



# Sistema de Gestión de Control de Libros para la Facultad 1

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

**Autor(es):** Rachel Marichal Casaña.

**Tutor(es):** Aux.MSc. Geidis Sánchez Michel

**Facultad 1**

La Habana, diciembre de 2022

Año 64 de la Revolución Cubana



## **DECLARACIÓN DE AUTORÍA**

El autor del trabajo de diploma con título “**Sistema de Gestión de libros para la Facultad 1.**”, concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los **<día>** días del mes de **<mes>** del año **2022**.

**Rachel Marichal Casaña**

---

Firma del Autor

**Geidis Sánchez Michel**

---

Firma del Tutor

## **DATOS DE CONTACTO**

Aux.MSc. Geidis Sánchez Michel, profesora de Redes y Seguridad Informática del departamento de Sistemas Digitales de la Facultad 1, master en Gestión de Proyectos Informáticos y Vicedecana de Economía y Administración.

## **AGRADECIMIENTOS**

A todos los que de una forma u otra hicieron posible este logro, principalmente:

1. A mi familia por tanto esfuerzo, paciencia, dedicación, por su guía infalible y por enseñarme el valor del trabajo, los amo.
2. A mi novio por sus consejos, ayuda incondicional y desinteresada, por escucharme y resistirme todo este tiempo.
3. A mis hermanos, cuñados y sobrinos por formar parte de esta aventura y su ayuda inmediata en respuesta a cualquiera de mis necesidades.
4. A mi tutora por su trabajo y tiempo empleado en esta tarea, a pesar de su apretada agenda fue una guía importante durante el proceso.
5. A todos mis compañeros, juntos logramos alcanzar este logro y no cabe duda de la influencia de cada uno de ustedes en el proceso.
6. A todos esos profesores que todos los días lo dan todo en las aulas y contribuyen en la formación de cientos de ingenieros.
7. A la Universidad de Ciencias Informáticas por darme la oportunidad de cumplir mi sueño.

## **DEDICATORIA**

A las estrellas de mi universo, mi madre, por confiar siempre en mí, motivarme a convertirme en una mejor persona, por apoyar cada uno de mis pasos, por su entrega y dedicación y por ser sencillamente especiales. A mis hermanos, por convertirse en mis consejeros, por ser las personas que son y por hacerme saber que siempre pase lo que pase puedo confiar en ellos.

## **RESUMEN**

En la actualidad los sistemas de gestión son herramientas dedicadas a la optimización de los procesos de negocio. Ellos permiten un desarrollo sostenible en la gestión de políticas y procedimientos, que garantizan a la organización cumplir determinadas metas. La presente investigación está encaminada a solucionar los problemas referentes a la gestión del proceso de entrega y recepción de libros en la Facultad 1 de la Universidad de las Ciencias Informáticas, a partir de un sistema que te permitirá el control de libros. Para dar cumplimiento al objetivo se plantean los conceptos fundamentales referentes a la investigación, se realiza una investigación sobre los antecedentes de la gestión de control de libros a nivel nacional e internacional. Se describen las herramientas utilizadas, el sistema de gestión de contenidos o CMS sobre el que se implementó el sistema y los elementos necesarios para su construcción. Así como las técnicas de validación aplicadas a la propuesta de solución.

**PALABRAS CLAVE:** Gestión de libro, sistema de gestión, entrega y recepción.

## **ABSTRACT**

Currently, management systems are tool dedicated to optimizing business processes. They allow sustainable development in the management of policies and procedures, which guarantee the organization to meet certain goals. The present investigation is aimed at solving the problems related to the management of the process of delivery and reception of books in faculty 1 of the University of Informatics Sciences, from a system that will allow you to control books. In order to fulfill the objective, the fundamental concepts related to the investigation are proposed, an investigation is carried out on the antecedents of the management of book control at the national and international level. The tools used, the CMS on which the system was implemented and elements necessary for its construction are described. As well as the results of the same that obtained a more efficient and effective management.

**KEYWORDS:** Book management, management system, delivery and reception.

## TABLA DE CONTENIDOS

INTRODUCCIÓN .....	10
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO .....	15
Introducción al capítulo.....	15
I.1 Conceptos asociados al dominio del problema I.1 .....	15
1.2 Estudio de sistemas homólogos.....	17
1.2.1 Sistemas Homólogos Internacionales:.....	17
1.2.2 Sistemas Homólogos Nacionales: .....	18
I.3 Metodología, lenguaje, herramientas y entorno de desarrollo I.3 .....	20
1.3.1 Metodología de desarrollo de software .....	20
1.3.2 Lenguaje de modelado .....	21
1.3.3 Herramienta de modelado .....	22
1.3.4 Framework de desarrollo: .....	22
1.3.5 Lenguaje de Programación.....	24
1.3.6 Entorno de Desarrollo Integrado .....	25
1.3.7 Gestor de Base de Datos.....	26
1.3.8 Herramientas para validar el sistema.....	27
1.3.9 Servidor Web. ....	28
Conclusiones del capítulo.....	29
CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO .....	30
Introducción al capítulo.....	30
2.1. Modelo Conceptual: .....	30
2.2 Descripción de la Propuesta de solución: .....	31

2.3 Requisitos de la Propuesta de Solución: .....	31
2.3.1 Requisitos Funcionales: .....	32
2.3.2 Requisitos no funcionales: .....	34
2.4 Historia de Usuario .....	36
2.5 Arquitectura de Software .....	39
2.6 Patrones de Diseño .....	41
2.6.1 Descripción de los patrones para asignar responsabilidades “Patrones GRASP”: .....	41
2.7 Diagrama de clases de diseño:.....	43
2.8 Diagrama de Secuencias:.....	46
2.9 Modelo de Datos.....	49
Conclusiones del capítulo.....	51
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	52
3.1 Diagrama de Componente.....	52
3.2 Estándares de Codificación .....	53
3.3 Diagrama de Despliegue .....	55
3.4 Estrategia de prueba.....	56
3.4.1 Pruebas funcionales.....	56
3.4.3 Pruebas de rendimiento .....	60
3.5 Validación científica de la propuesta de solución .....	62
CONCLUSIONES FINALES.....	66
RECOMENDACIÓN.....	67
REFERENCIAS BIBLIOGRÁFICAS.....	68
ANEXOS .....	74

**ÍNDICE DE TABLAS**

Tabla 1 : Resumen de Sistemas Homólogos.....	19
Tabla 2 . Requisitos Funcionales.....	322
Tabla 3 . HU_7 Solicitar Libro .....	37
Tabla 4 . HU_10 Adicionar Libro.....	38
Tabla 5 . HU_9 Listar Solicitudes.....	39
Tabla 6 . Estándares de Codificación .....	533
Tabla 7 . Caso de prueba RF_7 Solicitar Libro .....	577
Tabla 8 . Resultado de las pruebas de rendimiento.....	62
Tabla 9. Relación de la satisfacción individual con la escala de satisfacción.....	63

## ÍNDICE DE FIGURAS

1Figura #1 Modelo conceptual (Elaboración propia) .....	30
2 Figura #2 Arquitectura de Software .....	40
3Figura # 3 Ejemplo del patrón controlador .....	41
4Figura # 4 Ejemplo del patrón alta cohesión .....	42
5Figura # 5 Ejemplo del patrón bajo acoplamiento .....	48
6Figura # 6 DCD con estereotipos web Solicitar Libro (elaboración propia).....	49
7Figura # 7 DCD con estereotipos web Modificar Libro (elaboración propia) .....	45
8Figura # 8 DCD con estereotipos web Listar Libros (elaboración propia).....	46
9Figura # 9 DCD con estereotipos web Listar Usuario (elaboración propia) .....	48
10Figura # 10 DS con estereotipos web Solicitar .....	479
11Figura # 11 DS con estereotipos web Modificar Libro (elaboración propia).....	50
12Figura # 12 Modelo de Datos (elaboración propia).....	50
13Figura # 13 DC con estereotipos web Solicitar Libro (elaboración propia) .....	52
14Figura # 14 Diagrama de despliegue (elaboración propia) .....	55
15Figura # 15 Gráfico resumen de las pruebas funcionales (elaboración propia) .....	59
16Figura # 16 Gráfico resumen de las pruebas de seguridad (elaboración propia) .....	60

## INTRODUCCIÓN

La evolución de la tecnología ha permitido que tanto la captura de los datos, como la gestión del conocimiento en las organizaciones sea más ágil. La gestión del conocimiento, son los procesos y acciones de detección, selección, organización, filtrado, presentación y uso de la información por parte de los actores de una organización (Torres, 2015).

Como parte del proceso de gestión del conocimiento las universidades juegan un papel fundamental, como espacio para el intercambio académico y científico.

En las universidades cubanas se realiza a través de una buena gestión bibliográfica, la cual es de vital importancia para el proceso de enseñanza y aprendizaje de los estudiantes, ya que es imprescindible para su conocimiento, como base para las investigaciones científicas y futuras tesis de maestrías y doctorados, además, eleva la calidad y pertinencia en las funciones sustantivas (Díaz, 2021).

En este sentido los libros de texto constituyen un componente esencial en la gestión de la bibliografía para las asignaturas en las universidades.

Algunos autores se enfocan en cuatro premisas para la gestión de libros en el escenario actual: la estructura informacional vigente, la capitalización de los recursos humanos, la disponibilidad de herramientas y plataformas tecnológicas que satisfacen las necesidades requeridas para la gestión de la información y el conocimiento, y las innovaciones requeridas para facilitar el proceso de dirección en las organizaciones (Álvarez 2015; Blas y Planells 2021; Ortiz y Nagles, 2013).

Imponiendo a las universidades la necesidad de llevar un control de los recursos bibliográficos en formato de libro de texto. Como parte de ese control la Universidad de las Ciencias Informáticas (UCI) cuenta con un almacén de libros en cada facultad, donde los estudiantes adquieren los volúmenes de las asignaturas que cursan como apoyo al proceso docente-educativo.

Dentro de estos procesos se encuentra la entrega y recepción de libros que se llevan a cabo cada semestre; encontrándose a disposición de los estudiantes y profesores, tanto del curso

regular diurno como el curso por encuentros. Este proceso se va ejecutando durante el transcurso de las actividades docentes, ya que se puede solicitar un préstamo o devolver un libro en cualquier momento del curso.

Para realizar este control es necesario realizar un inventario donde se relacione la información referente a la cantidad de libros y títulos disponibles, así como la cantidad de libros en préstamo.

El proceso de actualización y control de la cantidad de libros en préstamos, así como la disponibilidad de libros en el local se realiza de forma manual y gradual a medida que se solicita un préstamo o se devuelve un libro, esto genera una considerable cantidad de documentos en formato duro para archivar debido a la cantidad de estudiantes y trabajadores que atiende el Vicedecanato, lo cual conlleva a que en ocasiones se cometan errores al registrar o controlar un préstamo o devolución de un libro, hace que la labor de control, sea lenta, extensa y agotadora. Además, da origen a la duplicidad de la información y la pérdida de datos de la base material (libros).

Por lo anteriormente expuesto surge el siguiente **problema de investigación**: ¿Cómo facilitar el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1?

El **objeto de estudio** se centra en los procesos de gestión de entrega y recepción de libros, comprendidos en el **campo de acción** entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1.

Como **objetivo general** se plantea desarrollar un sistema web que facilite el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1.

Para guiar el cumplimiento del objetivo general planteado, se formulan las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos relacionados con el uso de herramientas informáticas empleadas en el desarrollo del sistema para el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1 de la UCI?
2. ¿Cuál es el estado actual de las herramientas informáticas estudiadas para el desarrollo del sistema para el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1 de la UCI?
3. ¿Qué elementos deben tenerse en cuenta para llevar a cabo el análisis y diseño del sistema para el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1 de la UCI?
4. ¿Cómo materializar, en términos de componentes y código fuente, los elementos especificados para el sistema para el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1 de la UCI?
5. ¿Qué resultados se obtendrán al validar, a través de una estrategia de pruebas de software, el sistema para el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1 de la UCI?

Para dar cumplimiento a las preguntas científicas se trazaron las siguientes **Tareas de Investigación**:

- 1- Sistematización de los referentes teóricos que sustentan la investigación, relacionados con el uso de las herramientas informáticas para el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1 de la UCI.
- 2- Selección de las herramientas y tecnologías a utilizar para el desarrollo del sistema web que facilite el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1.
- 3- Especificación de los recursos funcionales y no funcionales para un sistema web que facilite el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía

y Administración de la Facultad 1.

4- Implementación de un sistema web que garantice que facilite el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad

5- Validación, a través de una estrategia de prueba, del sistema web que facilite el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1.

Para el desarrollo de la investigación se emplean los siguientes **métodos de investigación:**

**Métodos teóricos:**

- **Analítico-Sintético:** este método se utiliza para arribar a conclusiones prácticas y teóricas de la investigación que se desarrolla. Mediante este se realiza un análisis de los documentos, las publicaciones, bibliografías y en general toda la información relacionada con sistemas de gestión para control de libros y sistemas web que se emplean para tal fin.
- **Modelación:** se utiliza para realizar el modelado del sistema web a desarrollar, a través de los artefactos correspondientes a las fases: Modelo del Negocio, Requisitos, Análisis y Diseño e Implementación.

**Métodos Empíricos:**

- **Entrevista:** se usa con el objetivo de conocer la estructura organizativa del proceso de gestión para control de libros en la Facultad 1.
- **Observación:** se emplea para obtener una información más precisa de cómo se ejecuta el proceso de gestión para control de los libros.
- **Encuesta:** Se emplea para la recopilación de información sobre el proceso de gestión para control de libros en la Facultad 1 de la UCI.

La estructura del documento está conformada por los siguientes capítulos:

### Capítulo I: Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio

En el capítulo se exponen los fundamentos teóricos asociados al dominio del problema. Se realiza un análisis de los principales sistemas para la gestión de control de libros. Se expone la metodología, las herramientas, tecnologías y lenguajes de programación seleccionados para el desarrollo de la aplicación web que permita la gestión para el control de libros en la Facultad 1.

### Capítulo II: diseño de la solución propuesta al problema científico

Este es el capítulo para desarrollar un sistema informático es importante la utilización de métodos y técnicas que permitan la solución de problemas existentes a lo largo del ciclo de vida de un software. Dentro de este capítulo se obtienen un conjunto de artefactos generados a partir de la utilización de la metodología AUP\_UCI; incluyendo el ajuste de los planes del proyecto, considerando los requisitos y la arquitectura.

### Capítulo III: diseño de la solución propuesta al problema científico

En el capítulo se describe el diagrama de componentes, los estándares de codificación y el diagrama de despliegue. Además, de documentar los resultados obtenidos durante el desarrollo de la estrategia de pruebas.

## **CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO**

### **Introducción al capítulo**

En el presente capítulo se exponen los fundamentos teóricos asociados al dominio del problema. Se realiza un análisis de los principales sistemas para la gestión de control de libros. Se expone la metodología, las herramientas, tecnologías y lenguajes de programación seleccionados para el desarrollo de la aplicación web.

### **I.1 Conceptos asociados al dominio del problema I.1**

El libro nace oficialmente desde el momento en que el hombre inventó la escritura, desde ese instante se utilizó para registrar los conocimientos más importantes de la cultura de los pueblos (El Universo, 2019).

El libro es una obra impresa, manuscrita o pintada en una serie de hojas de papel, pergamino, vitela u otro material, unidas por, un lado, y protegidas con tapas, también llamadas cubiertas. Un libro puede tratar sobre cualquier tema. Según la definición de la Unesco, un libro debe poseer veinticinco hojas mínimo, pues de veinticuatro hojas o menos sería un folleto; y de una hasta cuatro páginas se consideran hojas sueltas (en una o dos hojas) (Morales, 2016).

Para la Real Academia Española un libro es una obra científica, literaria o de cualquier otra índole con extensión suficiente para formar volumen, que puede aparecer impresa o en otro soporte (ASALE y RAE, 2021).

Por su parte (Definista, 2022) plantea que un libro es el conjunto de hojas de papel, vitela, u otra sustancia, manuscritas o impresas, colocadas en el orden en que se han de leer, y que reunidas o encuadernadas forman un volumen.

Al analizar estos conceptos se puede concluir que un libro es una obra que puede tener carácter científico, literaria o de cualquier otra índole, que se expresa a través de volúmenes de información que pueden tener un carácter impreso o digital.

Los libros en el proceso de enseñanza y aprendizaje constituyen un referente de imprescindible prioridad para los estudiantes, permitiendo la correcta formación de representaciones históricas, el desarrollo de la independencia cognoscitiva y la formación de métodos de trabajo científico. Por lo que constituyen un soporte bibliográfico para la impartición de las asignaturas (Porta, 2015).

En tal sentido el Ministerio de Educación Superior (MES) de la República de Cuba en su Resolución No.47/22 Reglamento Organizativo del Proceso Docente y de Dirección del Trabajo Docente y Metodológico para las Carreras Universitarias, establece en los artículos 140.1 y 147.1 las responsabilidades que tienen los colectivos de disciplinas y de asignaturas de proponer las bibliografías básicas y complementarias para la impartición de las asignaturas. Además, en los artículos 236 y 245 establece que para el diseño de los programas de la disciplina y analíticos de las asignaturas la bibliografía forma parte de su estructura (Resolución47-2022, 2022).

Es una realidad indiscutible que los textos escolares han sido, desde el comienzo de su existencia, un medio básico en educación. A pesar de la proliferación de modernos medios tecnológicos continúa prevaleciendo en las universidades el uso de los libros de texto como material didáctico (Ríos, 2002).

Siendo necesario en este sentido, que las universidades lleven un control de los libros de textos que tienen en préstamo. Entiéndase por préstamo de un libro la autorización que se otorga a los usuarios para llevar fuera de la biblioteca, de manera gratuita, los libros de su interés a fin de que puedan leerlos en el momento y lugar que deseen (Sousa, 2004).

Otra visión del concepto de préstamo de un libro lo realiza (El Cantarano, 2022) el cual lo considera como el movimiento de un libro de la biblioteca para facilitar al lector su uso fuera del recinto bibliotecario. Se considera como un contrato tácito por el que una biblioteca permite a un usuario la utilización de un documento durante un período de tiempo determinado.

