Universidad de las Ciencias Informáticas

Facultad de Ciencias y Tecnologías Computacionales

Título: Sistema de gestión para el almacén de libros de la Facultad de Ciencias y Tecnologías Computacionales

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

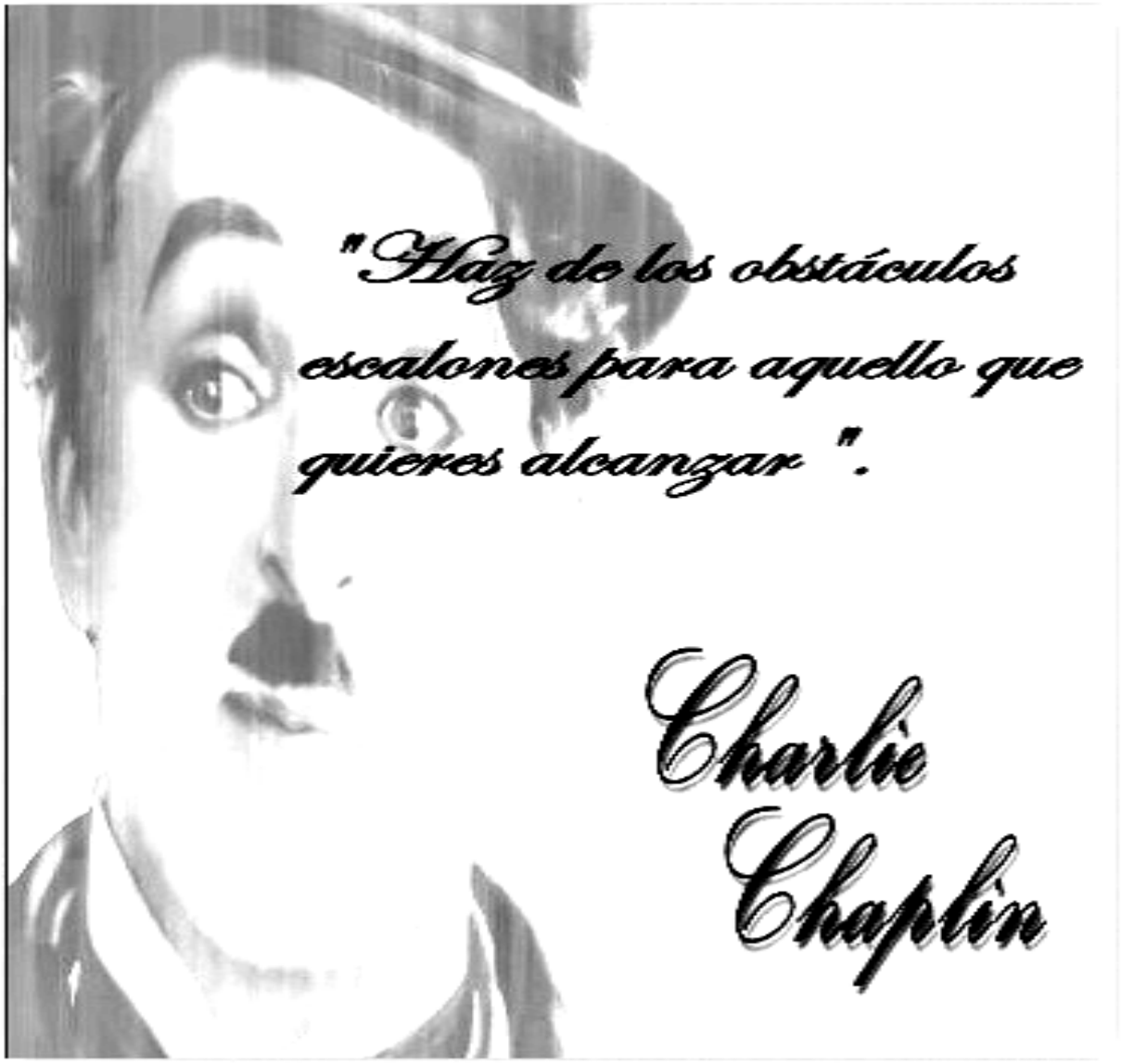
Autores: Asiel Riverón Hernández
Carlos Leandro Milán González

Tutores: MSc. Omar Mar Cornelio
Ing. Alex Brito de las Cuevas

La Habana, junio de 2017

"Año 59 de la Revolución"

Pensamiento:



DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 junio del año 2017.

Asiel Riverón Hernández

_____
Firma del Autor

Carlos Leandro Milán González

_____
Firma del Autor

MSc. Omar Mar Cornelio

_____
Firma del Tutor

Ing. Alex Brito de las Cuevas

_____
Firma del Tutor

DATOS DE CONTACTO

Autor:
Asiel Riverón Hernández
Universidad de las Ciencias Informáticas (UCI)
Correo electrónico: ariveronh@estudiantes.uci.cu

Autor:
Carlos Leandro Milán González
Universidad de las Ciencias Informáticas (UCI)
Correo electrónico: clmilan@estudiantes.uci.cu

Tutor:
MSc. Omar Mar Cornelio
Universidad de las Ciencias Informáticas (UCI)
Correo electrónico: omarmar@uci.cu

Tutor:
Ing. Alex Brito de las Cuevas
Universidad de las Ciencias Informáticas (UCI)
Correo electrónico: adelascuevas@uci.cu

AGRADECIMIENTOS

Asiel Riverón Hernández:

Agradecer ante todo a dios por ser mi pie de apoyo para lograr todas mis metas.
A mi papá por ser mi guía, por cada gota de sudor que derramó para que yo pudiera cumplir mis sueños, por formarme rodeado de felicidad y amor, siempre preocupado, trabajador, empeñado en que no nos faltara nada, donde quiera que esté sé que está muy orgulloso de su hijo. A mi mamá por su apoyo incondicional, por siempre apoyarme en todo, por su paciencia infinita, porque ha sido madre y padre a la vez y a pesar de eso ha sabido llevarme por el buen camino, porque sin ti hoy mis sueños no se estarían realizando. A mi hermana Linet por ser la mejor hermana del mundo, y aunque siempre nos peleábamos por cualquier cosa no la cambiaría por ninguna, a mi sobrino que, aunque su corazoncito late en el interior de un vientre ya significa mucho para mí. A mis abuelitas Ida y Ondina que siempre me dieron todo su amor y apoyo para continuar mi carrera. A mis tías y tíos Niurka, Mayito, Dairy, Lety, Pichi, Gelvis, el Bolo, Carlos, Marlenis que me han ayudado siempre a seguir adelante con sus consejos y su apoyo. A mis primos y primas que no menciono nombres porque son muchos, gracias por siempre darme alegría y estar para mí. A Caruchi y Omar por su cariño, agradecido toda la vida pues siempre han estado presentes brindándome su apoyo. ¡Gracias!! A mi familia en general a los que siempre llevo presente. A los que ya no están entre nosotros en especial a mis dos abuelos.
A mi compañero de tesis Leandro que a pesar de sacarme del paso algunas veces siempre enfrentamos las dificultades juntos para poder realizar nuestros sueños. A todas mis amistades de Camajuaní y de la UCI, en especial a Anita, Jorgito, Luis, Yairelis, Ginari por siempre estar presentes para mí en los momentos buenos y malos. A mis vecinos del barrio por siempre estar al tanto de mi carrera.
A mi grupo porque de cada uno aprendí durante estos años de carrera, en especial a Yanelis (Coli) que desde que entramos a esta universidad estamos juntos. A todos los profesores que tuvieron que ver con mi formación durante estos años de estudio por sus enseñanzas.
A mis amigos de la radio y la televisión de la UCI por permitirme ser uno más de ustedes y hacer realidad otro de mis grandes sueños.
A mis tutores, al oponente y al tribunal por su ayuda, sus consejos y sus exigencias para terminar con éxito nuestro trabajo de diploma.
A la UCI por acogerme, por hacerme sentir como en mi casa, por la formación recibida.
A todas las personas que he conocido en estos años y que de una forma u otra me ayudaron y me apoyaron. Muchas gracias.

Carlos Leandro Milán González:

De la UCI mis agradecimiento y mis recuerdos para Teudys, Saidy, Miguel Antonio, Rolando, Yosvany, Lester Nieto, Reinier, Marlon, Yazmin, Christopher, Mirislaydis, María Elena, María Cristina Hidalgo, Grendy, Adrián Carmona, Rolyen, Yunior, Rodolfo, Jorge Luis Espinosa, Rubert, Osmín, Karel, Cano, Idel, Suinny, Lenia, Lizsandra, José y Diana, Raimel, Jorgito, León, Medina, Ariam Lázaro, Asley, Adolfo N., L.Y. Maceo, Gisell, Luis M. (Yonky), Maibol, Milagros, Ramiro, Midalis y Yanelis, Lijandy, Aliú y Elena, Yisslen, Niurka, Yudeisy, Aldy, Leidy, Virmary, Laura Elena, Alexey, Ariel S. Cruz, Magdiel, Raciel, Carlos Luis, Dieter, Odiel, Enier, Ester Fabiola, Zayas, Raymundo, Irina, Ismael, Jorge Emilio, Jorge Luis, Dianet, José Carlos, José Leandro, Lázaro, Leonardo, Castaño, Argota, Manuel, María Cristina, Vilmavis, Willian, Elenny, Yanet Parra, Yanelis Ramírez, Yoamel, Yordany, Yulina, Armando, Yudiel Larrazabal, Yurima, Yunior Mesa, Lisbeth, Puebla, Zaralety, Dayron, Tellez, Themis, Eridniel,

Frank Alain, Yulién Miguel Ángel, Daynelis, Alfredo Trujillo, Jairo, entre otros. // A Alex Brito, Omar Mar, Alberto Garnache, Félix Glez., Flavio, Dayana, Yadira Robles, Liniuska, Ariel Manresa, Glennis, Irela, Barroso y Karel Rguez.
A mis profes Lázara, Xiomara, Francisco, Annia, Flavio, entre otros. A René, por ayudarme en todas las matemáticas desde la secundaria hasta la universidad, gracias por siempre decir que sí.
A mi amigo, compañero de cuarto y de tesis, Asiel, gracias por estos meses donde se consolidó nuestra amistad y por el documento.
A Patricia (mi Kitty) y Yandy gracias por su amistad y acompañamiento en los momentos más duros, por su cariño; los quiere, el Padri.
A Ismary por su apoyo durante la carrera, amor, incondicionalidad, por Manguito y por tu familia, por la compañía, etc, por todo, gracias.
A mis amistades y su familia: Popó (Juan Carlos), Pedro Luis, Mercedes, Yoel Fundora, Miguel Domínguez, Dyeter Díaz, Lina Vera, Dayron Herrara, Alexis Núñez, Alexander Nieto, Lester Castillo y Magalis, Ariam Gamel y su mamá, Mairelis, Alex Lafargue, Marimar, Yanet Ávila, a mi doctor y amigo Luis Alberto, Elaine y Queta; a Alberto y María Elena, a Alberto Valladares y Odalys, a Roberto y Deisy, Verdecia, Ángel Luis, Mireya Carballo, Yusimí Caraballo, Ana y De Jongh, Alejandrito, Marta, Diosdado, Caridad (fotógrafa), Lázaro, Yaquelín y Deisy, Chamizo, Torres, Consuelo, Martell, Lien.
A mis tíos Omar y Marieta, mi querida profe Sonia Morales, a Pedro, Amarilis, Danielys (Titi), Justina, Cristina, Irene, Victoria, Barbarita.
A mi familia de Campechuela (Clara y Roberto, Ana, Yola, Luis Miguel, Miguelito, Roxana); a la familia de Alfre; a Grisel, Isora, a José Luis gracias por mi infancia, a Lilian mi hermana, a mi gente de San Miguel del Padrón, a Jorge Correoso, a mis terceros abuelos Jorge y Maritza, a Gustavito, Aliche, Gustavito Jr., mis tíos Sandra e Idael (el hombre que más quiero), a mi abuelita Felina y a mi prima-hermana bella Lisandra y mi sobrina Paola a todos los quiero mucho, mucho, mucho. ¡A mi madre María Cristina González Martínez, la mujer más importante de mi vida y la que más quiero, le debo todo desde la vida hasta el empeño de que fuera universitario, a ti porque siempre estarás a mi lado, por la vida feliz que he tenido, por tu amor y paciencia, gracias!

DEDICATORIA:

Dedico esta tesis a mi papá y a mi mamá porque sin ellos no sería quién soy hoy. Gracias por ser los mejores padres del mundo. **Asiel**

A mi familia, a mi madre y a todas aquellas personitas que ya no están, mi primo Miguel, mi abuelo Gustavo, a Marita, a Naivi y Félix Daniel. **Carlos Leandro**

RESUMEN

Con el desarrollo de las nuevas tecnologías de la informática y las comunicaciones, diversos procesos que se llevan a cabo en las organizaciones se han favorecido, convirtiéndose en un factor determinante en la eficiencia y organización del trabajo. La presente investigación descriptiva comprende el registro, análisis e interpretación de las características y naturaleza actual de los procesos del almacén de libros de la Facultad de Ciencias y Tecnologías Computacionales (CITEC). En dicho almacén se han detectado varias deficiencias que en su conjunto influyen negativamente en las tareas que se llevan a cabo en dicha área. El control y gestión de los textos de las asignaturas y los procesos comprendidos en la tramitación de préstamos y devoluciones de libros se realizan manualmente manifestándose en ocasiones, múltiples problemas de tiempo en la atención al cliente y la veracidad de la información. La presente investigación tiene como objetivo realizar el análisis, diseño, implementación y evaluación de una solución que integre e informatice estos procesos para favorecer la eficiencia y el apoyo a la toma de decisiones. Para llevar a cabo la propuesta de solución se utilizó como metodología de desarrollo ágil AUP en su variación UCI, como Entorno de Desarrollo Integrado NetBeans IDE 8.0, como lenguaje de modelado UML 2.0, como Sistema Gestor de Base de Datos PostgreSQL 9.4 y como principal tecnología de desarrollo el marco de trabajo Symfony 2.8 haciendo uso de PHP 5.5 como lenguaje de programación.

PALABRAS CLAVE: Libros, gestión, préstamos, toma de decisiones.

ABSTRACT

With the development of new information and communication technologies, various processes carried out in organizations have been favored, becoming a determining factor in the efficiency and organization of work. This descriptive research includes the recording, analysis and interpretation of the characteristics and current nature of the book store processes of the Faculty of Sciences and Computational Technologies (CITEC). In this warehouse there have been several deficiencies that together have a negative impact on the tasks carried out in this area. The control and management of the texts of the subjects and the processes included in the processing of loans and book returns are carried out manually, sometimes manifests, multiple problems of time in the attention to the client and the veracity of the information. The objective of this research is to analyze, design, implement and evaluate a solution that integrates and informatizes these processes to favor efficiency and support to decision making. In order to carry out the proposal of solution, it was used as an agile AUP development methodology in its UCI variation, as NETBEANS IDE 8.0 Integrated Development Environment, as a UML 2.0 modeling tool, as PostgreSQL 9.4 Database Management System and as the main technology Development of the Symfony 2.8 framework by making use of PHP 5.5 as a programming language.

Key Words*: Books, management, loans, decision making.*

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

A lo largo de la historia el hombre ha necesitado almacenar y transmitir datos e información, para esto se ha valido de distintos herramientas y mecanismos. Estos mecanismos se fueron perfeccionando con los avances de la ciencia y la tecnología, y evolucionan hacia un desarrollo tecnológico que puso de manifiesto la necesidad, el interés y el ingenio del hombre en el camino evolutivo. Con el desarrollo humano surge la informática, ciencia que estudia el tratamiento automático de la información en computadoras, dispositivos electrónicos y sistemas informáticos (Alegsa, 2016).

En Cuba desde hace varios años se lleva a cabo un proceso paulatino de informatización. Ejemplo claro lo constituye la Universidad de las Ciencias Informáticas (UCI), con el objetivo principal de formar jóvenes; y aplicar sus conocimientos en función de informatizar la sociedad cubana. Para lograr dicho objetivo, se planifica dentro del proceso de formación del estudiante en la UCI el uso de las nuevas tecnologías de la informática y las comunicaciones (TIC). Un gran apoyo a esta formación lo constituye igualmente los libros, ya que la bibliografía básica de las asignaturas se encuentra en formato físico y se cuenta con más de 100 títulos en función para la formación de estudiantes y profesores de esta casa de altos estudios. Por esa razón, existe en cada facultad un almacén de libros donde los estudiantes adquieren los volúmenes de las asignaturas que cursan como apoyo al proceso docente-educativo.

El almacén perteneciente a la Facultad CITEC cuenta con una especialista la cual conduce el proceso de gestión de los libros en su mayoría de forma manual; es por eso que la gestión de los procesos como préstamo y devolución de libros, despacho, entrega y recepción de materiales o reportes administrativos resultan lentos y en ocasiones ineficientes si se tiene en cuenta la posible ocurrencia de errores de naturaleza humana o la falta de materiales debido a que la manualidad impera en la forma de trabajo actual.

Al no existir procesamiento informatizado, existe un grupo de informaciones con las que se puede hacer más efectivo el control de los libros como puede ser la cantidad de volúmenes por año académico, profesor o departamento, el estado de los libros por un número que lo caracterice, la cantidad de libros en manos de estudiantes y profesores, el número de títulos nacionales y extranjeros entre otros; que para su obtención se necesitaría de un enorme esfuerzo del personal que labora en el almacén pues habría que consultar cada unidad y repetir una y otra vez los procedimientos. El Vicedecanato de Administración de la Facultad CITEC también necesita este tipo de información para realizar controles sistemáticos y obtener estadísticas de consumo. Para realizar estos análisis, es necesario buscar en distintos documentos almacenados para: realizar estudios y tomar decisiones a partir del análisis de los indicadores mencionados anteriormente. Estos datos no se encuentran

almacenados en ningún sistema informático con un formato común que los caracterice y en muchas ocasiones se encuentran duplicados, debido a lo complejo que resulta trabajar con grandes armarios llenos de documentación.

La actual estructura de trabajo conlleva una alta dependencia al papel y otros elementos de oficina provocando altos costos para la universidad. Esto se debe a la multitud de procesos internos, desde la recepción hasta el archivado. La disminución de la dependencia al papel y otros elementos de oficina traería un notable ahorro de recursos para la universidad.

Los argumentos planteados anteriormente arrojan un grupo de irregularidades presentes en el almacén de libros de la facultad como son:

- El actual proceso de gestión de la información, que funciona en el depósito de libros, no proporciona exactitud en las estimaciones de los recursos necesarios para el trabajo de oficina provocando un gasto mayor de materiales y tiempo.
- La solicitud de libros al almacén central se realiza a partir de los reportes de inventarios que informa el almacén sobre sus necesidades, lo que propicia pedidos de libros innecesarios.
- El proceso de búsqueda o actualización de alguna información en las diversas tarjetas de control es un trabajo lento y engorroso provocando atrasos y dificultad en el trabajo.

Teniendo en cuenta los elementos planteados anteriormente se define como **problema de la investigación:** ¿Cómo contribuir al proceso de gestión de la información en el almacén de libros de la Facultad de Ciencias y Tecnologías Computacionales para un control más eficiente de los procesos? En tanto, el **objeto de estudio** se centra en el desarrollo de Sistemas de Gestión; mientras que el **campo de acción** se enmarca en los sistemas para la gestión de activos fijos tangibles.
Por lo anterior expuesto se formula como **objetivo general**: Desarrollar un sistema para la gestión de los procesos del almacén de libros de la Facultad CITEC.

Para dar cumplimiento al objetivo general se definen los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Realizar el modelado del sistema.
- Implementar un sistema informático que cumpla con los requisitos definidos.
- Validar el sistema mediante pruebas para garantizar la fiabilidad y calidad del producto.

Para guiar el proceso investigativo se plantean las siguientes **preguntas científicas:**

- ¿Cuáles son los principales fundamentos teóricos-metodológicos en referencia a los sistemas de gestión de libros?

- ¿Cómo modelar un sistema para la gestión de la información referente a los libros que se ajuste a las características de los procesos que se llevan a cabo en el almacén de libros de la Facultad CITEC?
- ¿Qué tecnologías y herramientas de desarrollo guiarán el proceso de implementación?
- ¿Qué técnicas de pruebas se pueden utilizar para la validación de la solución y como aplicarlas?

Con el propósito de dar cumplimiento a lo anteriormente planteado, se elaboraron las siguientes **tareas de la investigación**.

- Elaborar el estudio del arte sobre los sistemas de gestión como herramientas para mejorar los procesos de un negocio.
- Caracterizar las metodologías, herramientas y tecnologías de desarrollo más utilizadas para escoger cual aplicar en el proceso de implementación.
- Realizar el modelado del negocio para una mejor comprensión de los procesos del almacén.
- Identificar las principales funcionalidades con que contará el sistema para la posterior implementación del mismo.
- Investigar los diferentes tipos de pruebas funcionales, para su posterior aplicación sobre el sistema desarrollado.

