figura 14: Resultados de las pruebas de caja negra aplicadas al componente Referencias Bibliográficas | 48 |
| Figura 15: Fragmento de código del método make_Bibliografía..... | 51 |
| Figura 16: Representación del grafo de flujo de camino básico del método make_Bibliografía | 51 |



Introducción

Actualmente con el desarrollo de las Tecnologías de la Información y las Telecomunicaciones (TIC) las bibliotecas han tenido mayor auge en la búsqueda de información en formato digital a través de los Sistemas Integrados de Gestión Bibliotecaria (SIGB por sus siglas en español). Los SIGB son una herramienta tecnológica que permiten automatizar los procesos y servicios bibliotecarios más comunes. Típicamente abarcan la catalogación en línea, la circulación, la consulta y la adquisición de materiales (Moreno y Alonso 2014).

Específicamente para la consulta de materiales se utilizan los catálogos en línea, también conocidos como OPAC (del inglés *Online Public Access Catalog*). OPAC hace posible que las bibliotecas puedan ofrecer servicios de información y materiales que se encuentren en línea electrónica durante las 24 horas. Para las instituciones es importante utilizar los catálogos en línea pues permiten la búsqueda, la consulta y la visualización de los registros bibliográficos (Montoya 2015).

Los catálogos están compuestos por los registros bibliográficos. Los registros bibliográficos son una construcción descriptiva estructurada y normalizada creada para proporcionar al usuario la información bibliográfica de un documento al que se puede acceder de alguna forma desde la biblioteca. Desde los registros bibliográficos se pueden obtener referencias bibliográficas, pues contiene una descripción de todos los elementos y detalles considerados esenciales para identificar un documento, una publicación o una parte de ella. Las inclusiones de las referencias bibliográficas en un trabajo científico deben elaborarse con sumo cuidado y cumpliendo los requisitos técnicos implantados. Su adecuada preparación revela no solo rigor profesional, sino también la tenencia de una cultura general esencial para cualquier investigador de una materia, interesado en divulgar sus hallazgos o experiencias (Fernández 2009).

Para la realización de un trabajo científico, el usuario tiene a su disposición decenas de bases de datos, fuentes de información, miles de sitios web, repositorios, blogs, documentos a texto completo, archivos digitalizados de todo tipo, pero se encuentra con la dificultad de la organización y optimización de la información desde el punto de vista de la investigación. Con el objetivo de resolver los problemas asociados a la gestión, recopilación, sistematización e integración se crean los gestores de referencias bibliográficas.

Un gestor bibliográfico forma parte de una de las herramientas de mayor utilidad para estudiantes, investigadores y profesionales de la documentación pues a través de ellos, se garantizan la rigurosidad y el carácter científico de una publicación. El uso de dichos gestores supone gran ahorro de tiempo, porque



facilita el manejo de grandes cantidades de referencias de una manera eficaz, también permiten la inserción automática de citas y la aplicación de diferentes formatos para generar bibliografías (Boretto 2018).

En muchas universidades se usan herramientas para la gestión de las referencias bibliográficas (Chile 2014; Cadiz 2018; Rivera y León 2013; Biblioteca Universitaria 2015; Alicante 2018). En el caso de la Universidad de las Ciencias Informáticas (UCI), específicamente en el centro de desarrollo para la Informatización de la Gestión Documental (CIGED por sus siglas en español) se cuenta con el proyecto de Automatización de Bibliotecas y Centros de Documentación (ABCD). En el mismo se encuentra el sistema ABCD 3.0. La función principal de dicho sistema es la gestión bibliotecaria, pues ofrece herramientas para la gestión de bases de datos, entrada de datos, estadísticas, la circulación, la catalogación, control de publicaciones periódicas, funciones de búsqueda, entre otras funcionalidades. El sistema tiene además un módulo de catálogo en línea, el que permite gestionar los procesos correspondientes a la gestión de materiales, incluyendo consultas, lo que dinamiza el acceso de los usuarios a la hora de realizar las búsquedas (Medina 2015).

Una de las limitaciones identificadas en este sistema es que con frecuencia los usuarios realizan consultas referentes a temas específicos, las cuales son útiles para su investigación. Sin embargo, el resultado de estas búsquedas no puede ser generado, ni exportado a través del OPAC hacia los gestores de referencia bibliográfica existentes, por lo que el usuario debe realizar esta gestión de forma manual. Esta situación provoca que el proceso de la gestión de las referencias bibliográficas se pueda realizar con errores, ya que no existe un buen aprovechamiento de los datos a la hora de insertarlos. Además, puede perderse información relevante de los campos al no introducirlos de la forma en que lo exige cada una de las normas. Y por último, se convierte dicho proceso aún más lento, engorroso y complicado cuando son varias las fuentes consultadas. Tampoco incluye una funcionalidad que permita el envío de las referencias bibliográficas, previamente conformadas por el usuario hacia el correo electrónico. Esto provoca que al crear las referencias bibliográficas de un material específico no se pueda guardar en ese momento y se pierda la información referenciada de los sitios buscados.

Por lo anteriormente planteado, se propone el siguiente **problema a resolver**: ¿Cómo contribuir al proceso de gestión de las referencias bibliográficas en el catálogo en línea del sistema de Automatización de Bibliotecas y Centros de Documentación en su versión 3.0?

El **objeto de estudio** que se identifica es: el proceso de gestión de las referencias bibliográficas.



Para la solución de este problema se traza como **objetivo general**: Desarrollar un componente para realizar la gestión de las referencias bibliográficas en el OPAC. Enmarcado en la gestión de las referencias bibliográficas para el catálogo en línea de sistemas informáticos como **campo de acción**.

Para el cumplimiento del objetivo general se definen las siguientes **tareas de la investigación**:

1. Elaboración de los fundamentos teóricos metodológicos de la investigación relacionados con la gestión de las referencias bibliográficas y soluciones informáticas que las implementan.
2. Análisis de los requisitos asociados al negocio de la propuesta de solución para el sistema ABCD v3.0.
3. Diseño de los requisitos asociados al negocio de la propuesta de solución para el sistema ABCD v3.0.
4. Desarrollo del componente para la gestión de las referencias bibliográficas en el sistema ABCD 3.0.
5. Validar la propuesta de solución a partir de los métodos de pruebas definidos en la investigación.

En el transcurso de la investigación se utilizaron los siguientes **métodos teóricos**:

1. Analítico-Sintético: para descomponer el problema de investigación en elementos separados y profundizar en el estudio de cada uno de ellos, para luego dar la solución propuesta.
2. Inductivo-Deductivo: para la identificación de la problemática y de las soluciones.
3. Modelación: para con la herramienta Visual Paradigm realizar los diferentes diagramas que propone la metodología de desarrollo de software seleccionada.

Además, se empleó la entrevista como **método empírico**. Fue realizada a los especialistas y estudiantes del proyecto ABCD para entender la problemática (ver anexo 1).

La presente investigación está estructurada en tres capítulos, organizándose la información contenida de la siguiente manera:

Capítulo 1: Fundamentación teórica sobre la gestión de referencias bibliográficas. Se incluye el estudio conceptual de los principales elementos que se deben tener en cuenta para el desarrollo del trabajo y una



descripción detallada del proceso de gestión de las referencias bibliográficas, con la finalidad de comprender el negocio. Se presenta un estudio de sistemas similares para la gestión de referencias bibliográficas y las principales características de los gestores bibliográficos. Se determina la metodología, herramientas y tecnologías a utilizar en el desarrollo de la solución propuesta.

Capítulo 2: Descripción del componente para la gestión de referencias bibliográficas. Se realiza una caracterización del componente a desarrollar, así como también de las funcionalidades a implementar. Se define el modelo de dominio, los requisitos funcionales y no funcionales, así como los actores y casos de usos del sistema. Además, se realiza la descripción de los casos de usos propuestos.

Capítulo 3: Implementación y validación del componente para la gestión de referencias bibliográficas. Se presentan, el modelo de despliegue y los componentes que conforman la solución. Se realiza la validación de la misma, exponiendo el resultado final de las pruebas realizadas. Además se muestran los estándares de codificación utilizados para la solución del componente.



Capítulo 1: Fundamentación teórica sobre la gestión de referencias bibliográficas

En este capítulo se abordan los conceptos fundamentales para comprender el desarrollo del componente propuesto. Se explica que son las referencias bibliográficas, los gestores bibliográficos, estilos bibliográficos, así como un análisis de distintas herramientas para la gestión de referencias bibliográficas. Se estudian las características esenciales de los estilos seleccionados para regir dichas referencias. Se investiga acerca de los sistemas que implementan componentes para la gestión de referencias bibliográficas donde se enfatiza en sus características favorables y sus limitantes. Por último, se definen metodologías, tecnologías y herramientas a utilizar en la posterior implementación del sistema.

1.1. Conceptos asociados al tema

A continuación, se relacionan algunos de los conceptos relacionados a la investigación realizada, para permitir una mayor comprensión del funcionamiento de la propuesta de solución apoyando la investigación en curso.

Referencias bibliográficas

Según (Flacso 2010), las referencias bibliográficas “son un conjunto de indicaciones precisas y detalladas que permiten la identificación de un documento”. Los datos esenciales que identifican un documento son: el autor, título, fecha, editorial, lugar, URL, páginas, formato.

La función de un gestor de referencia bibliográfica es gestionar de forma automatizada las referencias bibliográficas, por lo que a continuación se expresa una definición del mismo.

Gestores bibliográficos

A su vez (Castañeda 2017), los gestores de referencias bibliográficas “son una aplicación informática de base de datos que permiten buscar y organizar de forma ordenada las fuentes de información en cualquier área, y exportarlas a manera de listas de referencias y citas”.

Basados en la norma ISO 690, los gestores permiten organizar la información de varias fuentes bibliográficas e incluso de la web, creando así la bibliografía de consulta para anexarla directamente al



documento con el que se trabaja. Entre las ventajas de usar un gestor bibliográfico se destacan la facilidad para integrar citas y bibliografías en el texto evitando errores en los datos. Además, la organización de la información por medio de bibliotecas personalizadas que pueden ser compartidas con otros usuarios y almacenadas por grupos temáticos manteniendo siempre una secuencia de orden. También los gestores permiten elaborar bibliografías y listas de lecturas recomendadas en un formato estándar. Asimismo sirven para desarrollar de forma automática un catálogo de referencias bibliográficas de los documentos citados y las notas a pie de página al final de un documento de texto (Gallegos, Peralta y Guerrero 2017).

Existen variadas herramientas para la gestión de las referencias bibliográficas, actualmente las principales son: *Zotero*, *Mendeley*, *RefWorks*, *EndNote*, *Reference Manager*, *BibText*, *RefMAN*, *CiteUlike* y *Rebase* (Navarro et al. 2013).

Después de haber analizado cada una de las características de las herramientas anteriormente mencionadas se ha optado por presentar una descripción de cinco gestores de uso gratuito que por sus funcionalidades y versatilidad gozan de gran aceptación en la actualidad; estos son *Zotero*, *Endnote*, *Mendeley*, *Reference Manager* y *CiteUlike*.

Zotero: Es un software de gestión bibliográfica gratuito, de acceso libre. Es producido por el Centro para la Historia y Nuevos Medios de Comunicación de la Universidad George Mason (GMU) y se distribuye bajo la Licencia Pública General Affero GNU¹. Sus características más importantes son: se puede integrar en la web, permite la sincronización de varios equipos en línea, genera citas, notas al pie y bibliografía en el texto, y se puede integrar con los procesadores *MS Word*, *LibreOffice/OpenOffice*. Entre sus funciones permite exportar e importar distintos formatos: RDF, MODS XML, Bibtext, RIS, TXT, Consulta/Biblix, Dublin, Endnote, Refer. Reconoce todos los estilos de mayor popularidad como *Chicago*, *MLA*, *APA*, *Vancouver*,

¹ La **licencia pública general de Affero** es una licencia derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red. La razón de esta licencia es proteger a las personas que creen un servicio en la red y publiquen su código, de manera que si otra persona publica un servicio basado en ese código esté obligada a publicar también sus fuentes, y no se pueda aprovechar sobre el servicio inicial, mejorándolo sin compartirlo.



entre otros (Carreño 2014).

Mendeley: Es un software de gestión bibliográfica gratuito de acceso libre. Permite extraer los metadatos de los documentos PDF, sincronizar automáticamente su biblioteca a través de escritorio web y móvil. Ofrece la opción de descargar la aplicación en iPhone y iPad. Permite integrarse con los procesadores de *MS Word*, *LibreOffice/OpenOffice* y *LaTeX* desarrollado por la editorial Elsevier y con licencia propia. Mendeley importa documentos de varias bases de datos tales como *Google Scholar* o *Scopus*. También exporta e importa ficheros bibliográficos de otros software como: *Zotero*, *RefWorks* y *EndNote*. Además, exporta e importa al formato *Bibtex*, *RIS* y archivos XML y añade directamente desde el escritorio a la biblioteca archivos PDF y carpetas (Carreño 2014).

Endnote: Es un gestor de versión gratuita desarrollado por la empresa *Thompson Reuters*. (Castañeda 2017). Posee entre sus funcionalidades importar referencias guardadas o bajadas desde un CD-ROM o desde una base de datos bibliográfica en nuestro navegador. Además, importa ficheros: *BibTeX*, *CSA*, *Endnote/Refer/BibIX*, *ISI*, *Medline*, *Ovid*, *PubMed*, *RIS*, *SciFinder*. Incorpora estilos bibliográficos muy utilizados en el ámbito académico y de la investigación como: *APA*, *Chicago*, *Harvard*, *MLA*. Posee formatos de archivo de listas de referencias: *HTML*, *RTF*, texto plano y otros como *clipboard* y *XML*. Se integra con los procesadores de texto *Microsoft Word*, *Open Office* y *Libre Office writer* (Cordón García, Martín Rodero y Alonso Arévalo 2009).

CiteUlike: Es un gestor de referencias bibliográficas gratuito que permite almacenar, organizar y compartir los artículos que esté leyendo. Es un servicio web donde la colección de referencias está guardada en el servidor de *CiteUlike*, de forma que garantiza el acceso desde cualquier ordenador con conexión a Internet. *CiteUlike* dispone de un perfil público que todos los usuarios pueden ver y al que puede acceder cualquier usuario incluso sin necesidad de registrarse. En este perfil se puede navegar por las referencias de otros investigadores, consultar los documentos más populares, hacer listas de seguimiento y copiar las referencias a nuestra biblioteca. Además, exporta e importa y crea bibliografía a formatos de salida: *RIS*, *Bibtex*, *TXT*, *RTF* (Rodríguez y Gutiérrez, 2011).

Reference Manager: Es un gestor de referencias comercializado por *Thomson Reuters*, diseñado para trabajar sobre el sistema operativo *Windows*. Exporta referencias directamente desde diversas bases de datos, como *Current Contents*, *Science Direct*, *ISI Web of Science*, *EBSCOhost* y *Discovery*. Para aquellos casos en los que no es posible la exportación/importación directa, permite importar ficheros *RIS*, *TXT* y *HTML*. Incorpora múltiples estilos de citas y nos permite crear y compartir cualquier estilo personalizado.



Nos permite gestionar localmente las referencias: crear bibliografías, realizar búsquedas locales, detectar duplicados, establecer sinónimos de palabras clave, nombres de autores y revistas, personalizar campos, personalizar vistas, establecer vínculos con Pdf, direcciones web (Vidal, 2010).

A continuación, se muestra la tabla 1 donde se evidencian los distintos formatos que importan los gestores de referencias bibliográficas anteriormente expuestos :

Tabla 1: Formatos que importan los gestores de referencias bibliográficas

| Gestores | Formato de archivo que importan los gestores de referencias bibliográficas | | | | |
|-------------------|--|------|-----|-----|-----|
| | TXT | HTML | RTF | XML | RIS |
| Zotero | ✓ | | | | ✓ |
| Mendeley | ✓ | | | ✓ | ✓ |
| EndNote | ✓ | ✓ | ✓ | ✓ | ✓ |
| CiteUlike | ✓ | | ✓ | | ✓ |
| Reference Manager | ✓ | ✓ | | | ✓ |

El análisis de los gestores bibliográficos permitió seleccionar los formatos: TXT, HTML, RTF y RIS. Estos formatos se escogen por la compatibilidad y similitud que existe con algunos de los gestores bibliográficos para manejar las referencias bibliográficas. El uso de estos formatos proporciona gran ventaja al componente a desarrollar pues la mayoría de los gestores importan a los formatos antes mencionados. A continuación, se describen los formatos de archivos antes mencionados.