Con el análisis realizado a estos conceptos se puede concluir que el préstamo de un libro es el movimiento de un libro de la biblioteca o local de almacenamientos, de manera autorizada, durante un periodo de tiempo determinado.

## **1.2 Estudio de sistemas homólogos.**

A continuación, se analizan sistemas de gestión de libros en el ámbito internacional y nacional, con el objetivo de identificar sistemas capaces de solucionar la problemática en su totalidad o que permita reutilizar algunas funcionalidades para brindar una solución al problema planteado.

La búsqueda realizada para la selección de los sistemas en análisis, descartó aquellos sistemas de los cuales no se tenía acceso a las funcionalidades o se encontraban en idiomas diferentes del inglés y el español.

### **1.2.1 Sistemas Homólogos Internacionales:**

#### **CodeAchi Library Management System**

CodeAchi Library Management System, también conocida como CodeAchi LMS, es un software para la gestión digital de una biblioteca. Los módulos disponibles están diseñados con la guía de la Organización Internacional de Estándares para Bibliotecas y Biblioteconomía. Este producto puede satisfacer cada consulta individual relacionada con la información de su biblioteca. Es un sistema de pago, aunque presenta cierta flexibilidad a la hora de comprarlo. Presenta prioridad de soporte, posventa y seguridad de los datos. Entre sus funciones principales se encuentran: búsqueda, categorización, gestión de archivos y gestión de usuarios. Este software solo es compatible con los sistemas operativos, Windows 7 o superiores (Capterra, 2020).

#### **Calibre**

Calibre es un software de gestión de libros electrónicos debido a su excelente conjunto de funciones que hacen posible a usuarios de Windows administrar su biblioteca virtual. Este software de administración de libros electrónicos ofrece numerosas opciones de administración para mantener organizada su biblioteca de libros electrónicos. Incluye una búsqueda avanzada de libros electrónicos y clasificación, que descarga todos los metadatos de un libro electrónico

(metadatos como título, autor, editor, ISBN, etiquetas, portada o resumen), transferencia de e-book de un dispositivo a otro (de forma inalámbrica o con un cable), admite todas las principales vistas de formato de libro electrónico y biblioteca de respaldo para libros electrónicos.

Calibre tiene un vasto repositorio de complementos. Además, el software de gestión Calibre e-book es compatible con Windows 7 o superiores (Vázquez, 2015).

### **Resource Mate**

Es un software de gestión bibliotecaria diseñado para iglesias, sinagogas, escuelas, profesionales, corporaciones y organizaciones sin ánimo de lucro. Entre sus funciones principales se encuentran: el préstamo y la devolución, búsqueda, categorización, gestión de archivos, contenidos, usuarios y circulación y agrupación. Además, permite administrar libros de textos y mostrar información sobre la actualización de los cambios realizados. Es un software de carácter propietario, con facilidad de pago. Solo puede ser utilizado en países como Canadá, Estados Unidos, Irlanda y Reino Unido ( Capterra, 2022).

### **1.2.2 Sistemas Homólogos Nacionales:**

#### **Sistema Integral de Gestión Cedrux:**

Tiene como objetivo garantizar la disponibilidad de los libros; así como ejecutar los procesos de gestión de manera eficiente y dinámica, promoviendo el uso racional de recursos. Facilita el registro de los productos, sus existencias, la gestión y control de lotes; así como los movimientos de entrada y salida de los libros en el almacén.

Este sistema, aunque cuenta con una aceptación positiva entre los usuarios, se encuentra en explotación solamente en la base de almacenes perteneciente al Ministerio de las Fuerzas Armadas Revolucionarias de Cuba. Se puede utilizar en sistemas operativos como Linux o Windows (González, Camejo y Gainza, 2012).

#### **ABC.uci.cu (Catálogo)**

Catálogo es un sistema que posibilita el trabajo; en la biblioteca de la Universidad de las Ciencias Informática; con la existencia de préstamos y devoluciones de libros tanto a estudiantes

como a trabajadores, controlando así el movimiento de los libros, con el fin de garantizar una organización exitosa. Tiene como objetivo garantizar el mejoramiento del proceso y asegurarle a la comunidad los recursos necesarios para el adecuado uso de sus fuentes. Este permite la administración eficaz y eficiente de funciones como: circulación, consulta, adquisición y descarte.

Entre sus características se encuentran que utiliza software libre para su desarrollo, se puede utilizar en sistemas operativos como Linux o Windows. Es un software que presenta problemas de soporte y seguridad de datos, por lo que está en constante actualización (Entrevista a la bibliotecaria).

A continuación, en la Tabla 1 se presenta una tabla resumen donde se recogen los datos más significativos de cada uno de los sistemas analizados:

*Tabla 1: Resumen de Sistemas Homólogos*

Herramienta	Código abierto	Multiplataforma	Gestión de Préstamo	Exportar información personalizada
CodeAchi Library Management System	No	No	No	No
Calibre	No	No	No	No
Resource Mate	No	Si	Si	No
Sistema Integral de Gestión Cedrux	No	Si	Si	No
Catálogo	Si	Si	Si	No

Los sistemas estudiados en el ámbito internacional y nacional permitieron identificar funcionalidades que pueden ser adoptadas en la propuesta de solución, tales como: gestionar usuarios, realizar búsquedas, registrar libros, realizar préstamo y devolución de los libros, entre otros.

Estos sistemas tienen como desventaja que, excepto Catálogo el acceso a su código fuente es privativo, por lo que no cumplen con el paradigma de independencia tecnológica que hoy aboga el país. Sistemas como: CodeAchi Library Management System y Calibre no son multi-plataforma y no realizan el proceso de gestión de préstamos. En su totalidad los sistemas carecen de la funcionalidad de exportar información, siendo de gran importancia para el negocio.

El estudio de los sistemas descritos, permitió afirmar que los mismos no pueden ser utilizados para satisfacer las necesidades actuales del Vicedecanato de Economía y Administración de la Facultad 1 de la UCI, por lo que se propone desarrollar un sistema web que facilite el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1.

### **I.3 Metodología, lenguaje, herramientas y entorno de desarrollo I.3**

A continuación, se exponen la metodología, lenguajes, herramientas y entorno de desarrollo utilizados en la propuesta de solución.

#### **1.3.1 Metodología de desarrollo de software**

La metodología de desarrollo, en Ingeniería de Software, es un conjunto de herramientas, técnicas, procedimientos y soporte documental encaminados a estructurar, planificar y controlar el proceso de desarrollo de forma organizada y lógica, que tiene como objetivo apoyar a los desarrolladores en la creación de un nuevo software (Pressman, 2010).

Se considera por (Santander, 2020) que la metodología de desarrollo son los pasos de fabricación del software con la tarea de trazar una línea secuencial de las tareas y actividades a realizarse, haciendo uso de herramientas, métodos, modelos y una buena gestión del equipo de desarrollo para que este aporte a la calidad final del sistema.

Existen varios tipos de metodologías como: Unified Rational Process (RUP), Extreme Programming (XP), Agile Unified Process (AUP) y Agile Unified Process versión UCI (AUP\_UCI). Esta última describe de una manera simple y fácil de entender la manera de desarrollar aplicaciones de software de negocio utilizando técnicas ágiles.

La metodología AUP\_UCI propone tres fases para el desarrollo del software (Inicio, Ejecución y Cierre) y siete disciplinas (Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación), además, existen tres modos de encapsular los requerimientos: Casos de Uso del Sistema (CUS), Historias de Usuario (HU) y Descripción de Requerimientos por Proceso (DRP) y estos están agrupados en cuatro escenarios (Rodríguez, 2018).

Para la propuesta de solución se propone utilizar la metodología AUP-UCI, a través del escenario 4, teniendo en cuenta que dicho escenario se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta manera modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda esta combinación de escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

Además, el cliente estará siempre acompañando al equipo de desarrollo para conveniar los detalles de los requerimientos y así poder implementarlos, probarlos y validarlos. El tratamiento de las HU es dinámico y flexible, lo que permite que estas puedan reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas de una manera más fácil, sin invertir tanto tiempo y esfuerzo.

### **1.3.2 Lenguaje de modelado**

Un lenguaje de modelado provee un vocabulario y conjunto de reglas centradas en la representación conceptual y física de un sistema. El Lenguaje Unificado de Modelado (Unified Modeling Language), es un lenguaje informático, gráfico o textual que sigue un conjunto semántico de reglas y marcos para el diseño y construcción de estructuras y modelos. Se utiliza para visualizar, especificar, construir y documentar software orientados a objetos. Permite a los

creadores de sistemas generar diseños que capturen sus ideas en una forma fácil de comprender y comunicar a otras personas (Amaya, 2017).

La elección de UML 2.5 se sustenta en su facilidad para especificar, documentar, visualizar y comprender los artefactos generados en las diferentes fases del desarrollo de software, así como sus ventajas para la comunicación entre el cliente y el equipo de desarrollo.

### **1.3.3 Herramienta de modelado**

Las herramientas de modelado son el nombre con que se identifica la herramienta empleada para apoyar las actividades del proceso de software. Facilitan la interacción entre los miembros de un equipo al hacer que la diagramación sea un proceso iterativo y dinámico.

Los analistas de sistemas se apoyan en estas herramientas desde el principio hasta el fin del ciclo de vida, para incrementar la productividad, comunicarse de manera eficiente con los usuarios y desarrolladores, e integrar el trabajo que desempeñan en el sistema (Martínez, et., 2014).

### **Visual Paradigm 16.3**

Visual Paradigm es una herramienta CASE que admite UML 2.5, SysML y notación de modelado de procesos de negocios del grupo administración de objetos. Además del soporte de modelado, proporciona capacidades de generación de informes e ingeniería de códigos, incluida la generación de códigos. Puede realizar ingeniería inversa de diagramas a partir del código y proporcionar ingeniería de ida y vuelta para varios lenguajes de programación (Verencia y Fonseca, 2015).

Se propone como herramienta de modelado para la propuesta de solución debido a que es multiplataforma. Contribuye al desarrollo de sistemas de software fiables, mediante un enfoque orientado a objetos, y soporta el ciclo completo de desarrollo de software, generando documentación en diferentes formatos, empleando UML como lenguaje de modelado.

### **1.3.4 Framework de desarrollo:**

Un marco de trabajo (framework) es una gran librería o conjunto de librerías, donde además de facilitar funciones para su uso, dispone de una sintaxis o metalenguaje específico del marco

de trabajo y una forma de organización de su código. Para usar un marco de trabajo no basta con conocer el nombre de la función a utilizar: hay que saber qué sintaxis emplea, qué obligaciones impone (a la hora de organizar el código, archivos) y su lógica o filosofía de trabajo. El empleo de marcos de trabajo se ha extendido debido a la gran complejidad de las aplicaciones que se desarrollan, ya que facilitan su organización y mantenimiento (Gómez, et., 2016).

### **Django 4.1**

Es un marco de desarrollo en Python, de código abierto y muy ligero, que permite la creación rápida de páginas y aplicaciones web. Pretende ser sencillo y rápido, lo que posibilita concentrarse principalmente en escribir la aplicación.

Las principales características que presenta este marco de trabajo son:

- ✧ Basado en la filosofía DRY (Don't Repeat Yourself, no te repitas). Muchas aplicaciones web y proyectos digitales comparten numerosas líneas de código unos con otros. Django es el marco de desarrollo de refactorización de código casi por excelencia. Permite reutilizar programación de unas aplicaciones a otras sin la obligación de tener que repetir las mismas líneas de código entre distintos proyectos.
- ✧ Es de alto nivel basado en el patrón arquitectónico Modelo-Vista-Plantilla. Apuesta por la sencillez, la rapidez y la reutilización de código.
- ✧ Viene integrado con SQLite, una base de datos usada por compañías tan importantes como Facebook o Bloomberg.
- ✧ Contiene una API (Application Programming Interface) propia para el desarrollo de proyectos digitales (Django, 2019).

Se decide utilizar Django porque permite construir una aplicación en poco tiempo, que funciona y es estable. Adaptar el código a las necesidades, ya que se puede encontrar una solución tanto en la comunidad como en algún paquete de Python o en el propio framework. Es muy seguro ya que este implementa por defecto algunas medidas de seguridad clásicas tales como: que la aplicación no pueda ser atacada a través de inyecciones SQL mediante la utilización de

métodos como: `execute`, `extra ()`, entre otros o la utilización del método `POST` para evitar falsificaciones de solicitudes en el sitio. Además, presenta un panel de administración fácil e intuitivo (Camino, 2018).

### **1.3.5 Lenguaje de Programación**

#### **Python 3.10.5**

Lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado intérprete, por lo que no es necesario “traducirlo” a lenguaje máquina. Python es un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano.

Además, se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar software sin límites. Con el paso del tiempo, Python ha ido ganando adeptos gracias a su sencillez y a sus amplias posibilidades, sobre todo en los últimos años, ya que facilita trabajar con inteligencia artificial, big data, machine learning y data science, entre muchos otros campos en auge (Santander, 2021).

#### **HTML 5**

Es la quinta revisión del estándar HTML, con esta versión es posible organizar y estructurar el contenido de la web para que sea fácilmente localizable por los buscadores. Permite organizar las imágenes, texto y todo lo que desees poner en la pantalla de forma que sea fácil de entender. Puedes ajustar el ancho y alto del contenido según la pantalla. Permite la personalización y se pueden hacer ajustes y modificaciones simplemente cambiando el código sin necesidad de permisos especiales o plug-ins. Se puede acceder desde diversos buscadores, plataformas y sistemas operativos al curso.

Además, HTML5 es el estándar utilizado para crear cursos interactivos, atractivos, multiplataforma y multidispositivo (E-learning Masters, 2018). En la presente investigación se utilizará para estructurar y presentar contenido en el sistema web que se propone desarrollar.

### 1.3.6 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o IDE (Integrated Development Environment) es un espacio de trabajo virtual que se utiliza para el desarrollo y programación de aplicaciones de software. Gracias a las herramientas y mecanismos que aporta un IDE la tarea de programar es mucho más sencilla, ahorrando tiempo y consiguiendo que la productividad y eficiencia de los programadores y desarrolladores sea mucho más alta (Hosting Plus, 2021).

Se decide utilizar PyCharm 2021.1.1, puesto que es un entorno de desarrollo que ayuda a los desarrolladores de Python a ser más productivos, tener más confianza y escribir mejores códigos. Es un IDE compatible con el flujo de trabajo completo de Python desde el primer momento, incluyendo en los marcos web, las tecnologías frontend, las bases de datos y las herramientas científicas. PyCharm proporciona una finalización del código inteligente, inspecciones del código, indicación de errores sobre la marcha y arreglos rápidos, así como refactorización de código automático y completas funcionalidades de navegación.

Además, la enorme colección de herramientas pre configuradas de PyCharm incluye un depurador y un ejecutor de pruebas integrados, perfilador Python, un terminal integrado, integración con los principales VCS y herramientas de base de datos integradas, capacidades de desarrollo remoto con intérpretes remotos, un terminal ssh integrado e integración con Docker y Vagrant.

También ofrece soporte de primer nivel para varios marcos de trabajo de desarrollo web, Python, lenguajes de plantilla específicos, JavaScript, CoffeeScript, TypeScript, HTML/CSS, AngularJS, Node.js y más. Puede ser utilizado en varias plataformas como Windows, macOS y Linux con una única clave de licencia. Precisamente buena parte del potencial de esta herramienta llega de la mano de las extensiones. PyCharm tiene un buscador de extensiones que ofrece en los mismos resultados la posibilidad de instalarlas sin necesidad de salir del programa (JetBrains, 2022).

### 1.3.7 Gestor de Base de Datos.

Data Base Management System o Sistema de Gestión de Base de Datos (SGBD), es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (PowerData, 2019).

#### **SQLite 3.36.0:**

Es un sistema de gestión de bases de datos relacionales más conocidas. Básicamente, funciona como un servidor propio e independiente, ya que el Sistema de Gerencia de Base de Datos o SGBD, se puede ejecutar en la misma instancia, eliminando así las consultas y procesos separados («Introducción al sistema gestor de base de datos (SGBD)» 2020). Por lo tanto, la biblioteca SQLite se genera y almacena directamente en el archivo de la base de datos. Además, es la opción ideal para los dispositivos integrados destinados al Internet de las Cosas y también de uso común en smartphones y consolas de videojuegos, ya que es un sistema multiplataforma y funciona en varios sistemas operativos.

Para el desarrollo del sistema se decide utilizar SQLite debido a que los sistemas que lo utilizan se benefician de una mayor resistencia y tiempos de desarrollo más cortos. SQLite es atómico y transaccional, por lo que puede evitar eventos de éxito parcial, si una operación en una transacción falla, las exitosas también se revierten, devolviendo la base de datos a su estado original.

Se centra en proporcionar una potente base de datos, sin gastos generales ni dependencias. Dado que sus bases de datos, son archivos simples, son muy portátiles y fáciles de respaldar. La ausencia de un componente de servidor también hace que SQLite sea mucho más fácil de configurar, incluso si no se encuentra en un entorno de desarrollo completo. No hay un proceso largo para monitorear, reiniciar o inspeccionar por problemas de seguridad.

Además, es resistente a fallas de almacenamiento y escenarios de falta de memoria. Las bases de datos se pueden recuperar de fallas del sistema y cortes de energía, lo que ayuda a mantener los datos seguros (Diario Informe, 2021).

### **1.3.8 Herramientas para validar el sistema**

Las herramientas de gestión de pruebas ayudan a tener todo organizado, almacenan los casos y resultados de las pruebas, gestionan el flujo de trabajo de los defectos y proporcionan informes útiles para analizar las tendencias y el progreso (In learning, 2017).

#### **JMeter 5.5**

JMeter de Apache es una herramienta de prueba de código abierto diseñada para cargar, probar el comportamiento y medir el rendimiento de aplicaciones web. Es un IDE de prueba con todas las funciones que permite un plan de prueba rápido grabación (desde navegadores o aplicaciones nativas), compilación y depuración. Presenta la Capacidad para cargar y probar el rendimiento de muchas aplicaciones/servidores/tipos de protocolo diferentes: web: http y https y servicios web soap/rest.

Una persona con pocos conocimientos técnicos también puede interpretar los resultados proporcionados por JMeter. Los planes de prueba se almacenan en formato XML y se generan mediante un editor de texto. JMeter también se utiliza para pruebas funcionales y automatizadas (Apache JMeter, 2022).