Durante la investigación se utilizaron los siguientes **métodos científicos**:
Los métodos teóricos se utilizan en la adopción de la teoría científica y en el enfoque general para abordar los problemas científicos, posibilitando la interpretación conceptual de los datos empíricos y la explicación de los hechos. Están presentes en los diversos momentos del proceso de investigación y durante el análisis y comprensión de la información utilizada para el proceso de desarrollo (Coello González & Hernández León, 2011).

Métodos Teóricos

- Histórico Lógico: Se realiza un estudio del estado del arte sobre los proyectos, soluciones y resultados en el mundo bibliotecario que contribuyan a definir y guiar la realización de este trabajo.
- Analítico Sintético: Análisis de la documentación y en diversas fuentes bibliográficas sobre la implementación de Sistemas de Gestión y el uso de herramientas y tecnologías en función de ellos.
- Modelación: se crean abstracciones con el propósito de explicar la realidad. El modelo es el sustituto del objeto de investigación. Se realizan los diagramas correspondientes a los modelos de análisis, diseño e implementación del Módulo de Control para el sistema de gestión en cuestión.

Métodos Empíricos

- Entrevistas: Se realizan varias entrevistas de tipo estructuradas con el cliente y personas a fin con el negocio para identificar procesos existentes en el mismo y capturar los requisitos que son fundamentales en la implementación. Se anexa en el documento los cuestionarios realizados.

La estructura del Trabajo Investigativo quedó constituida en tres capítulos, conclusiones, referencias bibliográficas y anexos.

Capítulo 1: "Fundamentación Teórica de los sistemas de gestión de libros"

En este capítulo se exponen los principales conceptos asociados a la solución en cuestión. Se realiza un análisis de las principales soluciones existentes para así identificar posibles características de las mismas que puedan aportar al diseño y desarrollo de la aplicación. Por último, se define la metodología, así como las herramientas que se utilizarán en la implementación y el futuro despliegue.

Capítulo 2: "Análisis y Diseño de la Solución: Sistema de gestión para el almacén de libros de la CITEC"

En este capítulo se definirán las funcionalidades que se van a informatizar, se define el modelado del negocio y los diagramas para lograr una mejor comprensión del mismo. Además, se determinan y especifican los requisitos funcionales y no funcionales, se modela el sistema y se identifican los casos de uso con sus respectivos diagramas para conocer las funcionalidades del sistema. Se especificarán además los patrones utilizados, así como otras características fundamentales como parte del diseño, generándose además un grupo de artefactos teniendo en cuenta la metodología a utilizar.

Capítulo 3: "Implementación y pruebas del Sistema de gestión para el almacén de libros de la CITEC"

En este capítulo se describe el proceso de implementación y pruebas de la solución. Se especifican la selección, diseño y aplicación de las pruebas realizadas en correspondencia a la tipología de la solución. Finalmente se muestran los resultados obtenidos una vez concluida la ejecución de pruebas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE GESTIÓN DE LIBROS

En este capítulo se describen los principales aspectos y conceptos que se asocian al marco teórico referencial de la investigación. Se realiza una valoración de las principales soluciones existentes y se hace una descripción de la metodología, así como las herramientas y tecnologías que se emplean en el desarrollo de la solución.

1.1 Conceptos asociados al dominio del problema

A continuación, se exponen los principales conceptos relacionados con el marco teórico de la investigación, con el objetivo de profundizar en los distintos puntos de vista y asumir una posición al respecto.

1.1.1 Gestión de la información

Son múltiples los conceptos que existen y describen este término. La Dra. Gloria Irene Ponjuán Dante, profesora de la Universidad de la Habana aborda que *la Gestión de la información se refiere a la gestión que se desarrolla en un Sistema de Información, si se trata de que el sistema tenga como propósito obtener salidas informacionales*. Además, la define como el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales), para manejar información dentro y para la sociedad a la que sirve. Tiene como elemento básico la gestión del ciclo de vida de este recurso y ocurre en cualquier organización. Es propia también de unidades especializadas que manejan este recurso en forma intensiva, llamadas unidades de información (Dante, 1998).

Elusa Morales Flores plantea que *la finalidad de la Gestión de la información es ofrecer mecanismos que permitieran a la organización adquirir, producir y transmitir, al menos coste posible, datos e informaciones con una calidad, exactitud y actualidad suficientes para servir a los objetivos de la organización* (Flores, 2014).

A partir del análisis de los conceptos asociados se asume como gestión de la información a la correcta explotación de la información para el trabajo con la misma en las organizaciones enmarcada en su creación o adquisición, procesamiento y divulgación siendo determinante en la adecuada toma de decisiones.

Análisis de la información: Se conoce como Análisis de la información al proceso mediante el cual se definen las necesidades del estudio, se busca información, se validan las fuentes, se procesa la información, se realiza el análisis, la integración y se presenta el resultado (Actores, 2014).

Sistema de Gestión Estadística: Existen varios tipos de sistemas de gestión de información, que se clasifican de acuerdo a la información que procesan y los fines específicos de cada uno en particular. Una de estas clasificaciones es la correspondiente a los sistemas que gestionan información estadística, los cuales basan su funcionamiento en la colección e interpretación de datos cuantitativos obtenidos en un estudio dado, con el objetivo de presentar de manera ordenada los datos a través de tablas y gráficos estadísticos, resumirlos a un reducido número de medidas estadísticas para poder establecer comparaciones y finalmente estimar la probabilidad de éxito que tiene cada una de las decisiones posibles.

El Sistema de gestión de la información referente a los libros en el depósito existente en la Facultad CITEC asimila gran cantidad de reportes estadísticos como cantidad de clientes con préstamo de libros otorgados por año, número de libros en poder de los clientes o reportes administrativos como el volumen de libros a disposición del almacén.

1.2 Estado actual de la temática

A continuación, se exponen algunas de las soluciones existentes que facilitan la gestión de la información referente a los libros en el ámbito nacional e internacional y que permitirá conocer las características de los sistemas y el posible aporte a la solución.

1.2.1 Soluciones similares en la gestión de libros

En el mundo existen diversos sistemas que han tributado a la informatización de las bibliotecas. Para determinar las características que debe cumplir la aplicación que se desea desarrollar es necesario realizar un estudio y posterior análisis de los sistemas existentes tanto a nivel nacional e internacional como en la propia universidad. Dicha investigación tuvo como resultado la localización de varios sistemas existentes en universidades tanto de Cuba como en el extranjero y que tienen puntos en común con el sistema a desarrollar. Es necesario acotar que, en la investigación realizada no se encontró con sistemas que se especialicen en un almacén o depósito de libros en particular. A continuación, se realiza una descripción de estos referentes aportando las razones que avalan la necesidad de crear un nuevo sistema adaptado a las características y requisitos exigidos.

Sistema para la gestión de la biblioteca en la Universidad Católica del Ecuador

El sistema informático presente en la biblioteca de la Universidad Católica del Ecuador con sede en Santo Domingo de los Colorados optimiza el control y gestión de libros. La institución posee una biblioteca con más de 5000 ejemplares, la misma presta sus servicios a los docentes, estudiantes y también al resto de la sociedad. El proceso con que cuenta este sistema se enfoca en la gestión de

registros y préstamos de libros proporcionando mayor control de este proceso cuando se brinda a los usuarios (Sánchez & Sangucho, 2015).

Este software informático es privativo, no cumple con la mayor parte de los requisitos que requiere el sistema a desarrollar, además, a pesar de la calidad para la institución en la que funciona no se encuentra acorde al almacén de libros de la Facultad CITEC pues este solo ofrece servicios a los estudiantes y trabajadores de la facultad.

Biblioteca Visual

Es una herramienta web creada en la Universidad Carlos Rafael Rodríguez de Cienfuegos que automatiza los procesos sustantivos que se originan en la biblioteca en cuestión; dígase selección y adquisición de libros, catalogación y clasificación de las fuentes bibliográficas que se reciben en la biblioteca, estadísticas que se generan, entre otros.

El sistema informático organiza el trabajo atendiendo a las diferentes áreas que cuenta la biblioteca, facilitando así un mayor control para acceder a la información, permite un preciso control estadístico, cuenta con facilidades de búsquedas y una interfaz fácil de utilizar por cualquier usuario. Es importante destacar que gracias a este sistema se definen correctamente algunas áreas que no existían como es el caso del procesamiento estadístico (Hernández, 2012).

A pesar de contar con grandes ventajas, sobre todo en el área estadística de una biblioteca, este sistema no cumple con todos los requisitos deseados pues no tiene implementado la gestión del proceso por pérdidas de libros. El sistema está pensado para la biblioteca en cuestión, además, no cuenta con la posibilidad de realizar reportes, requisito fundamental de la nueva aplicación para el correcto control de los procesos presentes en el almacén de libros de la facultad.

Sistemas en la UCI

La investigación realizada en la Universidad de las Ciencias Informáticas arrojó la realización de dos proyectos estudiantiles que implementaron sistemas con algunas características similares en cuanto a su objetivo en comparación con el sistema a desarrollar. Ambos sistemas fueron implementados en cursos anteriores por estudiantes de las Facultades 2 y 5 especializándose en la Gestión de la información de los libros en la antigua Área de Aseguramiento Técnico Material (ATM). Con el paso de los años las tecnologías y herramientas utilizadas en aquel entonces se consideran obsoletas, a eso se le suma la opción de reportes con que contará el sistema y que las antiguas soluciones no tenían incorporadas, en la actualidad estos sistemas no están disponibles.

Valoración de los sistemas estudiados

Una vez analizadas cada una de las soluciones existentes planteadas se constató que las mismas, aunque presentan funcionalidades útiles que se pueden tomar como referencia para la solución en cuestión, no satisfacen todas las necesidades ya que de forma general:

- No cuentan con la posibilidad de generar reportes para un mayor control en el almacén.
- En el caso del sistema presente en la universidad de Ecuador se enfoca en brindar servicio a cualquier persona ajena o no la institución siendo además un sistema privativo por lo que su uso no es adaptable al contexto de la problemática.
- Las soluciones desarrolladas en la UCI no están disponibles en la actualidad debido a que fueron realizadas hace varios años por estudiantes que hoy no se encuentran en la institución, además venían enfocadas hacia el Área de Aseguramiento Técnico Material (ATM) cuando este realizaba la función de almacén para toda la universidad.

A partir de los análisis realizados se concluye con la necesidad desarrollar un sistema informático que permita un mayor control en los procesos y trámites que se realizan en el almacén de libros de la Facultad CITEC.

1.3 Metodología de desarrollo

Desarrollar productos de software con alta calidad depende de las actividades que conllevan a su construcción, donde la selección de la metodología más adecuada juega un papel importante. La correcta selección es fundamental para la planificación, gestión, control y evaluación de forma sistemática, lo que garantiza grandes probabilidades de éxito.

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente, donde predecir no significa perder la capacidad adaptativa, no significa evitar la introducción de cambios en los requisitos, ni evitar que nuevos requisitos surjan sino definir un camino reproducible para obtener resultados confiables. Definen, además, una representación que permite facilitar la manipulación de modelos, la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema (Gacitúa, 2013).

En la actualidad existen dos tipos de metodologías para el desarrollo de software, primeramente, se crearon las metodologías tradicionales que se caracterizan por una planificación total del trabajo y una rigurosa definición de artefactos, roles, actividades, así como una extensa documentación limitando incluso la habilidad del equipo de desarrollo.

En el caso de las metodologías ágiles se centran más en el factor humano. Su base está en la simplificación sin renunciar a las prácticas esenciales que aseguran la calidad del producto, esto se refleja en el hecho de ser la persona, el principal factor de éxito en un proyecto, la habilidad está en centrarse más en los cambios que puedan surgir que seguir estrictamente un plan.

Teniendo en cuenta los elementos planteados anteriormente se propone en la Tabla 1 la comparación entre ambas metodologías.

Tabla 1: Comparación entre metodologías.

Metodologías Tradicionales	Metodologías Ágiles
El trabajo se desarrolla en grupos numerosos de personas en ocasiones de forma distribuida.	El trabajo se enfoca a grupos de pocas personas.
El cliente no forma parte del equipo de desarrollo, solo está presente a través de reuniones planificadas.	El cliente si forma parte del equipo y del proceso de desarrollo con reuniones frecuentes e incluso informales.
Cierta resistencia a los cambios	Pueden aparecer cambios en el proceso de desarrollo y se toman si es necesario.
Los proyectos presentan gran cantidad de documentación y de forma detallada	La documentación se genera a medida que se hace necesaria y de manera general no es amplia ni rigurosa en su estructura.

Dada la comparación anterior y teniendo en cuenta todos los elementos planteados sobre ambas metodologías se considera y define para la propuesta de solución enmarcarse en una metodología ágil. Tal decisión se debe a que se tiene un equipo de desarrollo pequeño, el cliente es parte fundamental del equipo y tiene la disposición de mantener una comunicación fluida y constante durante el proceso de desarrollo. Además, se estará al tanto de todos los cambios que se puedan generar a medida que avance la ejecución de las tareas tomando las decisiones necesarias para bien del producto.

Actualmente el desarrollo de software dentro de la actividad productiva de la UCI se caracteriza por el uso de diferentes metodologías de desarrollo entre robustas y ágiles, entre las que se listan a continuación:

- XP
- OPEN UP
- RUP

- BPM
- DAC
- KIMBALL
- SXP
- SCRUM
- NOVA OPEN UP
- AUP

A pesar de la variedad de metodologías usadas, se ha comprobado que muy pocos proyectos la aplican en su totalidad. La diferencia entre estas metodologías no radica únicamente en los productos de trabajos que proponen o en sus roles, sino en su forma de planificar el proyecto y realizar las estimaciones del tiempo. Factor determinante en la culminación exitosa de todo desarrollo de software, por lo que uno de los principales problemas detectados es que sin importar la metodología que se usa se está planificando con un único cronograma tipo, además de forzar el método de estimación definido en la Universidad y que responde en su gran mayoría a la metodología RUP (Sánchez, 2014).

Es por esto que surge en la Universidad de las Ciencias Informáticas lo que se llama una variación al Proceso Unificado Ágil (AUP) de Scott Ambler la cual usa técnicas ágiles. Esta variación de AUP que se utiliza en la universidad cuenta con cuatro fases que transcurren de forma consecutiva.

1. **Inicio:** El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema.

2. **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.

3. **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.

4. **Transición:** El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación.

La metodología AUP en su variación UCI se adapta a la configuración y ciclo de vida de la actividad productiva de la UCI. El lenguaje de trabajo es totalmente entendible logrando con su ejecución un producto de software con la calidad requerida. Consta de los principales aspectos positivos de la

mayoría de las metodologías ágiles adaptándose correctamente a equipos de desarrollo pequeños; es por esto que se decide utilizar la metodología AUP en su variación UCI.

1.4 Herramientas y tecnologías a utilizar

A continuación, se describen las principales herramientas y tecnologías que se utilizarán para el desarrollo de la investigación y la implementación de la solución

1.4.1 Lenguaje de modelado UML 2.0

Para el desarrollo de la aplicación se utilizará UML 2.0 como el lenguaje de modelado con el que se realizarán los artefactos necesarios en el proceso de desarrollo del software.

UML es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Entrega una forma de modelar sucesos conceptuales como lo son procesos de negocio y funciones de sistema, además de sucesos concretos como lo son escribir clases en un lenguaje determinado, esquema de base de datos y componentes de software reusables (Histchfeld & Salinas Caro, 2014).

Entre sus principales características se encuentra:

- Permite modelar sistemas utilizando técnicas de programación orientadas a objeto.
- Permite documentar todos los artefactos de un proceso de desarrollo dígase requisitos, arquitectura, pruebas entre otros.
- Se logra especificar todas las decisiones de análisis, diseño e implementación mediante la construcción de modelos precisos y no ambiguos.

La utilización de este lenguaje gráfico viene dada por las bondades que brinda, además de utilizar como versión 2.0 la cual es soportada perfectamente para ser utilizada por Visual Paradigm para UML 8.0.

1.4.2 Herramienta de modelado

La herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software asistida por Computadora) para el modelado se selecciona el Visual Paradigm para UML 8.0, la misma soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue y contribuye a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Se caracteriza por el uso de un lenguaje estándar común al equipo de trabajo, que

facilita la comunicación entre sus integrantes, es una herramienta fácil de instalar y actualizar, genera código para varios lenguajes de programación y exporta en formato HTML, es una tecnología libre y está disponible en varios idiomas. Uno de los elementos importantes que le aportan distinción es la posibilidad que brinda de soportar aplicaciones web (Pressman, 2002).

Las principales características que avalan el uso de esta herramienta son:

- Entorno de creación de diagramas para UML 2.0
- Uso de un lenguaje estándar común para el equipo de desarrollo.
- Diseño enfocado en el negocio y centrado a casos de uso.
- Interoperabilidad entre diagramas permitiendo a partir de un diagrama obtener otro que guarde relación con este.
- Disponible en múltiples plataformas.
- Capacidad de Ingeniería directa e inversa.

1.4.3 Sistema Gestor de Base de Datos PostgreSQL 9.4

Un sistema gestor de base de datos (SGBD) es el programa o conjunto de programas que gestionan y mantienen consistentes los datos almacenados en la base de datos. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad (Bachman, 2015).

PostgreSQL es un sistema de gestión de bases de datos relacional orientada a objetos y de software libres (Martínez, 2010). Entre sus características se destacan:

- Gran escalabilidad. Capaz de ajustarse al número de Unidades Centrales de Procesamiento (del inglés *Central Processing Units* CPU) y a la cantidad de memoria que posee el sistema de forma óptima.
- Manejo de gran volumen de datos (GROUP, 2011).
- Ofrece la posibilidad de ejecutar y trabajar varios procesos al mismo tiempo sobre la misma tabla sin ser dañada.
- Tiene gran soporte para vistas, procedimientos ubicados en el servidor, transacciones y características orientadas a objetos

Se define esta herramienta para el desarrollo de la propuesta de solución debido a la estabilidad, potencia, robustez y facilidad de administración que provee.

1.4.4 Herramienta de administración de bases de datos PgAdmin III

PgAdmin III es una herramienta para la administración de bases de datos para PostreSQL de código abierto y cuenta con una interfaz gráfica que soporta todas las características. Esta herramienta simplifica en gran medida la administración.

Esta herramienta de software libre fue diseñada para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL (*Structured Query Language*) a la elaboración de bases de datos complejas. La conexión del servidor se puede realizar mediante TCP/IP o Unix Domain Sockets y puede ser cifrado mediante SSL (*Secure Sockets Layer*) por seguridad.

Dentro de las principales ventajas que se tuvo en cuenta para escoger esta herramienta se encuentran su seguridad de cara a los usuarios. Está diseñado para responder a las necesidades de todos los usuarios y permite hacer desde búsquedas SQL hasta desarrollar toda la base de datos de forma muy fácil e intuitiva; directamente desde la interfaz gráfica.

1.4.5 Servidor Web Apache 2.0

Un servidor web es un programa que gestiona en forma de páginas Web, hipertextos o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo HTTP (*Hypertext Transfer Protocol*). Con esto, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; conocido como un Navegador Web (Perea, 2009).

Apache 2.0 es un servidor web flexible, rápido y eficiente, actualizado continuamente y adaptado a los nuevos protocolos. Entre sus principales características destacan que es libre de costo, multiplataforma, soporte para las conexiones seguras SSL. Posee una arquitectura que permite ser adaptado a diferentes entornos y necesidades. Tiene ventajosas características entre las cuales se encuentran:

- Es robusto y seguro en su funcionamiento.
- Es totalmente gratuito y se distribuye bajo la licencia *Apache Software License*.
- Permite la modificación del código.
- Permite personalizar la respuesta ante los posibles errores que se pueden dar en el servidor.
- Admite una gran cantidad de lenguajes de script.

1.4.6 Entorno de Desarrollo Integrado NetBeans IDE 8.0

Un Entorno de Desarrollo Integrado (del inglés, *Integrated Development Environment*) es un programa compuesto por un conjunto de herramientas para un programador. Provee un marco de trabajo amigable para la mayoría de los lenguajes de programación, puede utilizarse en el mismo varios lenguajes de programación (Morán, 2010). Para la construcción del sistema fue necesario escoger un entorno integrado de desarrollo que proporcionara el uso y la integración de todas las tecnologías y lenguajes mencionados en este capítulo.

NetBeans IDE 8.0 es un producto libre, gratuito, sin restricciones de uso y con un importante número de módulos para extenderlo. Entre sus características están las administraciones de ventanas, almacenamiento, interfaces y configuraciones de usuario. Además, empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones (NETBEANS, 2017).

Se selecciona NetBeans IDE en su versión 8.0 pues ofrece un excelente entorno y proporciona un rendimiento superior para los desarrolladores de PHP, proporcionando editores y herramientas integrales para sus tecnologías relacionadas. Este IDE es extensible, permitiendo a través de *pluggings,* editar código escrito en los lenguajes XML, HTML, PHP y Java Script, además tiene un acertad balance entre una interfaz con múltiples opciones y un aceptable completamiento de código.