Formatos de archivos

La extensión TXT (del inglés *textfile*) es un archivo informático que estructura series de líneas de texto. Los archivos TXT sirven como almacenes de información a la vez que evitan las complicaciones propias de otros formatos de archivo. Los archivos TXT son universales debido a que cualquier procesador de texto puede leerlos. Estos archivos pueden utilizar *Unicode* para facilitar su lectura a usuarios de distintos idiomas. Estos archivos están compuestos de bytes que representan caracteres ordinarios como letras, números y signos de puntuación también incluye algunos caracteres de control como tabulaciones, saltos de línea y retornos de carro². Los archivos de texto plano carecen de información destinada a generar formatos (negritas, subrayado, cursivas, tamaño, entre otros) y tipos de letra (por ejemplo, Arial, Times, Courier, entre otras). Esta simplicidad permite que una gran variedad de programas pueda leer y editar ese contenido (Corel 2018).

Hypertext Markup Language (por sus siglas en inglés HTML), estos archivos se implementan en su mayoría en forma de páginas estáticas de sitios web. HTML es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (*hyperlinks*) que conducen a otros documentos o fuentes de información relacionadas. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, entre otros) así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) (Corel 2018).

Los archivos de la extensión RTF (del inglés *Rich text format*) son documentos de texto que se pueden crear, abrir, ver, editar o modificar con el uso de aplicaciones de procesamiento de textos populares para los equipos basados en *Windows Mac* y *Microsoft*. Los archivos RTF proporcionan a los usuarios un apoyo de compatibilidad cruzada amplia, que es el objetivo principal para el desarrollo de la tecnología de formato de texto. Además, estos archivos .rtf se pueden abrir, visualizar y utilizar con aplicaciones de bases de datos. Las imágenes, tablas o gráficos pueden ser introducidas en los archivos RTF. Los documentos de texto guardados en la extensión .rtf tienen soporte para muchos tipos de estilos o formatos. Estos archivos

² Retorno de carro: es uno de los caracteres de control de la codificación *ASCII* o *Unicode*, que hace que se mueva el cursor a la primera posición de una línea.



pueden utilizar el formato de texto básico que incluye la negrita, cursiva y subrayado, así como cambios de fuente, color, tamaño y alineación del texto. (Corel 2018).

RIS (del inglés *Research Information Systems*) es un formato de archivo para representar las citas bibliográficas. El contenido de estos archivos RIS suelen incluir el título, el material consultado, la fecha de publicación, el autor, el nombre de la editorial, la página de inicio y final, así como el número de la edición del material, entre otros datos de interés que se pueden añadir por el usuario. Los archivos con extensión RIS fueron creados para estandarizar el intercambio de información de citas que comúnmente se hace entre las bibliotecas digitales.

Los gestores bibliográficos para exportar las referencias bibliográficas utilizan diferentes estilos bibliográficos, a continuación se expresa la definición del mismo.

Estilos bibliográficos

Según (Flacso 2010), un estilo bibliográfico “es el conjunto de normas que indican la información que se ha de incluir, así como el orden en que debe presentarse al citar cuando se elabora un texto y hacer una bibliografía”.

A pesar de haber estilos internacionales de redacción de bibliografías no hay un único estilo adoptado globalmente. Un investigador o estudiante cuando selecciona el estilo bibliográfico lo hace en dependencia del trabajo a realizar, dichos estilos pueden venir recomendados por el propio profesor al que se le debe entregar el trabajo o simplemente estén predefinidos por instituciones o universidades a la que pertenecen. Entre la gran diversidad de estilos existentes se presentan a continuación los más utilizados y compatibles con los gestores estudiados en el epígrafe anterior son:

Harvard: Este estilo se desarrolló en la Universidad de Harvard, Estados Unidos en los años 50. Es utilizado en Física y en las Ciencias Naturales y Sociales. Este estilo conocido también como sistema de autor-fecha permite citar la fuente en el cuerpo principal del texto al señalar para cada referencia en particular, el apellido(s) del autor, el año de edición y opcionalmente la página o páginas citadas entre paréntesis (Alicante 2005).

MLA: El estilo Asociación de Lengua Moderna de América (del inglés *Modern Language Association of America*, por sus siglas MLA). Es utilizado para las humanidades e indica no sólo el estilo para hacer las citas sino también el estilo para escribir: tipografía, tamaño, calidad de papel, formato de párrafos, de



enlaces, puntuación, especialmente para los escritos de lenguas modernas, crítica literaria, escritos culturales (Manager y Wendling 2012). Esquema básico utilizado para citar publicaciones en este estilo es: nombre y apellidos, el título, la ciudad editorial, el año de publicación, el tipo de medio de la publicación, la versión y la dirección electrónica.

APA: Es el estilo de citas bibliográficas oficial de la Asociación Psicológica de América (del inglés *America Psychological Association*, por sus siglas APA). Fue creado 1929 y establece el formato para todo tipo de citas y documentos en Psicología y Ciencias Sociales (Manager y Wendling 2012). La forma básica para representar las referencias bibliográficas según el estilo APA es: el autor, el año de publicación, el título, el lugar de publicación y la editorial.

Vancouver: Es el estilo de los Requisitos de Uniformidad para Manuscritos Enviados a Revistas Biomédicas. Toma su nombre de una reunión celebrada en Vancouver (Canadá) en 1978, que llevó a la creación del Comité Internacional de Editores de Revistas Médicas (ICMJE por sus siglas en inglés). Este fue desarrollado por la Biblioteca Nacional de Medicina de los EE.UU., cuya versión “debe ser considerado como el estilo autoritario”, según la Asociación Médica Británica (BMA por sus siglas en inglés) (Ortiz Chacha 2013). El Esquema general de las referencias del estilo Vancouver son: autor(es) del artículo, título del artículo, título de la revista, fecha de publicación [fecha de actualización, fecha de consulta], volumen (número), localización, disponibilidad y notas.

Chicago: Es el estilo más antiguo, su historia se remonta desde 1891, es utilizado principalmente en materias de historia, ciencias sociales, arte, musicología y literatura (Ortiz Chacha 2013). Chicago es toda una normativa de estilo de edición, las recomendaciones tratan de puntuaciones, organización del texto, pie de página, notas y citas. La estructura básica de las referencias bibliográficas en el estilo Chicago son: el autor, el título, el editor, el traductor, el volumen y la edición.

ISO-690: Es una norma de la Organización Internacional de Normalización (del inglés *International Organization for Standardization* por sus siglas ISO) que especifica los elementos que deben ser incluidos en las referencias bibliográficas de materiales publicados, como monografías y publicaciones seriadas, capítulos, artículos de publicaciones seriadas (como revistas y diarios), materiales cartográficos, grabaciones sonoras, fotografías, obras audiovisuales y documentos de patentes (Córdova 2015). Las especificaciones de los metadatos del estilo ISO-690 son: el autor, el título, tipo de medio, edición lugar de publicación, fecha de actualización, fecha de citación, disponibilidad y número normalizado.



A continuación, se muestra la tabla 2 donde se evidencia la compatibilidad de los estilos estudiados con los gestores de referencias bibliográficas anteriormente expuestos.

Tabla 2: Compatibilidad de los estilos bibliográficos con los gestores de referencia

| Gestores | Estilos bibliográficos | | | | |
|-------------------|------------------------|-----|-----------|---------|---------|
| | MLA | APA | Vancouver | Harvard | ISO-690 |
| Zotero | ✓ | ✓ | ✓ | ✓ | ✓ |
| Mendeley | ✓ | ✓ | ✓ | ✓ | ✓ |
| Endnote | ✓ | ✓ | ✓ | ✓ | ✓ |
| CiteUlike | ✓ | ✓ | ✓ | ✓ | ✓ |
| Reference Manager | ✓ | ✓ | ✓ | ✓ | ✓ |

La selección de los estilos bibliográficos aporta gran utilidad al componente a desarrollar pues la mayoría de los gestores bibliográficos utilizan estos estilos para crear las referencias bibliográficas.

1.2. Sistemas que utilizan un componente para la gestión de referencias bibliográficas

Se realizó una investigación a este tipo de sistemas y se llegó a la conclusión de que los que más se asemejan a la solución del componente Referencias Bibliográficas son los siguientes:



Citeproc-Java

Es un procesador CSL (del inglés *Citation Style Language*) para Java. Interpreta los estilos de CSL y genera citas y bibliografías. Citeproc-Java está licenciado bajo la Licencia Apache, Versión 2.0. Algunos de los aspectos más destacados en Citeproc-Java son (Krämer 2016):

- Admite una amplia gama de formatos de archivos como html, txt, rtf.
- Implementa varios estilos bibliográficos como IEEE, APA, Vancouver, ISO-690, MLA, Harvard, entre otros.
- Utiliza la herramienta de línea de comandos para ejecutar la biblioteca sin configurar un entorno de desarrollo.
- Contiene conectores remotos para Mendeley y Zotero que le permiten utilizar sus recursos en línea como entrada para citas o bibliografías.

Esta biblioteca a pesar de ser recomendada para generar las referencias bibliográficas no permite el envío de las mismas por correo electrónico.

Google Académico

Es un sistema que permite buscar bibliografía especializada de una manera sencilla. Desde un solo sitio se pueden realizar búsquedas en un gran número de disciplinas y fuentes como por ejemplo: estudios revisados por especialistas, tesis, libros, resúmenes y artículos de fuentes como editoriales académicas, sociedades profesionales, depósitos de impresiones preliminares, universidades y otras organizaciones académicas. Además, ayuda a encontrar el material más relevante dentro del mundo de la investigación académica.

Google Académico ordena los resultados de las búsquedas por orden de relevancia así al igual que sucede con las búsquedas web en Google, las referencias más útiles aparecerán al inicio de la página. La tecnología de ranking de Google toma en consideración el texto completo de cada artículo, así como el autor, dónde fue publicado y con qué frecuencia ha sido citado en otras fuentes especializadas (Ruiz 2007).

Google Académico posee un componente que tiene características esenciales como por ejemplo: permite la exportación de citas a formatos Bibtex, RIS y EndNote. Como toda plataforma presenta restricciones, en



este caso es un software cerrado por lo que no se tiene acceso a su código fuente. Además, no permite el envío de las referencias por correo electrónico.

Sistema para repositorios digitales REPXOS 3.0

Es un sistema desarrollado por el centro CIGED, donde se centraliza la documentación digital que contiene la producción científico-técnica e intelectual de la UCI. Dicho sistema cuenta con varias funcionalidades y módulos que permiten realizar búsquedas sobre los documentos que almacena, como por ejemplo el módulo Referencias Bibliográficas. Este módulo permite al usuario que acceda a REPXOS 3.0 gestionar las referencias bibliográficas sin tener que utilizar métodos manuales para conformar las mismas. Entre los múltiples beneficios que aporta el módulo se encuentra: permitir el envío de las referencias bibliográficas por correo electrónico, la exportación y generación de las mismas sin tener que hacer usos de gestores bibliográficos externos. Implementa una gran gama de estilos bibliográficos y formatos de archivos como son XML, BibTeX, RIS, RTF, TXT y HTML, que son adaptables a cualquier institución independientemente de su campo de especialización (Guevara y Soler 2015). El módulo Referencias Bibliográficas tiene como dificultad que no permite exportar las referencias bibliográficas sin que antes no fuesen creadas.

Sistema Integrado de Gestión Bibliotecaria

Este SIGB está basado en Koha³. Fue desarrollado por el Centro de Informatización Universitaria en la UCI. Entre las características favorables que presenta este sistema se encuentran, que permite exportar las referencias bibliográficas a varios formatos de archivos de los más utilizados mundialmente, además estas referencias pueden ser enviadas por correo electrónico a un usuario determinado (Rivera 2012). Una de sus limitantes es que tiene una gama reducida de estilos bibliográficos para la conformación de las referencias bibliográficas.

Conclusiones sobre las soluciones informáticas existentes

Al concluir el estudio de las soluciones informáticas existentes para la gestión de referencias bibliográficas, se constató que los sistemas actuales no cumplen todas las características que requiere el componente a implementar, por lo que se hace necesario desarrollar un componente que satisfaga las necesidades de los

³ Koha es un SIGB bajo licencia pública que funciona tanto en Linux como en Windows.



usuarios de OPAC para la generación de referencias bibliográficas. No obstante, se utilizarán algunas funcionalidades de los sistemas Citeproc, REPXOS 3.0 y SIGB, las mismas son: crear referencias bibliográficas, exportar las referencias a un archivo (RIS, TXT, HTML y RTF) y enviar las referencias bibliográficas por correo electrónico.

1.3. Metodología, herramientas y lenguajes utilizados

Para la realización de todo sistema informático se hace uso de las herramientas y tecnologías existentes. La correcta selección de la misma va en dependencia de la calidad del producto final, del equipo de desarrollo, o del propio sistema en sí.

El componente Referencias Bibliográficas se debe integrar al sistema ABCD 3.0, por lo que esto trae consigo que se utilizarán las pautas, técnicas, herramientas y metodología definidas para dicho sistema.

1.3.1. Proceso Unificado Ágil (AUP)

Las Metodologías de Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto. Su principal función es guiar a los desarrolladores al crear un nuevo proyecto, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del mismo (Carrillo Perez 2008).

La metodología de desarrollo de software *Agil Unified Process* (por sus siglas *AUP*) para la UCI, es una variación de la metodología ágil AUP creada por la universidad.

Fases Variación AUP-UCI (Rodríguez 2014):

- 1. Inicio:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- 2. Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño,



se implementa y se libera el producto. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software.

- 3. Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Descripción de las disciplinas:

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 8 disciplinas, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos, Análisis y Diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación y la disciplina Despliegue se considera opcional. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI-DEV v1.3 para el nivel 2, serían Gestión de la configuración (CM), Planeación de proyecto (PP) y Monitoreo y control de proyecto (PMC) (Rodríguez 2014).

Para la realización de este trabajo se empleó el escenario No 2 de la metodología AUP_UCI. Este escenario se aplica para proyectos que modelen el negocio con Modelo Conceptual (MC) solo pueden modelar el sistema con Caso de Uso del Sistema (CUS), el mismo plantea:

Escenario No 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información (Rodríguez 2014).

1.3.2. Lenguaje de programación

Un lenguaje de programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora. Cada lenguaje posee sus propias sintaxis. También se puede decir que un programa es un conjunto de órdenes



o instrucciones que resuelven un problema específico basado en un lenguaje de programación (Ramírez 2010).

Java es un lenguaje de programación creado por Sun Microsystems, (empresa que posteriormente fue comprada por Oracle) para poder funcionar en distintos tipos de procesadores. Su sintaxis es muy parecida a la de C ó C++, e incorpora como propias algunas características que en otros lenguajes son extensiones: gestión de hilos, ejecución remota, etc. El código Java, una vez compilado, puede llevarse sin modificación alguna sobre cualquier máquina, y ejecutarlo. Esto se debe a que el código se ejecuta sobre una máquina hipotética o virtual, la *Java Virtual Machine*, que se encarga de interpretar el código (ficheros compilados .class) y convertirlo a código particular de la CPU que se esté utilizando (siempre que se soporte dicha máquina virtual) (All 2012).

Java es el lenguaje de programación del lado del servidor que se va a utilizar para el desarrollo del componente. Su uso permitió la programación de las clases asociadas a la capa lógica de negocio, almacenamiento y presentación.

1.3.3. Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés *Integrated Development Environment*), es un entorno de programación empaquetado como un programa de aplicación, consiste en un editor de código, compilador, depurador y constructor de interfaz gráfica de usuario. Los IDEs pueden ser aplicaciones por sí solos o pueden ser parte de aplicaciones existentes (Jimenez 2012). En esta investigación se utilizará específicamente Eclipse Kepler.