#### **Acunetix Web Vulnerability Scanner 15.0**

Acunetix realiza automáticamente auditorías a aplicaciones web comprobando vulnerabilidades de Inyección SQL, Cross site scripting y otras vulnerabilidades que puedan ser explotadas por hackers. Ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor (Acunetix, 2013).

Se decide utilizar Acunetix Web Vulnerability Scanner, ya que es la herramienta más confiable del mercado para asegurar su página web. Es un scanner que trabaja en cualquier clase de

código, incluyendo HTML5, JavaScript y Python. Permite ubicar y arreglar la vulnerabilidad rápidamente gracias a la habilidad de entregar más información sobre la vulnerabilidad, tal como el número de la línea de código fuente, query SQL afectado, entre otros. Reducir significativamente los falsos positivos al escanear un sitio web; de esta forma, entender mejor el comportamiento interno de la aplicación web (MultySystem, 2014).

Además, se utiliza JMeter que es una herramienta ejemplar para realizar pruebas de rendimiento en aplicación web. Su utilización es bastante sencilla y asequible, presenta gran cantidad de documentación y tutoriales disponibles. Es una herramienta que sirve para realizar pruebas funcionales, pero también sirve para realizar pruebas de regresión en aplicaciones web. Es gratuita y permite la ejecución de pruebas distribuidas entre distintos ordenadores, que actuaran como clientes (Izertis, 2016).

### 1.3.9 Servidor Web.

Un servidor web es un programa informático que procesa aplicaciones realizando conexiones bidireccionales y/o unidireccionales, síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. Es quien gestiona la mayor parte de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores web son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones (Library, 2022).

**Apache 2.4.53** es un servidor web que implementa el protocolo HTTP, es software libre y se puede instalar en los sistemas operativos Windows y Linux. Es de fácil configuración, pues su estructuración en módulos permite al usuario utilizar los servicios y funcionalidades que ofrece. Se define Apache como software para el servidor de aplicaciones en su versión 2.4.53, por su flexibilidad, rapidez, porque es gratuito, multiplataforma e interpreta varios lenguajes como Python y HTML.

Estas características promueven la eficiencia y rendimiento del sistema que debe ser capaz de dar respuestas a las peticiones con un nivel aceptable de desempeño, teniendo en cuenta la concurrencia que pueda existir; debe prestar servicios sin que se amplíen los rangos de tiempo de repuesta (Apache, 2018).

## Conclusiones del capítulo

En el capítulo se expusieron los fundamentos teóricos asociados al dominio del problema. Se realizó un análisis de los principales sistemas para el proceso de gestión de entrega y recepción de libros, los cuales permitieron afirmar que, no pueden ser utilizados para satisfacer las necesidades actuales del Vicedecanato de Economía y Administración de la Facultad 1 de la UCI, por lo que se propuso desarrollar un sistema web que facilite el proceso de gestión de entrega y recepción de libros.

Para guiar el desarrollo se propuso la metodología AUP\_UCI en el escenario 4, utilizando como herramienta para el modelado Visual Paradigm 16.3, como framework de desarrollo Django 4.1, lenguajes de programación Python 3.10.5 y HTML 5, IDE PyCharm 2021.1.1, gestor de datos SQLite 3.36, herramientas de pruebas Acunetix 15.0 y JMeter 5.5 y como servidor web Apache 2.4.53. Quedando así definido el ambiente tecnológico para dar paso al desarrollo de la solución.

## CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

### Introducción al capítulo

Para desarrollar un sistema informático es importante la utilización de métodos y técnicas que permitan la solución de problemas existentes a lo largo del ciclo de vida de un software. Dentro de este capítulo se obtienen un conjunto de artefactos generados a partir de la utilización de la metodología AUP\_UCI; incluyendo el ajuste de los planes del proyecto, considerando los requisitos y la arquitectura.

### 2.1. Modelo Conceptual:

Un modelo conceptual es un artefacto de la disciplina de análisis, construido con las reglas UML. Tiene como objetivo comprender y describir las clases más importantes, así como, identificar y explicar los conceptos significativos en el dominio del problema, identificando los atributos y las asociaciones existentes entre ellos (Sommerville, 2011).

A continuación, en la figura 1 se representa el modelo conceptual relacionado a los procesos que se llevan a cabo en el Vicedecanato de Economía y Administración de la Facultad 1.

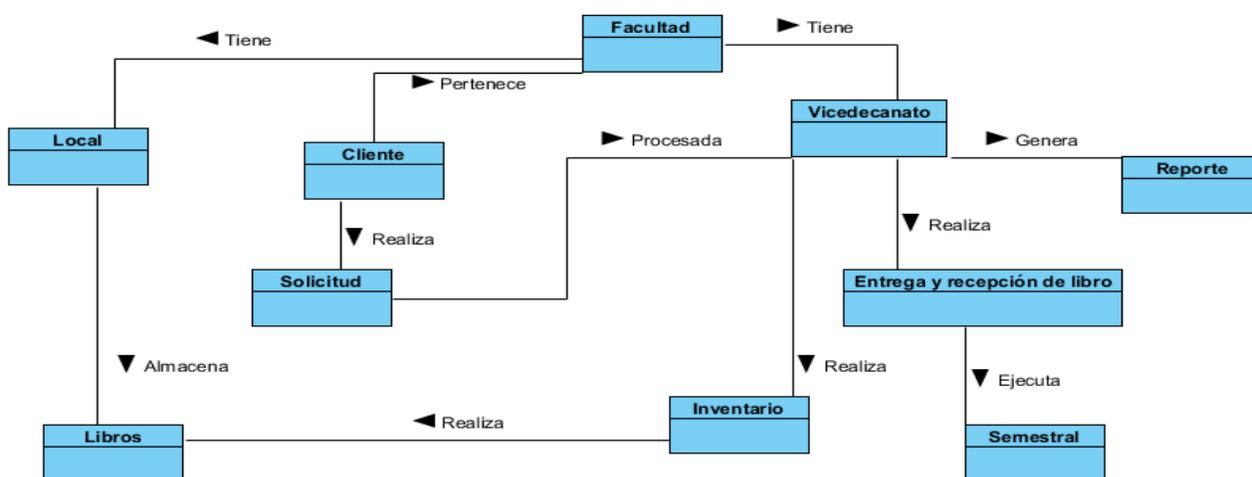


Figura #1 Modelo conceptual (Elaboración propia)

Los conceptos representados se describen a continuación:

**Cliente:** estudiante o profesor que realiza las solicitudes de libros.

**Local:** lugar de almacenamiento de libros.

**Vicedecanato:** encargado de realizar los inventarios, procesar solicitudes, generar reportes y entregar y recepcionar libros.

**Facultad:** lugar al que pertenece el cliente y que tiene un local y un vicedecanato.

**Entrega y recepción de libros:** realizado por el Vicedecanato y forma parte principal del negocio.

**Semestral:** frecuencia con que se realizan las entregas y recepciones de libros.

**Inventario:** es para controlar los libros en almacén.

**Reportes:** es para brindar información sobre los libros en el almacén.

**Libro:** material básico y forma parte principal del negocio.

## **2.2 Descripción de la Propuesta de solución:**

La propuesta de solución se basa en el desarrollo del Sistema de Gestión de control de libros que será utilizado en la Facultad 1 de la UCI, encargado de gestionar los procesos de gestión de entrega, recepción y reporte de libros que posee dicha facultad. Además, permite la gestión de usuarios, donde el acceso a la información se realiza basado en roles.

El sistema gestiona como propiedades principales del libro su título, número de folio, estado, asignatura y curso al que pertenece. Los reportes generados en formato pdf son fundamentales para apoyar la gestión del Vicedecanato.

## **2.3 Requisitos de la Propuesta de Solución:**

Los requisitos son la base para un desarrollo exitoso, así como para una plena conformidad con el entregable final. La especificación de los requisitos de software establece una base sólida para su diseño e implementación. Estos se definen las necesidades, describen los es-

cenarios de uso, delinear las funciones y características e identifican las restricciones del proyecto (Pressman, 2010). Para llevar a cabo el proceso de captura de requisitos se aplicó la técnica que a continuación se expone:

- ✧ Entrevista: se realizaron entrevistas, sustentado en una serie de preguntas, con el objetivo de obtener la mayor cantidad posible de información y comprender los objetivos generales de la solución.

### 2.3.1 Requisitos Funcionales:

Los requisitos funcionales (RF) son aquellos directamente relacionados con las funciones y las reacciones que el sistema debe proporcionar. Son asignados directamente a elementos o características del sistema de software (Sommerville, 2011).

A continuación, en la tabla 2 se muestran los requisitos funcionales:

*Tabla 2. Requisitos Funcionales*

No.	Nombre	Descripción	Prioridad
RF1	Autenticar usuario	Permite a los usuarios acceder al sistema mediante usuario y contraseña.	Alta
RF2	Cerrar sesión	Permite al usuario cerrar su sesión para salir del sistema.	Baja
RF3	Crear usuario	Los administradores podrán crear usuarios dentro del sistema.	Alta
RF4	Eliminar usuario	Permite al administrador eliminar un usuario registrado en el sistema.	Media
RF5	Modificar usuario	Permite al administrador modificar un usuario registrado en el sistema.	Media

RF6	Listar usuario	El sistema permite hacer un listado de todos los usuarios registrados.	Media
RF7	Solicitar libro	Permite al usuario solicitar un libro en el sistema.	Alta
RF8	Eliminar solicitud	Permite al administrador eliminar una solicitud del sistema.	Alta
RF9	Listar solicitudes	Lista las solicitudes realizadas.	Alta
RF10	Adicionar libro	Permite adicionar un libro a la base de datos.	Alta
RF11	Eliminar libro	Permite eliminar un libro registrado en la base de datos.	Media
RF12	Modificar libro	Permite modificar un libro registrado en la base de datos.	Alta
RF13	Listar libros	Permite listar los libros registrados en la base de datos.	Alta
RF14	Crear rol	El sistema permite al administrador crear roles.	Media
RF15	Modificar rol	Permite al administrador modificar un rol registrado antes en el sistema.	Media
RF16	Listar rol	Lista todos los roles creados en el sistema.	Media
RF17	Mostrar rol	Permite al administrador ver los detalles de un rol.	Media
RF18	Asignar rol	Permite asignar un rol a un usuario registrado en el sistema.	Baja
RF19	Quitar rol	Permite quitar un rol a un usuario registrado en el sistema.	Baja

RF20	Generar reporte estadístico de la cantidad de libro en el local	Permite generar un reporte de la cantidad de libros que se encuentran en el local.	Media
RF21	Generar reporte estadístico de los préstamos de libros	Permite generar un reporte de los libros que están prestados.	Media
RF22	Generar un modelo de préstamo de libros	Permite generar un modelo de préstamos de libros por cada usuario.	Media

### 2.3.2 Requisitos no funcionales:

Los requisitos no funcionales (RnF) son restricciones de las funciones ofrecidas por el sistema. Generalmente, se aplican al sistema en su totalidad y surgen de las necesidades del usuario. Pueden especificar también la utilización de una herramienta en particular o un lenguaje de programación (Ramos, 2015).

Para la solución propuesta se identificaron 13 RnF los cuales fueron agrupados por las diferentes categorías:

#### Seguridad:

RnF 1: El usuario debe estar autenticado antes de realizar alguna acción en el sistema.

RnF 2: Se necesita la presencia de roles que controlen las acciones que pueden realizar los usuarios.

RnF 3: El sistema muestra las funcionalidades de acuerdo a los permisos del usuario que esté activo.

RnF 4: El administrador es el que asigna los permisos de acceso al sistema.

#### Software:

Utilizar para pc\_cliente:

RnF 5: Se necesita instalado en la estación de trabajo un sistema operativo: Linux, Windows 7 o superior.

Utilizar para pc\_servidor:

RnF 6: Se requiere de la instalación del servidor web Apache en su versión 2.4.53.

RnF 7: Se requiere de un gestor de base de datos SQLite en su versión 3.36 o superior.

Hardware:

Utilizar para pc\_cliente:

RnF 8: Requerimientos mínimos: Procesador Intel(R) Core (TM) i3-4160, CPU 3.60 GHz con 2 MB de memoria RAM.

Utilizar para pc\_servidor:

RnF 9: Requerimientos mínimos: Procesador Intel(R) Core (TM) i3-4160, CPU 3.60 GHz de velocidad de procesamiento, 4 GB de memoria RAM y una capacidad de 80 GB de disco duro.

Usabilidad:

RnF 10: El sistema de gestión de control de libros debe ser intuitivo, facilitando la interacción del usuario con la aplicación.

RnF 11: Los dispositivos cliente que utilizarán la herramienta deben contar con navegadores web que soporten HTML5 y Python 3.10.5. (Chrome, Mozilla Firefox, Opera)

Eficiencia.

RnF 12: Toda funcionalidad del sistema de gestión tecnológica debe responder al usuario en menos de 5 segundos.

RnF 13: La información almacenada debe ser confiable en cuanto a su veracidad e integridad desde su recopilación a través de la validación de los formularios.

## 2.4 Historia de Usuario

Las historias de usuario son una herramienta que agiliza la administración de requisitos, reduciendo la cantidad de documentos formales y tiempo necesarios. Se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal (Menzinsky, 2020).

A continuación, se muestra en las siguientes tablas (tablas 3, 4 y 5) varias historias de usuarios, las cuales fueron generadas a partir de los requisitos funcionales y las mismas se estructuran de la siguiente forma:

**Número:** A cada HU se le asigna un número para facilitar su identificación por parte del equipo de desarrollo.

**Nombre:** Nombre descriptivo de la HU.

**Prioridad:** Grado de prioridad que le asigna el cliente a la HU en dependencia del valor en el negocio. Los valores que puede tomar son (Alta, Media o Baja).

**Iteración Asignada:** Número de la iteración en la cual será implementada la HU.

**Programador:** Nombre del programador encargado de desarrollar la HU.

**Tiempo estimado:** Esfuerzo estimado por el equipo de desarrollo para darle cumplimiento a la HU.

**Tiempo real:** Plazo real que el equipo de desarrollo tiene para dar cumplimiento a la HU.

**Descripción:** Descripción simple sobre lo que debe hacer la funcionalidad a la que se hace referencia.

**Riesgo en Desarrollo:** Grado de complejidad que le asigna el equipo de desarrollo a la HU luego de analizarla. (Alto, Medio o Bajo).

Tabla 3. HU\_7 Solicitar Libro

Historia de Usuario	
<b>Número: 7</b>	<b>Usuario:</b> Cliente
<b>Nombre de historia:</b> Solicitar libros	
<b>Prioridad en negocio (Alta, Medio, Baja):</b> Alta	<b>Riesgo en desarrollo (Alta, Medio, Bajo):</b> Medio
<b>Tiempo estimado: 4</b>	<b>Iteración asignada: 1</b>
<b>Programador Responsable:</b> Rachel Marichal Casaña	
<b>Descripción:</b> Permite al usuario solicitar un libro en el sistema. Para lo cual son necesario los datos que se introducen en el formulario: nombre y apellido, usuario, categoría, asignatura y el libro. Después de rellenar los campos del formulario brinda la opción del botón enviar que permite realizar la solicitud.	
<b>Validación:</b> El usuario puede solicitar un libro con satisfacción	
<b>Interfaz:</b> <div style="text-align: center; margin-top: 20px;"> <p>Nombre y apellidos:</p> <input type="text"/>  <p>Usuario:</p> <input type="text"/>  <p>Seleccione su categoria</p> <p><input type="radio"/> Estudiante <input type="radio"/> Profesor</p> <p>Asignatura</p> <input type="text"/>  <p>Libro</p> <input type="text"/>  <p><input type="button" value="Enviar"/></p> </div>	

Tabla 4. HU\_10 Adicionar Libro

Historia de Usuario	
<b>Número: 10</b>	<b>Usuario:</b> Cliente
<b>Nombre de historia:</b> Adicionar Libro	
<b>Prioridad en negocio (Alta, Medio, Baja):</b> Alta	<b>Riesgo en desarrollo (Alta, Medio,Bajo):</b> Alto
<b>Tiempo estimado:</b> 6	<b>Iteración asignada:</b> 1
<b>Programador Responsable:</b> Rachel Marichal Casaña	
<p><b>Descripción:</b> Permite adicionar un libro a la base de datos. Para lo cual son necesario los datos que se insertan en el formulario: nombre del libro, autor, asignatura, tomo, folio, parte, volumen y cantidad disponible. Después de rellenar los campos del formulario brinda la opción del botón aceptar que permite adicionar el libro y ofrece la opción de cancelar en caso de no desear realizar la acción.</p>	
<b>Validación:</b> Adiciona un libro a la base de datos del sistema	
<p><b>Interfaz:</b></p> 	

Tabla 5. HU\_9 Listar Solicitudes

Historia de Usuario																												
<b>Número:</b> 19	<b>Usuario:</b> Cliente																											
<b>Nombre de historia:</b> Listar solicitudes																												
<b>Prioridad en negocio(Alta, Medio, Bajo):</b> Alta	<b>Riesgo en desarrollo(Alta, Medio, Bajo):</b> Alta																											
<b>Tiempo estimado:</b> 3	<b>Iteración asignada:</b> 2																											
<b>Programador Responsable:</b> Rachel Marichal Casaña																												
<b>Descripción:</b> Lista la información referente a las solicitudes realizadas por los usuarios recogida en los campos: nombre, usuario, libro, fecha de entrega, fecha de devolución, categoría y asignatura; además de encontrarse la opción de eliminar en caso de que se desea realizar la acción.																												
<b>Validación:</b> Muestra un listado con todas las solicitudes realizadas																												
<b>Interfaz:</b>																												
Show <input type="text" value="10"/> entries <span style="float: right;">Search: <input type="text"/></span>																												
<table border="1"> <thead> <tr> <th>Id</th> <th>Nombre</th> <th>Usuario</th> <th>Libro</th> <th>FechaE</th> <th>FechaD</th> <th>Categoría</th> <th>Asignatura</th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Rachel Marichal Casaña</td> <td><a href="#">rachelmc</a></td> <td>1</td> <td></td> <td></td> <td>estudiante</td> <td>Matematica I</td> <td></td> </tr> <tr> <td>2</td> <td>Lester Díaz González</td> <td><a href="#">ldiaz</a></td> <td>1</td> <td></td> <td></td> <td>profesor</td> <td>Matematica I</td> <td></td> </tr> </tbody> </table>		Id	Nombre	Usuario	Libro	FechaE	FechaD	Categoría	Asignatura		1	Rachel Marichal Casaña	<a href="#">rachelmc</a>	1			estudiante	Matematica I		2	Lester Díaz González	<a href="#">ldiaz</a>	1			profesor	Matematica I	
Id	Nombre	Usuario	Libro	FechaE	FechaD	Categoría	Asignatura																					
1	Rachel Marichal Casaña	<a href="#">rachelmc</a>	1			estudiante	Matematica I																					
2	Lester Díaz González	<a href="#">ldiaz</a>	1			profesor	Matematica I																					
Showing 1 to 2 of 2 entries <span style="float: right;">Previous <input type="text" value="1"/> Next</span>																												

## 2.5 Arquitectura de Software

La Arquitectura de Software se refiere a la estructura de un sistema, compuesta de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. La misma es de especial importancia, ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema.