1.4.7 Lenguaje de programación PHP 5.5

PHP (Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Puede emplearse en los sistemas operativos que más se utilizan, incluyendo Linux, numerosas variantes de Unix, Microsoft Windows, Mac OS X y RISC OS (Pleva, 2013).

Dispone de una gran cantidad de características convirtiéndola en la herramienta ideal para la creación web:

- Soporte para gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle.
- Utiliza las conexiones persistentes a la base de datos con el objetivo de aumentar la eficiencia, soporta además el manejo de excepciones.
- Ofrece una solución simple y universal para las paginaciones dinámicas de la Web de fácil programación (Team, 2011).

1.4.8 Lenguaje de programación JavaScript

JavaScript es uno de los recursos surgidos en la programación para darle dinamismo y capacidad al lenguaje HTML. En la actualidad es una de las tecnologías más extendidas cuando se trata de enriquecer páginas web del lado del cliente. Es un lenguaje interpretado, por lo que no es necesario compilar los programas para ejecutarlos (Eguiluz, 2006).

Se hace imprescindible el uso de este lenguaje para la solución propuesta con la finalidad de validar los datos entrados por los usuarios al sistema permitiendo ejecutar instrucciones como respuesta a las acciones del usuario.

1.4.9 Marco de trabajo Symfony 2.8

Un *framework* o marco de trabajo es un término muy manejado en el mundo de la informática, se utiliza para referirse a un conjunto de bibliotecas usadas para implementar la estructura de un modelo para una aplicación en específico. Su utilización promueve la reutilización del código permitiendo el desarrollo de aplicaciones web robustas entre otros sistemas.

Symfony es uno de los marcos de trabajo que permite que los programadores sean mucho más productivos a la vez que crean códigos de más calidad y más fácil de mantener. Symfony es estable, profesional y está muy bien documentado (Zaninneti, 2008).

Este marco de trabajo cuenta con un grupo de características que influyen en su uso generalizado.

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- Fácil de extender permitiendo su integración con las bibliotecas de otros fabricantes.
- Es independiente del Sistema Gestor de Base de Datos.
- Resulta sencillo de usar en la mayoría de los casos permitiendo un mejor desenvolvimiento de los desarrolladores en el proceso de implementación.

Utilizar Symfony en su versión 2.8 permite contar con una estructura de carpetas más intuitivo, su uso permitiría en el futuro poder realizar con más facilidad y menos dificultades la migración del sistema a una nueva versión contando con el soporte necesario. La versión 2.8 ofrece además un componente propio para las aplicaciones que usen LDAP sin tener que utilizar componentes de terceros.

1.4.10 Marco de trabajo del lado de cliente JQuery 2.1.4

JQuery es una biblioteca JavaScript que ofrece una sólida infraestructura que brinda un grupo importante de ventajas y facilidades para la creación de aplicaciones de mediana y alta complejidad del lado del cliente, orientada a facilitar la labor en cuanto a la creación de interfaces de usuario, efectos dinámicos y otros aspectos. Es un producto estable, bien documentado, con un gran equipo de desarrolladores a cargo de la mejora y actualización del marco de trabajo, lo que hace fácil encontrar soluciones ya creadas en JQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos entre otros (Foundation, 2014).

Conclusiones Parciales

El análisis de la bibliografía y la investigación arrojó los elementos suficientes para corroborar la necesidad de implementar un sistema que sea capaz de ajustarse a las necesidades nutriéndose de las ventajas que presentan las aplicaciones analizadas. La utilización de las herramientas y tecnologías propuestas favorece rapidez, eficacia a lo largo del ciclo de vida del desarrollo de la solución y soberanía tecnológica teniendo en cuenta las características presentadas en el capítulo. Con el análisis de la bibliografía consultada y en correspondencia con las características de la solución a realizar se plantea la utilización de la metodología AUP en su variación UCI. Se utilizará Visual Paradigm para UML en su versión 8.0, utilizando como lenguaje de modelado UML 2.0 para el diseño de artefactos. El Sistema Gestor de Base de Datos a utilizar es PostgreSQL 9.4 y su administración a través de PgAdmin III, como marco de trabajo Symfony 2.8, utilizando como lenguajes de programación PHP 5.5 y JavaScript y como servidor web Apache 2.0, además la biblioteca JQuery 2.1.4.

CAPÍTULO 2: MODELADO DEL SISTEMA DE GESTIÓN PARA EL ALMACÉN DE LIBROS DE LA FACULTAD CITEC

En este capítulo se exponen las tareas correspondientes al modelado de la propuesta de solución. Se realiza el levantamiento de requisitos tanto funcionales como no funcionales y el modelo de casos de uso del sistema con su respectiva descripción textual y demás diagramas. La fundamentación de la arquitectura a utilizar y sus patrones de diseño serán tratados también, además se mostrará el diagrama Entidad-Relación el cual contiene las tablas de la base de datos. La realización de este capítulo tendrá como finalidad un mejor entendimiento de las características del sistema.

2.1 Modelado del negocio

Cuando se comienza la tarea de informatizar un proceso de cualquier índole se necesita comprender sobremanera el ambiente en el que se desarrolla el mismo y cuáles son sus características. Son estos, los principales objetivos de un modelo de negocio.

Un modelo de negocio describe los procesos de negocio de una empresa en términos de casos de uso del negocio y actores del negocio, que se corresponden con los procesos del negocio y los clientes, respectivamente. Al igual que el modelo de casos de uso para un sistema, el modelo de casos de uso del negocio presenta un sistema desde la perspectiva de su uso y esquematiza cómo proporciona valor a sus usuarios (Jacobson & Boocch, 2000).

2.1.1 Reglas del negocio

Las reglas de negocios definen y controlan la estructura, el funcionamiento y la estrategia de una organización, pueden estar formalmente definidas en manuales de procedimiento, contratos o acuerdos, o bien pueden existir como conocimiento o experiencia que tienen los empleados (Network, 2017).

- Cada título en el almacén de libros debe constar con un identificador en la tarjeta de estiba.
- Cada título puede tener más de una tarjeta de estiba.
- Solo cuando el cliente no tenga libros en su poder se emite el modelo de No Tenencia.
- Si el cliente pierde un libro entonces, se realiza el Reintegro por pérdida y no pierde el derecho a recibir los servicios del almacén.
- Cualquier estudiante o trabajador puede recibir los servicios del almacén de libros.

2.1.2 Procesos del negocio

En la actualidad, el trabajo en el almacén de libros de la Facultad CITEC incluye los siguientes procesos.

- Préstamos y devoluciones de libros.
- Reintegro por pérdida de libros
- Generación de modelos de no Tenencia de libros a clientes
- Solicitudes de información estadística
- Asignación de recursos

Existe una estrecha comunicación entre el vicedecanato de administración de la facultad, el cual se encarga de asignarle los recursos necesarios y solicitar un grupo de reportes que garantizan el correcto funcionamiento del almacén.

2.1.3 Actores y trabajadores del negocio

Identificados los procesos del negocio es posible determinar los actores y trabajadores que intervienen en dichos procesos. Se le llama actor a toda entidad externa al sistema que guarde una relación con este y que le demanda una funcionalidad (Rumbaugh, Jacobson, & Boocch, 2011). La Tabla 2 muestra los actores que intervienen en el negocio.

Tabla 2: Actores del negocio.

Actores	Descripción
Vicedecano	Máxima figura administrativa a la cual se subordina el almacén de libros, al cual le solicita reportes y le asigna recursos materiales y de oficina con que realiza su labor la encargada del mismo.
Cliente	Es la persona que solicita los servicios de préstamo o devolución y el pago de débitos al almacén de libros.

Los trabajadores del negocio constituyen abstracciones de personas, grupos de personas o un sistema automatizado que realiza operaciones dentro de las fronteras del negocio. La Tabla 3 muestra los trabajadores del negocio y los procesos donde intervienen

Tabla 3: Trabajadores del negocio.

Trabajadores	Descripción
Encargado(a) del almacén	Persona que realiza la gestión de los procesos de préstamos, devoluciones, inventarios del local y débitos.
Encargado(a) en Economía	Es la persona encargada de cobrar los débitos pendientes.

2.1.4 Diagrama de casos de uso del negocio

Un diagrama de casos de uso del negocio describe los procesos de un negocio vinculados al campo de acción, y cómo se benefician e interactúan los socios y clientes en estos procesos (González, 2013). Se visualiza en la Fig. **1** el diagrama de CUN correspondiente a la propuesta de solución.

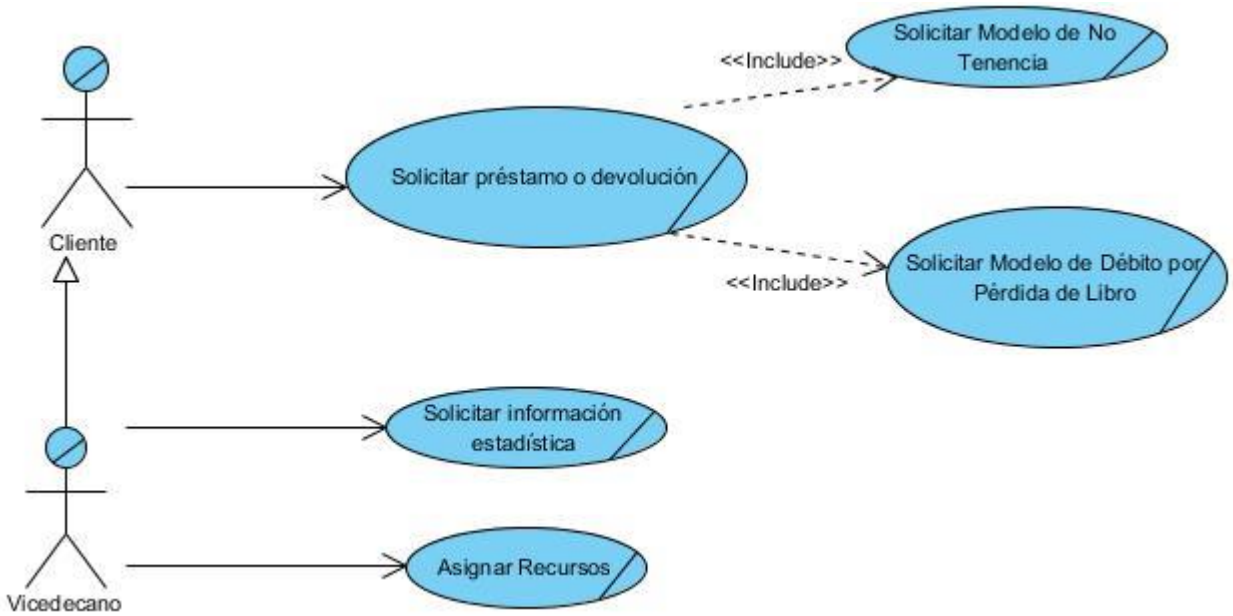


Fig. 1: Diagrama de casos de uso del negocio.

2.1.5 Descripción textual del CUN "Solicitar préstamo o devolución"

Tabla 4: Descripción del CUN "Solicitar Préstamo o Devolución".

Objetivo	Hacer entrega o recibo de un libro a partir de una solicitud.	
Actores	Cliente (inicia)	
Complejidad	Alta	
Prioridad	Crítico	
Resumen	El caso de uso se inicia cuando el cliente solicita a la encargada del almacén el préstamo o devolución de libros. El caso de uso finaliza con la entrega o devolución exitosa de libros.	
Flujo de eventos: "Solicitar Préstamo o Devolución"		
Flujo básico < Solicitar Préstamo o Devolución >		
	Actor	**Negocio**
1.	El cliente solicita el préstamo o devolución de uno o varios libros.	
2.		Verifica si el servicio es de préstamo y solicita el solapín al cliente para buscar su tarjeta de control. Si es devolución,

		ver el flujo alterno 1.1
3	Proporciona la información solicitada	
4		Una vez encontrada su tarjeta revisa si el cliente tiene débitos pendientes.
4		Si la verificación resultó negativa le pregunta al cliente los libros que desea. Si tiene débitos pendientes ver flujo alterno 4.1
5	El cliente nombra los títulos deseados	
6		La encargada verifica mediante tarjeta de estiba si hay disponibilidad de los títulos solicitados
7		La encargada registra la operación en la tarjeta de estiba y actualiza la tarjeta de control de entrega de libros.
8		La encargada busca los libros solicitados y los entrega al cliente.
9	El cliente recibe los libros solicitados y firma en la Tarjeta de control de entrega de libros	.
Flujos alternos: Devolución de libros		
	Actor	**Negocio**
1.1		La encargada pide los libros que el cliente va a devolver.
1.2	Entrega los libros a devolver	
1.3		La encargada actualiza la tarjeta de estiba y la tarjeta de control de entrega de libros.
1.4		Guarda los libros entregados en el almacén
Flujos alternos Débitos pendientes		
4.1		La encargada le alerta al cliente que tiene débitos pendientes y continúa el proceso.

Relaciones	**CU incluidos**	Solicitar Modelo de No Tenencia Solicitar Modelo de Débito por pérdida de libro
	CU extendidos	

2.1.5 Diagrama de Actividades

Un diagrama de actividades describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio (Larman, 2005). Para una mejor comprensión del caso de uso descrito anteriormente se representa en la Fig. **2** el diagrama de actividades para el caso de uso Solicitar Préstamo o Devolución.

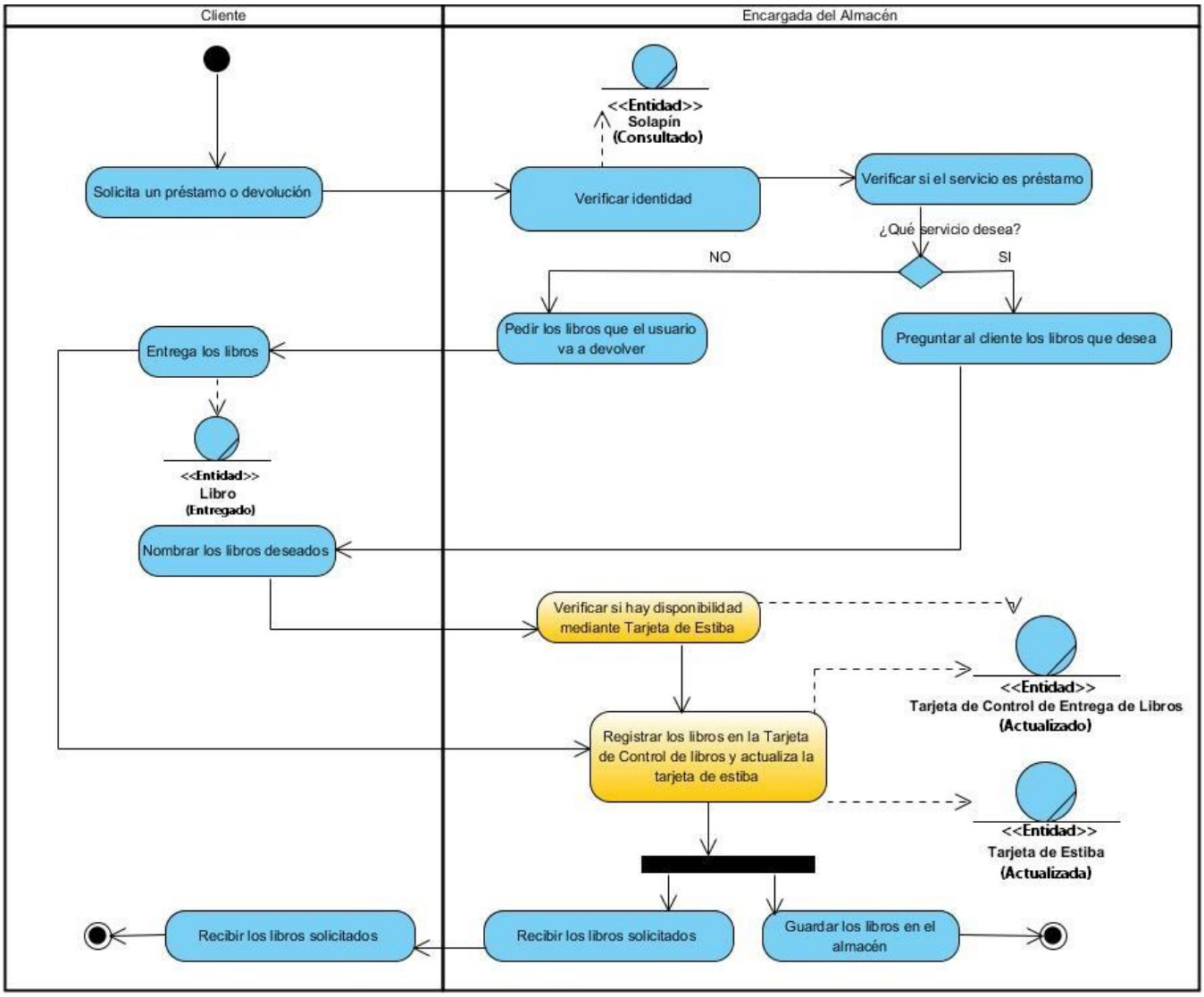


Fig. 2: Diagrama de actividades del caso de uso Solicitar préstamo o devolución.

2.1.6 Modelo de objetos

El modelo de objetos es un diagrama que describe la relación entre los trabajadores y las entidades que intervienen durante la realización de los CU del negocio. Las operaciones de préstamos y devoluciones de libros incluyen el trabajo con las entidades Tarjeta de Estiba y Tarjeta de control de Entrega de libros. Por su parte en la ocurrencia de un reintegro por pérdida de libros el Encargado del Almacén emite un Modelo de débito con el cual el estudiante o trabajador asiste a Economía donde el Encargado de dicha dependencia emite el modelo 1.01. A continuación, se muestra el modelo de objetos perteneciente al negocio y una tabla con la descripción de las entidades.

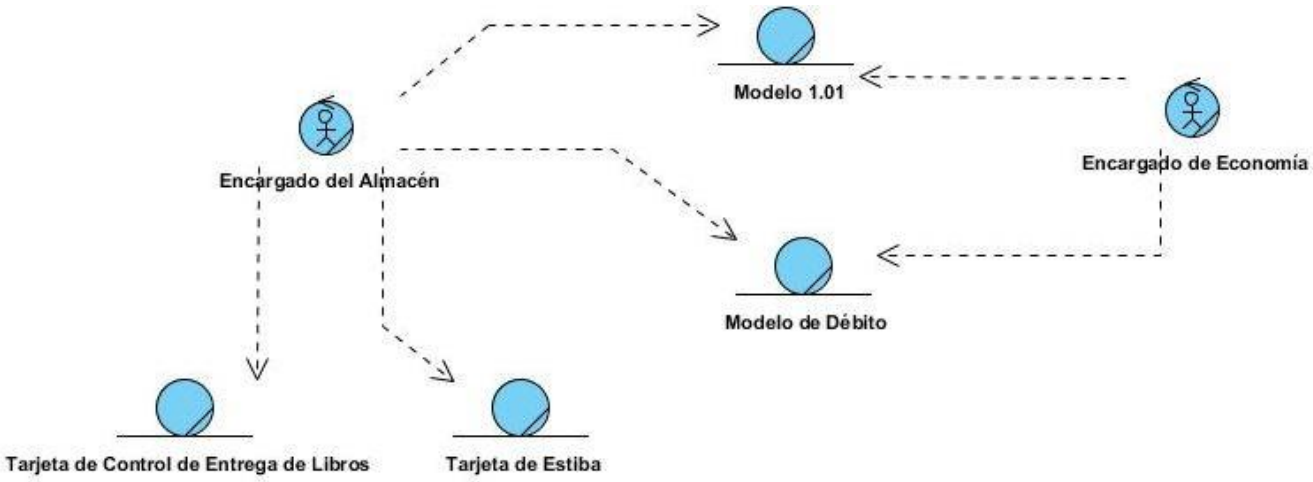


Fig. 3: Modelo de objetos del negocio.

Tabla 5: Descripción de las entidades.

Entidades	Descripción
Modelo de débito	Es un modelo que emite el encargado del almacén donde se hace constar el título y precio de cada libro que el cliente perdió para realizar el posterior pago en Economía
Modelo 1.01	Es un recibo de ingreso emitido por Economía como constancia de que el cliente depositó el importe como concepto de pérdida de libro. Este documento le sirve de constancia al encargado del almacén de que fue cancelada las deuda del cliente.
Tarjeta de Estiba	Es el soporte físico que registra la información de los libros que componen el inventario y a la vez constituye su control interno, donde se registran todos los movimientos que ocurren con dicho producto.
Tarjeta de control de entrega de libros	Es el documento con carácter personal donde se registran la fecha de entrega y devolución de un libro y la firma como constancia de la operación con el producto.

2.2 Levantamiento de Requisitos

Durante el levantamiento u obtención de requisitos, los desarrolladores del software trabajan directamente con los clientes y los usuarios finales del sistema, para determinar el dominio de la

aplicación, los servicios que debe proporcionar el sistema, el rendimiento requerido, las Restricciones de hardware, entre otros elementos, que de conjunto garantizarán la entrega de un producto final que satisfaga las necesidades planteadas por el cliente (Sommerville, 2005).

