

Universidad de las Ciencias Informáticas

Facultad 4



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

Título: Sistema informático para la gestión de los recursos materiales y libros.

Autor: *Leinier Caraballo Yanes*

Tutor: *M.Sc. Yordankis Matos López*

La Habana, diciembre de 2022

“Año 64 de la Revolución”



PENSAMIENTO:



## DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “Sistema informático para la gestión de los recursos materiales y libros” 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 firman la presente a los 7 días del mes de diciembre del año 2022.



Firma del autor

Leinier Caraballo Yanes



Firma del tutor

Yordankis Matos López

## **AGRADECIMIENTOS**

La amistad y el cariño que se reciben de aquellos que te rodean, son factores que determinan de manera decisiva el desarrollo de una persona. La cualidad más hermosa que debe poseer un ser humano es reconocer a quienes lo han ayudado a conseguir sus objetivos.

No es una tarea fácil, escribir en una hoja de papel, los nombres de todas aquellas personas que de una manera u otra me ayudaron en la realización de este trabajo, o que compartieron momentos inolvidables de mi vida, pero trataré de agradecerle a todos aquellos que con su apoyo y fuerza aportaron su granito de arena para poder realizar este trabajo.

Quiero comenzar dando un agradecimiento especial a mi familia, por su esfuerzo, cariño, y confianza, depositados en mí desde el inicio de la carrera.

A mi novia, que a pesar de sacarme de paso a veces, siempre ha estado a mi lado apoyandome.

A todas las personas que he conocido en los últimos 5 años, que de alguna manera u otra han estado para mí cuando más les he necesitado y han brindado su apoyo incondicional, en especial a mis “Negritos”, incluyendo los que hoy no se encuentran en la universidad, de los cuales no quiero mencionar nombres, para que no se me quede ninguno fuera.

A mi tutor, por ser mi guía durante todo el proceso de desarrollo de mi trabajo de diploma, y por dedicar un parte de su tiempo a atenderme y ayudarme.

Al tribunal por su ayuda, sus consejos, y sus exigencias para terminar con éxito mi trabajo de diploma.

A los profesores/as que contribuyeron a mi formación y a quienes me dieron su apoyo en los momentos difíciles, en particular a: Henry, Yonny, Yadira, Yasirys y Rafael.

¡Muchas Gracias a todos!

## RESUMEN

La presente investigación, comprende el registro, análisis e interpretación, de las características y naturaleza actual de los procesos de gestión de los recursos materiales y libros correspondientes a la Facultad 4. Surge, debido a la necesidad de información que presenta el Vicedecanato de administración para la toma de decisiones. En la facultad estos procesos se realizan de manera manual, haciendo que sea un proceso lento, debido a la gran cantidad de información almacenada, conduciendo en múltiples ocasiones a informaciones duplicadas, desaciertos en la veracidad de la información, y problemas de tiempo en la atención a estudiantes y trabajadores. Existe entonces la necesidad de buscar una solución, capaz de automatizar estos procesos. Una vez realizadas las entrevistas a los vicedecanos de administración y economía de todas las facultades y observación de los procesos, se identificaron 40 requisitos funcionales y 20 no funcionales. Además, se trazó como objetivo: concebir el estudio, análisis, diseño e implementación de una aplicación web para la gestión de los recursos materiales y libros de la Facultad 4, utilizando la metodología XP para guiar el proceso de desarrollo del sistema. Para su implementación, se utilizó el lenguaje de programación java, y para el desarrollo de las interfaces de usuario, el marco de trabajo Vaadin. El sistema incluye un catálogo de libros, información estadística del almacén de libros, creación de reportes y consumo de los servicios UCI. La solución fue validada mediante pruebas las unitarias y de aceptación, garantizando un correcto funcionamiento del sistema y la aceptabilidad del cliente.

**PALABRAS CLAVE:** automatizar, libro, sistema de gestión, recurso material, toma de decisiones.

## **ABSTRACT**

The present investigation includes the registration, analysis and interpretation of the characteristics and current nature of the management processes of material resources and books corresponding to Faculty 4. It arises due to the need for information presented by the Vice Dean of administration for decision making. In the faculty, these processes are carried out manually, making it a slow process, due to the large amount of information stored, leading on multiple occasions to duplicate information, errors in the veracity of the information, and time problems in attention. to students and workers. There is then the need to find a solution capable of automating these processes. Once the interviews with the vice deans of administration and economics of all the faculties and observation of the processes were carried out, 40 functional requirements and 20 non-functional ones were identified. In addition, the objective was: to conceive the study, analysis, design and implementation of a web application for the management of material resources and books of Faculty 4, using the XP methodology to guide the system development process. For its implementation, the Java programming language was used, and for the development of the user interfaces, the Vaadin framework. The system includes a catalog of books, statistical information on the book store, creation of reports and consumption of UCI services. The solution was validated through unit and acceptance tests, guaranteeing the correct functioning of the system and the acceptability of the client.