1.3.4. Lenguaje Unificado de Modelado (UML)

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño (Orallo 2010).

UML es un lenguaje estándar diseñado para visualizar, especificar, construir y documentar *software* orientado a objetos (Cinca, Callén y Molinero 2005). Sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas *software* como para la arquitectura *hardware* donde se ejecuten.



Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto (Orallo 2010).

1.3.5. Herramienta de modelado

Las herramientas de Ingeniería de Software Asistida por Computadoras (CASE por sus siglas en inglés *Computer Aided Software Engineering*) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software que facilitan al usuario el modelado de procesos. Dentro de estas herramientas se encuentran Visual Paradigm for UML, Rational Rose, Easy CASE y CASE Studio siendo la primera una de las más utilizadas (Paradigm 2010).

Visual Paradigm for UML es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Las ventajas que proporciona Visual Paradigm for UML son: (Paradigm 2014)

- **Dibujo:** Facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- **Corrección sintáctica:** Controla que el modelado con UML sea correcto.
- **Coherencia entre diagramas:** Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- **Integración con otras aplicaciones:** Permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.



1.3.6. Sistema Gestor de Base de Datos

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional y de código abierto, distribuido bajo la licencia BSD⁴ y con su código fuente disponible libremente. Se ejecuta en los sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc. Presenta documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios. Además, brinda soporte nativo para los lenguajes: Java, PHP, C, C++, Perl, Python, entre otros. Admite todas las características de una base de datos profesional (triggers, procedimientos almacenados, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas.) (Machado y Benítez 2016). En esta investigación se utilizará específicamente como base de datos relacional PostgreSQL v9.3. Además se hará uso de la herramienta J-ISIS, como base de datos no relacional.

1.4. Conclusiones parciales

En este capítulo se identificaron los principales conceptos asociados con las referencias bibliográficas y gestores bibliográficos. Se seleccionaron RIS, TXT, HTML y RTF como formatos de archivos y MLA, APA, Harvard, Vancouver y ISO-690 como estilos bibliográficos para la generación de las referencias bibliográficas. El estudio del estado del arte sirvió para identificar y caracterizar sistemas similares al componente, teniendo en cuenta ventajas y limitantes, escogiendo las características que más se ajustan a las condiciones actuales del sistema ABCD 3.0. Se estableció la metodología AUP-UCI específicamente el escenario número 2. Se eligieron las siguientes herramientas para el desarrollo del componente Referencias Bibliográficas: UML para el lenguaje de modelado, Visual Paradigm v8.0 como herramienta para modelar los procesos del negocio, el lenguaje de programación Java, Eclipse Kepler para implementar la propuesta de solución, J-ISIS como sistema gestor de base de datos no relacional, Postgre SQL v 9.4 como sistema gestor de base de datos relacional.

⁴ La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD, un tipo del sistema operativo. Es una licencia de software libre permisiva.



Capítulo 2: Descripción del componente Referencias Bibliográficas

En el presente capítulo se describe la propuesta de solución del componente Referencia Bibliográficas para el sistema ABCD 3.0. Se realiza la confección del modelo conceptual, también se definen los requerimientos funcionales y no funcionales del sistema. Se diseñan los casos de uso del sistema con sus descripciones textuales, también se muestran los diagramas de clases del diseño, diagrama de colaboración e interfaces de usuario.

2.1. Descripción de la propuesta de solución

El componente Referencia Bibliográfica debe aportar al sistema ABCD 3.0 un conjunto de funcionalidades que les permitan a los usuarios realizar una serie de acciones para lograr la gestión de las referencias bibliográficas. Estas funcionalidades permitirán a los usuarios crear las referencias bibliográficas de un documento o de una lista de documentos consultados que son considerados útiles para las investigaciones.

Para realizar el procedimiento el usuario seleccionará un estilo bibliográfico para la extracción y conversión de los metadatos de los documentos. Los estilos disponibles para conformar las referencias son los siguientes: Harvard, MLA, APA, Vancouver, Chicago, ISO-690.

Además, el componente podrá exportar las referencias bibliográficas a un formato de archivo. Los formatos de archivos que incluye el componente son: RIS, TXT, HTML y RTF. Igualmente, el componente podrá enviar las referencias bibliográficas previamente conformadas mediante el correo electrónico a un destinatario determinado por el usuario.



Figura 1: Descripción de la propuesta de solución



2.2. Modelo conceptual

El modelo conceptual (también conocido como modelo de dominio) se utiliza para capturar y expresar el entendimiento en un área bajo análisis como paso previo al diseño de un sistema, además de ser utilizado como un medio para comprender el sector de negocios al cual el sistema va a servir, tomándose como el punto de partida para el diseño del sistema (Pressman 2010).

El negocio que abarca esta investigación no tiene los procesos bien definidos, lo que propicia un modelado conceptual en lugar del negocio. El objetivo de este es capturar, comprender y describir las clases más importantes dentro del contexto del sistema. El modelo conceptual, como se muestra a continuación en la figura 2, ayuda a los usuarios a utilizar un vocabulario común.

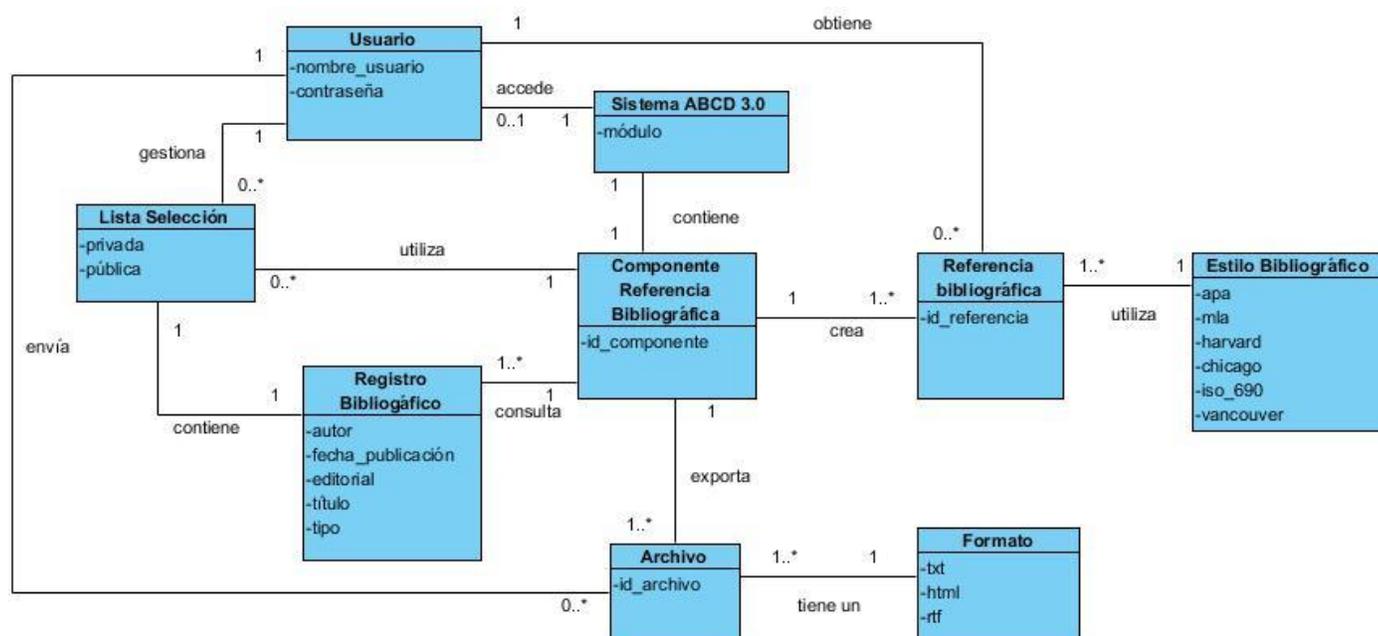


Figura 2: Modelo conceptual

2.2.1. Conceptos del modelo conceptual

Los conceptos del modelo conceptual describen textualmente las clases identificadas durante el modelado del dominio del problema. Estos conceptos sirven como glosario de términos y se muestran a continuación:

Usuario: es la persona que realiza el proceso de crear, exportar y enviar las referencias bibliográficas .



Sistema ABCD 3.0: es el sistema encargado de informatizar los procesos y servicios que el mismo brinda.

Componente Referencia Bibliográfica: es el componente que se integrará al sistema ABCD 3.0, permitirá crear y exportar un archivo que contiene las referencias bibliográficas.

Referencia bibliográfica: es la referencia creada con los metadatos extraídos de un documento, seleccionando un estilo bibliográfico.

Estilo bibliográfico: son los estilos que se podrán seleccionar para conformar las referencias.

Registro bibliográfico: es el documento al cual se le extraerán los metadatos que dependerán del estilo bibliográfico que sea seleccionado.

Archivo: es el archivo que se exporta según el formato que se seleccione. Estos archivos se pueden enviar por correo electrónico si así el usuario lo desea.

Formato: es el formato de archivo determinado por el usuario para exportar las referencias bibliográficas.

2.3. Requisitos del software

Los requerimientos para un software “son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas”. Estos requerimientos reflejan la necesidad de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos. Los requisitos se pueden clasificar en requisitos funcionales, no funcionales o como requerimientos del dominio (Sommerville 2005).

2.3.1. Requisitos funcionales

Un requisito funcional “define una función del sistema de software o sus componentes...”. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requisitos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir. Los requerimientos de comportamiento para cada requisito funcional se muestran en los casos de uso. Los requisitos funcionales establecen los comportamientos del sistema (Sommerville 2005). A continuación, se describen los siguientes requisitos funcionales (RF):



RF1. Crear referencia bibliográfica de un documento: El usuario crea las referencias bibliográficas, a partir de un documento, seleccionando un estilo de referencia que pueden ser Harvard, MLA, APA, Vancouver, Chicago, ISO-690.

RF2. Crear referencia bibliográfica de una lista de documentos: El usuario crea las referencias bibliográficas, a partir de una lista de documentos que fue creada, seleccionando un estilo de referencia que pueden ser Harvard, MLA, APA, Vancouver, Chicago, ISO-690.

RF3. Exportar referencia bibliográfica: El usuario tiene la opción de exportar las referencias bibliográficas conformadas a un archivo en los formatos RIS, TXT, HTML y RTF.

RF4. Enviar referencias bibliográficas por correo electrónico: El sistema permite que el usuario seleccione las referencias bibliográficas creadas con anterioridad y las envíe por correo electrónico.

2.3.2. Requisitos no funcionales

Los requerimientos no funcionales “son propiedades o cualidades que el producto debe tener, debe pensarse en propiedades que hacen al producto atractivo, usable, rápido o confiable. Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este, como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento” (Sommerville 2005). A continuación, se definen los siguientes requisitos no funcionales (RNF):

RNF1. Usabilidad: Por cada mensaje de error del sistema debe existir una correspondencia entre el fallo real ocurrido y el texto del mensaje mostrado.

RNF2. Operabilidad: Los campos de cada formulario se podrían recorrer en orden de izquierda a derecha.

RNF3. Confiabilidad: Las funcionalidades implementadas deberán cumplir con el manejo de excepciones de modo que se facilite la interpretación de las fallas ocurridas.

RNF4. Madurez: El sistema impone campos obligatorios para garantizar la integridad de la información que se introduce.

RNF5. Disponibilidad: El sistema debe garantizar el acceso a la información en todo momento siempre y cuando los servidores J-ISIS y Postgre se encuentren en funcionamiento.



RNF6. Requerimientos del Software: El servidor donde se instalará la aplicación debe cumplir con los siguientes requisitos: Sistema Operativo: GNU/Linux, Máquina Virtual de Java (JVM) v1.7 y PostgreSQL v9.3 o superior.

RNF7. Requerimientos del Hardware: La PC cliente debe de contar con una tarjeta de red. La PC servidor debe poseer un microprocesador Intel Core i3-4005 a una velocidad de 1.70 GHz, 4GB de memoria RAM y un espacio libre en disco duro de 20 GB para la instalación del sistema. El sistema debe tener una arquitectura 64 bit.

RNF8. Diseño e implementación: Se usará el lenguaje de programación Java. Las herramientas que se utilizan son de distribución bajo licencias libres.

2.4. Definición de los actores y casos de uso

Un actor es una idealización de una persona externa, de un proceso, o de una cosa que interactúa con un sistema, un subsistema, o una clase. Un actor caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema. En tiempo de ejecución, un usuario físico puede estar limitado a los actores múltiples dentro del sistema. Diferentes usuarios pueden estar ligados al mismo actor y por lo tanto pueden representar casos múltiples de la misma definición de actor (Ivar Jacobson 2000).

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un Caso de Uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema (Sparks 2004).

2.4.1. Actores del sistema

A continuación, se describe el actor del sistema del componente propuesto.

Tabla 3: Descripción del actor del sistema

| Actor | Descripción |
|---------|---|
| Usuario | Es la persona que posee permisos para crear o exportar las referencias bibliográficas. Además podrá enviar las referencias bibliográficas por correo electrónico. |



2.4.2. Casos de usos del sistema

A continuación, se describen los casos de uso.

- **CU1.** Crear referencia bibliográfica.
- **CU2.** Exportar referencias bibliográficas.
- **CU3.** Enviar referencias bibliográficas por correo electrónico.

2.4.3. Diagrama de caso de uso del sistema

A continuación, se muestra el diagrama de caso de uso del sistema.

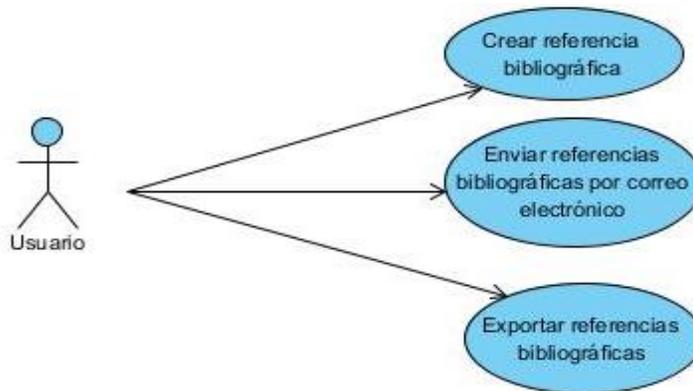


Figura 3: Diagrama de casos de uso del componente Referencias Bibliográficas

2.4.4. Descripción de los casos de uso

La descripción de casos de uso permite describir paso a paso el orden de los eventos que los actores siguen para completar un proceso mediante el sistema. A continuación, se muestra un resumen de los casos de uso del sistema.