2.2.1 Listado de Requisitos Funcionales

Tabla 6: Listado de Requisitos funcionales.

CU 1: Autenticar usuario	
RF 1: Autenticar usuario	Acción para comprobar autenticidad y controlar el acceso de los usuarios a los modelos, informaciones y funcionalidades del sistema
CU 2: Administrar usuario	
RF 2: Editar usuario	Acción donde se modifican permisos y visibilidad de las funcionalidades a los que puede acceder el usuario
RF 3: Eliminar usuario	Acción donde se elimina un usuario como tupla de la tabla de una base de datos, mediante la eliminación
RF 4: Listar usuarios	Acción donde se hace un listado o relación de datos de distintos elementos de tipo usuario
RF 5: Buscar usuario	Acción para encontrar o hallar un elemento que coincida con el criterio introducido por el usuario, dentro de una estructura de datos
RF 6: Visualizar trazas de usuarios	Acción para listar las trazas de un usuario en el sistema que incluyen fecha, ruta, operación e IP.
CU 3: Administrar préstamo	
RF 7: Adicionar préstamo	Acción donde se guardan las entregas de libros de un usuario en la base de datos
RF 8: Eliminar préstamo	Acción donde se realiza la devolución de un libro, teniendo como objetivo eliminar un préstamo
RF 9: Listar préstamo	Acción donde se hace un listado o relación de datos de distintos elementos de la relación persona-libro
RF 10: Generar Modelo de No Tenencia	Acción que exportar un modelo de No Tenencia de Libro creando dicho modelo en formato pdf, con la condición que el usuario no tenga deudas con el almacén
CU 4: Gestionar libro	
RF 11: Adicionar libro	Acción donde se crea un registro de libro en la base de datos
RF 12: Editar libro	Acción donde se modifican los datos del libro en la base de datos
RF 13: Eliminar libro	Acción donde se elimina un libro como tupla de la tabla de una base de datos

RF 14: Visualizar detalles del libro	Acción de búsqueda e interpretación de datos de un libro de forma que se transformen en información comprensible para el usuario
RF 15: Buscar libro	Acción para encontrar o hallar un elemento que coincida con el criterio introducido por el usuario dentro de una estructura de datos
RF 16: Listar libros	Acción donde se hace un listado o relación de datos de distintos elementos de tipo libro
CU 5: Visualizar módulo	
RF 17: Visualizar módulo de libros por asignatura	Acción de visualización de los libros pertenecientes a una asignatura en específico
RF 18: Visualizar módulo de libros por año docente	Acción de visualización de los libros pertenecientes a un año docente específico
CU 6: Gestionar asignatura	
RF 19: Adicionar asignatura	Acción donde se crea una asignatura en la base de datos
RF 20: Modificar asignatura	Acción donde se modifican los datos de la asignatura en la base de datos, a través de la actualización
RF 21: Buscar asignaturas	Acción para encontrar o hallar un elemento que coincida con el criterio introducido por el usuario dentro de una estructura de datos
RF 22: Listar asignaturas	Acción donde se hace un listado o relación de datos de distintos elementos de tipo asignatura
CU 7: Gestionar débito	
RF 23: Adicionar débito	Acción donde se adiciona un débito en la base de datos
RF 24: Modificar débito	Acción donde se modifican los datos del débito en la base de datos, a través de la actualización
RF 25: Obtener débito de usuario	Acción donde se obtiene lo datos referentes a los débitos de un usuario
RF 26: Eliminar débito	Acción donde se elimina un débito como tupla de la tabla de una base de datos
CU 8: Gestionar disciplina	
RF 27: Adicionar disciplina	Acción donde se crea una disciplina en la base de datos
RF 28: Editar disciplina	Acción donde se modifican los datos de las disciplinas mediante la actualización en la base de datos
RF 29: Listar disciplinas	Acción donde se hace un listado o relación de datos de distintos elementos de tipo disciplina
RF 30: Eliminar disciplina	Acción donde se elimina una disciplina como tupla de la tabla de una base de datos
RF 31: Buscar disciplina	Acción para encontrar o hallar un elemento que coincida con

	el criterio introducido por el usuario dentro de una estructura de datos
CU 9: Exportar modelos a PDF	
RF 32: Exportar modelos a PDF	Acción que exportar un modelo creando un documento en formato pdf, que la misma aplicación no podrá editar luego de ser guardado por el usuario
CU 10: Generar reporte	
RF 33: Generar listado de libros con porciento de uso y cantidades	Acción que genera un listado con él por ciento de uso y cantidades de los libros, dado un año académico y un parámetro de cuantía determinado
RF 34: Generar listado de personas con préstamos de libro	Acción que genera un listado de personas con la cantidad de préstamos, dado el libro y un parámetro de cuantía determinado
CU 11: Administrar notificación	
RF 35: Visualizar notificación	Acción donde se hace un listado o relación de datos de distintos elementos de tipo Notificación, con su nombre, valor y la acción ¨Editar¨
RF 36: Editar notificación	Acción donde se modifica el mecanismo de envío y los valores: Asunto, Cuerpo y Dirección de correo de los campos del correo que envía el sistema
RF 37: Enviar notificaciones por correo	Acción que permite utilizar este servicio que brinda la universidad, mediante redes, para enviar a usuarios determinadas notificaciones
CU 12: Administrar alerta	
RF 38 : Visualizar listado de alertas	Acción donde se hace un listado o relación de datos de distintos elementos de tipo Alerta, con su nombre, valor y la acción ¨Editar¨. Las alertas se generan cuando se cumple con una característica atípica en la solución
RF 39: Editar alertas	Acción donde se modifican valores, fechas o se activa un campo que dispararía una alerta al usuario
CU 13: Visualizar libros prestados	
RF 40: Visualizar libros prestados	Acción donde se hace un listado o relación de datos de distintos elementos de tipo libro, que el usuario tenga registrado como entregados en su Tarjeta de Control de Entrega
CU 14: Listar catálogo de libros	
RF 41: Listar catálogo de libros	Acción donde se hace un listado o relación de datos de todos los objetos de tipo libro, que estén registrados en la base de datos

CU 15: Gestionar Tarjeta de estiba	
RF 42: Adicionar Tarjeta de estiba	Acción donde se crea una Tarjeta de estiba en la base de datos
RF 43: Modificar Tarjeta de estiba	Acción donde se modifican los datos de la Tarjeta de estiba en la base de datos, a través de la actualización
RF 44: Buscar Tarjeta de estiba	Acción para encontrar o hallar un elemento que coincida con el criterio introducido por el usuario dentro de una estructura de datos
RF 45: Eliminar Tarjeta de estiba	Acción donde se elimina una tarjeta de estiba como tupla de la tabla de una base de datos
RF 46: Listar Tarjeta de estiba	Acción donde se hace un listado o relación de datos de todos los elementos de tipo tarjeta de estiba, que estén registrados en la base de datos
CU 16: Gestionar asiento	Asiento: Fila dentro de la Tarjeta de estiba
RF 47: Adicionar asiento	Acción donde se crea una asiento en la base de datos
RF 48: Modificar asiento	Acción donde se modifican los datos de la asiento en la base de datos, a través de la actualización
RF 49: Buscar asiento	Acción para encontrar o hallar un elemento que coincida con el criterio introducido por el usuario dentro de una estructura de datos
RF 50: Eliminar asiento	Acción donde se elimina un asiento como tupla de la tabla de una base de datos
RF 51: Listar asiento	Acción donde se hace un listado o relación de datos de todos los objetos de tipo asiento, que estén registrados en la base de datos
CU 17: Visualizar información	
RF 52: Visualizar información	Acción que realiza rol Revisor y que permite visualizar información sin poder editarla

2.2.2 Requisitos No Funcionales

Se entiende como Requisitos No Funcionales (RNF) al conjunto de propiedades o cualidades que el producto debe poseer. Luego de analizado el entorno de desarrollo y teniendo en cuenta el futuro despliegue de la solución propuesta, se detallan a continuación el listado de RNF con que contará el sistema:

Usabilidad:

RNF 1: El sistema se caracterizará por ser intuitivo con un diseño iconográfico ajustado a las funciones que representan garantizando facilidad para el trabajo de los usuarios.

RNF 2: El diseño del sistema es agradable, coherente y aporta confianza a los usuarios, sin enlaces rotos o páginas de error.

Eficiencia:

RNF 3: El tiempo de respuesta del sistema estará dado por la cantidad de información a procesar de acuerdo a la funcionalidad procesada, como promedio menos de 5 segundos con 200 usuarios conectados simultáneamente, validado mediante pruebas de rendimiento.

Seguridad:

RNF 4: El sistema garantiza la gestión de usuarios controlando la asignación de roles permitiendo que cada usuario tenga los privilegios necesarios de acuerdo a su rol.

Interfaz: Las siguientes tablas muestran los requisitos de interfaces de hardware y software

Tabla 7: Características de Hardware.

Para la computadora cliente	Para el servidor
• **RNF 5:** 512 MB de RAM como mínimo • **RNF 6:** Una tarjeta de red con velocidad de 100 Mbps • **RNF 7:** Un procesador de 1 GHZ como mínimo	• **RNF 8:** 2 GB de memoria RAM como mínimo • **RNF 9:** Una tarjeta de red con velocidad de 100 Mbps • **RNF 10:** Al menos 80 GB de disco duro • **RNF 11:** Un procesador de 2 GHZ como mínimo

Tabla 8: Características de Software.

Para la computadora cliente	Para la computadora servidor
• **RNF 12:** Un navegador como mozilla Firefox 16.0 o superior, Sistema operativo GNU Linux Ubuntu Server 12.04 o superior o Windows.	• **RNF 13:** Sistema operativo GNU Linux Ubuntu Server 12.04 o superior, servidor web Apache en su versión 2.0 y PostgreSQL 9.4.

2.3 Modelo de casos de uso del sistema

El modelo de casos de uso permite describir los RF del sistema, dando lugar a un acuerdo entre el cliente y los desarrolladores de la aplicación. Proporciona una explicación clara y consistente de lo que debería hacer el software de modo que el modelo se use a lo largo del proceso de desarrollo. Posibilita que se obtenga una base para realizar verificaciones del producto informático, siendo la entrada principal para el análisis, el diseño y las pruebas (Larman, 2005).

Casos de Uso

Los casos de uso son descripciones funcionales del sistema que permiten definir los límites del software y las relaciones entre la aplicación y el entorno; describen como los actores pueden usar un sistema. Especifican una secuencia de acciones que debe devolver algún resultado de valor a un actor (Schmuller, 2010).

2.3.1 Actores del sistema

Un actor es una entidad externa al sistema representada por un ser humano, una maquina o un software que interactúa con el sistema. Representa un tipo particular de usuario del negocio más que un usuario físico, debido a que varios usuarios físicos pueden realizar el mismo papel en relación al negocio (Schmuller, 2010).

Tabla 9: Actores del Sistema.

Actores	Descripción
Usuario	Persona que puede solicitar préstamos y hacer devolución de libros. Además, puede ver los que tiene en su poder y los catálogos de libros.
Encargado del almacén	Persona encargada de la gestión de los préstamos y devoluciones, la gestión de las operaciones en el almacén (entrada, salida, alta y baja) y gestionar los débitos de usuarios.
Administrador	Persona encargada de gestionar los usuarios. Además heredará las los privilegios tanto del Encargado del almacén como el de Usuarios.
Revisor	Persona que solo tiene acceso a la visualización y revisión de información. Hereda los privilegios del actor usuario.

2.3.2 Diagrama de Casos de Uso del Sistema

Un diagrama de Casos de Uso del Sistema es el encargado de mostrar el comportamiento de un sistema desde el punto de vista de los usuarios. Se utiliza para describir las especificidades de un software y documentar su comportamiento, mostrando la relación que existe entre el usuario final y las funcionalidades (Schmuller, 2010).

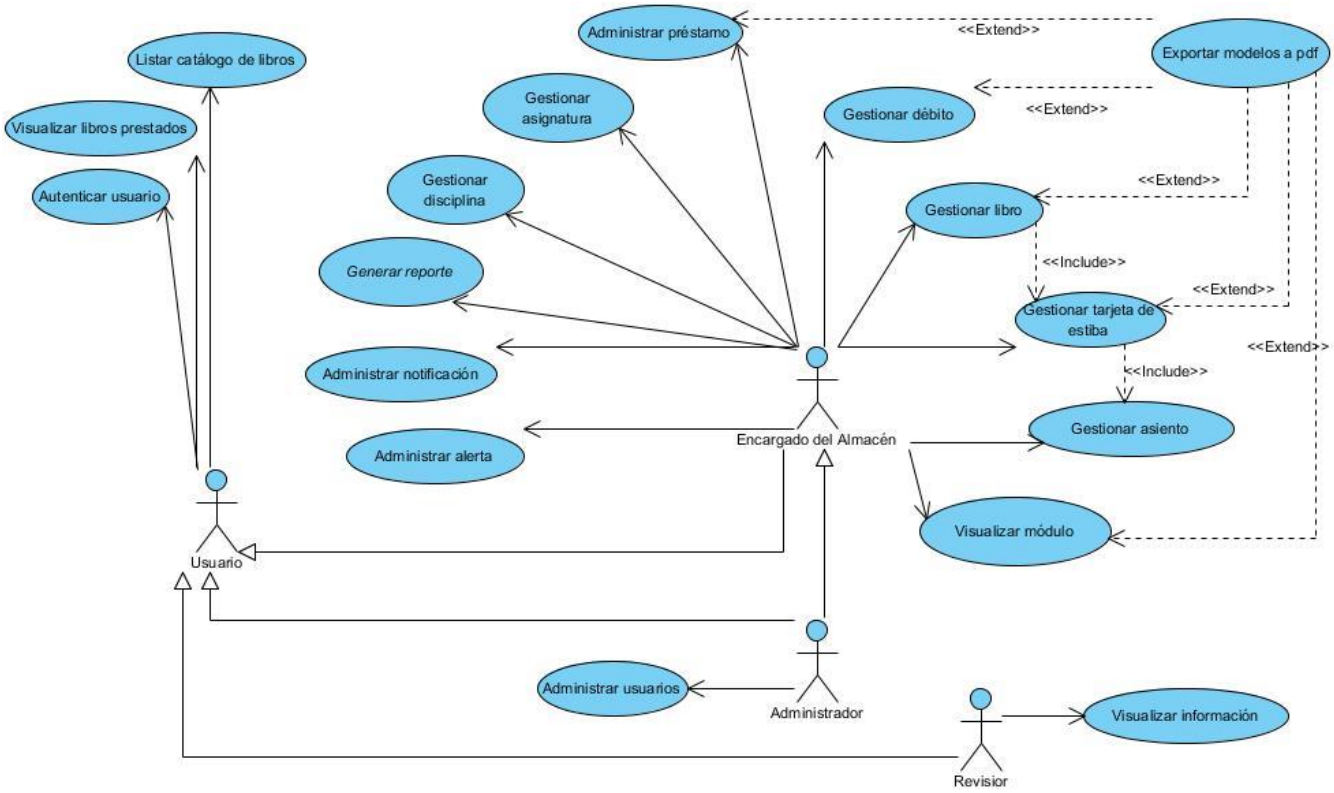


Fig. 4: Diagrama de CU del Sistema para la gestión de libros en el almacén de libros de la CITEC.

2.3.3 Descripción del Caso de Uso del Sistema "Administrar préstamo"

Tabla 10: Descripción del CU Administrar préstamo.

Caso de uso	Administrar préstamo
Objetivo	Este CU se desarrolla con el objetivo de realizar un préstamo o devolución de un cliente.
Actores	Encargada del almacén (inicia) Administrador (inicia)
Complejidad	Alta
Prioridad	Crítico
Precondiciones	La Encargada del almacén o Administrador deben estar autenticados en el sistema y

	tener los permisos necesarios para administrar un préstamo.	
Poscondiciones	Al finalizar el CU el registro de préstamo se añade o se elimina con la devolución	
Flujo de eventos: < Administrar préstamo>		
Flujo básico < Administrar préstamo >		
	Actor	**Sistema**
1	Selecciona la pestaña “Servicios del Almacén”	
2		Despliega un menú donde aparece la pestaña “Entrega y devolución”
3	Selecciona la pestaña “Entrega y devolución”	
4		Muestra la vista Préstamos y Devoluciones con la opción de buscar a un cliente mediante el usuario o el solapín.
5	Introduce los datos del usuario que se desea y presiona buscar.	
6		Comprueba que no existan campos vacíos, si existen campos vacíos ver el Flujo Alterno 6.1 Comprueba que se inserte un usuario o solapín válido, si existen datos no válidos ver el Flujo Alterno 6.2.
7		Se actualiza la vista Préstamos y devoluciones con los datos del usuario y los campos de acciones que permiten los préstamos y devoluciones. Adicionar préstamo: Ver sección 1 Eliminar préstamo: Ver sección 2 Listar préstamo: Ver sección 3

Fig. 5: Prototipo de pantalla: Préstamos y devoluciones

Flujo Alterno al paso 6		
No.	**Actor**	**Sistema**
6.1		Informa al usuario que debe llenar los campos vacíos mediante un mensaje de error.
6.2		Informa al usuario sobre la existencia de datos no válidos mediante un mensaje de error.

Fig. 6: Prototipo de pantalla: Mensaje de error

Sección 1: "Adicionar préstamo"		
Flujo Básico: < Adicionar préstamo>		
1	Selecciona los libros a entregar marcando en caja de selección múltiple correspondiente y selecciona la opción guardar	

2		Se muestra la vista Resumen de operaciones con los títulos involucrados en la operación.
3	Selecciona la opción enviar para guardar los cambios.	
4		Guarda la operación con el mensaje “Se han salvado los cambios satisfactoriamente” y se actualiza visualmente el estado del libro.

Fig. 7: Prototipo de pantalla: Préstamo satisfactorio

Sección 2: “Eliminar préstamo”		
Flujo Básico: < Eliminar préstamo>		
1	Selecciona los libros que el usuario desea devolver marcando en la caja de selección correspondiente y selecciona la opción guardar	
2		Se muestra la vista Resumen de

		operaciones con los títulos involucrados en la operación
3	Selecciona la opción enviar para guardar los cambios.	
4		Guarda la operación mostrando el mensaje “Se han salvado los cambios satisfactoriamente” y se actualiza visualmente el estado de los libros.

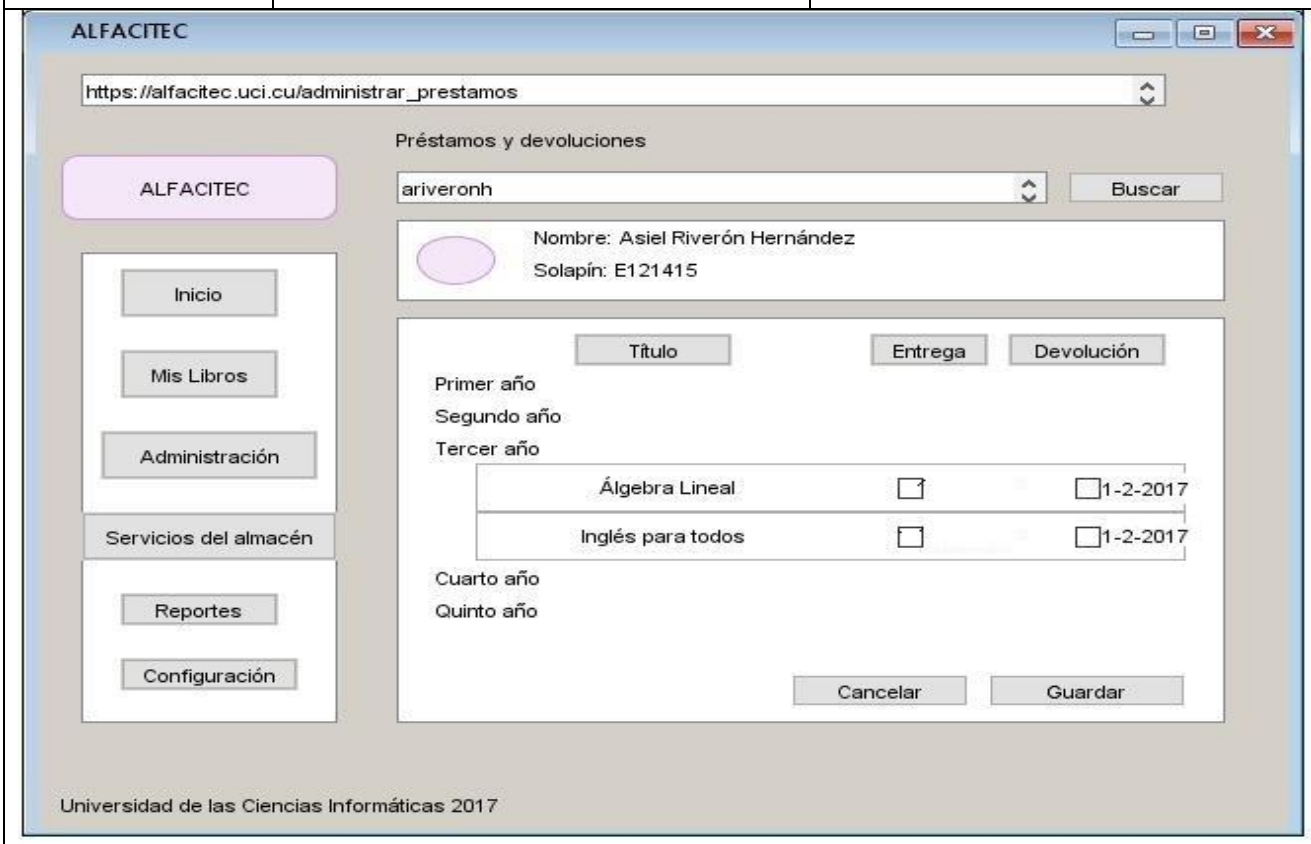


Fig. 8: Prototipo de pantalla: Eliminar préstamo

Sección 3: “Listar préstamos”		
Flujo Básico: < Listar préstamos>		
1	Introduce los datos de un usuario del sistema y selecciona la opción buscar.	
2		Comprueba que no existan campos vacíos, si existen campos vacíos ver el Flujo Alterno 2. El sistema comprueba que se inserte un usuario o solapín válido, si existen datos no válidos ver el flujo alterno 2.1
3		Muestra la interfaz con la lista de libros por años y los libros con la operación

		“préstamo” del usuario seleccionado.
Flujo alterno al paso 2		
No.	**Actor**	**Sistema**
2		Informa al usuario que debe llenar los campos vacíos mediante un mensaje de error.
2.1		Informa al usuario sobre la existencia de datos no válidos mediante un mensaje de error.