Ejemplos de atributos de calidad son el desempeño, que tiene que ver con el tiempo de respuesta del sistema a las peticiones que se realizan, la usabilidad, que tiene que ver con qué tan sencillo les resulta a los usuarios realizar operaciones con el sistema, o bien la edificabili-

dad, que tiene que ver con qué tan simple resulta introducir cambios en el sistema. Los atributos de calidad son parte de los requerimientos (no funcionales) del sistema y son características que deben expresarse de forma cuantitativa (Cervantes, 2018).

A continuación, en la figura 2 se representa la arquitectura de software utilizada en la propuesta de solución:

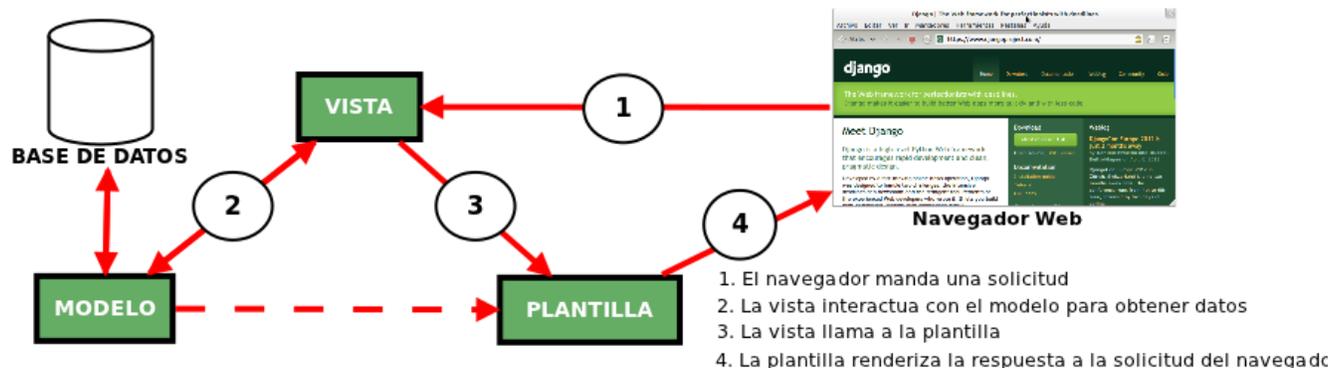


Figura #2 Arquitectura de Software

Django basa su trabajo en la arquitectura Modelo-Vista-Plantilla, como se muestra en la figura 2, implementado de forma tal que el desarrollo sea rápido y sencillo, pues, en la capa plantilla que es donde se ocupan de las decisiones relacionadas a la presentación: cómo son mostradas algunas cosas en la página web u otro tipo de documento. El modelo es la responsable del procesamiento de datos, en éste se manejan todas las transacciones relacionadas con los datos: cómo acceder, cómo verificar la validez, el comportamiento que tienen y las relaciones entre los datos. La Vista la cual contiene la lógica para acceder al modelo y llamar a la plantilla adecuada, funcionan así como un puente entre el modelo y la plantilla (Arango, Arango y Pérez 2022).

En el sistema de propuesta de solución, la capa plantilla se ve evidenciada en los archivos `admin_list.html`, `user_list.html`, `login.html`, entre otros. El modelo que no es más que la capa de acceso a datos se ve evidenciado en el archivo `models.py`. Además, la capa vista que es la que accede al modelo y delega a la plantilla apropiada, evidenciada en el archivo `views.py`.

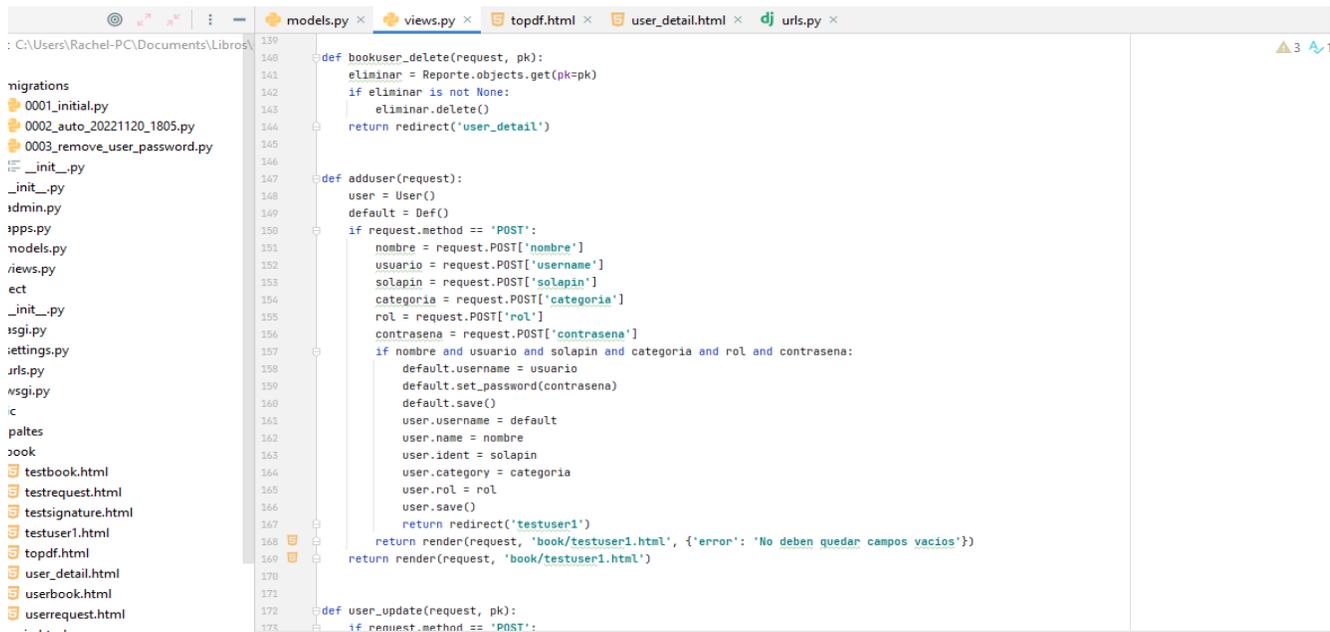
## 2.6 Patrones de Diseño

Según el análisis realizado por (Castañeda, 2016) un patrón de diseño describe problemas comunes de software y detalla la solución de este, la cual debe ser reutilizable. Son descripciones de comunicación entre objetos y clases que pueden adaptarse para resolver un problema de diseño general en un contexto particular. Durante la implementación de la propuesta de solución se utilizaron los siguientes patrones de diseño:

### 2.6.1 Descripción de los patrones para asignar responsabilidades “Patrones GRASP”:

Los patrones GRASP se utilizan para la asignación de responsabilidades como sus siglas en inglés lo indican (General Responsibility Assignment Software Patterns), pero más que patrones se consideran una serie de buenas prácticas de aplicación recomendable en el diseño de software (Larman, 2003).

**Controlador:** es el patrón encargado de las conexiones entre la interfaz y el algoritmo que la implementa, también es el que controla el flujo de datos de las funcionalidades que se le asignan y los eventos relacionados con estas. Este patrón se pone de manifiesto en la clase `views.py`, que se refleja en la figura 3.



```

139
140 def bookuser_delete(request, pk):
141     eliminar = Reporte.objects.get(pk=pk)
142     if eliminar is not None:
143         eliminar.delete()
144     return redirect('user_detail')
145
146
147 def adduser(request):
148     user = User()
149     default = Def()
150     if request.method == 'POST':
151         nombre = request.POST['nombre']
152         usuario = request.POST['username']
153         solapin = request.POST['solapin']
154         categoria = request.POST['categoria']
155         rol = request.POST['rol']
156         contrasena = request.POST['contrasena']
157         if nombre and usuario and solapin and categoria and rol and contrasena:
158             default.username = usuario
159             default.set_password(contrasena)
160             default.save()
161             user.username = default
162             user.name = nombre
163             user.ident = solapin
164             user.category = categoria
165             user.rol = rol
166             user.save()
167             return redirect('testuser1')
168         return render(request, 'book/testuser1.html', {'error': 'No deben quedar campos vacios'})
169     return render(request, 'book/testuser1.html')
170
171
172 def user_update(request, pk):
173     if request.method == 'POST':

```

Figura # 3 Ejemplo del patrón controlador

**Alta Cohesión:** si sus métodos están relacionados entre sí, tienen un contenido claro y temática común, trabajan con tipos similares, entre otros. Todo bien encerrado dentro de la clase, y perfectamente delimitado (Pascual, 2019). Este patrón se pone de manifiesto en la clase `models.py`, como se muestra en la figura 4.

```

14
15
16 class Asignatura(models.Model):
17     name = models.CharField(max_length=250, verbose_name='Nombre')
18     year = models.CharField(max_length=250, verbose_name='Año')
19     semestre = models.CharField(max_length=250, verbose_name='Semestre')
20
21     def __str__(self):
22         return self.name
23
24
25 class Libro(models.Model):
26     user = models.ForeignKey(User, null=True, blank=True, on_delete=models.CASCADE)
27     signature = models.ForeignKey(Asignatura, null=True, blank=True, on_delete=models.CASCADE)
28     name = models.CharField(max_length=250, verbose_name='Titulo')
29     author = models.CharField(max_length=250)
30     tomo = models.IntegerField()
31     folio = models.IntegerField()
32     parte = models.IntegerField()
33     volumen = models.IntegerField()
34     cantidad = models.IntegerField()

```

*Figura # 4 Ejemplo del patrón alta cohesión*

**Bajo Acoplamiento:** es el patrón encargado de mantener las clases lo menos relacionadas posible; de tal forma que, a la hora de realizar modificaciones en alguna de ellas, se tenga la mínima dependencia entre las clases, potenciando la reutilización. Este patrón se evidencia en todas las aplicaciones web que funcionen sobre Django, pues cada pieza de las aplicaciones tiene un propósito clave, que puede modificarse sin afectar otras piezas.

Por ejemplo, se puede cambiar la URL de cierta parte de la aplicación sin tener que afectar la implementación subyacente o se puede modificar el HTML de una página sin tener que tocar el código Python que la renderiza. Este patrón se pone de manifiesto en la clase `views.py`, como se muestra en la figura 5.

```

264     return redirect('testbook')
265     return redirect('testbook')
266
267
268 def book_delete(request, pk):
269     eliminar = Libro.objects.get(pk=pk)
270     if eliminar is not None:
271         eliminar.delete()
272     return redirect('testbook')
273
274
275 def addsignatura(request):
276     asignatura = Asignatura()
277     if request.method == 'POST':
278         name = request.POST['name']
279         year = request.POST['year']
280         semester = request.POST['semester']
281         if name and year and semester:
282             asignatura.name = name
283             asignatura.year = year
284             asignatura.semestre = semester
285             asignatura.save()
286             return redirect('testsignature')
287         return render(request, 'book/testbook.html', {'error': 'No deben quedar campos vacios'})
288     return render(request, 'book/testbook.html')
289

```

Figura # 5 Ejemplo del patrón bajo acoplamiento

## 2.7 Diagrama de clases de diseño:

Un Diagrama de Clase del Diseño (DCD) muestra la especificación de las clases de una aplicación, sus asociaciones, atributos y métodos, interfaces, navegabilidad y dependencias. El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos. Con el fin de facilitar la comprensión del diagrama, se pueden incluir paquetes como elementos del mismo, donde cada uno de ellos agrupa un conjunto de clases (Pressman, 2010).

A continuación, en las figuras 6, figura 7 y figura 8, se exponen diagramas de clase del diseño relacionados con las funcionalidades: solicitar libro, adicionar un libro y listar solicitudes.

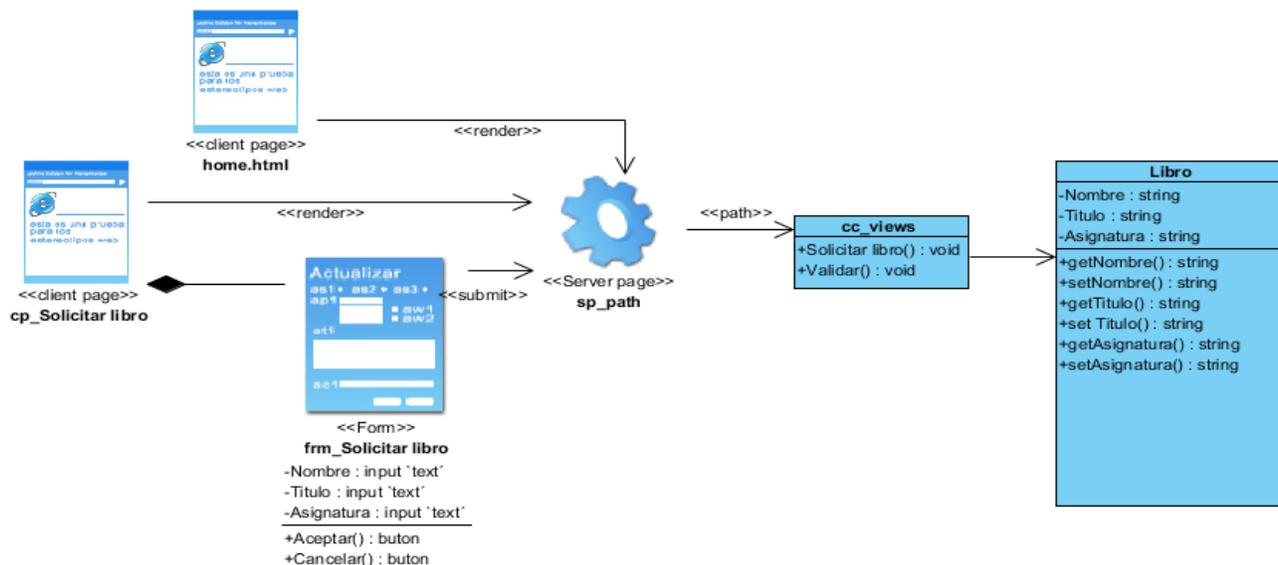


Figura # 6 DCD con estereotipos web Solicitar Libro (elaboración propia)

El DCD está compuesto por un server\_page que se corresponde con la clase principal del contenido de solicitudes, un client\_page que muestran la funcionalidad de solicitar libros, el formulario que contiene los campos, botones y filtros asociados a los contenidos, la clase controladora que contiene la funcionalidad del server\_page, la clase que permite la conexión con la base de datos y la clase de la base de datos con los campos y funciones asociados al contenido planificación.

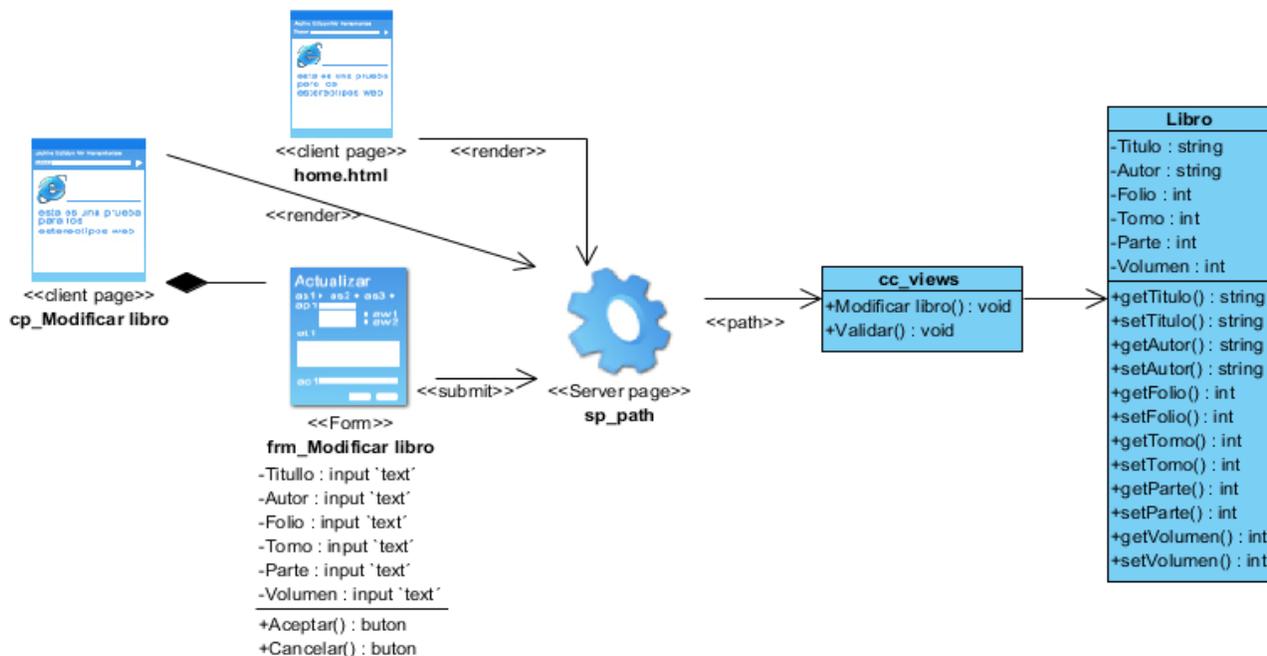


Figura # 7 DCD con estereotipos web Modificar Libro (elaboración propia)

El DCD está compuesto por un server\_page que se corresponde con la clase principal del contenido libro, un client\_page que muestran la funcionalidad modificar, un formulario que contiene los campos, botones y filtros asociados a los contenidos, la clase controladora que contiene la funcionalidad del server\_page, la clase que permite la conexión con la base de datos y la clase de la base de datos con los campos y funciones asociados al contenido planificación.