Tabla 4: Descripción del caso de uso Crear Referencia Bibliográfica

| | |
|-----------------|---|
| Objetivo | Crear las referencias bibliográficas de las listas de los documentos seleccionados |
| Actores | Usuario (inicia): crea las referencias bibliográficas de las listas de los documentos seleccionados escogiendo un estilo bibliográfico. Los estilos |



| | | |
|--|---|--|
| | bibliográficos son Harvard, MLA, APA, Vancouver, Chicago, ISO-690. | |
| Resumen | El caso de uso inicia cuando el usuario selecciona el estilo para crear las referencias bibliográficas de los documentos seleccionados. El caso de uso finaliza con una notificación para informar al usuario que las referencias fueron creadas correctamente. | |
| Complejidad | Media. | |
| Prioridad | Media. | |
| Precondiciones | El usuario tiene que estar autenticado en el sistema. La lista de documentos debe estar creada. | |
| Postcondiciones | Las referencias bibliográficas fueron creadas correctamente. | |
| Flujo de eventos | | |
| Flujo básico Sección 1: Crear las referencias bibliográficas de las listas de selección | | |
| | Actor | Sistema |
| 1. | Selecciona las listas a las cuales se les desea crear las referencias bibliográficas. | |
| 2. | | Permite seleccionar el estilo bibliográfico para crear las referencias bibliográficas. Los estilos son: -Harvard -MLA -APA -Vancouver -Chicago -ISO-690 |
| 3. | | Permite crear las referencias bibliográficas a partir de un documento. |
| 4. | Selecciona el estilo bibliográfico y presiona Aceptar. | |
| 5. | | Valida la selección y crea la referencia bibliográfica. |
| Flujos alternos | | |
| 3.a. Crear las referencias bibliográficas de los documentos seleccionados | | |
| | Actor | Sistema |
| 1. | Selecciona el documento al cuales se le desea | |



| | | |
|---------------------------|--|--|
| | crear las referencias bibliográficas. | |
| 2. | | Permite seleccionar el estilo bibliográfico para crear las referencias bibliográficas. Los estilos son: -Harvard -MLA -APA -Vancouver -Chicago -ISO-690 |
| 3. | Selecciona el estilo bibliográfico y presiona Aceptar. | |
| 4. | | Valida la selección y crea la referencia bibliográfica. |
| 5.a. Campos vacíos | | |
| | Actor | Sistema |
| 1. | | Muestra un mensaje de error: "Existen campos vacíos obligatorios", señalando dichos campos. |

Tabla 5: Descripción del caso de uso Exportar Referencia Bibliográfica

| | |
|------------------------|---|
| Objetivo | Exportar referencias bibliográficas |
| Actores | Usuario (inicia): exporta las referencias bibliográficas a un formato de archivo determinado por el usuario. Los formatos que pueden ser seleccionados son RIS, TXT, HTML y RTF. |
| Resumen | El caso de uso inicia cuando el usuario crea las referencias bibliográficas de los documentos seleccionados, presiona la opción "Exportar" ubicada la parte inferior derecha y puede seleccionar uno de los formatos determinados. El caso de uso terminará cuando se realice este proceso. |
| Complejidad | Media. |
| Prioridad | Media. |
| Precondiciones | El usuario tiene que estar autenticado en el sistema. La lista de documentos debe estar creada. |
| Postcondiciones | Se exportaron correctamente las referencias bibliográficas al formato |



| | | |
|--|---|---|
| | seleccionado por el usuario. | |
| Flujo de eventos | | |
| Flujo básico Sección 1: Exportar referencias bibliográficas | | |
| | Actor | Sistema |
| 1. | Selecciona el o los documentos a los cuales se les desea exportar las referencias bibliográficas. | |
| 2. | | Permite seleccionar el estilo bibliográfico para crear las referencias bibliográficas. Los estilos son: -Harvard -MLA -APA -Vancouver -Chicago -ISO-690 |
| 3. | | Permite seleccionar el formato de archivo para exportar las referencias bibliográficas. Los formatos son: RIS, TXT, HTML y RTF. |
| 4. | Selecciona el estilo bibliográfico, el formato de archivo y presiona Aceptar. | |
| 6. | | Valida la selección y exporta las referencias bibliográficas. |
| Flujos alternos | | |
| 6.a. Campos vacíos | | |
| | Actor | Sistema |
| 1. | | Muestra un mensaje de error: "Existen campos vacíos obligatorios", señalando dichos campos. |



Tabla 6: Descripción del caso de uso Enviar Referencia Bibliográfica por correo electrónico

| | | |
|---|--|---|
| Objetivo | Enviar las referencias bibliográficas por correo electrónico | |
| Actores | Usuario (inicia): envía las referencias bibliográficas de los documentos seleccionados a una dirección de correo electrónico. | |
| Resumen | El caso de uso inicia cuando el usuario selecciona uno o varios documentos contenidos en una lista, después selecciona la opción "EnviarPorCorreo" ubicada en la parte inferior derecha. El caso de uso termina cuando se envían las referencias bibliográficas. | |
| Complejidad | Media. | |
| Prioridad | Media. | |
| Precondiciones | El usuario tiene que estar autenticado en el sistema. La lista de documentos debe estar creada. | |
| Postcondiciones | Las referencias fueron enviadas correctamente. | |
| Flujo de eventos | | |
| Flujo básico Sección 1: Enviar las referencias bibliográficas por correo electrónico | | |
| | Actor | Sistema |
| 1. | Selecciona el o los documentos a los cuales se les desea enviar las referencias bibliográficas. | |
| 2. | | Muestra una interfaz para recoger los datos: -Usuario del destinatario -Comentario |
| 3. | Introduce los datos. | |
| 4. | | Valida los datos: -Usuario del destinatario -Comentario |
| 5. | | Envía las referencias bibliográficas. |
| Flujos alternos | | |
| 4.a. Campos vacíos | | |
| | Actor | Sistema |
| 1. | | Muestra un mensaje de error: "Existen campos vacíos obligatorios", señalando dichos campos. |
| 5.a. Conexión incorrecta | | |



| | Actor | Sistema |
|----|-------|--|
| 1. | | Muestra mensaje de error: "No se pudo realizar la conexión a la base de datos con los parámetros definidos". |
| 2. | | Regresa al paso 2 del flujo básico. |

2.5. Modelo de diseño

El modelo de diseño proporciona detalles sobre arquitectura del software, estructuras de datos, interfaces y componentes que se necesitan para implementar el sistema. El diseño permite modelar el sistema o producto que se va a construir. Este modelo se evalúa respecto de la calidad y su mejora antes de generar código; después, se efectúan pruebas y se involucra a muchos usuarios finales. El diseño es el lugar en el que se establece la calidad del software (Pressman 2004).

2.5.1. Diagrama de clases de diseño

El diagrama de clases del diseño (DCD) muestra las propiedades y funcionalidades de cada clase en el lenguaje de desarrollo y tecnologías seleccionados. Expresa la colaboración y responsabilidades de cada clase entorno al sistema que conforman y muestra cómo quedaría implementada toda la aplicación en términos lógicos (Pressman 2004). A continuación, se muestran los diagramas de clases del diseño con estereotipos web por cada caso de uso.

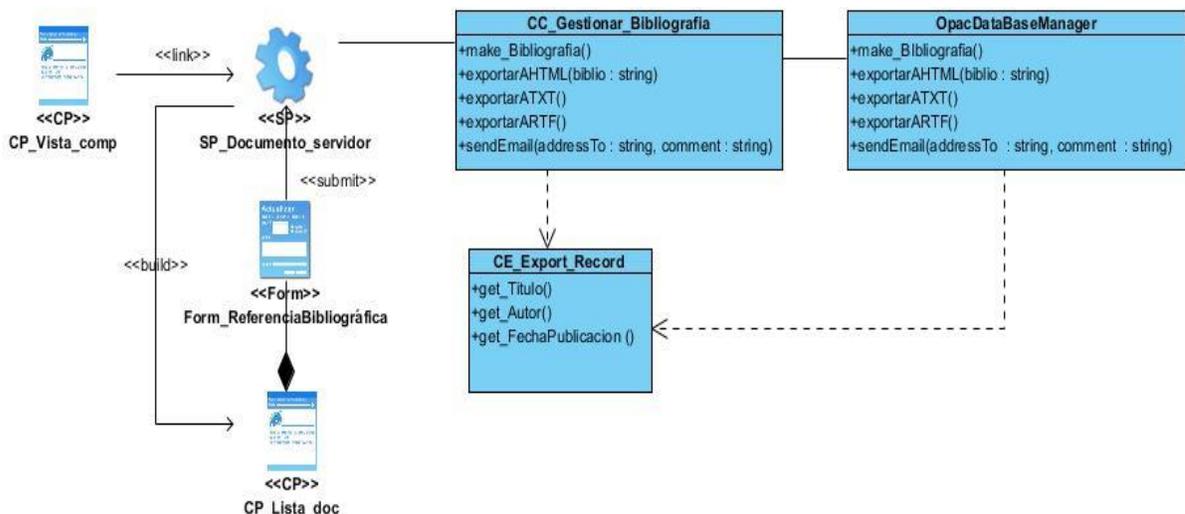


Figura 4: Diagrama de clases de diseño del caso de uso Crear Referencia

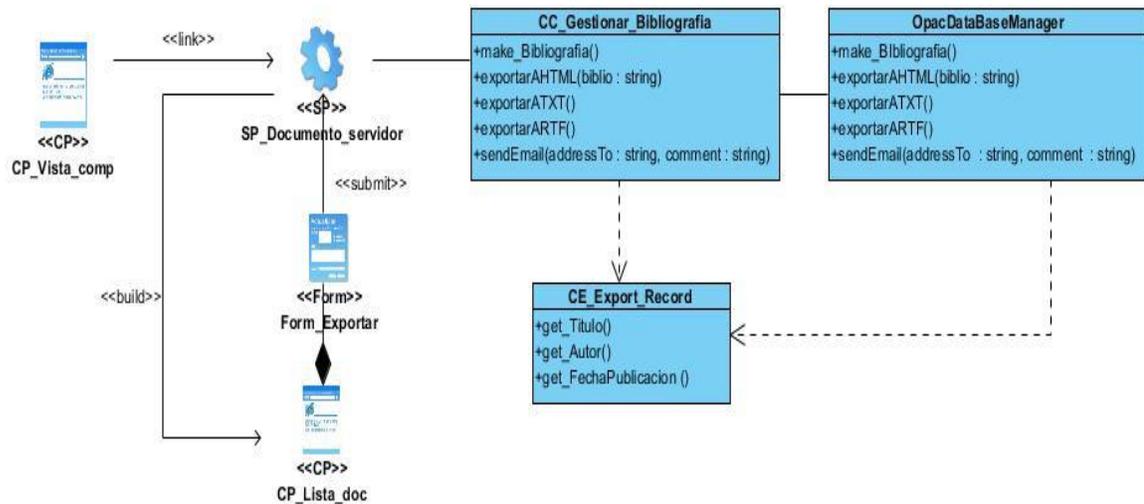


Figura 5: Diagrama de clases de diseño del caso de uso Exportar Referencia

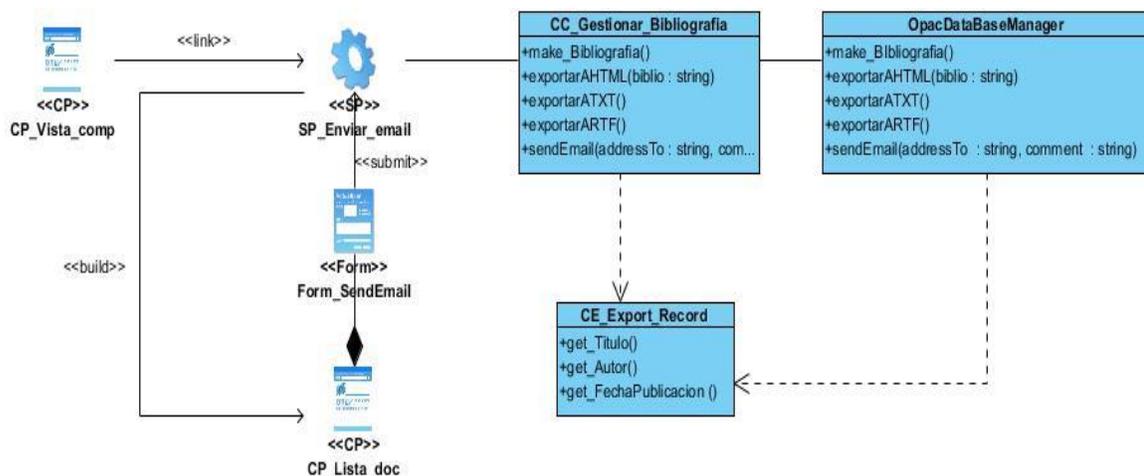


Figura 6: Diagrama de clases de diseño del caso de uso Enviar Referencia por correo electrónico

2.5.2. Diagrama de colaboración

El diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. En este tipo de diagrama se muestran las interacciones entre objetos. Estas son organizadas en torno a los objetos y los enlaces entre ellos, además de añadir mensajes (Pressman 2010).

A continuación, se muestran los diagramas de colaboración del análisis para los distintos casos de uso pertenecientes al componente Referencias Bibliográficas.



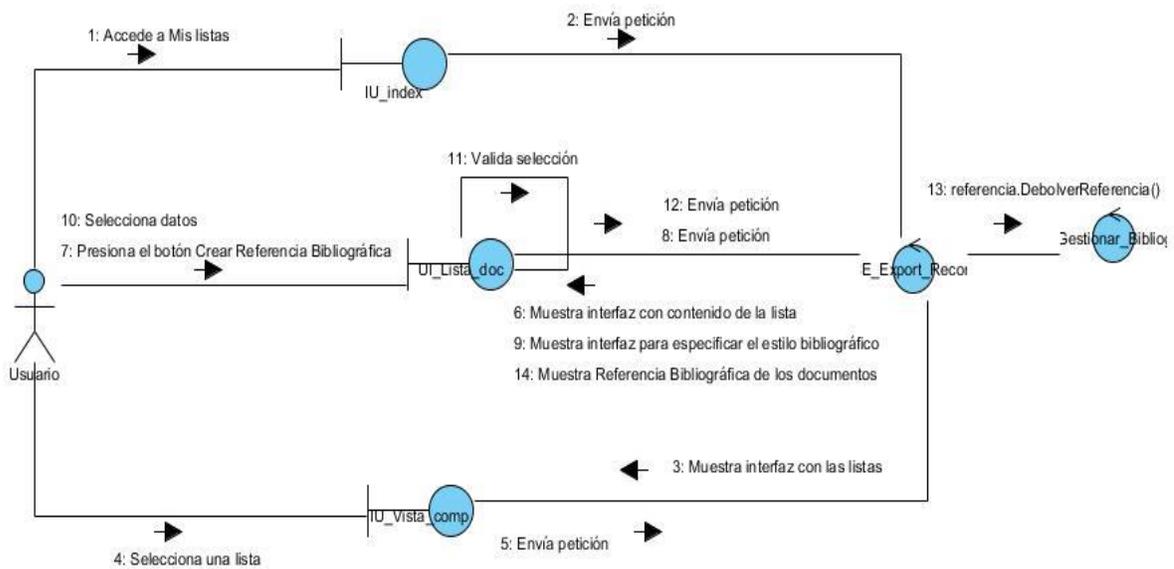


Figura 7: Diagrama de colaboración del requisito funcional Crear referencias bibliográficas

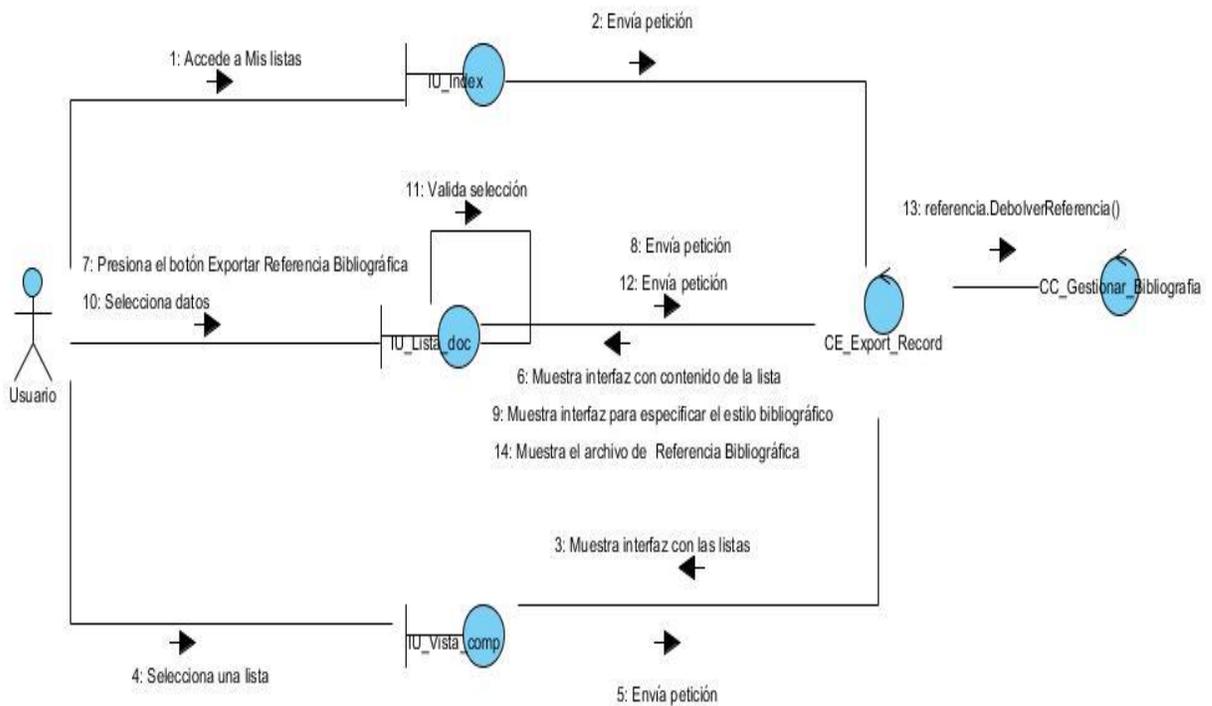


Figura 8: Diagrama de colaboración del requisito funcional Exportar referencias bibliográficas