Fig. 9: Prototipo de pantalla: Mensaje de error en búsqueda de usuario

Relaciones	CU incluidos	No procede
	CU extendidos	Exportar modelos a PDF
Asuntos Pendientes		No procede

2.3.4 Patrones de Casos de Uso

Para la construcción del diagrama de CUS se utilizaron los patrones de CU que se detallan a continuación:

CRUD Completo:

El patrón CRUD (Creating, Reading, Updating, Deleting) de tipo Completo normalmente utilizado en CU llamados Gestionar, modela todas las operaciones que pueden ser realizadas sobre una parte de la información como puede ser la creación, lectura, actualización y eliminación. Este patrón se pone de manifiesto en los CU Gestionar libro, Gestionar disciplina, Gestionar asignatura, Gestionar tarjeta de estiba y Gestionar asiento. A continuación, se muestra un ejemplo del uso de este patrón.

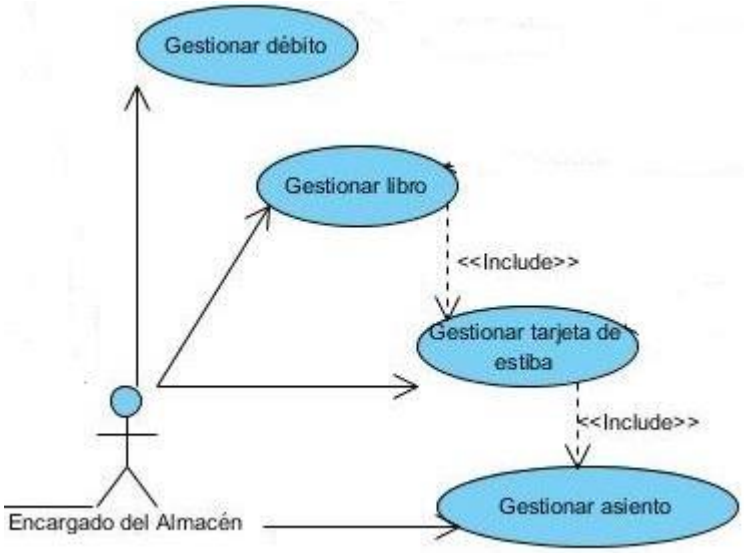


Fig. 10 Ejemplo de CRUD Completo.

CRUD Parcial:

Este patrón es considerado una versión del CRUD Completo. La diferencia está en que puede no incluir una o varias operaciones presentes en los CRUD Completo. Este patrón se pone de manifiesto en los CU Administrar préstamo, Administrar notificaciones y Administrar usuario. La siguiente figura muestra un ejemplo del uso de este patrón.

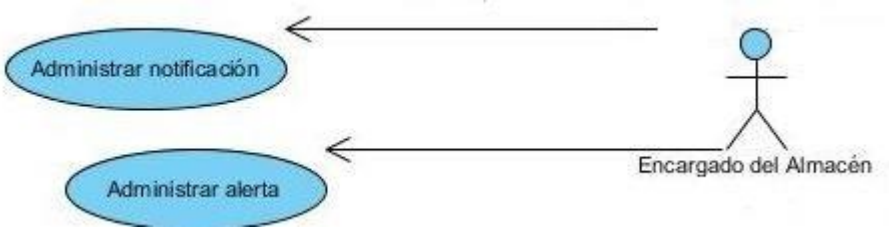


Fig. 11 Ejemplo de CRUD Parcial.

Generalización / Especialización entre Actores:

Este patrón permite agrupar varios actores que comparten similar rol con respecto a un CU en particular. Una relación de generalización de una clase hija de actor a otra clase padre de actor indica que el hijo hereda el rol que la clase padre puede jugar respecto a un Caso de Uso (Schmuller, 2010). El diagrama de CUS muestra la relación Generalización Especialización entre los actores Encargada de almacén, Revisor y Administrador los cuales asumen el rol del actor Usuario. Este patrón también se pone de manifiesto cuando el actor Administrador asume el rol de Encargada del almacén.

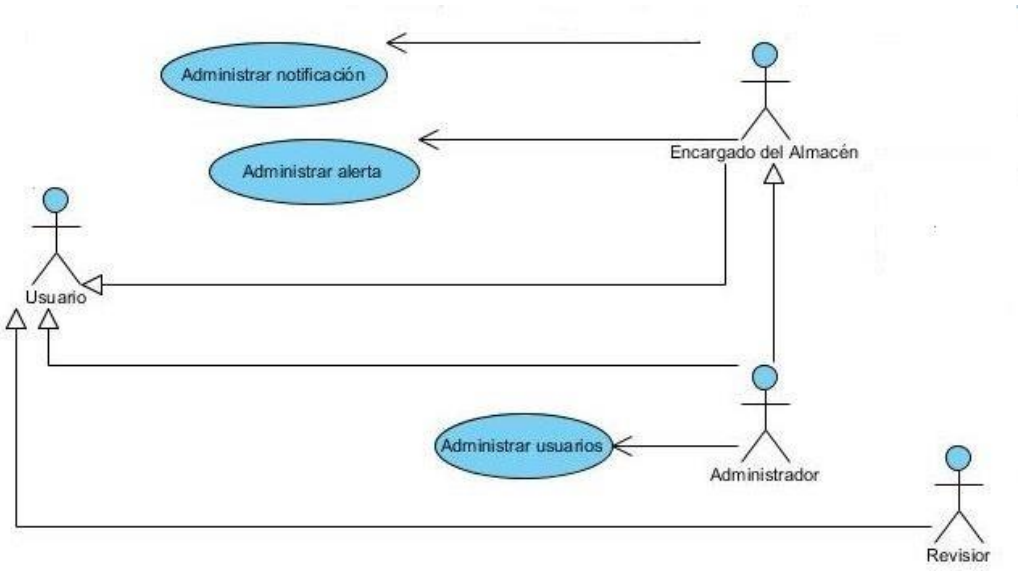


Fig. 12 Ejemplo de Generalización Especialización.

Extensión Concreta o Inclusión:

Este patrón se divide en Extensión concreta o Inclusión concreta. La Extensión concreta presenta un CU que extiende de algún CU base con el correspondiente comportamiento que este asimila y bajo determinadas condiciones donde el segundo extiende como una opción para satisfacer el CU base sin que este deje de cumplir su objetivo, en el caso de la inclusión concreta para que el caso de uso base funcione correctamente el CU incluido es parte esencial del primero. En el diagrama de CU del sistema se tiene presente la Inclusión concreta en la relación existente entre Gestionar Libro con Gestionar tarjeta de estiba y esta última con Gestionar asiento. A continuación, un ejemplo de uso de este patrón.

Fig. 13 Ejemplo de Inclusión concreta.

Concordancia de adición

Este patrón se utiliza cuando existen más de dos CU los cuales extienden de un CU base compartiendo la subsecuencia de acciones. Este patrón se puede observar mediante el Exportar modelos a PDF, donde varios CU necesitan esta función para su correcta realización.

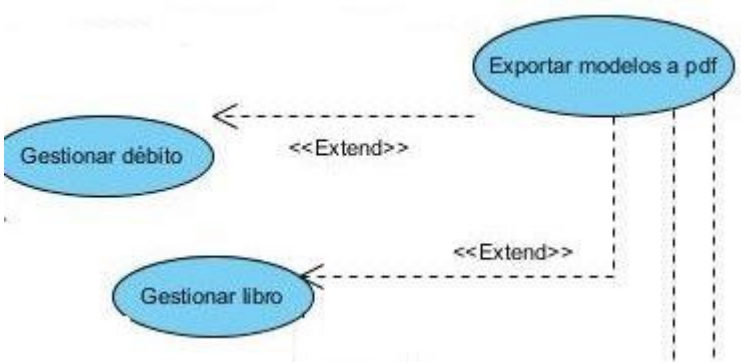


Fig. 14 Ejemplo de Concordancia de adición.

2.4 Elementos del diseño y la arquitectura del sistema

Una arquitectura de software define la estructura del sistema. Esta estructura se constituye de componentes-módulos o piezas de código que nacen de la noción de abstracción, cumpliendo funciones específicas, e interactuando entre sí con un comportamiento definido. Los componentes se organizan de acuerdo a ciertos criterios, que representan decisiones de diseño. De esta manera, la arquitectura de software puede ser vista como la estructura del sistema en función de la definición de los componentes y sus interacciones, constituyendo un puente entre los requisitos del sistema y la implementación (Fowler, 2013).

2.4.1 Estilo Arquitectónico

Un estilo arquitectónico es una lista de tipos de componentes que describen los patrones o las interacciones a través de estos. Un estilo influye en toda la arquitectura del software y puede combinarse en la propuesta de solución. Los estilos ayudan a un tratamiento estructural que concierne más bien a la teoría, la investigación académica y la arquitectura en el nivel de abstracción más elevado, expresando la arquitectura en un sentido más formal y teórico (Almeira, 2007).

En el caso de la solución propuesta se determina el empleo del estilo llamada y retorno, ya que permite obtener una estructura de programa fácil de modificar persiguiendo la escalabilidad del sistema. Según Pressman, *el empleo de este estilo posibilita además la comunicación, la coordinación y cooperación entre los componentes y las restricciones que definen como se integran para conformar el sistema, así como los modelos semánticos que facilitan al diseñador el entendimiento de todas las partes del*

sistema, evitando que las variaciones realizadas a funcionalidades o componentes específicos afecten el funcionamiento general (Pressman, 2008).

2.4.2 Patrón arquitectónico Modelo Vista Controlador

Un patrón arquitectónico brinda la descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. El esquema de solución se especifica mediante la descripción de los componentes que lo constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí. El patrón arquitectónico es quién define la estructura básica de la aplicación, de ahí la medular importancia de su correcta elección (Camacho, Cardeso, & Nuñez, 2004).

Para el desarrollo de la solución se define el patrón arquitectónico MVC, el mismo se basa en la posibilidad de reutilizar el código, separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos: modelo, vista y controlador; esto facilita el desarrollo y mantenimiento de las aplicaciones.

Fig. 15: Funcionamiento MVC.

A continuación, se describen las tres capas del patrón:

Modelo: Se encarga de la representación específica de toda la información con la cual el sistema va a trabajar; la lógica de datos puede llegar a asegurar la integridad de estos y permitirá derivar nuevos datos.

Vista: Presenta la interfaz con la que va a interactuar el cliente. Captura eventos del usuario y se los envía al sistema a través del controlador, luego éste envía una respuesta y muestra la información al usuario.

Controlador: Responde a eventos, normalmente son acciones que el usuario invoca. Sirve de enlace entre las vistas y el modelo respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación.

En el caso de la solución en cuestión se tiene una base de datos donde se van a guardar objetos en cada una de las tablas y esos objetos se corresponden a un determinado modelo. Por otro lado, se cuenta con la vista, el aspecto atractivo, la interfaz que va a ver el usuario dentro de la aplicación.

Por su parte, el controlador se ocupa de recuperar los modelos interpretados de la base de datos (BD) y luego pasárselos a la vista, o sea se recibe una serie de HTTP Request y el Controlador se ocupa de distribuirlos y enviarlos al Modelo. También, por el contrario, puede recepcionar una serie de HTTP Request que va a solicitar un grupo de datos de la BD y el controlador se ocupa de pedir un modelo para ese identificador, la BD se los devuelve al controlador y este se los pasa a la vista.

Se presenta el diagrama de paquetes del diseño correspondiente al CU Administrar préstamo donde se presenta los elementos que conforman al modelo, la vista y el controlador.

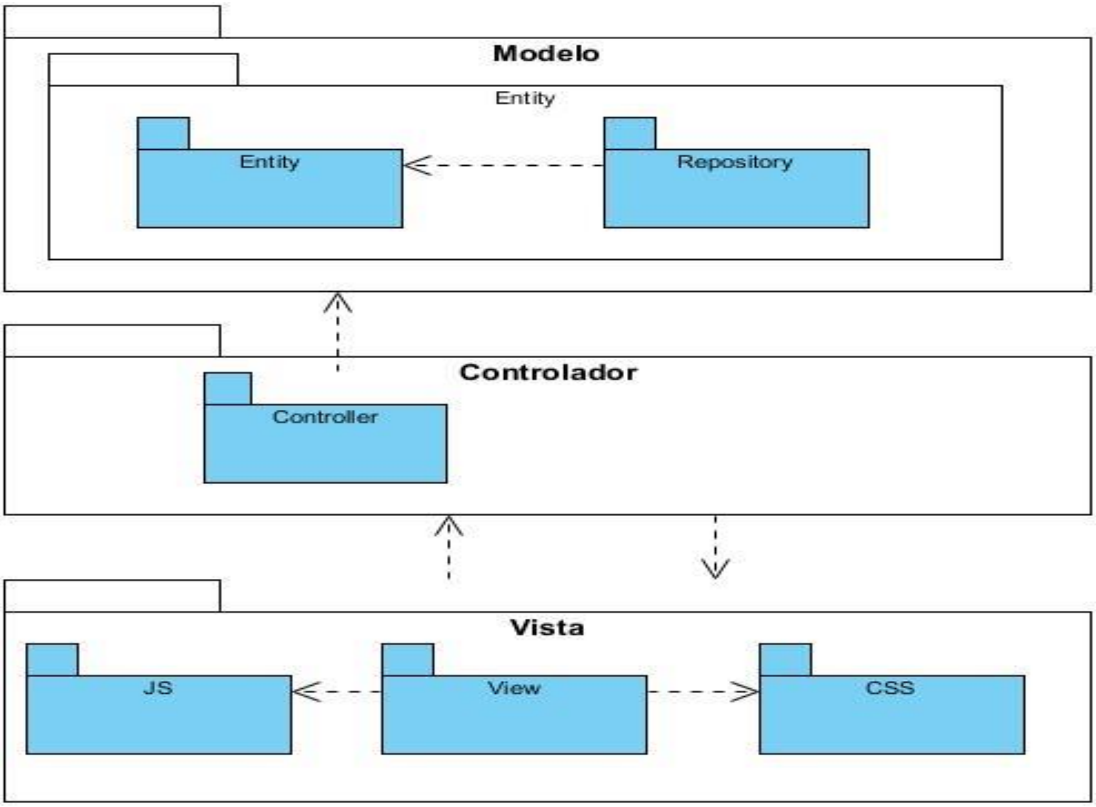


Fig. 16: Diagrama de paquetes del diseño del CU Administrar préstamo.

La capa correspondiente a la vista contiene los paquetes correspondientes a las páginas y plantillas de la aplicación, además los paquetes “JS” y “CSS” engloban los archivos javascript y las hojas de estilo respectivamente. La capa Controlador engloba los controladores del sistema, dentro del paquete “Controller” se localizan las clases encargadas de manejar la lógica de control y se definen en ellas las funcionalidades que son invocadas, cuando el usuario hace alguna solicitud al sistema a través de acciones sobre los componentes de la interfaz. El modelo por su parte define dentro del paquete “Entity” todas las entidades de presentación propias del módulo que se generan por cada tabla de la base de datos, en tanto el paquete “Repository” incluye las clases responsables de los mecanismos de selección de datos.

2.5 Modelo de diseño

El modelo de diseño describe la realización física de los CU y se corresponde directamente con los elementos físicos del ambiente de implementación. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y/o subsistemas (Rumbaugh, Jacobson, & Boocch, 2011).

2.5.1 Diagramas de clases del diseño

Un Diagrama de Clases del Diseño (DCD) muestra gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación y permite modelar la vista de diseño del sistema. A

continuación, se muestra el DCD del CU Administrar préstamo donde se ilustra las relaciones existentes entre las clases. En el modelo se encuentra Book, PersonBook, BookRepository y PersonBookRepository que acceden a los datos y tienen relación con el Controlador BookController. Por su parte la vista contiene la clase Index principal encargada mediante el formulario de mostrar e insertar la información y relacionadas con los paquetes Twig, JavaScript y CSS, encargados de la visualización de las vistas.

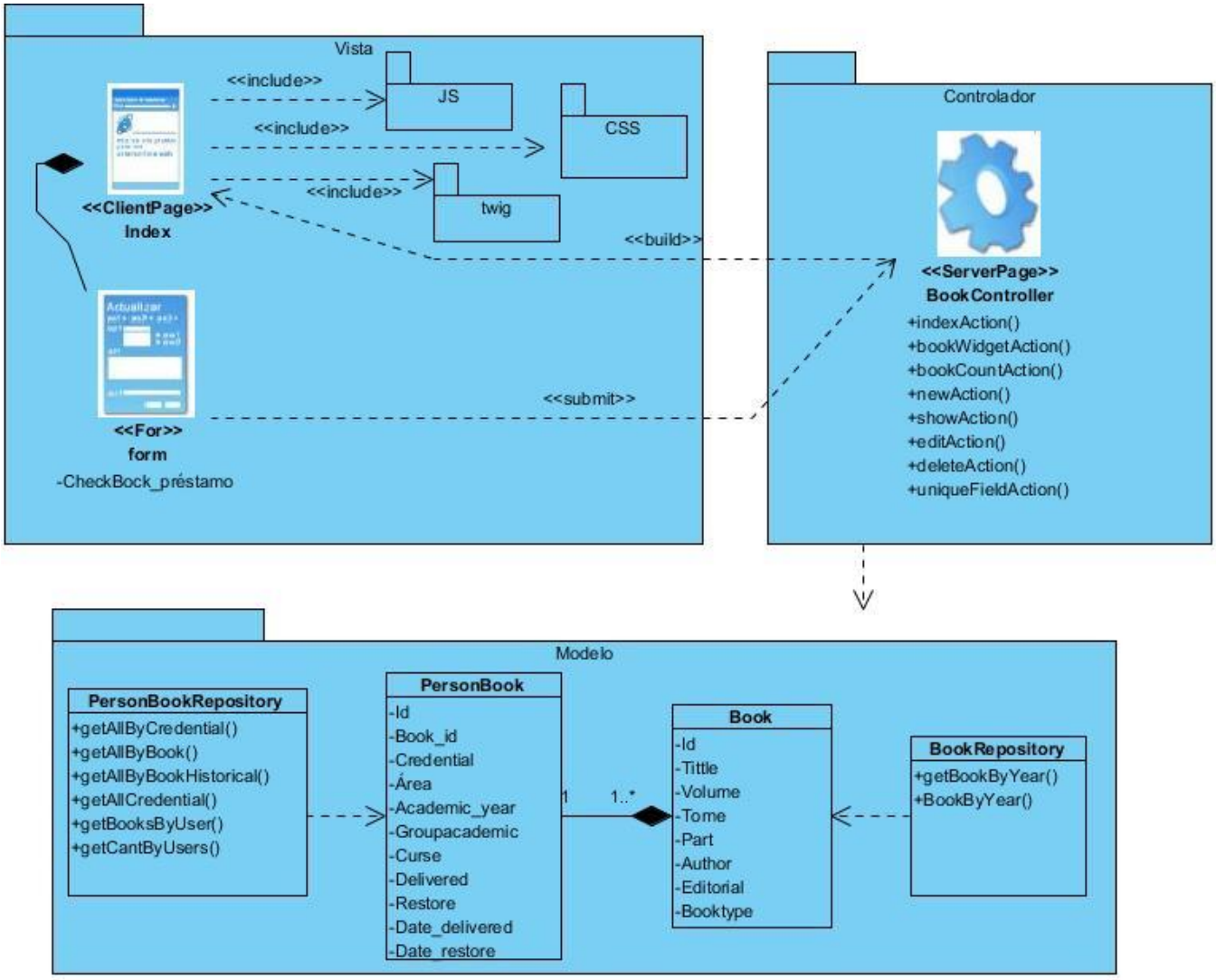


Fig. 17: Diagrama de clases del diseño del CU Administrar préstamo.