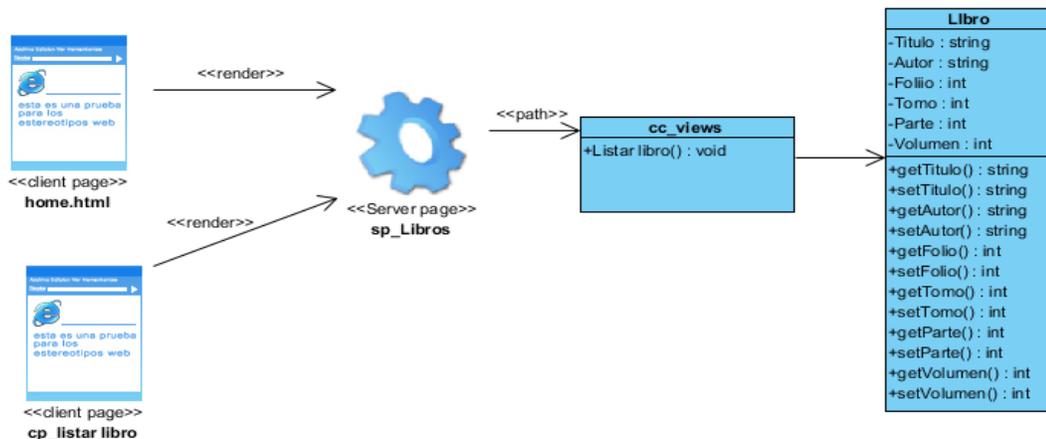


Figura # 8 DCD con estereotipos web Listar Libros (elaboración propia)

El DCD está compuesto por un server\_page que se corresponde con la clase principal del contenido libro, un client\_page que muestran la funcionalidad modificar, la clase controladora que contiene la funcionalidad del server\_page, la clase que permite la conexión con la base de datos y la clase de la base de datos con los campos y funciones asociados al contenido planificación.

## 2.8 Diagrama de Secuencias:

Los diagramas de secuencia (DS) en el UML se usan principalmente para modelar las interacciones entre los actores y los objetos en un sistema, así como las interacciones entre los objetos en sí (Sommerville, 2011). Para la presente investigación se generaron diagramas de secuencia relacionados con las funcionalidades: solicitar libro, adicionar libro y listar solicitudes, reflejados en las figuras 10, figura 11 y figura 12.

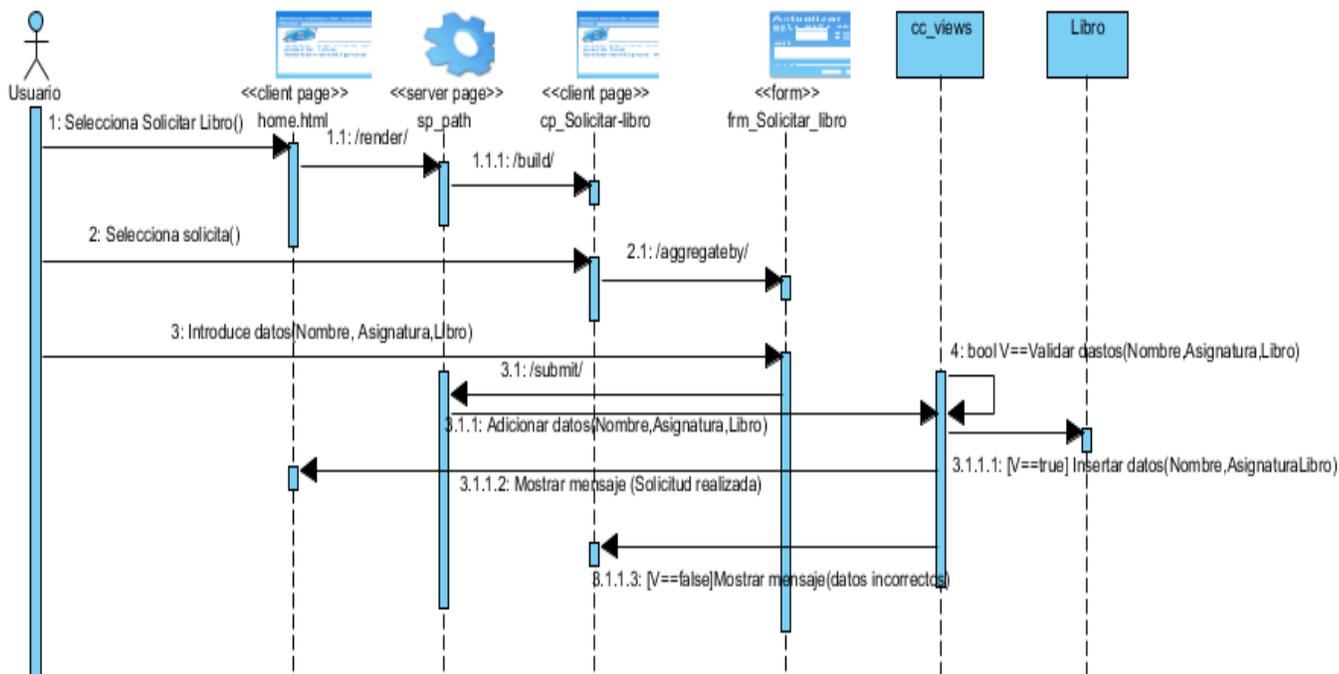


Figura # 10 DS con estereotipos web Solicitar

En caso de que los datos no hayan sido insertados de forma correcta se muestra un mensaje de error en la página cliente desde donde se creó la solicitud, en el caso contrario se muestra un mensaje de confirmación en la clase client\_page desde donde el actor realizó la solicitud inicial.

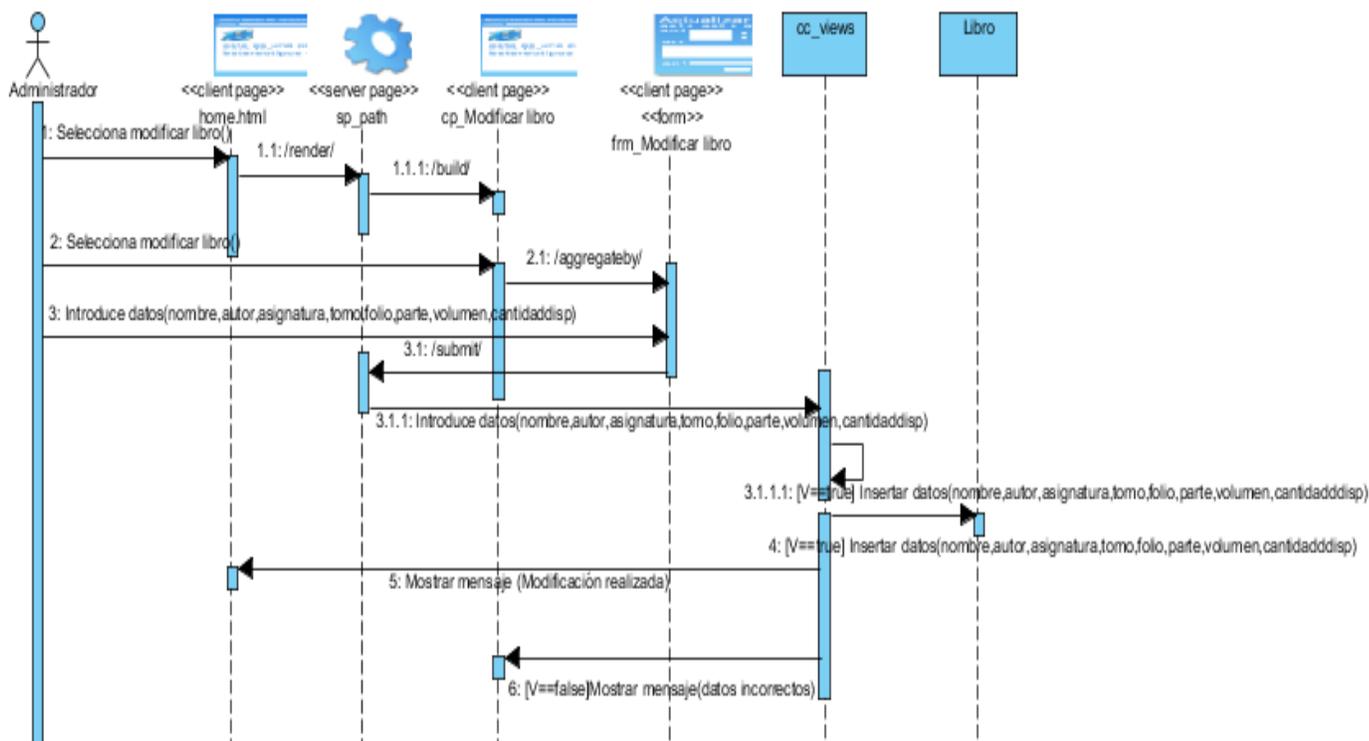


Figura # 11 DS con estereotipos web Modificar Libro (elaboración propia)

El flujo representado en el DS comienza cuando el actor (usuario) del sistema selecciona en el mismo la opción de modificar libro. La página cliente intermediaria hace la solicitud a server\_page (página controladora/servidora). La página controladora construye la client\_page Modificar libro (página cliente/vista) que permite modificar libro, esta hace link a la controladora que crea la página cliente con el formulario con los campos asociados a la modificación. El actor introduce los datos que se envían a server\_page (página controladora/servidora) y automáticamente se adiciona la solicitud en la clase controladora donde se validan que los datos estén correctos.

En caso de que los datos no hayan sido insertados de forma correcta se muestra un mensaje de error en la página cliente desde donde se creó la modificación, en el caso contrario se muestra un mensaje de confirmación en la clase desde donde el actor realizó la solicitud inicial.

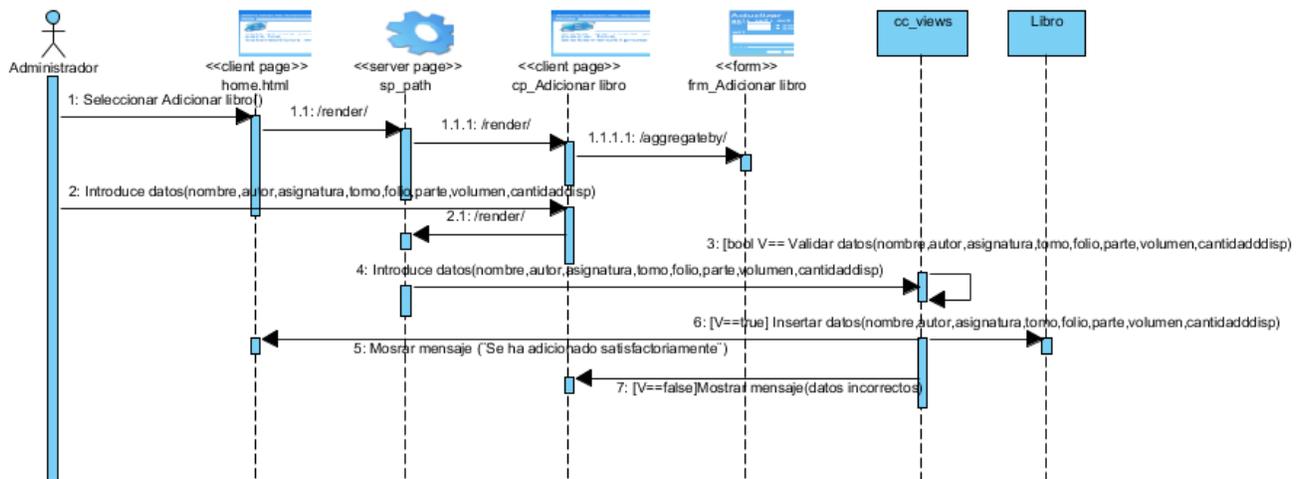


Figura # 12 DS con estereotipos web Adicionar Libro (elaboración propia)

El flujo representado en el DS comienza cuando el actor (administrador) del sistema selecciona en el mismo la opción de adicionar libro. La página cliente home.html intermediaria hace la solicitud a server\_page (página controladora/servidora). La página controladora construye la client\_page Adicionar libro (página cliente/vista) que permite adicionar libro, esta hace link a la controladora que crea la página cliente con el formulario con los campos asociados a la adición. El actor introduce los datos que se envían a server\_page (página controladora/servidora) y automáticamente se adiciona en la clase controladora donde se validan que los datos estén correctos.

En caso de que los datos no hayan sido insertados de forma correcta se muestra un mensaje de error en la página cliente Adicionar libro, en el caso contrario se muestra un mensaje de confirmación en la clase desde donde el actor realizó la solicitud inicial.

## 2.9 Modelo de Datos

Un modelo de datos es un conjunto de herramientas conceptuales encargadas de describir la representación de los datos en una base de datos. Los mismos comprenden aspectos relacionados con estructuras y tipos de datos, operaciones y restricciones (Ayudaley, 2022).

En la figura 12, se muestra el modelo de datos, el cual fue generado para la presente investigación:

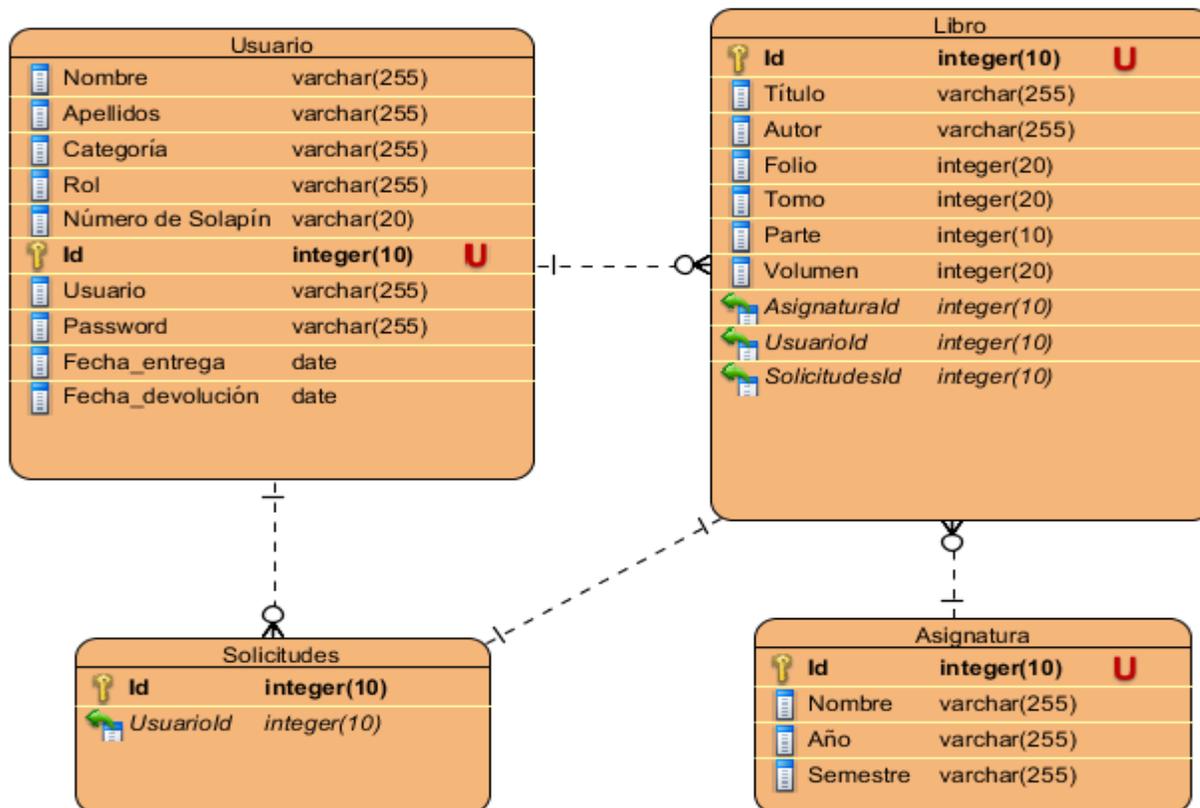


Figura # 12 Modelo de Datos (elaboración propia)

**Usuario:** Registra la información de los usuarios del sistema. Se almacena en la base de datos los atributos que lo identifican nombre, apellido, categoría, rol, número de solapín, id, usuario, password, fecha\_entrega y fecha\_devolución.

**Libros:** Registra la información de los libros en el sistema. Se almacena en la base de datos los atributos que lo identifican id, título, autor, folio, tomo, parte y volumen.

**Asignatura:** Registra la información de las asignaturas de los libros en el sistema. Se almacena en la base de datos los atributos que lo identifican id, nombre, año y semestre.

**Solicitudes:** Registra la información de las solicitudes realizadas por el usuario en el sistema.

## **Conclusiones del capítulo**

En este capítulo se evidenciaron las principales características y elementos significativos de la propuesta de solución, por lo que se puede corroborar que la delimitación de los requisitos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, garantizó que la solución responda a las necesidades del cliente. Los artefactos generados constituyeron una guía fundamental para el desarrollo del sistema de gestión de libros, que unido a la arquitectura MTV permitió precisar la estructura del software y su interrelación entre los diferentes componentes.

## CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se describe el diagrama de componentes, los estándares de codificación y el diagrama de despliegue. Además, de documentar los resultados obtenidos durante el desarrollo de la estrategia de pruebas.

### 3.1 Diagrama de Componente

El diagrama de componentes proporciona una visión física de la construcción del sistema. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos (Ramos, 2015). Basándose en la arquitectura de software que propone el marco de trabajo Django, seleccionado en la fase de análisis y diseño. A continuación, en la figura 14, se muestra el diagrama de componentes del requisito solicitar libro.