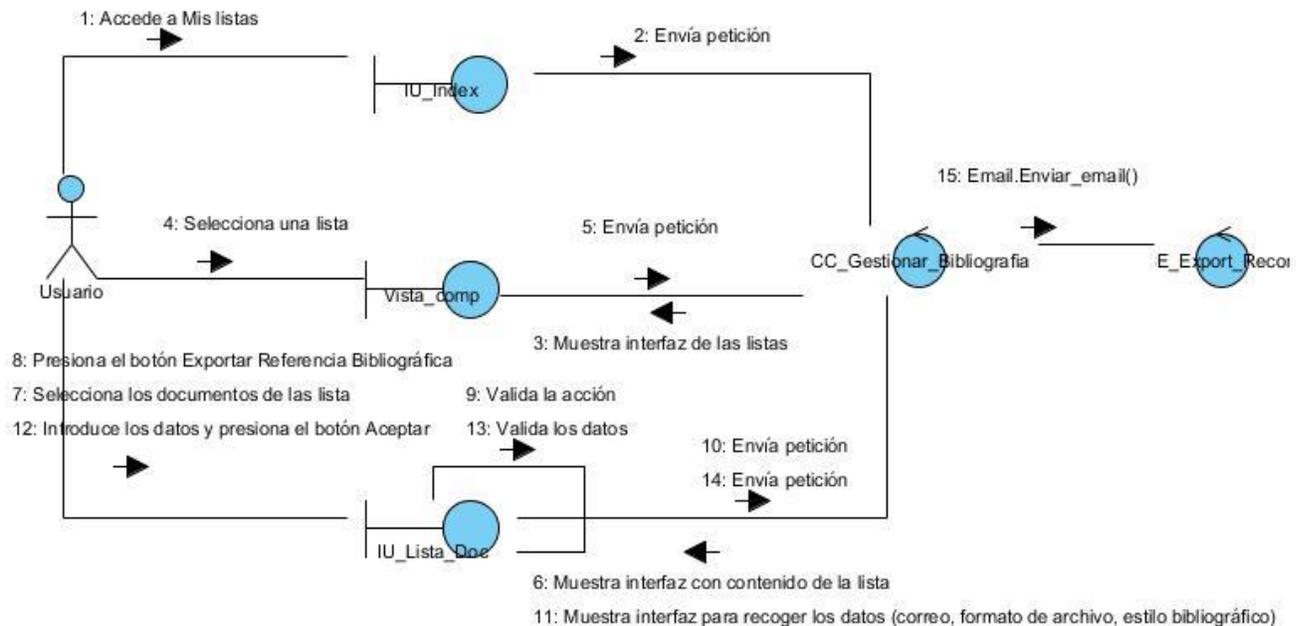


Figura 9 :Diagrama de colaboración del requisito funcional Enviar referencias bibliográficas por correo electrónico

2.6. Arquitectura del componente

La arquitectura de software se refiere a las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. El concepto de arquitectura de software se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad) y servir como guía en el desarrollo (Pressman 2010).

El sistema ABCD 3.0 utiliza el estilo arquitectónico n capas, lo que implica utilizar el mismo estilo de arquitectura para el componente Referencias Bibliográficas. Este estilo descompone la aplicación en cuantas capas sean necesarias para distribuir las subtarefas, donde cada capa proporciona servicios a la capa inmediata superior y se sirve de las prestaciones que le brinda la inmediata inferior. La arquitectura definida para el desarrollo de la solución se encuentra dividida en cuatro capas (Presentación, Lógica de negocio, Acceso a Datos e Infraestructura).

Capa de Presentación: contiene los objetos encargados de comunicar al usuario con el sistema mediante el intercambio de información, capturando y desplegando los datos necesarios para realizar alguna tarea. En esta capa están las interfaces de usuario.



Capa de Lógica de negocio: es aquí donde se definen las reglas que se deben cumplir para una correcta ejecución de la aplicación. En esta capa se encuentra toda la lógica del programa, así como las estructuras de datos y objetos encargados para la manipulación de los datos existentes, asimismo el procesamiento de la información ingresada o solicitada por el usuario en la capa de presentación.

Capa de Acceso a Datos: es la encargada de realizar transacciones con bases de datos y con otros sistemas para obtener o ingresar información al sistema. Es en esta capa donde se definen las consultas a realizar en la base de datos para la generación de reportes más específicos. Esta capa envía la información directamente a la capa de reglas de negocio para que sea procesada e ingresada en objetos según se necesite, esta acción se denomina encapsulamiento.

Capa de Infraestructura: añade gestión dinámica de permisos proporcionando un modelo de seguridad capaz de autenticar el código a través de la ubicación del *bundle* y de sus paquetes. Provee una infraestructura para desplegar y manejar aplicaciones que tienen que ejecutarse en un entorno seguro y controlado.

Según la especificación arquitectónica, el cliente interactúa con la capa más alta o capa de presentación, esta a su vez interactúa con la capa inmediata inferior o capa de lógica de negocio y esta con la capa más baja que es la de acceso a datos. Por último, se encuentra transversalmente la capa de infraestructura. Esta arquitectura soporta fácilmente la evolución del sistema y el encapsulamiento de datos. A continuación, se puede observar una imagen donde se expone lo antes mencionado.



Figura 10: Arquitectura en n Capas



2.7. Patrón de diseño

Un patrón de diseño es una buena práctica documentada de la solución de un problema que ha sido aplicado satisfactoriamente en múltiples entornos. Además, es una solución recurrente a un problema común observado o descubierto durante el estudio o construcción de numerosas aplicaciones. Su principal objetivo es incrementar la calidad del software en términos de reusabilidad, mantenimiento y extensibilidad (Pressman 2010).

Los patrones de diseño tienen como objetivo aumentar la eficiencia, disminuir el esfuerzo de mantenimiento y reducir el tiempo de desarrollo. Los patrones de diseño utilizados para la implementación del componente Referencias Bibliográficas son:

Patrones GRASP

GRASP es un acrónimo de *General Responsibility Assignment Software Patterns* (Patrones Generales de Software para Asignar Responsabilidades). Estos patrones describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones (Grosso 2011).

Experto: El patrón experto se basa en asignar una responsabilidad al experto en información: la clase que tiene la información necesaria para realizar la responsabilidad. De esta manera se logra un sistema más fácil de entender, mantener y ampliar y existen más oportunidades para utilizar componentes en futuras aplicaciones (Wahid y Ahmad 2017).

El patrón experto es usado en todas las clases ya que cada una posee la información correspondiente para realizar la tarea que le corresponde. Un ejemplo se evidencia en la clase *CE_Export_Record*, la cual contiene todo lo referente a las referencias, es la responsable de recoger todos los campos de los documentos.

Creador: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento (Wahid y Ahmad 2017).

Este patrón se pone de evidencia al momento de crear una referencia bibliográfica en el componente, ya que le permite al servidor mediante la clase *Gestionar_Bibliografía.java* construir en dependencia del estilo



seleccionado por el usuario en la vista de la funcionalidad, el objeto encargado de devolver la cadena con los datos del (los) documento (s) en cuestión.

Bajo acoplamiento: El patrón bajo acoplamiento impulsa la asignación de responsabilidades, de manera que su localización no incremente el acoplamiento, hasta un nivel que conlleve a los resultados negativos que puede producir un acoplamiento alto. El bajo acoplamiento soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. En otras palabras, debe haber poca dependencia entre las clases, de manera que se puedan extraer porciones de código de un modo independiente y reutilizarlas en otro proyecto (Mora 2007).

Se refleja en las interfaces en las cuales se declaran los métodos que mediante Spring se publican como servicios, la utilización de Spring mediante la inyección de dependencia, proporciona que las clases estén débilmente acopladas ejemplo *IOpac_OpacDataBaseManager*.

Alta Cohesión: Asigna una responsabilidad de modo que la cohesión siga siendo alta. La alta cohesión permite que la clase que tiene una responsabilidad pueda interactuar con otras clases para poder realizar las tareas encomendadas (Larman 2003).

Este patrón se demuestra en la clase *CE_Export_Record* es responsable de recoger los campos de los documentos para crear las referencias bibliográficas.

Data Access Object (DAO): Este patrón de diseño divide más las responsabilidades en la aplicación de tal forma que tendremos unas clases que se encargarán de la lógica de negocio y otras clases la responsabilidad de persistencia (Álvarez 2014).

En el componente Referencias Bibliográficas se crean DAO como *DataProvider* que es responsable de ejecutar las consultas a la base de datos necesarias para la obtención de los datos a mostrar en el sistema.

Patrones GOF

GOF es la sigla de *Gang of Four* (Banda de Cuatro). Estos patrones son considerados los fundadores de los demás patrones de diseño, su nombre se debe a que proceden del libro "*Design Patterns: Elements of Reusable Object Oriented Software*", escrito por cuatro autores: Erich Gamma, Richard Helm, Ralph



Johnson, and John Vlissides. Se clasifican según su propósito en tres grupos: creacionales, estructurales y de comportamiento (Larman 2003).

Singleton: Debe asegurar que una clase tenga una única instancia y proporcionar un punto de acceso global a la misma. El cliente llama a la función de acceso cuando se requiere una referencia a la instancia única (Burgos 2015).

Este patrón está presente en cada funcionalidad del componente, debido a que las clases de lógica del negocio atienden al criterio de crear únicas instancias para las operaciones que se realicen en el momento.

2.8. Conclusiones parciales

Se realizó la descripción del componente propuesto, aportando una solución a la problemática planteada. Se especificó el modelo conceptual que permitió definir los conceptos fundamentales del negocio. De esta forma se obtuvo una visión más clara del entorno sobre el cual se desarrolla el problema a resolver. Al identificar los requisitos funcionales y los requisitos no funcionales se obtuvo las características con las que debe contar el componente, se definieron 4 requisitos funcionales y 8 no funcionales. Se modeló el diagrama de casos de uso del sistema posibilitando un mejor entendimiento del proceso. Con las especificaciones de los casos de uso del sistema se establecen los flujos básicos de estos y se logra una descripción detallada para una mejor comprensión del componente que se implementará. La confección del diagrama de clases del diseño, permitió especificar correctamente las clases de *software* que se deben utilizar para dar paso a la implementación, y visualizar las relaciones entre las clases que involucran el sistema.



Capítulo 3: Implementación y validación del componente Referencias Bibliográficas

En este capítulo se representan los elementos necesarios para la implementación, a partir del resultado obtenido en el capítulo anterior. Además, se muestra la distribución del sistema en nodos mediante el diagrama de despliegue. Se explican los estándares de codificación para realizar buenas prácticas de programación. Se hace la validación del componente a través de las pruebas realizadas a cada caso de prueba.

3.1. Modelo de implementación

El modelo de implementación representa cómo los elementos del modelo de diseño se implementan en términos de componentes. De igual forma describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y la relación existente entre ellos (Ivar Jacobson 2000).

3.1.1. Estándares de codificación

Los estándares de código se refieren a un término que describe convenciones para escribir código fuente en ciertos lenguajes de programación. Los estándares de programación son frecuentemente dependientes del lenguaje de programación que se haya elegido para escribir. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad, es de gran importancia para la calidad del *software* y para obtener un buen rendimiento. A continuación, se enumeran los estándares a seguir en la codificación del componente Referencias Bibliográficas (Pressman 2010):

Nomenclatura de paquetes:

- Los paquetes están definidos como, por ejemplo: ***cu.uci.abcd.(Nombre del paquete)***.
- Los nombres de todos los paquetes están escritos en minúsculas y sin caracteres especiales.

Nomenclatura de las clases:

- Comenzar el nombre de la clase en letra inicial mayúscula; en caso de estar compuesto por más de una palabra se utilizan delimitadores (`_`) para discernir entre palabras.



Ejemplo: `public class Export_Record {`

`// código}`

- Los nombres de los métodos de estas clases deben comenzar con letra inicial minúscula; en caso de estar compuesto por más de una palabra se utilizan delimitadores (`_`) para discernir entre palabras.

Ejemplo: `public void make_Bibliografía {`

`// código}`

- Los nombres de variables referentes a los metadatos de los documentos deben comenzar con letra inicial minúscula; en caso de estar compuesto por más de una palabra, la inicial de esta debe aparecer en mayúscula.

3.2. Diagrama de despliegue

El diagrama de despliegue se utiliza para modelar el *hardware* utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Permiten modelar la disposición física o topología de un sistema. Muestran los dispositivos físicos usados y los componentes instalados en el mismo. Además, muestra las conexiones entre el *hardware* y las relaciones entre componentes. A continuación, se muestra en la siguiente figura el diagrama de despliegue del componente Referencias Bibliográficas, donde se visualiza la distribución de los componentes de *software* en los nodos físicos.



Figura 11: Diagrama de despliegue del componente Referencias Bibliográficas



El diagrama de despliegue refleja la disposición de los recursos necesarios para realizar el despliegue del sistema. A continuación, se describen características de los nodos:

Nodo PC_Cliente: representa las computadoras que utilizarán los usuarios para interactuar con la aplicación. Se comunica con el servidor de aplicaciones a través del protocolo HTTP por el puerto 8080.

Nodo servidor de aplicaciones: representa el servidor donde se encuentra instalado el sistema. Se comunica con el Servidor de Base de Datos PostgreSQL a través de la familia de protocolos TCP/IP por el puerto 5432.

Nodo servidor de Base de Datos PostgreSQL: es el servidor de base de datos relacional PostgreSQL donde se almacena la base de datos del sistema ABCD 3.0.

Nodo Servidor de Correo: representa el servidor de correo que se requiere para el envío de las referencias bibliográficas a un determinado usuario. La entidad determina qué servidor de correo desea utilizar. Realiza el envío del correo electrónico mediante el protocolo SMTP por el puerto 25.

3.3. Pruebas del *software*

Las pruebas de *software* son importantes porque aseguran el correcto cumplimiento de la funcionalidad del producto, ayudan a ganar confianza, confirman la fiabilidad del uso y previenen defectos en producción, lo cual tiene un impacto económico positivo en la empresa en cuestión. Las pruebas de *software* son una actividad primordial en el proceso de “aseguramiento de la calidad” (Chiu, Alberto y Carbajal 2015).

Realizar pruebas a una aplicación desarrollada es una actividad fundamental que toda empresa productora de *software* debería llevar a cabo. Las pruebas de *software* son un elemento clave en la garantía de la calidad del producto final.

3.4. Estrategias de pruebas

Una estrategia de prueba del *software* integra los métodos de diseño de caso de pruebas del *software* en una serie bien planeada de pasos que desembocará en la eficaz construcción del mismo. La estrategia proporciona un mapa que describe los pasos que se darán como parte de la prueba, indica cuándo se planean, cuándo se dan estos pasos, además de cuanto esfuerzo, tiempo y recursos consumirán. Por tanto, cualquier estrategia de prueba debe incorporar la planeación de pruebas, el



diseño de casos de pruebas, la ejecución de pruebas, la recolección y evaluación de los datos resultantes (Pressman 2010).

La estrategia de prueba debe indicar los niveles de pruebas (ciclos) que se aplican y la intensidad o profundidad a aplicar para cada nivel de prueba definido.