2.5.2 Patrones de diseño

Los patrones de diseño constituyen una descripción de un problema y la solución, la misma recibe un nombre y se puede aplicar a varios contextos. Muchos patrones proporcionan guías en el modo en que deberían asignarse las responsabilidades a los objetos, dada una categoría específica del problema (Larman, 2005).

Patrones de Principios Generales para Asignar Responsabilidades (GRASP)

Los patrones GRASP describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. A continuación, se detalla el uso de estos patrones en la solución:

Experto: Propone asignar las responsabilidades a las clases de acuerdo a la información que contienen las mismas cumpliendo así un principio básico e intuitivo de la programación orientada a objetos (Larman, 2005). En la presente solución se utiliza el mencionado patrón dentro de la capa de acceso a datos, ya que Symfony utiliza el objeto relacional de mapeo (ORM, por sus siglas en inglés) Doctrine en la capa del modelo en la clase “Book”, esto permite que se pueda manejar su información como un objeto de tipo entidad mapeada.

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos (Larman, 2005). Este patrón tiene como función la de encontrar un creador que necesite conectarse al objeto creado en alguna situación. Se observa en el sistema propuesto en la clase “BookController”, la cual crea instancias de la clase que provee la información necesaria para su propio manejo. De manera general las clases enmarcadas dentro de la capa controlador distribuyen las responsabilidades de comunicación con las instancias de la capa de presentación.

Bajo acoplamiento: Determina el nivel de dependencia de una clase con respecto a otras. Su uso potencia la reutilización, el mantenimiento y la mitigación de efectos a producirse en una clase al hacer cambios en otra. En la presente solución se pone de ejemplo la existencia de una baja interdependencia entre la clase controlador y el modelo o la vista. Esta característica evidencia la posibilidad de efectuar cambios en estas sin que las otras sufran grandes afectaciones.

Alta cohesión: Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Significa que las clases del sistema tienen asignadas solo las responsabilidades que les corresponde y mantienen una estrecha relación con el resto de las clases. Este patrón se evidencia en la propuesta de solución con las clases “Book” y “BookController” pues las mismas muestran responsabilidades relacionadas coherentemente, que se complementan entre sí. Esta situación garantiza que exista un bajo acoplamiento, favoreciendo un equilibrio al interactuar los objetos.

Controlador: El patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, manejando los eventos de entrada de dicha interfaz (Larman, 2005). En la aplicación se manifiesta el uso de este patrón con la creación de varios controladores que se

encargan de manejar eventos, el controlador “BookController” ejemplifica claramente el uso de este patrón.

Patrones de la banda de los cuatro (GoF)

Los patrones GoF (por sus siglas en inglés The Gang of Four) se utilizan para diseñar objetos y solucionar problemas de creación de instancias, ya que ayudan a encapsular y abstraer dicha creación. En la propuesta de solución se manifiestan los patrones:

Decorador: Es un patrón de tipo estructural que permite añadir dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender funcionalidades (Larman, 2005). Este patrón está presente en la propuesta de solución a través de las plantillas twig para las vistas. Dichas plantillas heredan de la plantilla “base.html.twig”, lo cual permite declarar bloques editables, dentro de los cuales se realizar modificaciones específicas manteniendo una homogeneidad en la apariencia.

Agente Remoto: Este patrón es utilizado cuando se requiere que el sistema en cuestión se comunique con un servicio externo y no se desea o no es posible acceder directamente a este (Larman, 2005). La utilización de este patrón en la solución se evidencia en la autenticación mediante el Web Service disponible para la UCI. Este servicio permite la autenticación y obtención de los datos referentes a los usuarios de la universidad utilizando también la clase mediadora “defaultController”.

2.5.3 Diagrama Entidad-Relación

El Diagrama Entidad Relación (DER) proporciona una herramienta para representar información del mundo real a nivel conceptual. Permite describir las entidades involucradas en una base de datos, así como las relaciones y restricciones de estás (Gaona, 2012). La figura muestra el DER de la solución correspondiente a la propuesta de solución. Se puede apreciar los paquetes donde se muestran las entidades y sus relaciones.

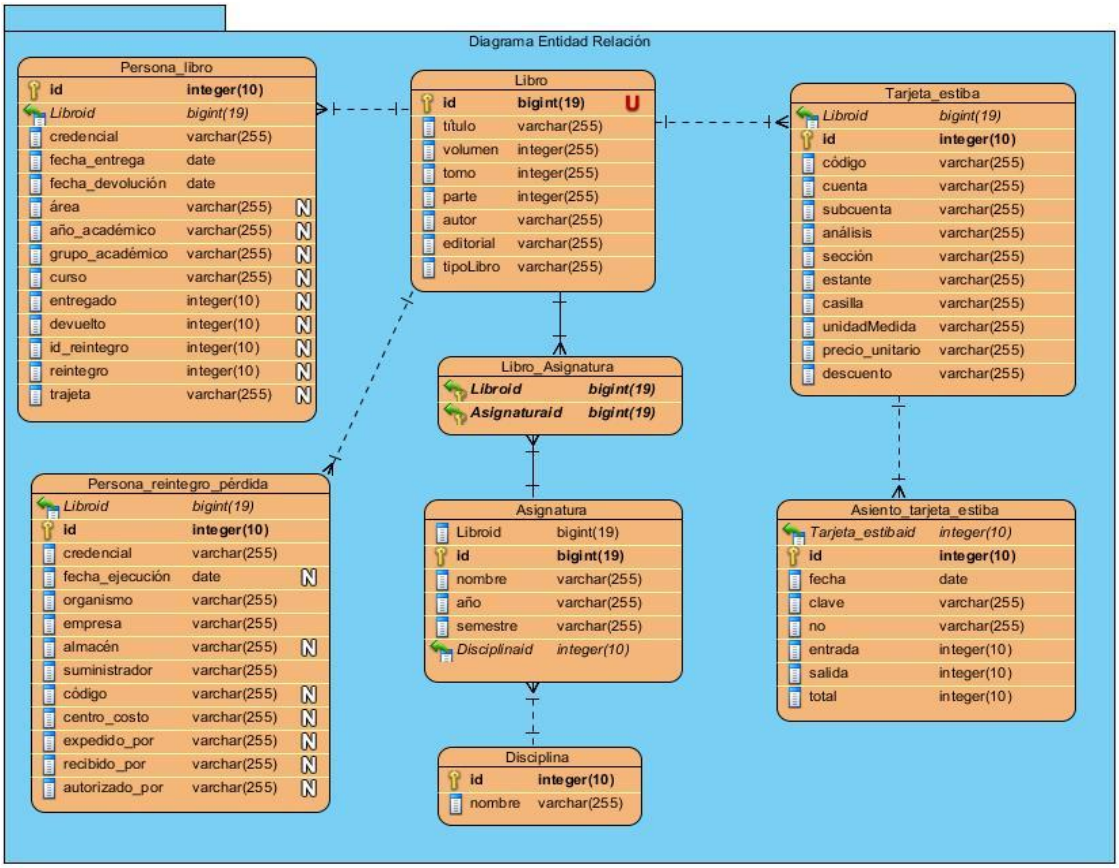


Fig. 18: Diagrama Entidad-Relación de ALFACITEC.

El diagrama Entidad Relación contiene las entidades fundamentales del sistema donde la entidad Libro tiene asociada una asignatura, una disciplina, una tarjeta de estiba y una asociación para los reintegros por pérdida. Cada entidad contiene sus llaves y atributos correspondientes a los datos que las caracterizan.

2.5.4 Diagrama de despliegue

El modelo de despliegue tiene como objetivo capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. Permite el mapeo de procesos dentro de los nodos, asegurando la distribución del comportamiento a través de aquellos que son representados, (Pressman, 2008). A continuación, se muestra el diagrama de despliegue propuesto para el sistema.

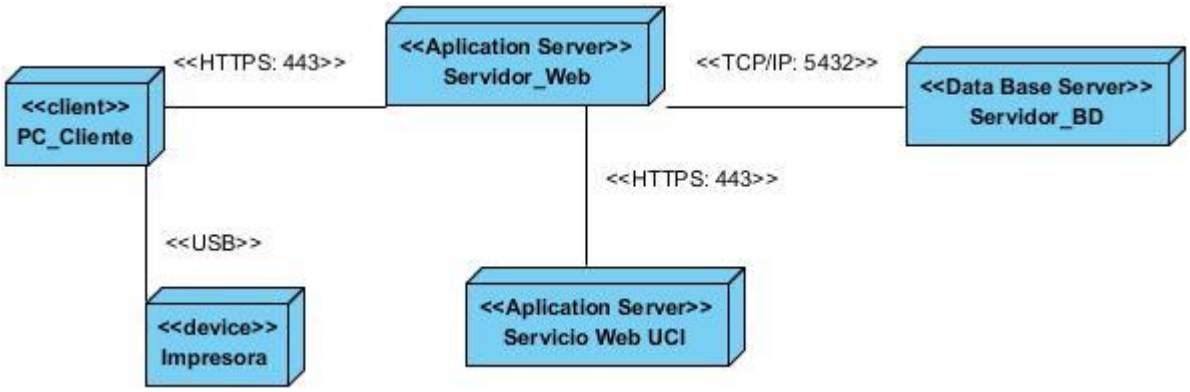


Fig. 19: Diagrama de despliegue de ALFACITEC.

En la figura 19 se muestra el diagrama de despliegue correspondiente a la propuesta de solución donde se representan físicamente los servidores de la aplicación y de la base de datos conectados por el protocolo TCP/IP mediante el puerto 5432. Por su parte se tiene la PC cliente la cual cuenta con una impresora vía USB para la impresión de modelos y se conecta mediante HTTPS al servidor web utilizando el puerto 443. Por su parte dicho servidor se conecta al servicio web LDAP mediante el puerto 443 utilizando el protocolo HTTPS.

Conclusiones Parciales

Durante el desarrollo de este capítulo se realizó el modelado del negocio y del sistema obteniendo 52 RF agrupados en 17 CU estructurados por distintos patrones como CRUD Total y Parcial entre otros. Se relacionó los distintos casos de uso mediante el diagrama de casos de uso del negocio y del sistema y se describió formalmente el CU más crítico a través de la especificación para una mejorar comprensión de los flujos del negocio y el sistema. Se conformó el Diagrama de paquetes en el cual se muestran los elementos del Modelo, la Vista y el Controlador. Por su parte el Diagrama de Clases del diseño presentado muestra las especificaciones de las clases de software y de las interfaces utilizando diversos patrones de diseño. Una vez realizado el diseño de la solución se obtuvo el DER del modelo de datos que muestra las distintas tablas de la BD, concluyendo con la obtención del modelo físico de despliegue encargado de mostrar la distribución física del sistema ALFACITEC.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DEL ALMACÉN DE LIBROS DE LA CITEC

En el presente capítulo se abordan los flujos de implementación y pruebas del sistema. Se describe y representa el sistema a nivel de componentes a través del diagrama del mismo nombre. Se especifican los estilos de programación y estándares de codificación empleados y finaliza con la definición del modelo de pruebas donde se refleja el resultado de la ejecución de dichas pruebas.

3.1 Modelo de implementación

El modelo de implementación describe como los elementos del modelo de diseño se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. El modelo de implementación describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros. El modelo brinda una mejor visualización del camino a seguir por los desarrolladores del sistema en el momento de enfrentar la implementación del mismo.

3.1.1 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos y sus relaciones en el entorno. Los componentes representan los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas, y las relaciones de dependencia se utilizan para indicar que un componente utiliza los servicios ofrecidos por otro componente. Además, agrupa varias partes de un sistema modular, desplegables y reemplazables, que encapsulan la implementación y expone un conjunto de interfaces como, por ejemplo: código fuente, binario o ejecutable (Larman, 2005).

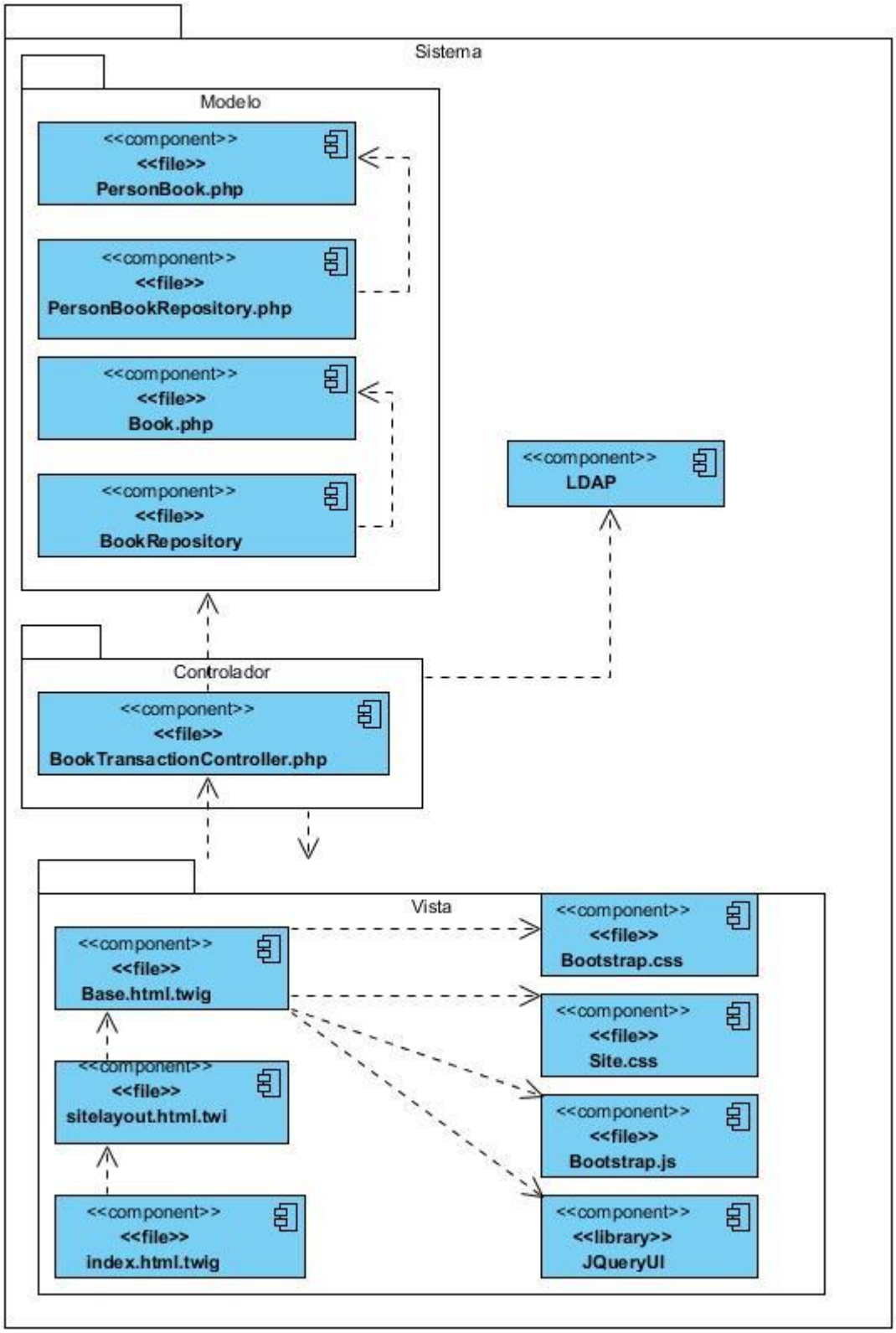


Fig. 20: Diagrama de componentes del CU Administrar préstamo.

El diagrama de componentes correspondiente al CU Administrar préstamo muestra las principales dependencias entre los principales elementos que conforman la estructura del sistema ubicados en el paquete “Sistema”. El paquete “Vista” se encarga de agrupar las vistas correspondientes a las

acciones que ejecuta el usuario. En el paquete “Modelo” se muestran los componentes encargados de manejar los elementos referentes al negocio. Por su parte el paquete “Controlador” se encarga de manejar la lógica de control del sistema. El componente LDAP se relaciona con el controlador y permite la utilización del servicio web UCI para la autenticación y búsqueda de usuarios.

3.2 Estándares de codificación

El código fuente es un conjunto de líneas que conforman un bloque de texto, escrito según las reglas sintácticas de algún lenguaje de programación destinado a ser legible por humanos. Es un programa en su forma original, tal y como fue escrito por el programador, no es ejecutable directamente por el computador, debe convertirse en lenguaje de máquina mediante compiladores, ensambladores o intérpretes (Pergamino Virtual, 2015).

El propósito fundamental de los estándares de codificación es organizar y estilizar consistentemente el sistema con independencia del autor para que resulte fácil de entender y de mantener. Teniendo en cuenta que la propuesta de solución se desarrolla utilizando el marco de trabajo Symfony 2.8, se define el uso de los estándares PSR-1 (PHP Standards Recommendation 1) y PSR-2 (PHP Standards Recommendation 2). Estos estándares comprenden un grupo de elementos a tener en cuenta a la hora de codificar y que tiene como característica una uniformidad de estilos y reglas a cumplir.

- Se debe emplear solamente la codificación UTF-8 para el código PHP.
- Debe haber una línea en blanco después de la declaración del *“namespace”* y otra después del bloque de declaraciones *“use”*.
- Se deben utilizar solamente las etiquetas <?php y <?=.
- El código debe usar 4 espacios como indentación, no tabuladores.
- Los paréntesis de apertura en las estructuras de control no deben tener un espacio después de ellos, y los paréntesis de cierre no deben tener un espacio antes de ellos.
- Las llaves de apertura de las estructuras de control deben estar en la misma línea, y las de cierre deben ir en la línea siguiente al cuerpo.

Se muestra un fragmento de código en la Fig. **21**.

```
public function newAction(Request $request) {
    $book = new Book();
    $form = $this->createForm('AppBundle\Form\BookType', $book);
    $form->add('send', 'submit', array('label' => 'Crear',
        'attr' => array(
            'class' => 'btn btn-primary pull-right',)));
    $form->handleRequest($request);
    if ($form->isSubmitted() && $form->isValid()) {
        try {
            $em = $this->getDoctrine()->getManager();
            $em->persist($book);
            $em->flush();
            $this->addFlash('success', 'El libro se ha registrado satisfactoriamente.');
            return $this->redirectToRoute('book_index');
        } catch (Exception $ex) {
            $this->addFlash('danger', $ex->getMessage());
            return $this->redirectToRoute('book_index');
        }
    }
    $items = array(
        array("Registro de libros", "ti ti-briefcase", "book_index", null),
        array("Registrar", null, null, null));
    $this->get('my_breadcrumb')->generateBreadcrumb($items);
    return $this->render('book/new.html.twig', array(
                'book' => $book,
                'form' => $form->createView(),));
}
```

Fig. 21: Fragmento del código fuente correspondiente a la implementación del RF Adicionar libro**.**

3.3 Proceso de pruebas

Las pruebas de software son el proceso de ejercitar el software con la intención de encontrar y corregir errores. Estas verifican que el software implemente una función específica de forma correcta y validan que las acciones que realiza respondan a los requisitos del cliente (Pressman, 2005).

Las pruebas del sistema se encargan del comportamiento de un medio completo. Son similares a las pruebas de integración, pero con un alcance mucho más amplio. Estas son consideradas comúnmente como las apropiadas para comparar el sistema con los requisitos no funcionales del sistema, como seguridad, velocidad, exactitud y confiabilidad. También se evalúan en este nivel, las interconexiones externas con otras aplicaciones, utilidades, dispositivos hardware o con el sistema operativo (SWEBOK, 2013).

Según (Pressman, Ingeniería del Software, 2008) se establece una serie de reglas que pudieran entenderse como los objetivos fundamentales del proceso de pruebas:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no

descubierto hasta entonces.

- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Los procesos de prueba para las aplicaciones web comienzan con pruebas que ejercitan el contenido y la funcionalidad de la interfaz que es inmediatamente visible para los usuarios finales. Conforme se realizan las pruebas, se ejercitan los aspectos de la arquitectura de diseño y la navegación. Finalmente, la atención se centra en las pruebas que ejercitan las capacidades tecnológicas referentes a la infraestructura de la aplicación y cuestiones de instalación e implementación (Pressman, 2002).

3.3.1 Tipos de pruebas

Existe toda una tipología de pruebas aplicables a distintos contextos de desarrollo de aplicaciones. Según (Pressman, 2008) existen siete tipos fundamentales de pruebas recomendadas sobre aplicaciones web las cuales son: pruebas de contenido, interfaz, navegación, componentes, configuración, seguridad y desempeño o rendimiento.

Para la presente solución se requiere la realización de un grupo de pruebas que redunden en la correcta validación del sistema, como son:

- **Pruebas unitarias**: Son pruebas que realiza el equipo de desarrollo y que, según Pressman: verifiquen que los componentes unitarios están codificados bajo condiciones de robustez, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada (Pressman, 2010).