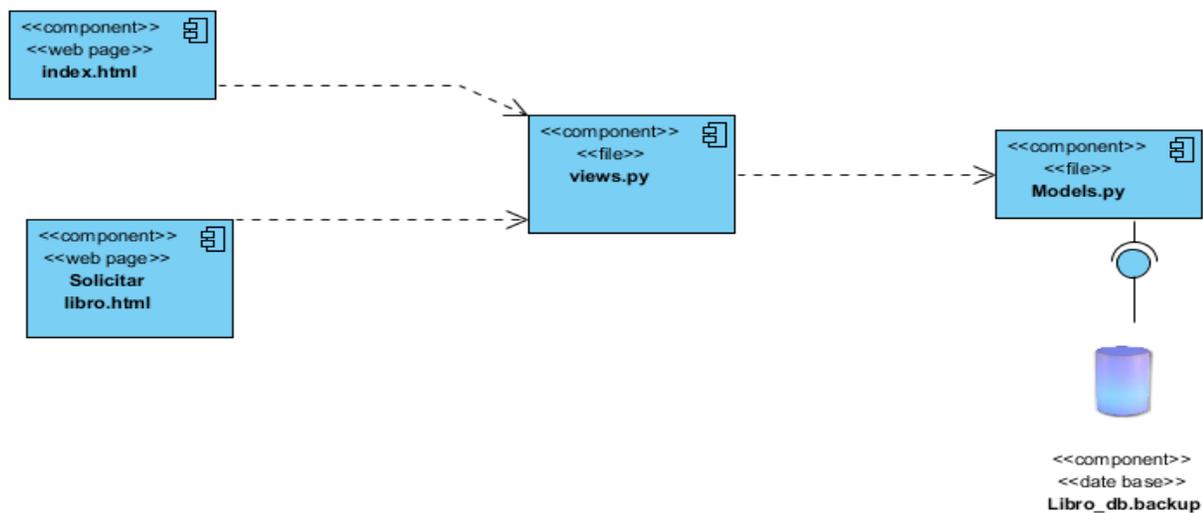


Figura # 14 DC con estereotipos web Solicitar Libro (elaboración propia)

En el diagrama de componente del sistema se representa los componentes index.html y solicitar libro.html como componentes visuales, el views.py como componente controlador y models.py y libro\_db. backup como los componentes de la base de datos del sistema.

### 3.2 Estándares de Codificación

Los estándares de codificación son un conjunto de convenciones para lograr uniformidad en el código fuente de un software. Permiten una mayor comprensión, modificación, calidad y mantenibilidad del código generado (Standards, 2018). En la siguiente tabla 7 se muestran los estándares de codificación utilizados en la propuesta de solución:

Tabla 6. Estándares de codificación

Estándar de codificación	Descripción
Organización del código	<p>La apertura de llaves para clases y funciones deben estar en la próxima línea después de su declaración. Las aperturas de llaves en las estructuras de control se colocan en la misma línea. El cierre de llaves para clases, funciones y estructuras de control deben estar en la próxima línea después del bloque de código.</p> <p>Ejemplo:</p> <pre data-bbox="456 989 1333 1087"> 194     {% for b in books %} 195         &lt;div class="modal fade" id="modal_updatebook{{ b.id }}" tabindex="-1" . 240     {% endfor %} </pre>
Líneas	<p>El tamaño máximo de una línea no excederá de ochenta (80) caracteres. Se deberá agregar una línea en blanco delante de la sentencia return. Ejemplo:</p> <pre data-bbox="456 1266 1468 1365"> 286         return redirect('testsignature') 287         return render(request, 'book/testbook.html', {'error': 'No deben quedar campos vacios'}) 288         return render(request, 'book/testbook.html') </pre>
Uso de comillas	<p>Se utilizan tanto comillas simples como la ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto (Gil, 2012). En la que se evidencia el uso de las comillas en la codificación de la propuesta de solución. Ejemplo:</p>

	<pre> 92     def solicitar(request): 93         solicitud = Solicitud() 94         if request.method == 'POST': 95             name = request.POST['name'] 96             categoria = request.POST['categoria'] 97             asignatura = request.POST['asignatura'] 98             libro = request.POST['libro'] </pre>
Línea en blanco	<p>Dos líneas en blanco entre las funciones de nivel de módulo y las definiciones de clase; línea en blanco entre las funciones de los miembros de la clase; puede utilizar varias líneas en blanco para separar varios conjuntos de funciones relacionadas. Las líneas en blanco se pueden utilizar en funciones para separar códigos relacionados lógicamente. Ejemplo:</p> <pre> 87     def sign_up_page(request): 88         users = User.objects.all() 89         return render(request, 'sign_up.html', {'users': users}) 90 91 92     def solicitar(request): 93         solicitud = Solicitud() 94         if request.method == 'POST':... 108        return render(request, 'book/userbook.html') 109 110 111     def request_delete(request, pk):... 112 </pre>
Declaración de importación	<p>La declaración de importación debe escribirse en líneas separadas y usar importación absoluta. Debe colocarse al principio del archivo, después de la descripción del módulo y la cadena de documentos, y antes de las variables globales; las declaraciones de importación deben organizarse en orden, separadas por una línea en blanco entre cada grupo. Al importar definiciones de clases de otros módulos, puede utilizar la importación relativa. Si se produce un conflicto de nombres, puede utilizar el espacio de nombres. Ejemplo:</p> <pre> 1     from django.contrib import auth 2     from django.contrib.auth.decorators import login_required 3     from django.contrib.auth.models import User as Def </pre>
Espacio	<p>Un espacio a cada lado del operador binario. En la lista de parámetros de la función, después de una coma debe tener un espacio. En la lista de parámetros de la función, no agregar espacios a ambos lados del signo de igual predeterminado. No agregue espacios adicionales después del paréntesis izquierdo y antes</p>

del paréntesis derecho. Sin espacios adicionales antes del paréntesis de apertura del objeto de diccionario. No use espacios adicionales para alinear las declaraciones de asignación. Ejemplo:

```
def sign_up_page(request):
    users = User.objects.all()
    return render(request, 'sign_up.html', {'users': users})
```

### 3.3 Diagrama de Despliegue

Un Modelo de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos (Siriwardhana, 2020). Un nodo es un elemento de hardware o software.

A continuación, en la figura 15 se muestra el diagrama de despliegue de la propuesta de solución.



Figura # 15 Diagrama de despliegue (elaboración propia)

En el diagrama de despliegue el proceso de solicitud a la aplicación web se inicia en el nodo pc\_cliente, a través del navegador utilizando el protocolo HTTPS, mediante el cual se emite una petición al servidor de aplicaciones en el cual se encuentra hosteada la aplicación web. El servidor web establece una conexión con la base de datos, la cual se encuentra integrada al servidor de aplicaciones. Además, el nodo pc\_cliente mantendrá una conexión con la impresora a través de un conector USB.

**Pc\_cliente:** representa la computadora que utiliza el usuario, para realizar consultas o actualizar la información que se encuentra en el server web. La comunicación entre el pc\_cliente y el server web se establece utilizando el protocolo HTTPS para el envío de paquetes de forma segura por la red, evitando que se creen rastros de la información enviada en el navegador o servidor web.

**Server Web:** recibe las peticiones realizadas por el pc\_cliente y efectúa las consultas necesarias, para validar que dicha información se encuentra en la base de datos, para brindar una respuesta. La base de datos es la encargada del almacenamiento de datos del sistema.

**Impresora:** Es un dispositivo periférico del ordenador que permite imprimir los reportes generados por el sistema de gestión de libros.

### 3.4 Estrategia de prueba

La validación es el proceso de revisión al que se somete el sistema informático para comprobar que cumple con todas sus especificaciones y es capaz de realizar su cometido en cumplimiento con la normativa aplicable (TEAM, 2020).

Para la realización de la estrategia de prueba, se emplea a nivel de pruebas de sistema: las funcionales, de seguridad, rendimiento y carga. Las pruebas funcionales se ejecutan mediante el método de caja negra empleando la técnica de partición de equivalencia.

#### 3.4.1 Pruebas funcionales

Las pruebas funcionales se realizan para comprobar que las funcionalidades implementadas cumplan con las especificaciones de los requisitos definidos. Para desarrollar esta prueba, el autor selecciona la técnica de Caja Negra o también llamadas pruebas de comportamiento, las cuales se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa (Pressman, 2010).

A continuación, en la tabla 8 se muestra un ejemplo de diseño de caso de prueba del requisito solicitar libro, utilizado para detectar errores en la aplicación.

Los valores de las variables pueden ser: Validos(V), Inválidos(I) o No Aplica (NA).

Tabla 7. Caso de prueba RF\_7 Solicitar Libro

Escenario	Descripción	Nombre y Apellido	Categoría	Asignatura	Libro	Respuesta del sistema	Respuesta de prueba
EC1	Solicitar libro	V	V	V	V	Muestra el mensaje de confirmación " La solicitud se ha realizado satisfactoriamente"	
		I	V	V	V	Muestra el mensaje " Dato nombre y apellido incorrecto "	No muestra el mensaje " Dato nombre y apellido incorrecto "
		V	I	V	V	Muestra el mensaje " Datos incorrectos "	No muestra el mensaje " Datos incorrectos "

		V	V	V	V	Muestra el mensaje " Debe seleccionar una categoría "	No muestra el mensaje " Debe seleccionar una categoría "
		V	V	I	Programación web	Muestra el mensaje " Datos incorrectos "	No muestra el mensaje " Datos incorrectos "
EC 1.2	Solicitar libro	Pedro González	pedrog	Programación		Muestra el mensaje " No debe dejar campos vacíos "	Muestra el mensaje " No debe dejar campos vacíos "

### Resultado de las pruebas funcionales

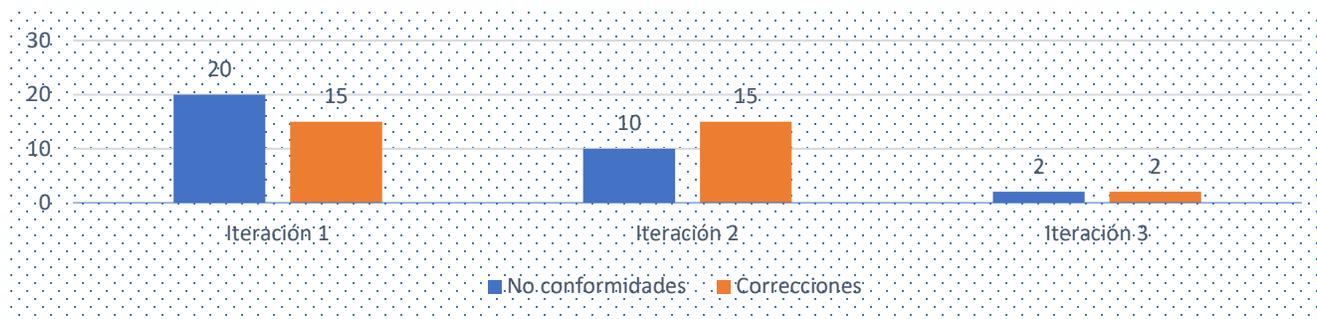
Se obtuvo en una primera iteración un total de veinte (20) no conformidades, divididas en seis (6) de ortografía, las cuales fueron corregidas al momento; trece (13) de funcionalidad relacionadas con la ausencia de la notificación al usuario cuando los datos de los campos son incorrectos, para resolverlo se incluyen los mensajes antes mencionados al código fuente.

Una no conformidad de excepción que se produce en la adición de un libro; al introducir incorrectamente los datos en los campos con caracteres extraños o letras, se muestra un error fatal que interrumpe el flujo normal del sistema, fue resuelta al validar que en los campos usuario y

nombre y apellido de dato sea de texto. De las deficiencias identificadas 15 fueron resueltas en la primera iteración.

En una segunda iteración en la cual se agregaron nuevas funcionalidades al sistema, se encontraron diez (10) no conformidades, de las cuales cinco (5) fueron errores ortográficos, los cuales fueron resueltos al momento, cinco (5) funcionalidades relacionadas con la ausencia de notificaciones al usuario cuando los datos son incorrectos o algún campo se encuentra vacío, para su solución se incluyeron al código los mensajes. Las deficiencias identificadas (10) y las cinco pendientes de la primera iteración fueron resueltas.

En una tercera iteración a la cual se le agregaron nuevas funcionalidades al sistema, se encontraron un total de dos (2) no conformidades, una relacionada con la funcionalidad eliminar, la cual no muestra el mensaje de alerta y la otra es una excepción que se produce con la adición de un libro, al introducir incorrectamente el dato solapín se muestra un error fatal que interrumpe el flujo normal del sistema, fue solucionada al validar que en el campo se aceptaran tanto letras como números. Las deficiencias fueron resueltas, obteniendo resultados satisfactorios y un sistema funcional. A continuación, en la figura 15 se muestra el gráfico de comportamiento del resultado de las pruebas funcionales.



*Figura # 15 Gráfico resumen de las pruebas funcionales (elaboración propia)*

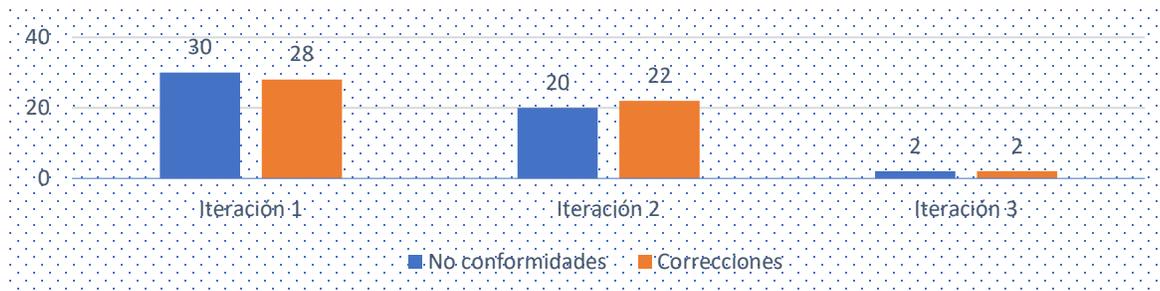
### 3.4.2 Pruebas de seguridad

Las pruebas de seguridad son el proceso de evaluar y probar la seguridad de la información de hardware, software, redes o un entorno de sistemas de información. Permite revisar y certificar el nivel de seguridad de un activo o instalación de tecnología de la información en relación con los pilares. Entre de los componentes clave de la seguridad de la información se encuentran : la disponibilidad, confidencialidad, integridad y autorización (Techopedia, 2022).

Para las pruebas de seguridad al sistema, se utiliza la herramienta Acunetix Web Vulnerability Scanner, con el objetivo de validar la seguridad de la solución propuesta.

Mediante estas pruebas se detectaron cincuenta y dos (52) alertas, de ellas cuarenta y seis (46) son de nivel medio y seis (6) de carácter informacional. De las vulnerabilidades de nivel medio, cuarenta y dos (42) están relacionadas con los mensajes de error producidos en el marco de desarrollo django y el acceso desde una página a otra a través de una url y se detectó un formulario que no está protegido contra la falsificación de solicitudes entre sitios (CSFR por sus siglas en inglés).

Las alertas de carácter informativo se dividen en cinco (5) enlaces rotos y una relacionada con el fichero views.py que contiene los atributos para la conexión con el sistema gestor de base de datos SQLite, este último relacionado con la obtención de los datos que provienen del método POST de los formularios. La alerta del formulario se soluciona manualmente como recomienda la herramienta Acunetix Web Vulnerability. Luego de ser solucionadas las alertas de vulnerabilidad encontradas se obtiene un sistema con mayor nivel de seguridad. A continuación, en la figura 16 se muestra el gráfico de comportamiento del resultado de las pruebas de seguridad.



*Figura # 16 Gráfico resumen de las pruebas de seguridad (elaboración propia)*

### 3.4.3 Pruebas de rendimiento

Las pruebas de rendimiento de software se centran en determinar la velocidad con la que el sistema bajo pruebas, realiza una tarea en las condiciones particulares del escenario de pruebas (Verona, Pérez y Torres, 2016). Donde las pruebas de carga se ocupan principalmente de

evaluar el rendimiento de los sistemas y el propósito de las pruebas de estrés es evaluar la disponibilidad y la estabilidad del sistema bajo una carga pesada.

### Resultados de las pruebas de rendimiento

Para las pruebas de rendimiento se utiliza el software Apache JMeter v2.8.4. Para ello se definen las propiedades de las computadoras cliente y servidor.

#### Hardware de prueba (PC cliente):

- Tipo de procesador: Intel(R) Core (TM) 2 Duo CPU P8400 @2.26GHz 2.27 GHz.
- RAM: 2 GB.
- Tipo de Red: Ethernet 10/100 Mbps.

#### Hardware de prueba (PC servidor):

- Tipo de procesador: Intel(R) Core (TM) 2 Duo CPU P8400 @2.26GHz 2.27 GHz.
- RAM: 4 GB.
- Tipo de Red: Ethernet 10/100 Mbps.

### Resultado de las pruebas de rendimiento

- **Etiqueta:** el nombre de la muestra (conjunto de muestras).
- **# Muestras:** el número de muestras para cada URL (peticiones).
- **Mín.:** el mínimo tiempo (milisegundos) transcurrido para las muestras de la URL dada.
- **Máx.:** el máximo tiempo (milisegundos) transcurrido para las muestras de la URL dada.
- **% Error:** porcentaje de las peticiones con errores.
- **Rendimiento:** rendimiento medido con base en peticiones respondidas por segundo/minuto/hora.

A continuación, en la tabla 9 se muestra el comportamiento del resultado de las pruebas de rendimiento.

Tabla 9. Resultado de las pruebas de rendimiento.

#Muestra	Min.	Max.	%Error	Rendimiento (peticiones/segundos)
100	3425ms	10456ms	0.00	3.6/seg
200	3425ms	10456ms	0,00	4,0/seg
300	3425ms	10456ms	0,00	4,5/seg

Según Nielsen, 1993 se considera un buen tiempo de respuesta entre 1.0 a 10.0 segundos para una aplicación dinámica. Este es aproximadamente el límite para que el flujo de pensamiento del usuario permanezca ininterrumpido, aunque el usuario notará la demora. Luego de analizados los datos de la tabla 9, se puede afirmar que el sistema de gestión de control de libros funciona correctamente y en un tiempo aceptable, teniendo en cuenta que el sistema no devuelve ningún error al realizarse las peticiones y responde en un tiempo aceptable.

### 3.5 Validación científica de la propuesta de solución

Con el objetivo de evaluar el desarrollo de un sistema web que facilite el proceso de gestión de entrega y recepción de libros en el Vicedecanato de Economía y Administración de la Facultad 1, se utiliza como técnica de validación, la técnica ladov.