3.5. Niveles de prueba

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Se distinguen los siguientes Niveles de Pruebas (Ivar Jacobson 2000):

- Prueba de Unidad: Un componente es la unidad más pequeña especificada de un sistema, las pruebas se llevan a cabo tras la construcción o realización de cada componente para verificar que la implementación se esté llevando conforme a los estándares acordados. El objetivo es comprobar que el sistema, entendido como una unidad funcional, está correctamente codificado.
- Prueba de Sistema: Son las pruebas que se hacen cuando el *software* está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de *software* y *hardware* han sido integrados.
- Prueba de Aceptación: Es la prueba final antes del despliegue del sistema. El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales.

Para realizar las pruebas en el componente Referencias Bibliográficas se desarrollarán las pruebas de sistema, las pruebas de unidad y las pruebas de aceptación.

3.6. Prueba de sistema

La prueba de sistema es una serie de diferentes pruebas cuyo propósito principal es ejercitar por completo el sistema basado en computadora. Aunque cada prueba tenga un propósito diferente, todo el sistema funciona para verificar que los elementos se hayan integrado de manera adecuada y que se realicen las funciones asignadas (Pressman 2010).



Tipos de prueba del sistema

Cada nivel de prueba engloba una técnica de prueba específica según los atributos de calidad que se deseen verificar con las pruebas al software. Entre las técnicas de pruebas que se realizan en el sistema está la que evalúa la funcionalidad de éste. En esta investigación se realiza el tipo de prueba funcional.

Pruebas funcionales

Este tipo de prueba se enfoca en validar la correcta implementación de las necesidades del cliente. La funcionalidad puede ser vinculada a los datos de entrada y de salida. Los datos de entrada serán ejecutados y mostrarán un resultado y dicho resultado será comparado con el resultado esperado (comportamiento), este proceso se muestra en la figura 12 (Chiu, Alberto y Carbajal 2015).

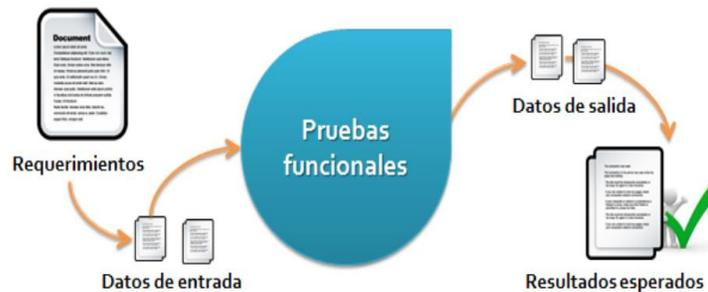


Figura 12: Pruebas funcionales

Entre las técnicas de pruebas que se realizan en el sistema está la que evalúa la funcionalidad de éste. Según los parámetros de evaluación que abarca la técnica se pueden distinguir distintos tipos de pruebas:

- Prueba de Función.
- Prueba de Seguridad.
- Prueba de Volumen.

Al componente Referencias Bibliográficas se le aplica la prueba de funcionalidad específicamente la de función.



Pruebas de función (Pressman 2010):

Las pruebas de función aseguran el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

Metas de estas pruebas son:

- Verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio.
- Verificar la apropiada aceptación de datos.

Las pruebas de función están enfocadas en los requisitos funcionales (Casos de Uso) y las reglas del negocio. Estas pruebas utilizan el método de prueba Caja Negra.

Las **pruebas de caja negra**, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. (Pressman 2010). Estas pruebas pretenden demostrar que las funcionalidades del sistema son operativas, las entradas se aceptan correctamente y que se producen los resultados esperados.

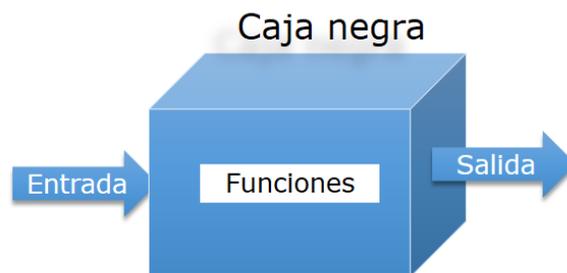


Figura 13: Método de prueba Caja negra

Para desarrollar las pruebas de caja negra existen varias técnicas. La técnica que se va a utilizar para el componente Referencias bibliográficas es la de partición de equivalencia.

La **partición de equivalencia** divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. El diseño de casos de prueba para la partición de equivalencia se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Por lo general, una



condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleana. Las clases de equivalencia, pueden desarrollarse y ejecutarse los casos de prueba para cada ítem de datos del dominio de entrada. Los casos de prueba se seleccionan de modo que se revise a la vez el número más grande de atributos de una clase de equivalencia (Pressman 2010).

Los **casos de prueba** deben diseñarse para descubrir errores debido a cálculos erróneos, comparaciones incorrectas o flujo de control inadecuado. Se diseñan para garantizar que: se satisfagan todos los requisitos de funcionamiento, se logran todas las características de comportamiento, todo el contenido es preciso y se presenta de manera adecuada, se logran todos los requisitos de rendimiento, la documentación es correcta y se satisfacen la facilidad de uso y otros requisitos (SOMMERVILLE 2005). A continuación, se muestran los casos de prueba del componente Referencias Bibliográficas.

Tabla 7: Prueba de caja negra del caso de uso Crear referencias bibliográficas

| Escenario | Descripción | Variables | | Respuesta del sistema | Flujo central |
|---|---|-----------------------------------|----------------------|---|--|
| | | Documentos seleccionados | Estilo bibliográfico | | |
| EC 1.1 Crear las referencias bibliográficas de las listas de selección. | Permite crear las referencias bibliográficas de las listas de selección en un estilo bibliográfico determinado correctamente. | V | V | Valida la selección y crea la referencia bibliográfica. | 1. Selecciona las listas a las cuales se les desea crear las referencias bibliográficas. 2. Selecciona el estilo bibliográfico y presiona el botón Aceptar. |
| | | Libro Patrones de Diseño en Java. | APA. | | |
| EC 1.2 Crear las referencias bibliográficas de los documentos. | Permite crear las referencias bibliográficas de los documentos seleccionados en un estilo bibliográfico | V | V | Valida la selección y crea la referencia bibliográfica. | 1. Selecciona el o los documentos a los cuales se les desea crear las referencias bibliográficas. |



| | | | | | | |
|----------------------|-----|---|-----------------------------------|------------|--|---|
| | | determinado, correctamente. | Libro Patrones de Diseño en Java. | APA. | | 2. Selecciona el estilo bibliográfico y presiona el botón Aceptar. |
| EC Campos vacíos. | 1.3 | Se valida que no existan campos obligatorios vacíos, es decir que se seleccione al menos un documento y se elija un estilo. | I | V | Muestra un mensaje de error: "Existen campos vacíos obligatorios" señalando dichos campos. | 1. Selecciona el o los documentos a los cuales se les desea crear las referencias bibliográficas. 2. No se selecciona un estilo bibliográfico. |
| | | | | Vancouver. | | |
| | | | V | I | | |
| | | | Libro Ingeniería de software. | | | |

Tabla 8: Prueba de caja negra del caso de uso Exportar referencias bibliográficas

| Escenario | Descripción | Variables | | | Respuesta del sistema | Flujo central |
|--|--|--------------------------|----------------------|--------------------|---|--|
| | | Documentos seleccionados | Estilo bibliográfico | Formato de archivo | | |
| EC 1.1 Exportar Referencias Bibliográficas. | Permite exportar las referencias bibliográficas de los documentos seleccionados a un formato de archivo determinado correctamente. | V | V | V | Valida la selección y exporta las referencias bibliográficas. | 1. Selecciona el o los documentos a los cuales se les desea exportar las referencias bibliográficas. 2. Selecciona el estilo bibliográfico, el formato de archivo y presiona Aceptar. |
| | | Somerville. | ISO-690. | TXT. | | |
| | | I | V | V | | |



| | | | | | | |
|--------------------------|---|--------------------------|----------|------|---|--|
| EC 1.2 Campos vacíos. | Se valida que no existan campos obligatorios vacíos, es decir que se seleccione al menos un documento y se elija un estilo. | | APA. | TXT. | Muestra un mensaje de error: "Existen campos vacíos obligatorios", señalando dichos campos. | 1. No se seleccionan los documentos. 2. Selecciona el estilo bibliográfico, el formato de archivo y presiona Aceptar. |
| | | V | I | V | | |
| | | Pressman. | | TXT. | | |
| | | V | V | I | | |
| | | Gestores bibliográficos. | Harvard. | | | |

Tabla 9: Prueba de caja negra del caso de uso Enviar referencias bibliográficas por correo electrónico

| Escenario | Descripción | Variables | | Respuesta del sistema | Flujo central |
|--|---|------------------------------------|------------|---|---|
| | | Correo electrónico | Comentario | | |
| EC 1.1 Enviar Referencias Bibliográficas por correo electrónico. | Permite enviar las referencias bibliográficas creadas de los documentos seleccionados por correo electrónico correctamente. | V | V | Valida la selección y envía las referencias bibliográficas. | 1. Selecciona el o los documentos a los cuales se les desea enviar las referencias bibliográficas. 2. Introduce los datos destinatario, archivo que contenga las referencias bibliográficas. |
| | | <i>malvarez@estudiantes.uci.cu</i> | "Hola". | | |
| | | I | V | | |



| | | | | | |
|--------------------------------|---|---------------------------------------|--|--|---|
| EC 1.2 Campos vacíos. | Se valida que no existan campos obligatorios vacíos, es decir que se seleccione al menos un documento y se elija un estilo. | V | I | Muestra un mensaje de error: "Existen campos obligatorios", señalando dichos campos. | 1. Selecciona el o los documentos a los cuales se les desea enviar las referencias bibliográficas. 2. No se selecciona un estilo bibliográficos. |
| | | <i>gyrosabal@uci.cu</i> | | | |
| EC 1.3 Conexión incorrecta. | Se realiza la conexión a la base de datos de forma incorrecta. | V | V | Muestra mensaje de error: "No se pudo realizar la conexión a las BD con los Parámetros definidos". | 1. Selecciona el o los documentos a los cuales se les desea enviar las referencias bibliográficas. 2. Introduce los datos destinatario, archivo que contenga las referencias bibliográficas. |
| | | <i>snmustelier@estudiantes.uci.cu</i> | "Registros bibliográficos recomendados". | | |

Tabla 10: Descripción de las variables

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|----|---------------------------|------------------------------|------------|--|
| 1 | Documentos seleccionados. | Campo de Selección múltiple. | No | Se refiere al documento o a la lista de documentos a la cual se les desea crear las referencias bibliográficas. |
| 2 | Estilo bibliográfico. | Lista desplegable. | No | Se refiere al estilo bibliográfico que se selecciona para crear referencias bibliográficas. Se pueden seleccionar algunos de los siguientes estilos: Harvard, MLA, APA, Vancouver, Chicago, ISO-690. |
| 3 | Formato de archivo. | Lista desplegable. | No | Se refiere al formato de archivo que se selecciona para exportar las referencias bibliográficas. Se puedes |



| | | | | |
|---|---------------------|-----------------|----|---|
| | | | | seleccionar los formatos de archivos RIS, TXT, HTML y RTF. |
| 4 | Correo electrónico. | Campo de texto. | No | Se refiere al correo electrónico del destinatario al que se le enviara el archivo con las referencias bibliográficas, admite solo letras. |
| 5 | Comentario. | Campo de texto. | No | Se refiere al comentario que se le enviara por correo electrónico al destinatario. Estos comentarios admiten caracteres alfabéticos. |

3.6.1. Resultado de las pruebas de sistema del componente Referencia Bibliográfica

A continuación, se muestran los resultados de las pruebas de caja negra que fueron realizadas al componente Referencias Bibliográficas, en las que se ejecutaron un total de 2 iteraciones. En la siguiente gráfica de barras se observa además el total de no conformidades encontradas y las que fueron resueltas, e igualmente se muestra la cantidad que no procedieron.

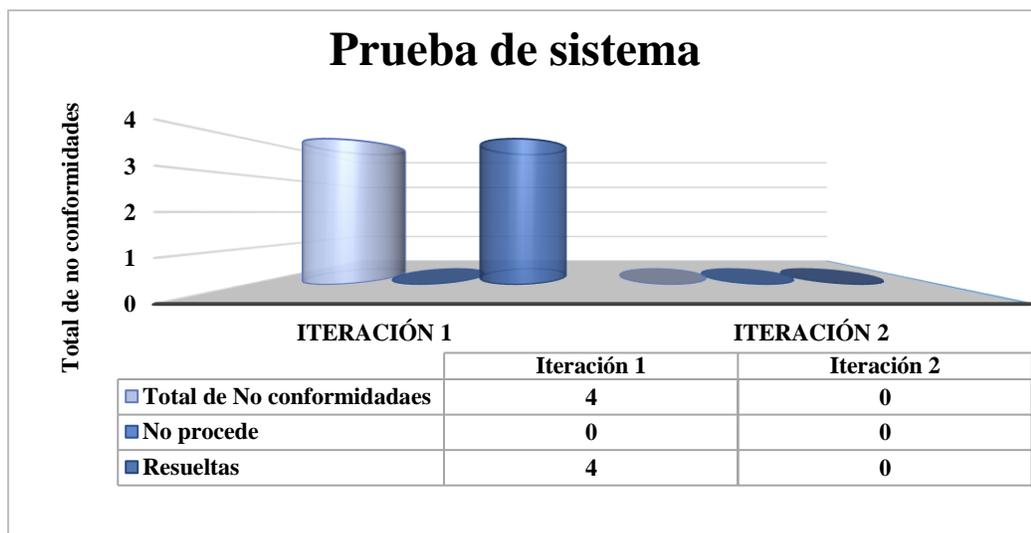


Figura 14: Resultados de las pruebas de caja negra aplicadas al componente Referencias Bibliográficas

En la primera iteración se detectaron un total de cuatro no conformidades. De ellas fueron resueltas las cuatro correctamente. En la segunda iteración no se encontraron no conformidades, validándose de esta manera el correcto funcionamiento del componente Referencias Bibliográficas para el sistema ABCD 3.0.



Las no conformidades son los errores que son detectados durante la revisión de la documentación del sistema. Se elabora un documento por cada revisión que se haga y se controlan a través de versiones según se vayan eliminando los errores, hasta que finalmente se hayan erradicado todos los defectos que posea el elemento que se prueba. A continuación, se muestra la tabla donde se hace una clasificación de las no conformidades.

Tabla 11: Clasificación de las No conformidades del componente Referencias Bibliográficas

| Clasificación | Observaciones |
|------------------|--|
| Significativa | Errores en la interpretación de los procesos de la entidad y de funcionalidad. |
| No significativa | Errores de terminología y de diseño de interfaz. |
| Recomendación | Errores de redacción y ortografía. |

Las no conformidades encontradas en la primera iteración tienen como clasificación:

1. Significativa: en el caso de prueba exportar referencias bibliográficas, describe que al presionar el botón Aceptar se obtenga un archivo en el formato seleccionado, que contenga las referencias bibliográficas, pero esta acción no se realiza de la forma correcta.
2. No significativa: en la descripción la variable “correo electrónico” se especifica que admite solo letras, pero además admite números, por lo que debería especificarse que admite caracteres alfanuméricos.
3. Recomendación: el mensaje “Conexión incorrecta” muestra error de concordancia.
4. Recomendación: el mensaje “Existen campos vacíos obligatorios” muestra falta de ortografías.

3.7. Pruebas de unidad

Se centra en el esfuerzo de verificación de la unidad más pequeña del diseño del *software*, el componente o sistema de *software*. Tomando como guía la descripción del diseño al nivel de componentes, se prueban importantes caminos de control para describir errores dentro de los límites del módulo. El alcance restringido que se ha determinado para las pruebas de unidad limita la relativa complejidad de las pruebas y los errores que estas descubren. Las pruebas se encuentran en la lógica del procedimiento interno y en las estructuras



de datos dentro de los límites de un componente. Este tipo de prueba se puede aplicar en paralelo a varios componentes (Pressman 2010).