- **Pruebas funcionales**: Ejercita el contenido y las unidades funcionales dentro de la aplicación y se enfocan sobre un conjunto de pruebas que intentan descubrir errores en las mismas.

- **Prueba de aceptación**: Representa aquella fase del ciclo de vida de desarrollo de software en el que el equipo de desarrollo y el área usuaria de un sistema de información tiene que garantizar que el sistema desarrollado se corresponde con los requerimientos definidos (Ponce, Dominguez, Gutierrez, & Escalona, 2014).

- **Prueba de integración**: Las pruebas de integración prueban la interacción entre dos o más elementos, que pueden ser clases, módulos, paquetes o subsistemas, entre otros; incluso la interacción del sistema con el entorno de producción. El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido

probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes (Pablo Navarrete, 2012).

- **Prueba de rendimiento**: Se aplican para descubrir problemas, debido a la falta de recursos del lado del servidor, ancho de banda de red inapropiado, poca capacidad en la base de datos o en el sistema operativo, funcionalidades mal diseñadas y otros conflictos de hardware o software que puedan inducir a un bajo desempeño cliente-servidor (Pressman, 2008).

3.3.2 Aplicación de las pruebas

No todos los productos de software requieren la aplicación de todos los tipos de pruebas que existen, ya que cada una de ellas es aplicada para comprobar un determinado objetivo en determinado momento del ciclo de vida del software. A continuación, se muestra la aplicación de las pruebas que se le efectuaron a la solución.

Pruebas Unitarias

Como parte de las pruebas unitarias se utilizó el método de caja blanca utilizando la técnica de camino básico. Denominadas en ocasiones como pruebas de caja de cristal, las pruebas de caja blanca es un método que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante esta prueba, el ingeniero de software puede obtener casos de prueba que, según (Pressman, 2008):

1- Garanticen que se ejercite por lo menos una vez todos los caminos independientes de cada módulo.
2- Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
3- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
4- Ejerciten las estructuras internas de datos para asegurar su validez.

Para la aplicación de la técnica de camino básico es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática. Este se realiza de tres formas diferentes:

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
- La complejidad ciclomática V(G) de un grafo de flujo G se define como: V(G)= A-N+2 donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.

- V(G) también se calcula como el resultado de P+1 donde P es el número de nodos predicados (nodos de los cuales parten dos o más aristas) que tiene contenido el grafo de flujo G.

Ejemplo de código de la funcionalidad newAction

```
public function newAction(Request $request) {
    $book = new Book();
    $form = $this->createForm('AppBundle\Form\BookType', $book);
    $form->add('send', 'submit', array('label' => 'Crear',
        'attr' => array(
            'class' => 'btn btn-primary pull-right',)));
    $form->handleRequest($request);
    if ($form->isSubmitted() && $form->isValid()) {
        try {
            $em = $this->getDoctrine()->getManager();
            $em->persist($book);
            $em->flush();
            $this->addFlash('success', 'El libro se ha registrado satisfactoriamente.');
            return $this->redirectToRoute('book_index');
        } catch (Exception $ex) {
            $this->addFlash('danger', $ex->getMessage());
            return $this->redirectToRoute('book_index');
        }
    }
    $items = array(
        array("Registro de libros", "ti ti-briefcase", "book_index", null),
        array("Registrar", null, null, null));
    $this->get('my_breadcrumb')->generateBreadcrumb($items);
    return $this->render('book/new.html.twig', array(
                'book' => $book,
                'form' => $form->createView(),));
}
```


Fig. 22: Código de la funcionalidad newAction.

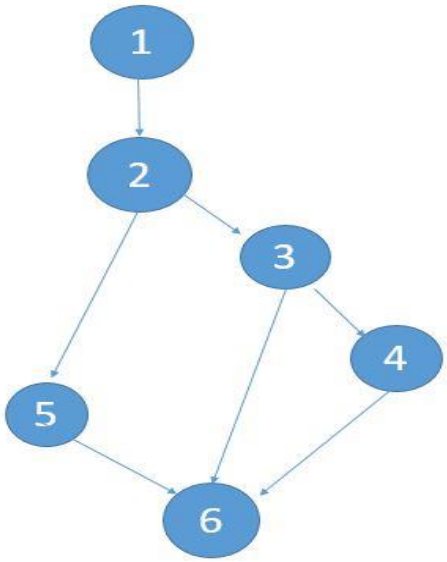


Fig. 23: Grafo de flujo asociado a la funcionalidad newAction.

Resultado del cálculo de la complejidad ciclomática:

V(G)= A-N+2=7-6+2=3

V(G)= P+1=2+1=3

A partir del resultado obtenido se determina que la funcionalidad presenta una complejidad ciclomática de 3, lo que deriva que existe a lo sumo 3 caminos lógicos por donde ejecutarse dicha funcionalidad. En la **Tabla 11** se muestran los caminos básicos.

Tabla 11: Caminos básicos.

No.	Camino básico
1	1-2-5-6
2	1-2-3-6
3	1-2-3-4-6

Diseño de casos de prueba. Caja Blanca:

Según (Peña, 2006) se le podría llamar caso de prueba al conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular, como, por ejemplo, ejercitar un camino concreto de un programa o verificar el cumplimiento de un determinado requisito.

A continuación, se muestra el Diseño de Casos de Prueba (DCP) correspondiente a los caminos básicos generados a través de las pruebas de caja blanca.

Tabla 12: DCP- Camino básico No. 1.

Caso de prueba	
Camino 1-2-3-6	
Descripción:	Adicionar un nuevo libro
Condiciones de Ejecución	Tiene que estar autenticado Tiene que tener los permisos pertinentes
Entrada	Se introducen todos los datos válidos
Resultados esperados	Se adiciona el libro correctamente

Tabla 13: DCP- Camino básico No. 2.

Caso de prueba	
Camino 1-2-5-6	
Descripción:	Adicionar un nuevo libro
Condiciones de Ejecución	Tiene que estar autenticado Tiene que tener los permisos pertinentes
Entrada	Se introducen datos válidos y se dejan uno o varios campos obligatorios vacíos
Resultados esperados	Se muestra un mensaje de error para que llene los campos obligatorios

Tabla 14: DCP-Camino No. 3.

Caso de prueba	
Camino 1-2-3-4-6	
Descripción:	Adicionar un nuevo libro
Condiciones de Ejecución	Tiene que estar autenticado Tiene que tener los permisos pertinentes
Entrada	Se introducen todos los campos obligatorios y de los no obligatorios se deja alguno vacío
Resultados esperados	El libro se adiciona correctamente

Pruebas Funcionales

Como parte de las pruebas funcionales se utilizó el método de caja negra utilizando la técnica Partición de equivalencia. Las pruebas de caja negra también conocidas como prueba de comportamiento, se centran en los requisitos funcionales del software, y permiten al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (Pressman, 2008). Este método busca encontrar errores en cinco categorías:

1- Funciones incorrectas o ausentes.
2- Errores de interfaz.
3- Errores en estructuras de datos o en accesos a bases de datos externas.
4- Errores de rendimiento.
5- Errores de inicialización y terminación.

El método de caja negra permite examinar los valores válidos e inválidos de las entradas existentes en el software. Además, esta técnica se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número de clases de prueba a desarrollar. Para su implementación se divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software a partir de las cuales pueden derivarse casos de prueba (Pressman, 2008).

Diseño de casos de prueba para caja negra:

A continuación, se muestra los DCP correspondiente al CU Administrar préstamo. Los DCP correspondientes a los demás CU se encuentran disponibles en los artefactos del expediente del proyecto.

DCP: CU Administrar préstamo

Descripción general

Este CU se inicia cuando el usuario desea recibir o entregar un libro y se realiza la búsqueda en el sistema por su usuario o solapín. Culmina con una de las siguientes acciones sobre un usuario, el préstamo, devolución y listado de libros.

Condiciones de ejecución

La Encargada del almacén debe estar autenticada en el sistema y debe tener permisos para realizar las mencionadas acciones. Debe buscar el usuario al que se le realizará un préstamo, una devolución o listar sus libros prestados. En el caso de la devolución, el usuario debe tener al menos un libro prestado.

Secciones a probar en el CU Administrar préstamo

Tabla 15: Secciones a probar en el CU Administrar préstamo

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: "Realizar préstamo"	**EC 1.1**: La encargada del almacén inserta los datos del usuario deseado y selecciona los libros deseados por el usuario.	El sistema almacena la dependencia usuario-libro y la fecha de préstamo guardando una entrega de libro a un usuario.	1. Click en "Servicios del almacén". 2. Click en "Entrega y devolución". 3. Inserta datos de usuario en el campo. 4. Selecciona libros a entregar. 5. Click en Guardar. 6. Selecciona enviar en la ventana de confirmación de operación.
	EC 1.2: La encargada del almacén deja campos vacíos.	El sistema muestra un mensaje indicando que existen campos vacíos.	1. Click en "Servicios del almacén". 2. Click en "Entrega y devolución". 3. Inserta datos de usuario en el campo. 4. Selecciona libros a entregar 5. Click en Guardar. 6. Selecciona enviar en la ventana de confirmación de operación.
	EC 1.3: La encargada del almacén introduce datos incorrectos.	El sistema muestra un mensaje indicando que existen datos incorrectos.	1. Click en "Servicios del almacén". 2. Click en "Entrega y devolución". 3. Inserta datos de usuario en el campo. 4. Selecciona libros a entregar. 5. Click en Guardar. 6. Selecciona enviar en la ventana de confirmación de operación.
	EC 2.1: La encargada del	El sistema elimina la	1. Click en "Servicios del almacén".

SC 2: "Eliminar préstamo"	almacén selecciona los libros que el usuario desea entregar.	dependencia existente entre usuario-libro y la fecha de préstamo.	2. Click en "Entrega y devolución". 3. Inserta datos de usuario en el campo. 4. Selecciona libros a devolver 5. Click en Guardar. 6. Selecciona enviar en la ventana de confirmación de operación.
	EC 2.2: La encargada del almacén deja campos vacíos.	El sistema muestra un mensaje indicando que existen campos vacíos.	1. Click en "Servicios del almacén". 2. Click en "Entrega y devolución". 3. Inserta datos de usuario en el campo. 4. Selecciona libros a devolver 5. Click en Guardar. 6. Selecciona enviar en la ventana de confirmación de operación.
	EC 2.3: La encargada del almacén introduce datos incorrectos.	El sistema muestra un mensaje indicando que existen datos incorrectos.	1. Click en "Servicios del almacén". 2. Click en "Entrega y devolución". 3. Inserta datos de usuario en el campo. 4. Selecciona libros a devolver 5. Click en Guardar. 6. Selecciona enviar en la ventana de confirmación de operación.
SC 3: "Listar préstamo"	**EC 3.1**: Mostrar listado de libros	El sistema muestra el listado de libros por años donde aparecen los libros prestados marcados por la fecha de entrega.	1. Click en "Servicios del almacén". 2. Click en "Entrega y devolución". 3. Inserta datos de usuario en el campo. 4. Despliega los años docentes visualizando los correspondientes libros.

Descripción de las variables

Tabla 16: Descripción de las variables.

No.	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Usuario	Campo de texto	No	Debe ser una combinación de caracteres perteneciente a un usuario del sistema uci.
2	Solapín	Campo de texto	No	Debe ser una combinación de caracteres perteneciente a un usuario del sistema uci.
3	Entrega	Campo Caja de selección	No	Debe marcarse el campo deseado.

		múltiple		
4	Devolución	Campo Caja de selección múltiple	No	Debe marcarse el campo deseado.
5	Año docente	Lista desplegable	No	Debe seleccionarse una de las opciones definidas en el sistema.

Matrices de datos.

Adicionar préstamo

Tabla 17: Matriz de datos Adicionar préstamo.

ID del Escenario	Escenario	Usuario	Solapín	Entrega	Devolución	Año docente	Respuesta del sistema	Resultado de la prueba
EC 1.1	La encargada del almacén inserta de forma correcta los datos.	V/ (ariveronh)	V/ (EH12415)	V/(Marcado)	V/ (No Marcado)	V/ (Quinto año)	El sistema adiciona una entrega de libro con la fecha de préstamo y muestra un mensaje que confirma que se guardó correctamente	Satisfactorio
EC 1.2	La encargada del almacén deja campos vacíos	I/ (vacío)	V/ (vacío)	V/(No Marcado)	V/ (No Marcado)	V/ (Quinto año)	El sistema muestra un mensaje indicando que existen campos vacíos que deben llenarse para realizar la operación	Satisfactorio
EC 1.3	La encargada del almacén introduce datos incorrectos	V/ (ariveronh78)	V/ (EH12415)	V/(No Marcado)	V/ (No Marcado)	V/ (Quinto año)	El sistema muestra un mensaje indicando que existen datos incorrectos.	Satisfactorio

Eliminar préstamo

Tabla 18: Matriz de datos. Eliminar préstamo

ID del Escenario	Escenario	Usuario	Solapín	Entrega	Devolución	Año docente	Respuesta del sistema	Resultado de la prueba
EC 2.1	La encargada del almacén inserta de forma correcta los datos.	V/ (ariveronh)	V/ (EH12415)	V/(No Marcado)	V/ (Marcado)	V/ (Quinto año)	El sistema realiza la devolución de uno o varios libros y muestra un mensaje de operación exitosa.	Satisfactorio
EC 2.2	La encargada del almacén deja campos vacíos	I/ (ariveronh)	V/ (E121415)	V/(No Marcado)	V/ (No Marcado)	V/ (Quinto año)	El sistema no permite activar el botón de guardar debido a que no hay ningún libro marcado para devolver.	Satisfactorio
EC 2.3	La encargada del almacén introduce datos incorrectos	V/ (ariveronh.78)	V/ (EH12415)	V/(Marcado)	V/ (No Marcado)	V/ (Quinto año)	El sistema muestra un mensaje indicando que existen datos incorrectos	Satisfactorio

Listar préstamo

Tabla 19: Matriz de datos. Listar préstamo

ID del Escenario	Escenario	Usuario	Solapín	Entrega	Devolución	Año docente	Respuesta del sistema	Resultado de la prueba
EC 2.1	La encargada del almacén inserta de forma correcta los datos.	V/ (ariveronh)	V/ (EH12415)	V/(No Marcado)	V/ (Marcado)	V/ (Quinto año)	El sistema muestra el listado de libros correspondiente a Quinto año	Satisfactorio
EC 2.2	La encargada del almacén deja campos vacíos	I/ (Vacío)	V/ (E121415)	V/(No Marcado)	V/ (No Marcado)	V/ (No desplegado)	El sistema no permite obtener el listado de libros del usuario y muestra un mensaje de campo vacío.	Satisfactorio
EC 2.3	La encargada del almacén introduce datos incorrectos	V/ (ariveronh.78)	V/ (EH12415)	V/(Marcado)	V/ (No Marcado)	V/ (Quinto año)	El sistema no permite obtener el listado de libros del usuario y muestra un mensaje de datos incorrectos.	Satisfactorio

Pruebas de rendimiento

Para el desarrollo de las pruebas de rendimiento se pueden aplicar pruebas de carga y de tensión. En la presenta investigación solamente se realiza las pruebas de carga debido a que el sistema no estará expuesto a grandes niveles de concurrencia significativa de usuarios. Las pruebas de carga son utilizadas para valorar y validar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales (Pressman, 2008).

Para la realización de las pruebas de rendimiento se utilizó la herramienta Apache JMeter en su versión 2.3.1. El sistema fue instalado en un entorno de prueba con las siguientes características en el servidor:

- Memoria RAM 2 GB
- Disco Duro: 500 GB
- Sistema Operativo: Ubuntu Server 12.04

Pruebas de integración

Para la realización de las pruebas de integración se utilizó la técnica de integración descendente donde se parte del módulo principal, en este caso el sistema en cuestión y se prueba la correcta relación al realizar consultas al componente Web Service de la UCI en la autenticación y búsqueda de diez usuarios diferentes en diversos momentos del día. Con la realización de estas pruebas no se obtuvieron No Conformidades quedando correctamente validada la integración. La siguiente figura muestra un ejemplo de las pruebas de integración realizadas.

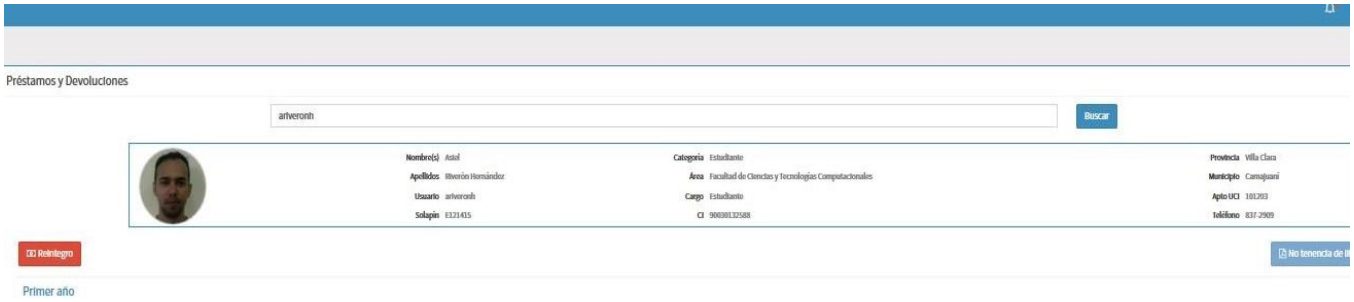


Fig. 24: Validación de las pruebas de integración.

Pruebas de aceptación

Para el desarrollo de las pruebas de aceptación se utilizaron las de tipo alfa, que son las que realiza el cliente, una vez recibido el producto final y su documentación, de conjunto con los desarrolladores del sistema. Este proceso se realizó en el Vicedecanato de Administración de la FCITEC en presencia de los desarrolladores y los especialistas del área quienes se encargaron de comprobar las funcionalidades y aprobar el producto. La realización de las pruebas de aceptación tuvo como salida la liberación de un acta de aceptación, la cual se anexa a este documento, con firma de los clientes de la solución.

3.3.4. Resultado de la aplicación de las pruebas

Como parte de la ejecución de las pruebas y teniendo en cuenta como objetivo esencial el de identificar en qué medida satisface la aplicación las funcionalidades implementadas, se realizó una

primera iteración de pruebas, donde fueron aplicados los diseños de casos de pruebas realizados. En esta primera iteración se identificaron en total de 13 No Conformidades, clasificadas en 9 no significativas y 4 significativas. Una vez corregidas se procedió a realizar una segunda iteración de pruebas donde se identificaron 3 nuevas No Conformidades de tipo no significativas. Luego de corregidas estas deficiencias se procedió a la realización de una tercera iteración de pruebas obteniendo resultados satisfactorios, por lo que se determina no realizar nuevas iteraciones dando por concluidas las pruebas.