Esta constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario (preguntas 1, 3, 5) del cuestionario aplicado y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el “Cuadro Lógico de ladov”.

El número resultante de la interrelación de las tres preguntas, indica la posición de cada sujeto en la escala de satisfacción, o sea su satisfacción individual. Permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 (Fabre, Ortega y Farrat 2020).

Mediante esta técnica se determinó el nivel de satisfacción individual y grupal a partir de un cuestionario elaborado según las exigencias pertinentes y aplicadas a una muestra de trabajadores a los que se le realizó la encuesta. Las preguntas cerradas fueron: ¿Considera que puede lograrse un cambio significativo en la gestión y control de libros en el almacén, sin disponer de un sistema informático específico para ello?, si necesitas producir un cambio positivo en la gestión y control de libros en la Facultad ¿emplearías el sistema que se propone? y ¿Cuál es su criterio definitivo sobre el diseño del software propuesto y su incidencia en la gestión de libros de la Facultad 1?

En la interacción de las tres preguntas los trabajadores demostraron una clara satisfacción, seis (6) trabajadores les gustó mucho y a uno (1) trabajador le gustó más de lo que le disgustó. A continuación, en la tabla 10 se muestra la relación de la satisfacción individual con la escala de satisfacción.

Tabla 10. Relación de la satisfacción individual con la escala de satisfacción

Escala	Significado	Satisfacción Individual	Porcentaje (%)
+1	Satisfecho	6	85
+0.5	Más satisfecho que insatisfecho	1	15
0	No definida y contradictorio	0	
-0.5	Más insatisfecho que satisfecho	0	
-1	Máxima insatisfacción	0	

Para obtener el índice de satisfacción grupal (ISG) se partió de asociar los diferentes niveles de satisfacción de los encuestados con una escala numérica que oscila entre +1 y - 1, con los resultados siguientes de la tabla 10. En esta investigación el ISG fue:  $ISG = 0.92$ , lo que representa un nivel de satisfacción alto y un reconocimiento a su utilidad, por parte de los entrevistados, que evidencia la calidad de la propuesta.

Además, contempla dos preguntas complementarias de carácter abierto, que permitió conocer cualitativamente mediante los criterios emitidos la importancia que se le concede a la investigación. Las preguntas abiertas fueron: Si tuvieras que referirte a los requisitos que se deben tener en cuenta para mejorar la gestión y control de libros en el almacén de libros de la Facultad 1 ¿cuáles mencionarías?, Si tuvieras que señalar algunas dificultades que se pueden presentar al desplegar el sistema informático para la gestión y control de libros de la Facultad 1 ¿cuáles mencionarías? y ¿cuáles son los resultados más significativos que se pueden obtener al aplicar el sistema para la gestión y control de libros en la Facultad 1?

Las respuestas a las preguntas abiertas del cuestionario, afirman que se cumplieron las expectativas y se sienten satisfechos con el sistema. No obstante, expresan recomendaciones como: la notificación por correo de la llegada de un libro nuevo al almacén y mostrar listado de libros entregados.

El proceso de validación mediante la Técnica de Iadov, resultó satisfactorio, expresado cuantitativamente en el alto Índice de Satisfacción Grupal (ISG = 0.92) y cualitativamente en los criterios emitidos donde se evidencia la importancia que le conceden a la investigación.

## Conclusiones del capítulo

En este capítulo se abordaron una serie de aspectos correspondientes a la implementación y validación del sistema, lo cual nos hace concluir que: la representación y descripción del diagrama de componentes permitió visualizar con más facilidad la estructura arquitectónica general del sistema. El diagrama de despliegue propuesto permitió exponer como se relacionan los elementos de hardware del sistema. La utilización de estándares de codificación permitió adoptar una estructura homogénea que facilita la comunicación y asegura el fácil mantenimiento del sistema.

Además, la aplicación de pruebas al sistema desarrollado permitió identificar las principales deficiencias en el desarrollo del sistema; así como las estrategias para solucionar los errores detectados y obtener un producto con una alta utilidad y mayor seguridad en el menor tiempo posible.

## CONCLUSIONES FINALES

Terminada la presente investigación, se puede concluir que:

- ✧ La investigación sobre las relaciones existentes entre los principales conceptos asociados al dominio de la presente investigación permitió una mayor comprensión de la propuesta de solución.
- ✧ El estudio del estado del arte de los sistemas de gestión de control de libros y el análisis de las diferentes herramientas y tendencias para la gestión de control de libros permitieron determinar las características que constituyen la base para el diseño de las funcionalidades que se definen en la propuesta de solución.
- ✧ La elaboración de los artefactos propuestos por la metodología de desarrollo y el levantamiento de requisitos permitieron una mayor comprensión, identificación de los procesos y características del sistema desarrollado.
- ✧ Las técnicas de validación aplicadas a la propuesta de solución permitieron la detección y corrección de las no conformidades detectadas y evidenciaron que el sistema constituye una solución funcional.

## RECOMENDACIÓN

Para el desarrollo de futuras investigaciones se recomienda:

- ✧ Desplegar el sistema web de gestión de control de libros en otras facultades de la universidad.

## REFERENCIAS BIBLIOGRÁFICAS

Acunetix | Web Application Security Scanner. [en línea], 2013. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.acunetix.com/>.

*Alvarez2015Implementacion.pdf* [en línea], [sin fecha]. S.l.: s.n. [Consulta: 23 noviembre 2022]. Disponible en: <https://archivo.cepal.org/pdfs/GuiaProspectiva/Alvarez2015Implementacion.pdf>.

AMAYA Q., L.A., 2017. FUNDAMENTOS DE LENGUAJE UNIFICADO DE MODELADO UML. Ing. Luis Armando Amaya Q. Laboratorio 1. Introducción a la Modelación del Sistema en UML - PDF Free Download. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://docplayer.es/94152788-Fundamentos-de-lenguaje-unificado-de-modelado-uml-ing-luis-armando-amaya-q-laboratorio-1-introduccion-a-la-modelacion-del-sistema-en-uml.html>.

APACHE, 2018. Welcome! - The Apache HTTP Server Project. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://httpd.apache.org/>.

Apache JMeter - Apache JMeter™. [en línea], 2022. [Consulta: 23 noviembre 2022]. Disponible en: <https://jmeter.apache.org/>.

ARANGO, C.G., ARANGO, J.P.V. y PÉREZ, H.A.P., 2022. PROTOTIPO DE APLICATIVO WEB ORIENTADO A LA MEJORA DE LA EXPERIENCIA DE USUARIO DENTRO DEL ÁMBITO DE LAS TERAPIAS ENERGÉTICAS Y VIBRACIONALES, pp. 82.

ASALE, R.- y RAE, 2021. libro | Diccionario de la lengua española. «*Diccionario de la lengua española*» - Edición del Tricentenario [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://dle.rae.es/libro>.

AYUDALEY, 2022. El Modelo base de datos: Definición y tipos. *Ayuda Ley Protección Datos* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://ayudaleyprotecciondatos.es/bases-de-datos/modelos/>.

BLAS, F. de A. y PLANELLS, J., 2021. Retos actuales de la educación técnico-profesional, pp. 159.

- CAMINO BUENO, P., 2018. Qué es Django y por qué usarlo | OpenWebinars. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://openwebinars.net/blog/que-es-django-y-por-que-usarlo/>.
- CAPTERRA, 2020. CodeAchi Library Management System. *Capterra* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.capterra.es/software/183633/codeachi-library-management-system>.
- CASTAÑEDA ROJAS, E.B., 2016. Propuesta de patrón de diseño de software orientado a prevenir la extracción automatizada de contenido web. En: Accepted: 2016-11-26T17:22:22Z, *Pontificia Universidad Católica del Perú* [en línea], Disponible en: <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/7513>.
- CERVANTES, H., 2018. Arquitectura de Software. *SG Buzz* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://sg.com.mx/revista/27/arquitectura-software>.
- DEFINISTA, 2022. ¿Qué es un Libro? Sus Características y Partes. *Concepto de - Definición de* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://conceptodefinicion.de/libro/>.
- DIARIO INFORME, 2021. ¿Qué es SQLite y por qué es tan popular? - Diario Informe. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://diarioinforme.com/que-es-sqlite-y-por-que-es-tan-popular/>.
- DÍAZ, L.M.L., 2021. Y SU POSIBLE CONTRIBUCIÓN AL DESARROLLO SOSTENIBLE DE LA UNIVERSIDAD, DESDE LA TEORÍA, pp. 15.
- Django. *Django Project* [en línea], 2019. [Consulta: 23 noviembre 2022]. Disponible en: <https://docs.djangoproject.com/es/4.1/>.
- EL CANTARANO, 2022. ALQUIBLA. WEB DE DIFUSIÓN CULTURAL. UNA MIRADA AL MUNDO DE LAS BIBLIOTECAS – El Cantarano. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://elcantarano.com/alquibla-web-de-difusion-cultural-una-mirada-al-mundo-de-las-bibliotecas/>.

EL UNIVERSO, 2019. La evolución del libro, desde la arcilla hasta el digital. *El Universo* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.eluniverso.com/entretenimiento/2019/04/23/nota/7290560/evolucion-libro-arcilla-hasta-digital>.

E-LEARNING MASTERS, 2018. Ventajas de utilizar HTML5 en cursos virtuales. *Comunidad eLearning Masters | edX* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <http://elearningmasters.galileo.edu/2018/05/04/html5-en-cursos-virtuales/>.

FABRE, A.F. de C., ORTEGA, N.S. y FARRAT, Y.R., 2020. El proceso de validación mediante la Técnica de ladov en cursos por encuentros. *Revista Ingeniería Agrícola* [en línea], vol. 10, no. 1. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.redalyc.org/journal/5862/586262449010/html/>.

GONZÁLEZ BRITO, H., CAMEJO, R. y GAINZA REYES, D., 2012. EXTENSIÓN DEL ERP CEDRUX CON EL MARCO DE TRABAJO SAUXE. CASO DE ESTUDIO: SUBSISTEMA POSTGRADO 1.0 S.I.: s.n., DOI 10.13140/RG.2.1.3430.0881.

HOSTING PLUS, F., 2021. Concepto de IDE y cuáles son sus características. *Hosting Plus* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.hostingplus.mx/blog/concepto-de-ide-y-cuales-son-sus-caracteristicas/>.

IBM Documentation. [en línea], 2022. [Consulta: 15 septiembre 2022]. Disponible en: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/control-desk/7.6.1.1?topic=incidents-resolving-incident>.

IN LEARNING, 2017. Conceptos sobre las pruebas de software - Instituto de Gestión de Proyectos (PMI)® | LinkedIn Learning, antes Lynda.com. *LinkedIn* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://es.linkedin.com/learning/fundamentos-de-la-gestion-de-proyectos-calidad/conceptos-sobre-las-pruebas-de-software>.

Introducción al sistema gestor de base de datos (SGBD). *IONOS Digital Guide* [en línea], 2020. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/sistema-gestor-de-base-de-datos-sgbd/>.

IZERTIS, por, 2016. JMeter, un viejo amigo en plena forma. *Solid GEAR* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://ahorasomos.izertis.com/solidgear/jmeter-un-viejo-amigo-en-plena-forma/>.

JET BRAINS, 2022. Funcionalidades - PyCharm. *JetBrains* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.jetbrains.com/pycharm/features/>.

LARMAN, C., 2003. *UML Y PATRONES*. S.I.: s.n. ISBN 978-84-205-3438-1.

LIBRARY, 2022. SITUACIÓN ACTUAL DE LAS APLICACIONES Y SERVIDORES. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://1library.co/article/situaci%C3%B3n-actual-de-las-aplicaciones-y-servidores.y964e8ly>.

MENZINSKY, A., LÓPEZ, G., PALACIO, J., SOBRINO, M.A., ÁLVAREZ, R. y RIVAS, V., 2020. *Historias de Usuario*, pp. 64.

MORALES, A., 2016. Partes de un libro. *Significados* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.significados.com/partes-libro/>.

MULTYSYSTEM, 2014. Los Beneficios de Acusensor de Acunetix | MultiSystem - Tecnología Superior. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <http://www.multisystem.cl/productos/acunetix-2/los-beneficios-de-acusensor-de-acunetix/>.

ORTIZ PABÓN, E. y NAGLES GARCÍA, N., 2013. *Gestión de Tecnología e Innovación. Teoría, proceso y práctica* [en línea]. 2. S.I.: Universidad EAN. [Consulta: 23 noviembre 2022]. ISBN 978-958-756-255-2. Disponible en: <http://hdl.handle.net/10882/1580>.

PASCUAL, J.R., 2019. Acoplamiento y Cohesión. *Disrupción Tecnológica* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.disrupciontecnologica.com/acoplamiento-y-cohesion/>.

POWERDATA, 2019. ¿Qué es un gestor de datos y para qué sirve? [en línea]. [Consulta: 24 noviembre 2022]. Disponible en: <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-un-gestor-de-datos-y-para-que-sirve>.

PRESSMAN, R.S., 2010. *Ingeniería del Software. Un Enfoque Practico.*, pp. 805.

RAMOS, D., 2015. Amazon.com: Curso de Ingeniería de Software (Spanish Edition): 9781515194804: Ramos, Daniel, Noriega, Raúl, Laínez, José Rubén, Durango, Alicia: Libros. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.amazon.com/Curso-Ingenier%C3%ADa-Software-Spanish-Daniel/dp/1515194809>.

*Resolución47-2022.pdf* [en línea], [sin fecha]. S.l.: s.n. [Consulta: 23 noviembre 2022]. Disponible en: <http://www.hospitalameijeiras.sld.cu/hha/sites/all/informacion/2022/Resoluci%C3%B3n47-2022.pdf>.

ResourceMate Pricing, Alternatives & More 2022 - Capterra. [en línea], 2022. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.capterra.com/p/40056/ResourceMate/>.

RÍOS, P., 2002. El libro de texto como recurso para el aprendizaje estratégico, vol. 16.

RODRIGUEZ, F., 2018. INGENIERIA DE SOFTWARE. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://docplayer.es/73925061-Ingenieria-de-software-ing-francisco-rodriguez.html>.

SANTANDER, U., 2020. Metodologías de desarrollo de software: ¿qué son? [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>.

SANTANDER, U., 2021. Python: qué es y por qué deberías aprender a utilizarlo. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.becas-santander.com/es/blog/python-que-es.html>.

SIRIWARDHANA, S., 2020. Tutorial de Diagrama de Despliegue | ¿Qué es un Diagrama de Despliegue? *Blog de Creately* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>.

SOMMERVILLE, I., 2011. Ingeniería de Software 7a Edición. [en línea], [Consulta: 23 noviembre 2022]. Disponible en: [https://www.academia.edu/15059886/Ingenieria\\_de\\_Software\\_Ian\\_Sommerville\\_7a\\_Edicion](https://www.academia.edu/15059886/Ingenieria_de_Software_Ian_Sommerville_7a_Edicion).

SOUSA, J.M. de, 2004. *Diccionario de bibliología y ciencias afines: (terminología relativa a la archivística, bibliofilia, bibliografía[...] paleografía, papirología, etc.)* [en línea]. S.l.: Fundación

Germán Sánchez Ruipérez. [Consulta: 23 noviembre 2022]. ISBN 978-84-86168-45-2. Disponible en: <https://dialnet.unirioja.es/servlet/libro?codigo=592296>.

TEAM, A., [sin fecha]. Guía definitiva de la validación de sistemas informáticos. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://www.ambit-bst.com/blog/guía-definitiva-de-la-validación-de-sistemas-informáticos>.

TECHOPEDIA, 2022. ¿Qué son las pruebas de seguridad? - definición de techopedia - Seguridad. *Icy Science* [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://es.theastrologypage.com/security-testing>.

TORRES LEBRATO, L., 2015. La gestión de información y la gestión del conocimiento. *Revista Archivo Médico de Camagüey*, vol. 19, no. 2, pp. 96-98. ISSN 1025-0255.

VÁZQUEZ, S., 2015. ¿Qué es Calibre? - Crea y Distribuye tu libro digital | Publicación Digital | Elabora, Distribuye y Promociona tus Libros Digitales. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: <https://digital-editorial.com/que-es-calibre/>.

VERONA MARCOS, S., PÉREZ DÍAZ, Y. y TORRES PÉREZ, L., 2016. Pruebas de rendimiento a componentes de software utilizando programación orientada a aspectos. [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1815-59362016000300006](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59362016000300006).

Carlos A. Guerrero, Johanna M. Suárez y Luz E. Gutiérrez. 2013. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. [En línea] 28 de enero de 2013. [www.researchgate.net](http://www.researchgate.net).