Pruebas de caja blanca

La prueba de caja blanca, en ocasiones llamada “prueba de caja de vidrio”, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que (Pressman 2010):

1. Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
2. Revisen todas las decisiones lógicas en sus lados verdadero y falso.
3. Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
4. Revisen estructuras de datos internas para garantizar su validez.

La prueba de ruta o trayectoria básica

Es una técnica de prueba de caja blanca propuesta por primera vez por Tom McCabe. El método de ruta básica permite al diseñador de casos de prueba derivar una medida de complejidad lógica de un diseño de procedimiento y usar esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para revisar el conjunto básico tienen garantía para ejecutar todo enunciado en el programa, al menos una vez durante la prueba (Pressman 2010).

A continuación, se muestra el caso de prueba correspondiente al método “make_Bibliografía”, el cual se localiza en la clase “Gestionar_Bibliografía”. El objetivo de este método es crear las referencias bibliográficas en los formatos de archivo TXT y HTML, a partir de las listas de selección. Se selecciona este método teniendo en cuenta la importancia que representa para el resultado de este trabajo. A continuación, se muestra el código fuente referente al método descrito anteriormente y cómo fue aplicada la Técnica de Camino Básico al mismo.



```

public List<String> make_bibliografia() throws IOException {
1  List<String> result = new LinkedList<>();
2  for (Record record : listaRecord) {
3      Export_Recod dato = new Export_Recod(record);
4      listItemData
5          .add(new CSLItemDataBuilder()
6              .id(dato.getControlNumber())
7              .type(tipo(dato.getRecordType()))
8              .title(dato.getTitle())
9              .author(new CSLNameBuilder().family(
10                 dato.getAuthor()).build())
11                 .issued(dato.getPublicationDate()).build());
12  }
13  ListItemDataProviderMIO item = new ListItemDataProviderMIO(listItemData);
14  citeProc = new CSL(item, norma);
15  getClass().getResource("/^*.csl");
16  switch (salida) {
17      case texto:
18          citeProc.setOutputFormat("text");
19          break;
20      case html:
21          citeProc.setOutputFormat("html");
22          break;
23  }
24  for (Record record : listaRecord) {
25      Export_Recod dato = new Export_Recod(record);
26      citeProc.makeCitation(dato.getControlNumber());
27  }
28  bibliografia = citeProc.makeBibliography();
29  for (String entry : bibliografia.getEntries()) {
30      result.add(entry);
31  }
32  return result;
33  }

```

Figura 15: Fragmento de código del método make_Bibliografía

Antes de considerar el método de ruta básica, debe introducirse una notación simple para la representación del flujo de control, llamado grafo de flujo (o grafo de programa) (Pressman 2010). Cada fragmento de código tiene un número que lo identifica en el grafo de flujo. A continuación, se muestra dicho grafo.

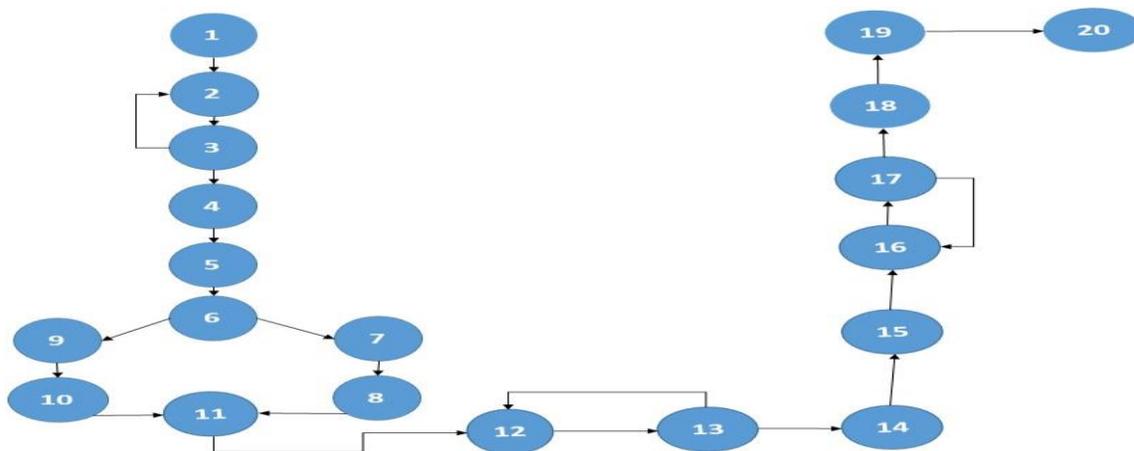


Figura 16: Representación del grafo de flujo de camino básico del método make_Bibliografía



La **complejidad ciclomática** es una medición de software que proporciona una evaluación cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba de la ruta básica, el valor calculado por la complejidad ciclomática define el número de rutas independientes del conjunto básico de un programa y le brinda una cota superior para el número de pruebas que debe realizar a fin de asegurar que todos los enunciados se ejecutaron al menos una vez. La complejidad ciclomática tiene fundamentos en la teoría de grafo y proporciona una medición de software extremadamente útil. La complejidad se calcula en una de tres formas (Pressman 2010):

1. El número de regiones del grafo de flujo corresponde a la complejidad ciclomática.
2. La complejidad ciclomática $V(G)$ para un grafo de flujo G se define como $V(G) = E - N + 2$ donde E es el número de aristas del grafo de flujo y N el número de nodos del grafo de flujo.
3. La complejidad ciclomática $V(G)$ para un grafo de flujo G también se define como $V(G) = P + 1$ donde P es el número de nodos predicado contenidos en el grafo de flujo G .

En el grafo de flujo de la figura 16, la complejidad ciclomática puede calcularse usando cada uno de los algoritmos recién indicados:

1. La cantidad de regiones del grafo de flujo es 5
2. $V(G) = 23$ aristas – 20 nodos + 2
 $V(G) = 5$
3. $V(G) = 4$ nodos predicados + 1
 $V(G) = 5$

Por tanto, la complejidad ciclomática del grafo de flujo en la figura 16 es 5.

Una **ruta independiente** es cualquiera que introduce al menos un nuevo conjunto de enunciados de procesamiento o una nueva condición en el programa. Cuando se establece como un grafo de flujo, una ruta independiente debe moverse a lo largo de al menos una arista que no se haya recorrido antes de definir la ruta (Pressman 2010). A continuación, se muestran las rutas del grafo de flujo de la figura 16.

Ruta 1: 1-2-3-4-5-6-7-8-11-12-13-14-15-16-17-18-19-20.



Ruta 2: 1-2-3-4-5-6-9-10-11-12-13-14-15-16-17-18-19-20.

Ruta 3: 1-2-3-3-4-5-6-7-8-11-12-13-14-15-16-17-18-19-20.

Ruta 4: 1-2-3-2-3-4-5-6-9-10-11-12-13-13-14-15-16-17-18-19-20.

Ruta 5: 1-2-3-4-5-6-7-8-11-12-13-14-15-16-17-17-18-19-20.

A continuación, se diseñan los casos de prueba que cubren las rutas independientes presentadas.

Tabla 12: Caso de prueba de caja blanca para la ruta 1

| Caso de prueba para la ruta 1 | |
|--------------------------------------|--|
| Entrada | Lista de selección con los registros bibliográficos de los documentos. Además de un estilo bibliográfico y un formato de salida (TXT, HTML). |
| Resultados esperados | Se obtienen las referencias bibliográficas. |
| Condiciones | El formato de salida para crear las referencias bibliográficas debe ser TXT. |

Tabla 13: Caso de prueba de caja blanca para la ruta 2

| Caso de prueba de la ruta 2 | |
|------------------------------------|--|
| Entrada | Lista de selección con los registros bibliográficos de los documentos. Además de un estilo bibliográfico y un formato de salida (TXT, HTML). |
| Resultados esperados | Se obtienen las referencias bibliográficas. |
| Condiciones | El formato de salida para crear las referencias bibliográficas debe ser HTML. |



Tabla 14: Caso de prueba de caja blanca para la ruta 3

| Caso de prueba de la ruta 3 | |
|-----------------------------|--|
| Entrada | Lista de selección con los registros bibliográficos de los documentos. Además de un estilo bibliográfico y un formato de salida (TXT, HTML). |
| Resultados esperados | Se obtienen las referencias bibliográficas. |
| Condiciones | El formato de salida para crear las referencias bibliográficas debe ser TXT. Se debe adicionar los registros bibliográficos según el estilo especificado. |

Tabla 15: Caso de prueba de caja blanca para la ruta 4

| Caso de prueba de la ruta 4 | |
|-----------------------------|---|
| Entrada | Lista de selección con los registros bibliográficos de los documentos. Además de un estilo bibliográfico y un formato de salida (TXT, HTML). |
| Resultados esperados | Se obtienen las referencias bibliográficas. |
| Condiciones | El formato de salida para crear las referencias bibliográficas debe ser HTML. Se deben crear las referencias bibliográficas de las citas realizadas. |

Tabla 16: Caso de prueba de caja blanca para la ruta 5

| Caso de prueba para la ruta 5 | |
|-------------------------------|--|
| Entrada | Lista de selección con los registros bibliográficos de los documentos. Además de un estilo bibliográfico y un formato de salida (TXT, HTML). |
| Resultados esperados | Se obtienen las referencias bibliográficas. |



| | |
|--------------------|---|
| Condiciones | El formato de salida para crear las referencias bibliográficas debe ser TXT. Se debe adicionar las referencias en una lista. |
|--------------------|---|

Luego de realizar las pruebas unitarias al componente Referencias Bibliográficas empleando la técnica de ruta básica al método (`make_Bibliografía`), se afirma que se obtuvo el resultado esperado, por lo que esta prueba concluye de manera satisfactoria.

3.8. Pruebas de aceptación

Cuando se construye un *software* para un cliente, se realiza una serie de pruebas de aceptación a fin de permitir al cliente validar todos los requerimientos. Una prueba de aceptación puede variar desde una “prueba de conducción” informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente. De hecho, la prueba de aceptación puede realizarse durante un período de semanas o meses, y mediante ella descubrir errores acumulados que con el tiempo puedan degradar el sistema. La mayoría de los constructores de productos de *software* usan un proceso llamado prueba alfa y prueba beta para descubrir errores que al parecer solo el usuario final es capaz de encontrar (Pressman 2010).

La prueba alfa se lleva a cabo en el sitio del desarrollador por un grupo representativo de usuarios finales. El *software* se usa en un escenario natural con el desarrollador “mirando sobre el hombro” de los usuarios y registrando los errores y problemas de uso. Las pruebas alfa se realizan en un ambiente controlado.

La prueba beta se realiza en uno o más sitios del usuario final. A diferencia de la prueba alfa, por lo general el desarrollador no está presente. Por tanto, la prueba beta es una aplicación “en vivo” del *software* en un ambiente que no puede controlar el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que se encuentran durante la prueba beta y los reporta al desarrollador periódicamente. Como resultado de los problemas reportados durante la prueba beta, es posible hacer modificaciones y luego preparar la liberación del producto de *software* a toda la base de clientes.

Para la realización de la prueba de aceptación en el componente Referencia Bibliográfica se utiliza la prueba alfa (ver anexo 2).



3.9. Conclusiones parciales

En el presente capítulo se realizó el diagrama de despliegue el cual logró especificar la plataforma sobre la que se ejecuta el software del sistema. La definición de los estándares de codificación permitió mejorar el entendimiento del código perteneciente al componente Referencias Bibliográficas del sistema ABCD 3.0 y alcanzar una uniformidad en el mismo. Trazando la estrategia de prueba se definió una serie de pasos bien planificados que dan como resultado una correcta construcción del sistema, entre los que se incluyen los tipos de prueba que se le realizaron al *software*, así como sus niveles. La aplicación del método de caja negra, permitió validar correctamente los requisitos funcionales del sistema utilizándose para ello la técnica de partición de equivalencia. Además, el método de caja blanca contribuyó a validar el código fuente del componente a través de los casos de pruebas usando la técnica de ruta básica. Sin embargo, la realización de la prueba de aceptación demostró que el usuario final está de acuerdo con el componente Referencia Bibliográfica.



Conclusiones generales

La presente investigación se centró en el objetivo general, el cual se basó en desarrollar un componente para el sistema ABCD 3.0 que permita exportar las referencias bibliográficas. Durante el avance del mismo se cumplieron las tareas propuestas y se arribó a los siguientes resultados:

1. El establecimiento de los fundamentos teóricos y metodológicos relevantes a las referencias bibliográficas, permitió sentar las bases del proceso investigativo y conocer la lógica del negocio, tomando como punto de partida las soluciones informáticas siguientes: Citeproc, Google Académico, REPXOS 3.0, SIGB.
2. Se caracterizaron las tecnologías y herramientas definidas por el proyecto ABCD, lo que permitió determinar dónde serían utilizadas efectivamente para el desarrollo de la solución propuesta.
3. Se propuso una solución, la cual posibilitó tener una concepción más clara de lo que se debía desarrollar y se especificaron las funcionalidades crear, exportar y enviar por correo electrónico las referencias bibliográficas que tenía que cumplir el componente.
4. El análisis y diseño del componente Referencia Bibliográfica posibilitó formalizar la estructura ingenieril con el uso de la metodología de desarrollo de *software* AUP-UCI.
5. Se realizaron las pruebas de sistema y de unidad para comprobar el correcto funcionamiento del componente Referencia Bibliográfica, las no conformidades encontradas se resolvieron satisfactoriamente, comprobándose que el mismo cumple con todas las funcionalidades especificadas.



Recomendaciones

A partir del análisis realizado en la presente investigación y teniendo en cuenta las experiencias adquiridas a lo largo de su desarrollo, se recomienda incorporar las siguientes funcionalidades:

1. Permitir generar las referencias bibliográficas de un documento que no esté contenido en las listas de selección, teniendo en cuenta las nuevas tecnologías que se desarrollan en la actualidad y de donde se implante el sistema.
2. Integrar el componente Referencias bibliográficas al módulo OPAC del sistema ABCD 3.0.
3. Hacer uso de un fichero externo donde se especifique el número de campos y subcampos del formato definido en la base de datos de ABCD.



Referencias bibliográficas

ALICANTE, I. de, 2005. Estilo Harvard. *Universidad de Alicante*, pp. 0-42.

ALICANTE, U. de, 2018. Gestores de Referencias Bibliográficas. [en línea]. [Consulta: 5 febrero 2018].
Disponible en: <https://biblioteca.ua.es/es/estudia-y-aprende/elaboracion-y-gestion-de-citas-bibliograficas/gestores-de-referencias-bibliograficas.html>.

ALL, I.A., 2012. *Introducción al lenguaje Java*. 2012. S.l.: s.n.

ÁLVAREZ, C., 2014. Patrones de Diseño (Active Record vs DAO). *GENBETA* [en línea]. [Consulta: 24 mayo 2018]. Disponible en: <https://www.genbetadev.com/java-j2ee/patrones-de-diseno-active-record-vs-dao>.

BIBLIOTECA UNIVERSITARIA, U. del P.V., 2015. Google Académico Guía De Uso. ,

BORETTO, J., 2018. Gestores de Referencias Bibliográficas. *Rev. Asoc. Argent. Ortop. Traumatol*, pp. i-i. ISSN 1852-7434.

BURGOS, L., 2015. Patrones de Diseño en Java. *GitHub* [en línea]. [Consulta: 24 mayo 2018]. Disponible en: <https://github.com/LuisBurgos/design-patterns#singleton>.

CADIZ, U. de, 2018. Gestores de Referencias Bibliográficas. [en línea]. Disponible en: <http://biblioteca.uca.es/aprendizajeinvestigacion/gestoresreferenciasbibliograficas>.

CARREÑO, M.L., 2014. Análisis comparativo de los gestores bibliográficos sociales Zotero, Docear y Mendeley: características y prestaciones. *Cuadernos de Gestión de Información*, vol. 4, pp. 51-66.

CARRILLO PEREZ, 2008. *No Title*. S.l.: s.n.

CASTAÑEDA, C.M.V., 2017. *Gestores bibliográficos: recomendaciones para su aprovechamiento en la academia*. Medellín, Colombia: s.n. ISBN 9789585623309.