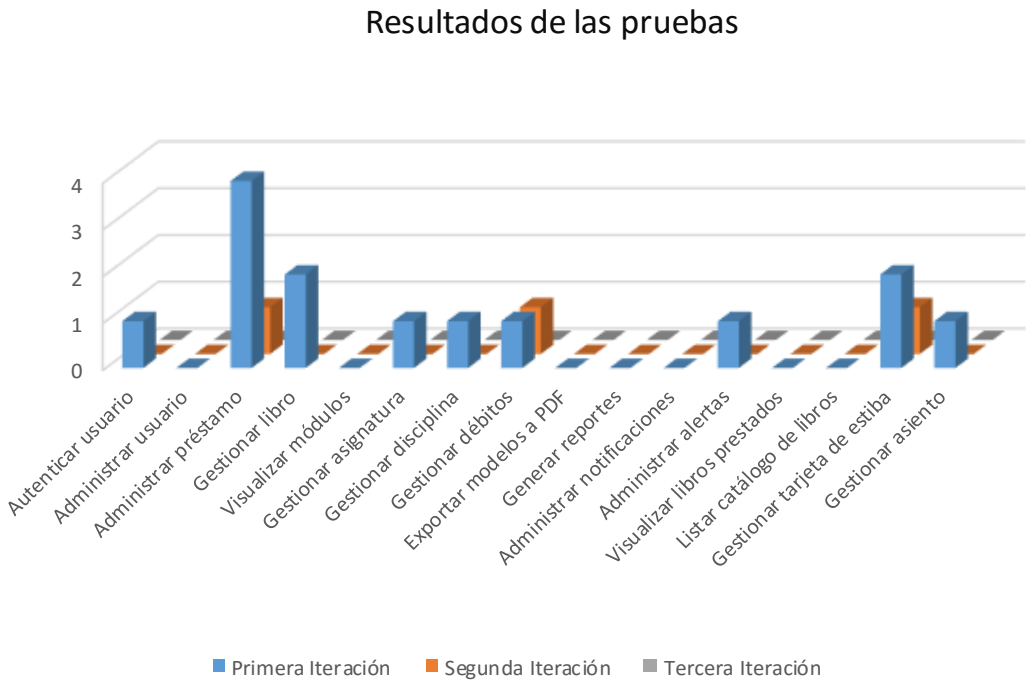
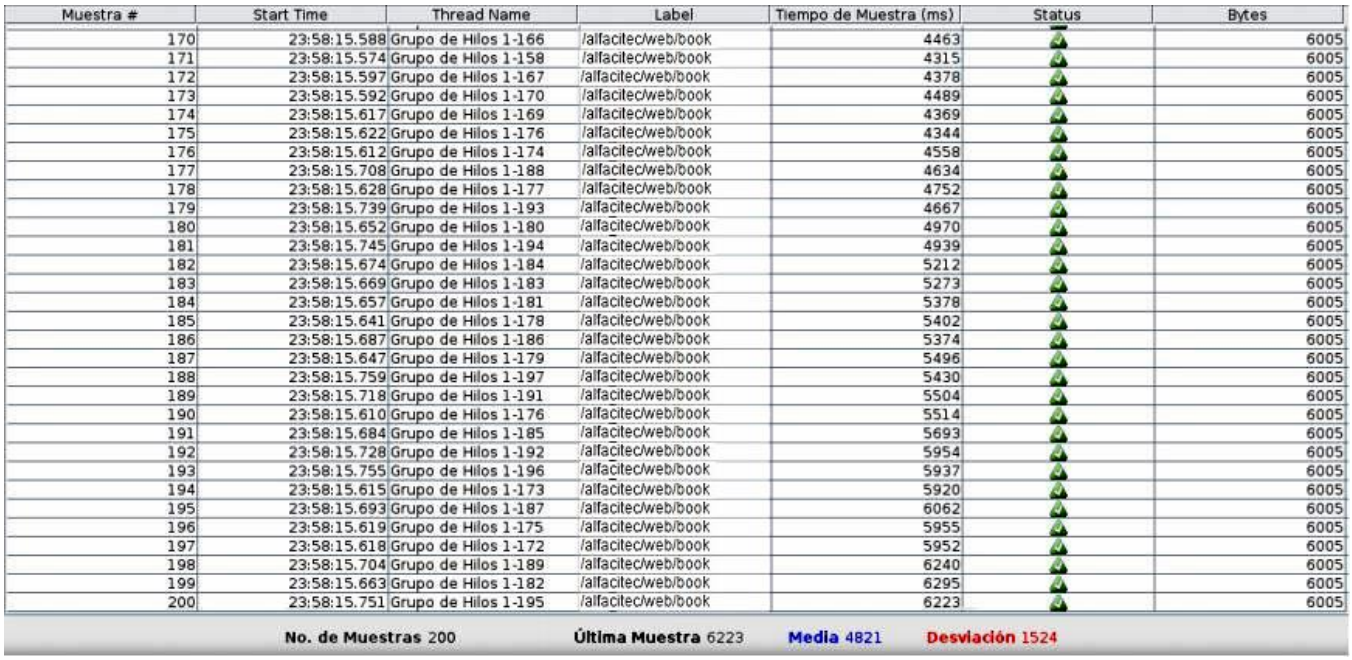


Fig. 25: Resultado de las pruebas

Con la realización de las pruebas de carga mediante la herramienta JMeter en su versión 2.3.1, se validó el RNF No. 3 el cual plantea que el tiempo promedio de respuesta del sistema no debe superar los 5 segundos con 200 usuarios conectados simultáneamente. El resultado de estas pruebas se muestra en la siguiente figura donde se observa un tiempo de respuesta para cada petición con una media de 4821 milisegundos, los errores detectados durante el proceso de petición y respuesta en Status y la desviación entre el mayor y el menor tiempo de respuesta.



Muestra #	Start Time	Thread Name	Label	Tiempo de Muestra (ms)	Status	Bytes
170	23:58:15.588	Grupo de Hilos 1-166	/alfacitec/web/book	4463		6005
171	23:58:15.574	Grupo de Hilos 1-158	/alfacitec/web/book	4315		6005
172	23:58:15.597	Grupo de Hilos 1-167	/alfacitec/web/book	4378		6005
173	23:58:15.592	Grupo de Hilos 1-170	/alfacitec/web/book	4489		6005
174	23:58:15.617	Grupo de Hilos 1-169	/alfacitec/web/book	4369		6005
175	23:58:15.622	Grupo de Hilos 1-176	/alfacitec/web/book	4344		6005
176	23:58:15.612	Grupo de Hilos 1-174	/alfacitec/web/book	4558		6005
177	23:58:15.708	Grupo de Hilos 1-188	/alfacitec/web/book	4634		6005
178	23:58:15.628	Grupo de Hilos 1-177	/alfacitec/web/book	4752		6005
179	23:58:15.739	Grupo de Hilos 1-193	/alfacitec/web/book	4667		6005
180	23:58:15.652	Grupo de Hilos 1-180	/alfacitec/web/book	4970		6005
181	23:58:15.745	Grupo de Hilos 1-194	/alfacitec/web/book	4939		6005
182	23:58:15.674	Grupo de Hilos 1-184	/alfacitec/web/book	5212		6005
183	23:58:15.669	Grupo de Hilos 1-183	/alfacitec/web/book	5273		6005
184	23:58:15.657	Grupo de Hilos 1-181	/alfacitec/web/book	5378		6005
185	23:58:15.641	Grupo de Hilos 1-178	/alfacitec/web/book	5402		6005
186	23:58:15.687	Grupo de Hilos 1-186	/alfacitec/web/book	5374		6005
187	23:58:15.647	Grupo de Hilos 1-179	/alfacitec/web/book	5496		6005
188	23:58:15.759	Grupo de Hilos 1-197	/alfacitec/web/book	5430		6005
189	23:58:15.718	Grupo de Hilos 1-191	/alfacitec/web/book	5504		6005
190	23:58:15.610	Grupo de Hilos 1-176	/alfacitec/web/book	5514		6005
191	23:58:15.684	Grupo de Hilos 1-185	/alfacitec/web/book	5693		6005
192	23:58:15.728	Grupo de Hilos 1-192	/alfacitec/web/book	5954		6005
193	23:58:15.755	Grupo de Hilos 1-196	/alfacitec/web/book	5937		6005
194	23:58:15.615	Grupo de Hilos 1-173	/alfacitec/web/book	5920		6005
195	23:58:15.693	Grupo de Hilos 1-187	/alfacitec/web/book	6062		6005
196	23:58:15.619	Grupo de Hilos 1-175	/alfacitec/web/book	5955		6005
197	23:58:15.618	Grupo de Hilos 1-172	/alfacitec/web/book	5952		6005
198	23:58:15.704	Grupo de Hilos 1-189	/alfacitec/web/book	6240		6005
199	23:58:15.663	Grupo de Hilos 1-182	/alfacitec/web/book	6295		6005
200	23:58:15.751	Grupo de Hilos 1-195	/alfacitec/web/book	6223		6005

No. de Muestras 200 **Última Muestra** 6223 **Media** 4821 **Desviación** 1524

Fig. 26: Interfaz de salida JMeter

Conclusiones parciales

Durante el desarrollo de este capítulo quedó definido el modelo de implementación del sistema de gestión mediante el diagrama de componentes lo que permitió representar las principales dependencias que conforman la estructura del sistema. La distribución física mediante el diagrama de despliegue ofreció desde la perspectiva física, la distribución de los elementos operativos para el despliegue. Los estándares de codificación y los estilos de programación utilizados fueron descritos para hacer más fácil el entendimiento del código y permitir un mejor mantenimiento de la aplicación en un futuro. Mediante el desarrollo del proceso de pruebas se pudo identificar y resolver los errores en la implementación aumentando la calidad del producto final obtenido. Se comprobó que el mismo satisface las funcionalidades requeridas y cumple con los Requisitos No Funcionales planteados.

CONCLUSIONES

Una vez culminada la investigación se puede afirmar que se les dio cumplimiento a los objetivos planteados, arribando a las siguientes conclusiones.

El estudio del arte sobre los sistemas de gestión de la información, enmarcados en los sistemas para la gestión de información referente a libros arrojó que estas soluciones no satisfacen la problemática, lo cual evidencia la necesidad de esta investigación.

Se caracterizó los principales lenguajes y herramientas empleados en el desarrollo de la aplicación fundamentando en cada caso la selección y permitiendo identificar el ambiente de desarrollo de acuerdo a las necesidades del proyecto.

El proceso de modelado del negocio y del sistema ejecutado tuvo como resultado los diagramas y artefactos necesarios para guiar el desarrollo del sistema.

La implementación de las funcionalidades viabilizó el cumplimento de los requisitos funcionales definidos para la implementación obteniendo como resultado un sistema funcional acorde a los requerimientos especificados.

Las pruebas realizadas y la corrección de las deficiencias detectadas en cada una de las iteraciones posibilitaron la terminación exitosa de un producto de calidad.

REFERENCIAS BIBLIOGRÁFICAS

1. Actores, C. d. (2014). *Análisis de la información.* Armenia: Facultad de Ciencias Humanas y Bellas Artes.
2. Alegsa, L. (2016). *Diccionario de informática y tecnología*. Obtenido de http://www.alegsa.com.ar/Dic/informatica.php
3. Almeira, A. S. (2007). *Arquitectura de Software: Estilos y Patrones.* Buenos Aires: Universidad Nacional de la Patagonia.
4. Bachman, C. (s.f.). *The programmer as navigator. Comunications of the ACM.*
5. Camacho, E., Cardeso, F., & Nuñez, G. (2004). *Arquitecturas de Software. Guía de estudio.*
6. Coello González, S., & Hernández León, R. (2011). Proceso de Investigación Científica. La Habana: Alianza Universidad.
7. Dante, G. P. (1998). *Gestión de información en las organizaciones: Principios, Conceptos y Aplicaciones.* Santiago de Chile.
8. Eguiluz, J. (2006). *Introducción a JavaScript.*
9. *El pensante. Educación*. (abril de 2016). Obtenido de Análisis de información. Definición: http://educacion.elpensante.com/analisis-de-informacion-definicion/
10. Flores, E. M. (2014). La gestión y los gestores de la información. *Bibliodocencia*.
11. Foundation, J. (2014). *What is query.* Obtenido de http://www.jquery.com
12. Fowler, M. (2013). *Patterns of Enterprise Application Architecture.*
13. Gacitúa, R. (2013). *Métodos de desarrollo de software. El desafío pendiente de la estandarización.*
14. Gaona, A. L. (2012). *El Modelo Entidad Relación.* México.
15. González, D. A. (2013). *Diagrama de casos de uso del negocio.* La Habana: Instituto Superior Politécnico Jose Antonio Echevarría.
16. GROUP. (12 de 09 de 2011). *PostgreSQL*. Obtenido de Equipo Global de Desarrollo de PostgreSQL: http://www.postgresql.org/about/press/presskit91/es/
17. Hernández, A. (2012). *Observatorio de la economía latinoamericana*. Obtenido de http://www.eumed.net/cursecon/ecolat/cu/12
18. Histchfeld, N., & Salinas Caro, P. (20 de abril de 2014). *Tutorial de UML*. Obtenido de http://users.dcc.uchile.cl/psalinas/uml/introduccion.html
19. Jacobson, I., & Boocch, G. (2000). *El proceso unificado de desarrollo de software.* Madrid: Pearson Education S.A.
20. Larman, C. (2005). *UML y Patrones.* California: Prentice Hall.
21. Martínez, R. (2 de octubre de 2010). *PostgreSQL-es*. Obtenido de Portal en español sobre PostgreSQL: http://www.postgresql.org.es/sobre_postgresql
22. Morán, J. M. (2010). *Framework para la capa de presentación de aplicaciones web.*

23. NETBEANS. (2015). *¿Qué es Netbeans?*

24. Network, D. (2017). *Crear y usar reglas de negocios.* Obtenido de https://msdn.microsoft.com/es-es/library/aa577691.aspx

25. Pablo Navarrete. (2012). *Pruebas de integración.* Recuperado el 15 de 05 de 2017, de https://es.slideshare.net/pablis001/

26. Peña, J. M. (2006). *Pruebas de Software.* Obtenido de Universidad Veracruzana: http://www.uv.mx/personal/jfernandez/material-de-curso

27. Perea, J. (2009). *Abadia Digital*. Obtenido de http://www.abadiadigital.com

28. Pergamino Virtual. (2015). *Pergamino Virtual*. Obtenido de http://pergaminoVirtual.com.ar/definicion/Codigo_Fuente.html

29. Pleva, J. (2013). *PHP: Hypertext Preprocessor.* Wisconsin: Universidad de Wisconsin.

30. Ponce, J. F., Dominguez, F., Gutierrez, J., & Escalona, M. (2014). Pruebas de aceptación orientadas al usuario. contexto ágil para un proyecto de gestión. *Ibersid: Revista de sistemas de información y comunicación*.

31. Pressman, R. (2005). *Software Engineering. Sixth edition.*

32. Pressman, R. S. (2002). *Ingeniería de software 5ta Edición.* McGrawHill.

33. Pressman, R. S. (2008). *Ingeniería del Software.*

34. Pressman, R. S. (2010). *Ingeniería de Software: Un enfoque práctico 7ma edición.*

35. Rumbaugh, J., Jacobson, I., & Boocch, G. (2011). *UML Reference Manual.*

36. Sánchez, E. J., & Sangucho, E. (2015). *Desarrollo de un sistema informático para la gestión de la biblioteca en la Unidad educativa del milenio en la ciudad de Santo Domingo de los colorados.* Santo Domingo de los Colorados: Pontificia Universidad Católica del Ecuador. Obtenido de http://www.puce.edu.ec/portal/content/Bibliotecas%20Virtuales/174

37. Sánchez, T. R. (2014). *Metodología de desarrollo para la actividad productiva de la UCI.* La Habana.

38. Schmuller, J. (2010). *Aprendiendo UML en 24 horas.*

39. Sommerville, I. (2005). *Ingeniería del Software. Séptima Edición.* Madrid: Pearson Addison Wesley.

40. SWEBOK. (2013). *IEEE Computer Society.* Obtenido de http://www.computer.org/portal/web/swebok

41. Team, S. T. (2011). *Características de PHP.*

42. Zaninneti, J. (2008). *Symfony: la guía definitiva.*

ANEXOS

Anexo 1: Entrevista realizada a: Elcides Mauro Rosa Figueredo: Técnico de la Dirección de Compras y Almacenes. Fecha: 15-02-2017

Objetivo: Identificar las principales características y comprender el funcionamiento del almacén central y los pedidos que se realizan desde las diversas facultades.

1- ¿Cómo funciona el proceso de solicitud de nuevos libros para la universidad?

2- ¿Cómo se realiza la distribución de los libros a las diversas facultades?

3- ¿Qué deficiencias usted considera, afecten el actual proceso de préstamos y devoluciones?

4- ¿Cómo se llenan los distintos campos presentes en los modelos oficiales y su significado?

5- ¿Cuál es la función principal del almacén central una vez que en cada facultad se crearon almacenes de libros?

Anexo 2: Entrevista realizada a: Bárbara Cornelio Gámez: Encargada del almacén de libros de la Facultad de Ciencias y Tecnologías Computacionales. Fecha: 17-02-2017

Objetivo: Comprender el funcionamiento del almacén de libros en cuanto a los servicios que allí se brindan.

1- ¿Descripción del flujo de pasos de los distintos procesos de préstamos, devoluciones, pérdidas, modelo de no tenencia?

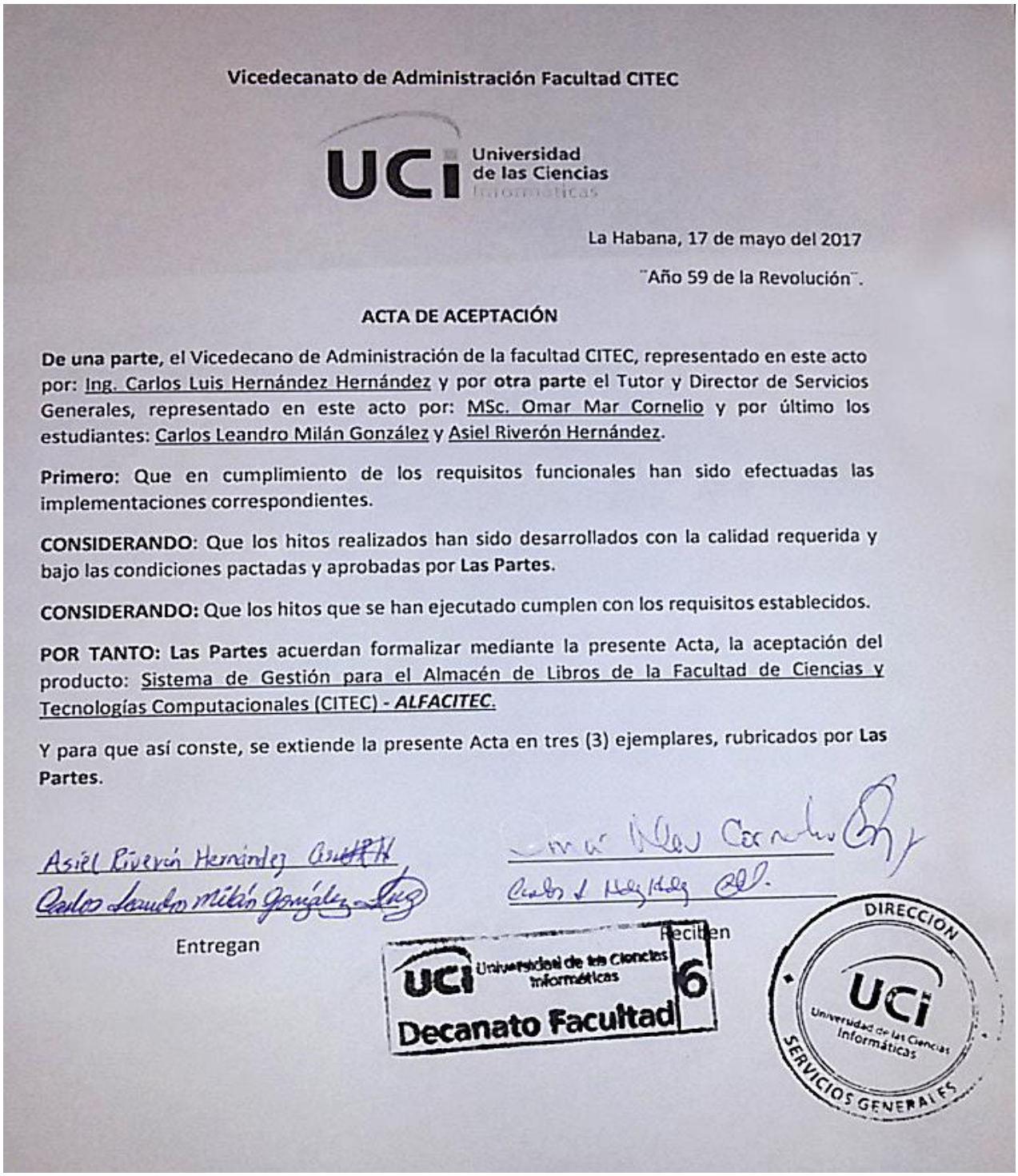
2- ¿Cuál es el tratamiento que tiene el proceso de pérdida de un libro y como se realiza el pago?

3- ¿Cuáles son las reglas o normas que usted como encargada del almacén debe cumplir con respecto al servicio que se brinda?

4- ¿Cuáles son las principales deficiencias que usted como encargada opina que existen?

Anexo 3: Acta de Aceptación de la solución propuesta.

Acta de aceptación emitida y aprobada por el MSc. Omar Mar Cornelio y Ing. Carlos Luis Hernández Hernández. Fecha 17-05-2017.



Vicedecanato de Administración Facultad CITEC

UCi Universidad de las Ciencias Informáticas

La Habana, 17 de mayo del 2017

"Año 59 de la Revolución".

ACTA DE ACEPTACIÓN

De una parte, el Vicedecano de Administración de la facultad CITEC, representado en este acto por: Ing. Carlos Luis Hernández Hernández y por **otra parte** el Tutor y Director de Servicios Generales, representado en este acto por: MSc. Omar Mar Cornelio y por último los estudiantes: Carlos Leandro Milán González y Asiel Riverón Hernández.

Primero: Que en cumplimiento de los requisitos funcionales han sido efectuadas las implementaciones correspondientes.

CONSIDERANDO: Que los hitos realizados han sido desarrollados con la calidad requerida y bajo las condiciones pactadas y aprobadas por **Las Partes**.

CONSIDERANDO: Que los hitos que se han ejecutado cumplen con los requisitos establecidos.

POR TANTO: Las Partes acuerdan formalizar mediante la presente Acta, la aceptación del producto: Sistema de Gestión para el Almacén de Libros de la Facultad de Ciencias y Tecnologías Computacionales (CITEC) - ***ALFACITEC***.

Y para que así conste, se extiende la presente Acta en tres (3) ejemplares, rubricados por **Las Partes**.

Asiel Riverón Hernández

Carlos Leandro Milán González

Entregan

Reciben

UCi Universidad de las Ciencias Informáticas Decanato Facultad 6

DIRECCION UCi Universidad de las Ciencias Informáticas SERVICIOS GENERALES