# ANEXOS

## Anexo 1. Diagramas de Diseño de Clases

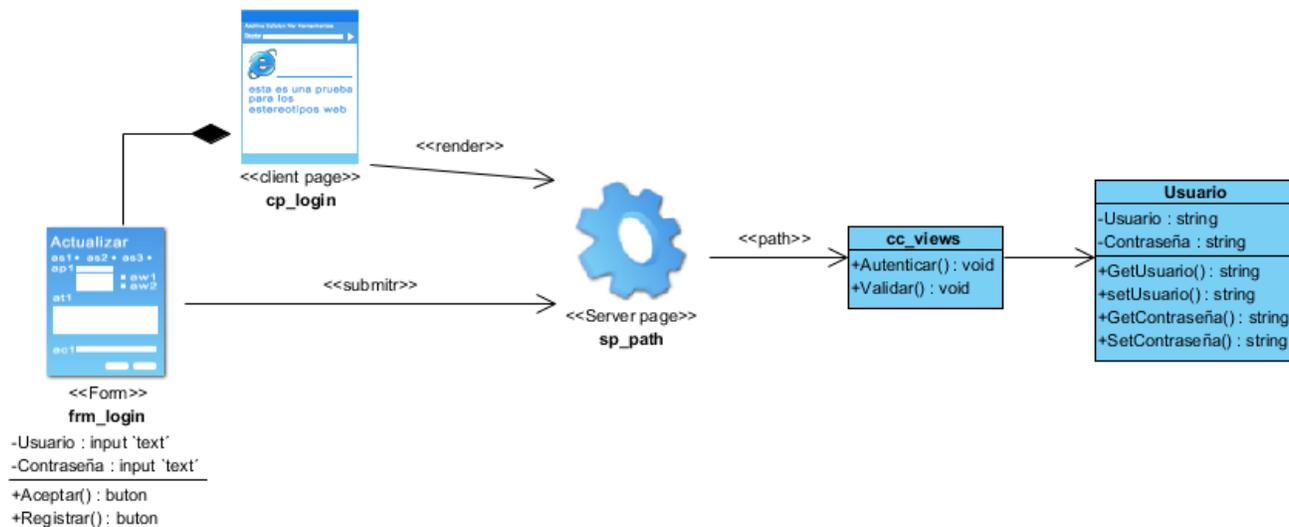


Figura # 11 DCD con estereotipos web Autenticar Usuario (elaboración propia)

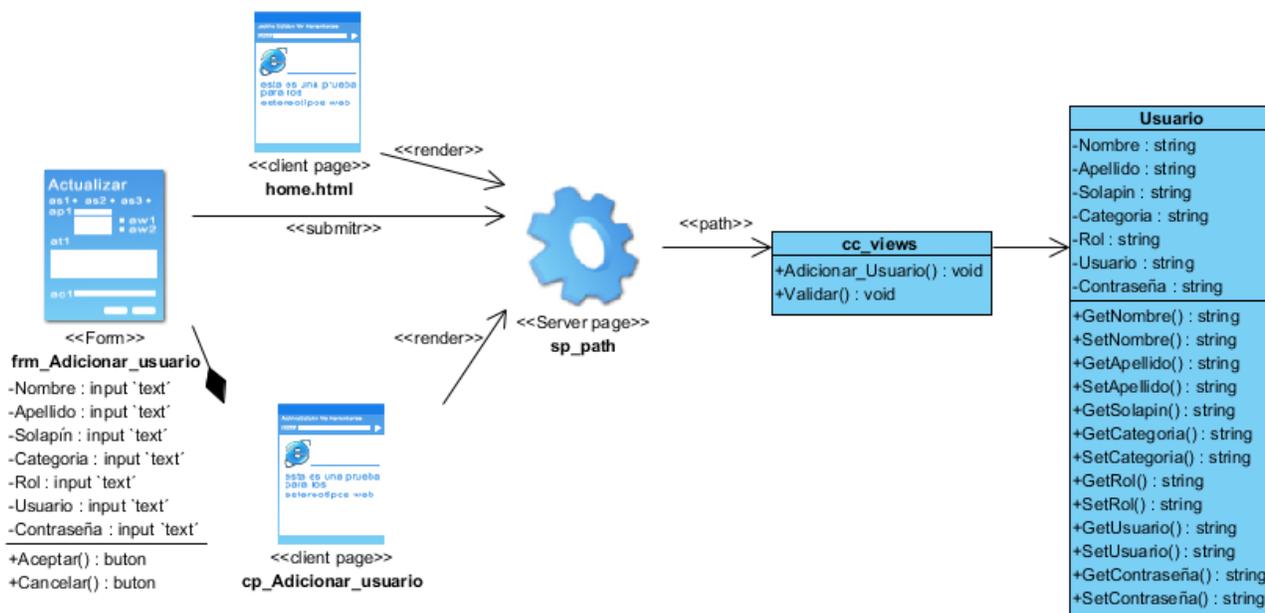


Figura # 12 DCD con estereotipos web Adicionar Usuario (elaboración propia)

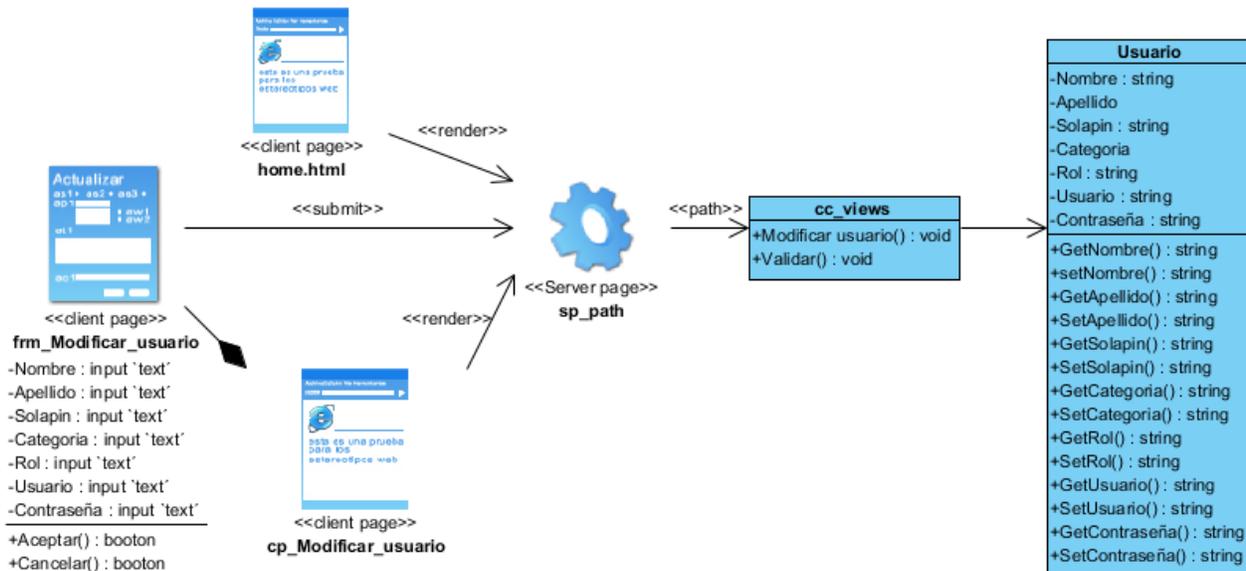


Figura # 13 DCD con estereotipos web Modificar Usuario (elaboración propia)

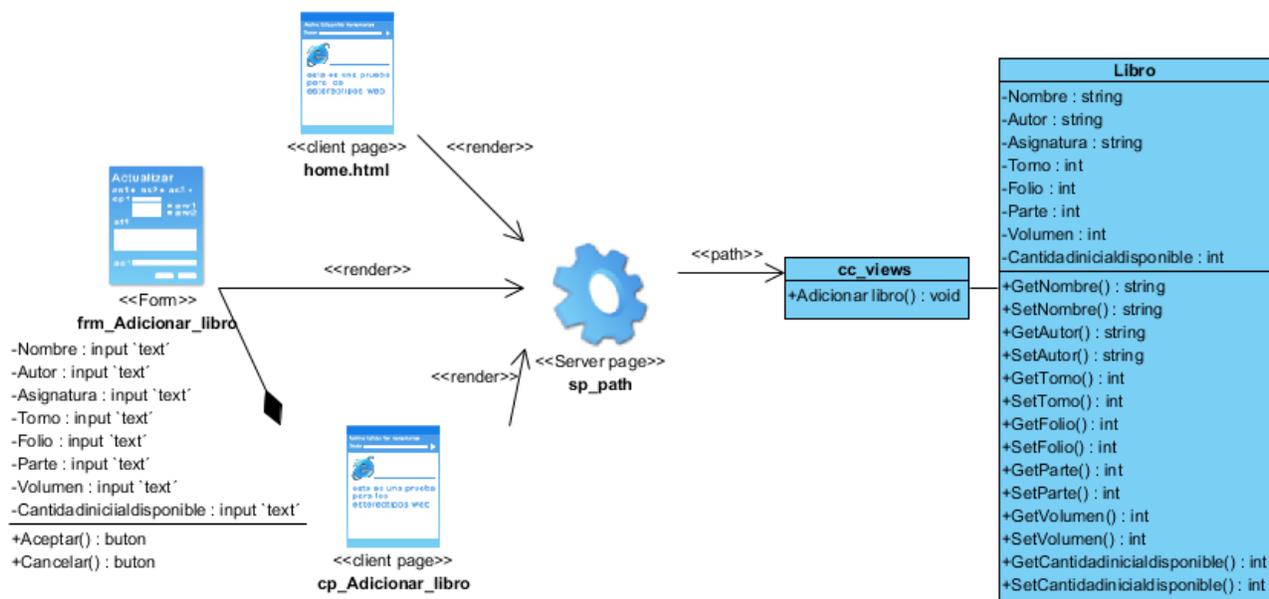


Figura # 14 DCD con estereotipos web Adicionar Libro (elaboración propia)

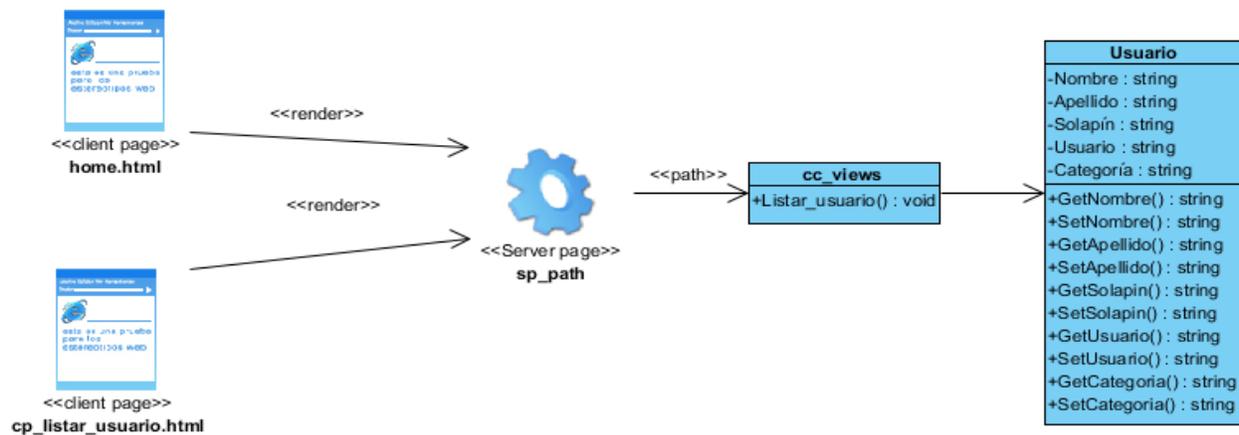


Figura # 15 DCD con estereotipos web Listar Usuario (elaboración propia)

**Anexo 2. Entrevista al cliente para conocer la necesidad del desarrollo de la propuesta de solución y definir los requisitos funcionales y no funcionales.**

Estimado profesor: Se necesita de su cooperación en una investigación para una tesis de pregrado. Por ello, sería de gran ayuda que respondiera lo siguiente:

1. ¿Cuántas personas intervienen en el proceso de entrega y recepción de libros?
2. ¿Cuál es la frecuencia con la que se realiza el proceso de entrega y recepción de libros?
3. ¿Este proceso de entrega y recepción de libros se le realiza tanto a profesores como a estudiantes?
4. ¿Todos los cursos que se realizan en la facultad tienen derecho al proceso de entrega y recepción de libros?
5. ¿Se realizan inventarios en el local de almacenamiento de libros, con frecuencia?
6. ¿Qué tipo de informe o reportes se generan después de haber realizado un inventario?
7. ¿Cómo se realiza el control de libros en el local de almacenamiento?

### Anexo 3. Cuestionario

Estimado participante: Le agradecemos que colabore con una investigación en proceso de Trabajo de Diploma, para fortalecer la gestión y control de libros del almacén de libros de la Facultad 1 de la UCI. Por ello, sería de gran ayuda que lea cuidadosamente las preguntas y responda lo siguiente:

NIVEL DE SATISFACCIÓN CON LA PROPUESTA						
Si tuvieras que referirte los requisitos que se deben tener en cuenta para mejorar la gestión y control de libros en el almacén de libros de la Facultad 1, ¿cuáles mencionarías?						
¿Considera que puede lograrse un cambio significativo en la gestión y control de libros en el almacén, sin disponer de un sistema informático específico para ello?	No	No sé	Si			
Si tuvieras que señalar algunas dificultades que se pueden presentar al desplegar el sistema informático para la gestión y control de libros de la Facultad 1, ¿cuáles mencionarías?						
Si necesitas producir un cambio positivo en la gestión y control de libros en la Facultad 1, ¿emplearías el sistema que se propone?	No	No sé	Si			
A tu consideración, ¿cuáles son los resultados más significativos que se pueden obtener al aplicar el sistema para la gestión y control de libros en la Facultad 1?						
¿Cuál es su criterio definitivo sobre el diseño del software propuesto y su incidencia en la gestión de libros Facultad 1?	Me gusta mucho	No me gusta tanto	Me da lo mismo	Me gusta más de lo que me gusta	No me gusta nada	No sé qué decir

--	--	--	--	--	--	--

## Anexo 4. Historias de Usuarios

Tabla 5. HU\_13 Listar Libro

Historia de Usuario																			
<b>Número: 13</b>	<b>Usuario:</b> Cliente																		
<b>Nombre de historia:</b> Listar libros																			
<b>Prioridad en negocio(Alta,Medio, Baja):</b> Alta	<b>Riesgo en desarrollo(Alta,Medio,Bajo):</b> Medio																		
<b>Tiempo estimado: 3</b>	<b>Iteración asignada: 2</b>																		
<b>Programador Responsable:</b> Rachel Marichal Casaña																			
<b>Descripción:</b> Permite listar los libros registrados en la base de datos. Donde se encuentra mostrada la información referente a los libros recogida en los campos: nombre, autor, tomo, folio, parte, volumen, cantidad y asignatura; además de encontrarse los botones modificar y eliminar en caso de querer realizar alguna de estas funciones.																			
<b>Validación:</b> El usuario puede ver un listado de libros																			
<b>Interfaz:</b>																			
<div style="display: flex; gap: 10px;"> <span style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 3px;">Libros</span> <span style="color: #007bff; padding: 2px 5px;">Usuarios</span> <span style="color: #007bff; padding: 2px 5px;">Solicitudes</span> </div> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Id</th> <th>Nombre</th> <th>Autor</th> <th>Tomo</th> <th>Folio</th> <th>Parte</th> <th>Volumen</th> <th>Cantidad</th> <th>Asignatura</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Programacion II</td> <td>Jonh Doe</td> <td>1</td> <td>1</td> <td>2</td> <td>2</td> <td>100</td> <td>Programacion II</td> </tr> </tbody> </table>		Id	Nombre	Autor	Tomo	Folio	Parte	Volumen	Cantidad	Asignatura	1	Programacion II	Jonh Doe	1	1	2	2	100	Programacion II
Id	Nombre	Autor	Tomo	Folio	Parte	Volumen	Cantidad	Asignatura											
1	Programacion II	Jonh Doe	1	1	2	2	100	Programacion II											

Tabla 7. HU\_6 Listar Usuarios

Historia de Usuario	
<b>Número: 6</b>	<b>Usuario:</b> Cliente
<b>Nombre de historia:</b> Listar usuarios	
<b>Prioridad en negocio(Alta,Medio, Baja):</b> Alta	<b>Riesgo en desarrollo(Alta,Medio,Bajo):</b> Alto
<b>Tiempo estimado: 6</b>	<b>Iteración asignada: 2</b>
<b>Programador Responsable:</b> Rachel Marichal Casaña	

<p><b>Descripción:</b> El sistema permite hacer un listado de todos los usuarios registrados en él. Donde se encuentra mostrada la información referente a los usuarios recogida en los campos: nombre, apellido, solapín, usuario y categoría; además de encontrarse los botones de eliminar y modificar en caso de querer realizar dichas funciones.</p>												
<p><b>Validación:</b> Muestra un listado de todos los usuarios en el sistema</p>												
<p><b>Interfaz:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="margin-right: 10px;">Libros</span> <span style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 3px;">Usuarios</span> <span style="margin-left: 10px;">Solicitudes</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Id</th> <th style="width: 20%;">Nombre</th> <th style="width: 20%;">Apellidos</th> <th style="width: 15%;">Solapin</th> <th style="width: 15%;">Usuario</th> <th style="width: 25%;">Categoría</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Serhio</td> <td>Perez Perez</td> <td>E234567</td> <td>pperez</td> <td>estudiante</td> </tr> </tbody> </table>	Id	Nombre	Apellidos	Solapin	Usuario	Categoría	1	Serhio	Perez Perez	E234567	pperez	estudiante
Id	Nombre	Apellidos	Solapin	Usuario	Categoría							
1	Serhio	Perez Perez	E234567	pperez	estudiante							

Tabla 4. HU\_12 Modificar Libro

Historia de Usuario	
<b>Número:</b> 12	<b>Usuario:</b> Cliente
<b>Nombre de historia:</b> Modificar libros	
<b>Prioridad en negocio(Alta, Medio, Baja):</b> Alta	<b>Riesgo en desarrollo(Alta, Medio, Bajo):</b> Medio
<b>Tiempo estimado:</b> 6	<b>Iteración asignada:</b> 2
<b>Programador Responsable:</b> Rachel Marichal Casaña	
<p><b>Descripción:</b> Permite modificar los libros registrados en la base de datos. Al presionar el botón modificar aparece un formulario con todos los campos rellenos con la información referente al libro que desea modificar. Después de haberse realizados las modificaciones necesarias aparece el botón aceptar si desea guardar las modificaciones y el botón cancelar en caso de desear guardarlas.</p>	
<b>Validación:</b> El usuario puede modificar los datos registrados del libros	
<b>Interfaz:</b>	

### Adicionar Libro

Nombre del Libro:

Nombre del Autor:

Asignatura:

Tomo:

Folio:

Parte:

Volumen:

Cantidad Inicial Disponible:

## Anexo. Pruebas funcionales

Tabla 8. Caso de Prueba RF\_12 Modificar Libro

ID	Des- cripción	Tí- tulo	Au- tor	Asigna- tura	Tomo	Folio	Parte	Vo- lu- men	Cant d.	Resp. sis- tema	Resp. prueb a
EC 2	Modifi- car libro	V	V	V	V	V	V	V	V	Mues- tra el men- saje “ Acción reali- zada satis- facto- ria- mente ”	
		V	V	I	V	V	V	V	V	Mues- tra el men- saje “ Asig- natura inco- rrecta”	

		V	V	V	V	V	I	V	V	Mues- tra el men- saje " " Datos inco- rrec- tos " "	No mues- tra el men- saje " " Datos inco- rrec- tos " "
		V	I	V	V	V	V	V	V	Mues- tra el men- saje " " Autor inco- rrecto " "	No mues- tra el men- saje " " Autor inco- rrecto " "
		Pro- gra- ma- ción web	Julio Cas- ta- ña	Progra- mación	46	140	1	II	100		