CHILE, U. de, 2014. Gestor de citas bibliográficas EndNote Basic. *Informatica y Bibliotecas*,

CHIU, C.C., ALBERTO, I. y CARBAJAL, T., 2015. Las Pruebas En El Desarrollo De Software. , pp. 46.



- CINCA, C.S., CALLÉN, Y.F. y MOLINERO, C.M., 2005. *Measuring DEA efficiency in Internet companies*. S.I.: s.n. ISBN 01679236 (ISSN).
- CORDÓN GARCÍA, J.A., MARTÍN RODERO, H. y ALONSO ARÉVALO, J., 2009. Gestores de referencias de última generación: análisis comparativo de RefWorks, EndNote Web y Zotero. *El Profesional de la Información*, vol. 18, no. 4, pp. 445-454. ISSN 1386-6710. DOI 10.3145/epi.2009.jul.14.
- CÓRDOVA, B. universitaria de, 2015. Gestores de referencia. Citas y referencias. ,
- COREL, 2018. ReviverSoft. [en línea]. [Consulta: 8 mayo 2018]. Disponible en: <https://www.reviversoft.com/es/file-extensions/ris>.
- FERNÁNDEZ, M., 2009. La importancia de las referencias bibliográficas y las citas en la elaboración de documentos y trabajos científicos y/o académicos. , pp. 5.
- FLACSO, B.I., 2010. Estilos Bibliográficos. *Biblioteca Iberoamericana-FLACSO*. S.I.:
- GALLEGOS, M. et all, PERALTA, C. y GUERRERO, W., 2017. Utilidad de los Gestores Bibliográficos en la Organización de la Información para Fines Investigativos. *Formación universitaria*, vol. 10, no. 5, pp. 77-87. ISSN 0718-5006. DOI 10.4067/S0718-50062017000500009.
- GROSSO, A., 2011. Patrones GRASP. ,
- GUEVARA, A.F.L. de y SOLER, Z.R., 2015. *Módulo Referencias Bibliográficas del Sistema para repositorios digitales REPXOS 3.0*. S.I.: Universidad de las Ciencias Informáticas.
- IVAR JACOBSON, 2000. *El Proceso Unificado de Desarrollo de Software (Spanish Edition)*. Addison, W. Madrid: s.n. ISBN 9788478290369.
- JIMENEZ, E.G., 2012. Desarrollo de software con netbeans. ,
- KRÄMER, M., 2016. citeproc-java. [en línea]. [Consulta: 22 mayo 2018]. Disponible en: <https://michelkraemer.github.io/citeproc-java/>.
- LARMAN, C., 2003. *UML y Patrones*. segunda. Madrid: s.n. ISBN 0130925691.
- MACHADO, A.V. y BENÍTEZ, J.H., 2016. Herramienta para el procesamiento en lote de documentos



digitales para el sistema REPXOS. , pp. 2015.

MANAGER, R. y WENDLING, D., 2012. Principales estilos de citas bibliográficas. , pp. 6-7.

MEDINA, A.C., 2015. *ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE*. 2015. La Habana: s.n.

MONTOYA, J.A.F., 2015. Más allá del OPAC: tecnología y nuevos soportes del catálogo de la biblioteca. *Cuadernos de documentación multimedia*, no. January 1997. ISSN 1575-9733.

MORA, R.C., 2007. Patrones de GRASP. *Adictos al trabajo* [en línea]. [Consulta: 24 mayo 2018].
Disponible en: <http://www.adictosaltrabajo.com/tutoriales/grasp/>.

MORENO, J.A.L. y ALONSO, F.Z., 2014. Sistema Integrado de Gestión Bibliotecaria para Extremadura en software libre. *Cuadernos de Documentación Multimedia*, vol. 25, no. 0, pp. 18-24. ISSN 1575-9733.
DOI 10.5209/rev_CDMU.2014.v25.47473.

NAVARRO, F., FRANCO, Y., AVILA, C., BACCA, J., BALDIRIS, S. y SALAS, D., 2013. Gestión de referencias bibliográficas dentro de una plataforma colaborativa basada en el enfoque de comunidades de práctica en el contexto educativo. *IV ENCUENTRO INTERNACIONAL VIRTUAL EDUCA COLOMBIA. 2013*. S.l.: s.n.,

ORALLO, E.H., 2010. Unificado de Modelado (UML). , pp. 1-6.

ORTIZ CHACHA, C.S., 2013. *Estilo de citación Chicago*. Patricia P. Xalapa: s.n. ISBN 1870 3267.

PARADIGM, M. de V., 2010. No Title. ,

PARADIGM, V., 2014. Guión Visual Paradigm for UML Índice. . S.l.:

PRESSMAN, R.S., 2004. *Ingeniería del software. Un enfoque práctico*. S.l.: s.n. ISBN 0196-2892 VO - 42.

PRESSMAN, R.S., 2010. Ingeniería de Software, Un enfoque práctico. ,

RAMÍREZ, V., 2010. *Lenguajes*. 2010. S.l.: s.n.

RIVERA, H.E.G., 2012. *Solución para la referencia bibliográfica en el Sistema Integrado de Gestión Bibliotecaria de la Universidad de las Ciencias Informáticas*. S.l.: Universidad de las Ciencias



Informáticas.

RIVERA, M.E.G. y LEÓN, P.P., 2013. Universidad Veracruzana. *Revista Medica de la Universidad Veracruzana*, vol. 13, no. 2, pp. 7-11.

RODRÍGUEZ, T., 2014. Metodología de desarrollo para la Actividad productiva de la UCI. , pp. 1-16.

RUIZ, L., 2007. Sistema Gestor de Referencias Bibliográficas Integrado. ,

SOMMERVILLE, I., 2005. *Ingrniería del software*. Madrid: s.n. ISBN 8478290745.

SPARKS, G., 2004. El modelo de casos de uso. , pp. 15.

WAHID, H. y AHMAD, S., 2017. Fundamentos de Ingeniería de Software. *Jurnal Ekonomi Malaysia*, vol. 51, no. 2, pp. 39-54. ISSN 01261962. DOI 10.1017/CBO9781107415324.004.



Bibliografía

- GALLEGOS, M. et all, PERALTA, C. y GUERRERO, W., 2017. Utilidad de los Gestores Bibliográficos en la Organización de la Información para Fines Investigativos. *Formación universitaria*, vol. 10, no. 5, pp. 77-87. ISSN 0718-5006. DOI 10.4067/S0718-50062017000500009.
- ALICANTE, I. de, 2005. Estilo Harvard. *Universidad de Alicante*, pp. 0-42.
- ALICANTE, U. de, 2018. Gestores de Referencias Bibliográficas. [en línea]. [Consulta: 5 febrero 2018]. Disponible en: <https://biblioteca.ua.es/es/estudia-y-aprende/elaboracion-y-gestion-de-citas-bibliograficas/gestores-de-referencias-bibliograficas.html>.
- ALL, I.A., 2012. *Introducción al lenguaje Java*. 2012. S.l.: s.n.
- ÁLVAREZ, C., 2014. Patrones de Diseño (Active Record vs DAO). *GENBETA* [en línea]. [Consulta: 24 mayo 2018]. Disponible en: <https://www.genbetadev.com/java-j2ee/patrones-de-diseno-active-record-vs-dao>.
- BIBLIOTECA UNIVERSITARIA, U. del P.V., 2015. Google Académico Guía De Uso. ,
- BORETTO, J., 2018. Gestores de Referencias Bibliográficas. *Rev. Asoc. Argent. Ortop. Traumatol*, pp. i-i. ISSN 1852-7434.
- BURGOS, L., 2015. Patrones de Diseño en Java. *GitHub* [en línea]. [Consulta: 24 mayo 2018]. Disponible en: <https://github.com/LuisBurgos/design-patterns#singleton>.
- CADIZ, U. de, 2018. Gestores de Referencias Bibliográficas. [en línea]. Disponible en: <http://biblioteca.uca.es/aprendizajeinvestigacion/gestoresreferenciasbibliograficas>.
- CARREÑO, M.L., 2014. Análisis comparativo de los gestores bibliográficos sociales Zotero, Docear y Mendeley: características y prestaciones. *Cuadernos de Gestión de Información*, vol. 4, pp. 51-66.
- CARRILLO PEREZ, 2008. *No Title*. S.l.: s.n.
- CASTAÑEDA, C.M.V., 2017. *Gestores bibliográficos: recomendaciones para su aprovechamiento en la academia*. Medellín, Colombia: s.n. ISBN 9789585623309.
- CHILE, U. de, 2014. Gestor de citas bibliográficas EndNote Basic. *Informática y Bibliotecas*,
- CHIU, C.C., ALBERTO, I. y CARBAJAL, T., 2015. Las Pruebas En El Desarrollo De Software. , pp. 46.
- CINCA, C.S., CALLÉN, Y.F. y MOLINERO, C.M., 2005. *Measuring DEA efficiency in Internet companies*.



S.I.: s.n. ISBN 01679236 (ISSN).

CORDÓN GARCÍA, J.A., MARTÍN RODERO, H. y ALONSO ARÉVALO, J., 2009. Gestores de referencias de última generación: análisis comparativo de RefWorks, EndNote Web y Zotero. *El Profesional de la Información*, vol. 18, no. 4, pp. 445-454. ISSN 1386-6710. DOI 10.3145/epi.2009.jul.14.

CÓRDOVA, B. universitaria de, 2015. Gestores de referencia. Citas y referencias. ,

COREL, 2018. ReviverSoft. [en línea]. [Consulta: 8 mayo 2018]. Disponible en:
<https://www.reviversoft.com/es/file-extensions/ris>.

FERNÁNDEZ, M., 2009. La importancia de las referencias bibliográficas y las citas en la elaboración de documentos y trabajos científicos y/o académicos. , pp. 5.

FLACSO, B.I., 2010. Estilos Bibliográficos. *Biblioteca Iberoamericana-FLACSO*. S.I.:

GALLEGOS, M. et all, PERALTA, C. y GUERRERO, W., 2017. Utilidad de los Gestores Bibliográficos en la Organización de la Información para Fines Investigativos. *Formación universitaria*, vol. 10, no. 5, pp. 77-87. ISSN 0718-5006. DOI 10.4067/S0718-50062017000500009.

GROSSO, A., 2011. Patrones GRASP. ,

GUEVARA, A.F.L. de y SOLER, Z.R., 2015. *Módulo Referencias Bibliográficas del Sistema para repositorios digitales REPXOS 3.0*. S.I.: Universidad de las Ciencias Informáticas.

IVAR JACOBSON, 2000. *El Proceso Unificado de Desarrollo de Software (Spanish Edition)*. Addison, W. Madrid: s.n. ISBN 9788478290369.

JIMENEZ, E.G., 2012. Desarrollo de software con netbeans. ,

KRÄMER, M., 2016. citeproc-java. [en línea]. [Consulta: 22 mayo 2018]. Disponible en: <https://michelkraemer.github.io/citeproc-java/>.

LARMAN, C., 2003. *UML y Patrones*. segunda. Madrid: s.n. ISBN 0130925691.

MACHADO, A.V. y BENÍTEZ, J.H., 2016. Herramienta para el procesamiento en lote de documentos digitales para el sistema REPXOS. , pp. 2015.

MANAGER, R. y WENDLING, D., 2012. Principales estilos de citas bibliográficas. , pp. 6-7.

MEDINA, A.C., 2015. *ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE*. 2015. La Habana: s.n.

MONTOYA, J.A.F., 2015. Más allá del OPAC: tecnología y nuevos soportes del catálogo de la biblioteca. *Cuadernos de documentación multimedia*, no. January 1997. ISSN 1575-9733.



- MORA, R.C., 2007. Patrones de GRASP. *Adictos al trabajo* [en línea]. [Consulta: 24 mayo 2018].
Disponible en: <http://www.adictosaltrabajo.com/tutoriales/grasp/>.
- MORENO, J.A.L. y ALONSO, F.Z., 2014. Sistema Integrado de Gestión Bibliotecaria para Extremadura en software libre. *Cuadernos de Documentación Multimedia*, vol. 25, no. 0, pp. 18-24. ISSN 1575-9733. DOI 10.5209/rev_CDMU.2014.v25.47473.
- NAVARRO, F., FRANCO, Y., AVILA, C., BACCA, J., BALDIRIS, S. y SALAS, D., 2013. Gestión de referencias bibliográficas dentro de una plataforma colaborativa basada en el enfoque de comunidades de práctica en el contexto educativo. *IV ENCUENTRO INTERNACIONAL VIRTUAL EDUCA COLOMBIA. 2013*. S.l.: s.n.,
- ORALLO, E.H., 2010. Unificado de Modelado (UML). , pp. 1-6.
- ORTIZ CHACHA, C.S., 2013. *Estilo de citación Chicago*. Patricia P. Xalapa: s.n. ISBN 1870 3267.
- PARADIGM, M. de V., 2010. No Title. ,
- PARADIGM, V., 2014. Guión Visual Paradigm for UML Índice. . S.l.:
- PRESSMAN, R.S., 2004. *Ingeniería del software. Un enfoque práctico*. S.l.: s.n. ISBN 0196-2892 VO - 42.
- PRESSMAN, R.S., 2010. Ingeniería de Software, Un enfoque práctico. ,
- RAMÍREZ, V., 2010. *Lenguajes*. 2010. S.l.: s.n.
- RIVERA, H.E.G., 2012. *Solución para la referencia bibliográfica en el Sistema Integrado de Gestión Bibliotecaria de la Universidad de las Ciencias Informáticas*. S.l.: Universidad de las Ciencias Informáticas.
- RIVERA, M.E.G. y LEÓN, P.P., 2013. Universidad Veracruzana. *Revista Medica de la Universidad Veracruzana*, vol. 13, no. 2, pp. 7-11.
- RODRÍGUEZ, T., 2014. Metodología de desarrollo para la Actividad productiva de la UCI. , pp. 1-16.
- RUIZ, L., 2007. Sistema Gestor de Referencias Bibliográficas Integrado. ,
- SOMMERVILLE, I., 2005. *Ingeniería del software*. Madrid: s.n. ISBN 8478290745.
- SPARKS, G., 2004. El modelo de casos de uso. , pp. 15.
- WAHID, H. y AHMAD, S., 2017. Fundamentos de Ingeniería de Software. *Jurnal Ekonomi Malaysia*, vol. 51, no. 2, pp. 39-54. ISSN 01261962. DOI 10.1017/CBO9781107415324.004.



Anexos

Anexos 1. Entrevista

Entrevista realizada el 23 de octubre del 2017 a especialistas del proyecto ABCD, en el Laboratorio de Producción 102 perteneciente al Centro CIGED.

1. ¿Qué procesos se desarrollan en el ABCD?
2. ¿Qué problemática presenta ABCD en cuanto a la generación de referencias bibliográficas?
3. ¿Qué funcionalidades son necesarias implementar para satisfacer esta problemática?
4. ¿Cuáles son las principales tecnologías y herramientas que se utilizaron para desarrollar el sistema ABCD 3.0?



Anexos 2. Prueba de aceptación



Acta de aceptación

ACTA DE ACEPTACIÓN

En cumplimiento del trabajo de diploma **Componente para la Gestión de las Referencias Bibliográficas en el sistema ABCD 3.0** se hace entrega de los productos que se relacionan a continuación:

- *Código del componente.*
- *Artefactos generados durante las fases como parte de la metodología AUP.*

El Jefe del proyecto ABCD y el jefe de Departamento Desarrollo de Componentes, luego de haber revisado los productos de trabajo determina que se **aceptan** y que el mismo cumple con los requisitos definidos y con la calidad necesaria para ser integrada al proyecto.

Entrega

Nombre y apellidos: Maydelin Alvarez Morales

Cargo: Autora del trabajo de diploma

Firma:

Recibe:

Nombre y apellidos: Luis Carlos Álvarez Fernández

Cargo: Jefe de Departamento Desarrollo de componentes

Firma:

Nombre y apellidos: Aniel Sánchez Verdecia

Cargo: Jefe de Proyecto ABCD.

Firma:

Fecha: 11 de junio 2018