**KEYWORDS:** automate, book, management system, material resource, decision making.

Índice

INTRODUCCIÓN .....	1
CAPÍTULO 1 Fundamentación teórica .....	7
1.1 Análisis de los procesos de negocio vinculados a la distribución y el control de los recursos materiales en la Facultad 4 .....	7
1.2 Análisis de los sistemas homólogos.....	13
1.2.1 Soluciones internacionales .....	13
1.2.2 Soluciones nacionales .....	15
1.3 Herramientas y tecnologías a utilizar .....	18
1.3.1 Lenguaje de programación .....	18
1.3.2 Frameworks.....	20
1.3.3 Servidores de aplicación web .....	22
1.3.4 Entorno de desarrollo integrado (IDE) .....	23
1.3.5 Sistema gestor de base de datos .....	25
1.3.6 Control de versiones.....	26
1.4 Metodologías de desarrollo de software y herramientas para el modelado	29
1.4.1 Metodologías de desarrollo de software .....	29
1.4.2 Lenguaje de modelado .....	34
1.4.3 Herramienta de modelado .....	36
Conclusiones parciales .....	36
CAPÍTULO 2 Características del sistema y propuesta solución.....	38
2.1 Fase 1 Exploración.....	38
2.1.1 Educción de requisitos.....	38
2.1.2 Requerimientos del sistema .....	40
2.2 Fase 2 Planificación de entregas .....	53
2.2.1 Historias de usuario .....	53
2.2.2 Estimación de tiempo por historias de usuario .....	56

2.2.3	Plan de iteraciones .....	59
2.2.4	Plan de entrega .....	62
2.2.5	Reuniones diarias de seguimiento .....	63
2.3	Fase 3 Etapa de diseño .....	64
2.3.1	Tarjeta CRC (Clase-Responsabilidad-Colaboradores) .....	64
2.3.2	Arquitectura Modelo-Vista-Controlador.....	65
2.4	Fase 4 Implementación .....	67
2.4.1	Estándares de codificación.....	67
2.4.2	Patrones de diseño.....	68
2.4.3	Modelo de datos .....	70
2.5	Diagrama de despliegue .....	71
	Conclusiones parciales .....	72
	Capítulo 3 Validación de la propuesta solución.....	74
3.1	Matriz de trazabilidad .....	74
3.2	Fase 5 Pruebas.....	74
3.2.1	Pruebas unitarias.....	75
3.2.2	Pruebas de sistema .....	79
3.2.3	Pruebas de aceptación .....	86
3.3	Resultados de las pruebas.....	90
	Conclusiones parciales .....	91
	Conclusiones generales .....	92
	Recomendaciones.....	93
	Bibliografía .....	94
	Anexos .....	101

Índice de las Figuras:

Figura 1 Estructura jerárquica de la facultad teniendo en cuenta los roles que intervienen en los procesos de distribución y control de los recursos materiales y libros en la Facultad 4 (Elaboración propia) .....	8
Figura 2 Descripción del subproceso: Entrega medios materiales (Elaboración propia) .....	9
Figura 3 Descripción del subproceso entregar libros a estudiantes o trabajadores (Elaboración propia) .....	10
Figura 4 Descripción del subproceso: Recepcionar libros entregados a estudiantes y trabajadores (Elaboración propia) .....	11
Figura 5 Descripción del subproceso: Controlar los libros en el almacén (Elaboración propia) .....	11
Figura 6 Metodología XP Fases (Sinnaps, 2020).....	33
Figura 7 Funcionamiento del patrón Modelo Vista Controlador (Elaboración propia) .....	66
Figura 8 Diagrama Entidad-Relación (Elaboración propia) .....	71
Figura 9 Diagrama de despliegue (Elaboración propia) .....	72
Figura 10 Código del método de prueba salvar libro .....	77
Figura 11 Grafo de flujo.....	77
Figura 12 Gráfico de no conformidades .....	90

Índice de las Tablas:

Tabla 1 Comparación de los lenguajes de programación Python, Java y JavaScript.....	18
Tabla 2 Comparación entre los frameworks de desarrollo .....	21
Tabla 3 Comparación de sistemas gestores de base de datos .....	26
Tabla 4 Sistemas de control de versiones más populares - ventajas y desventajas .....	27
Tabla 5 Comparación entre las metodologías .....	29
Tabla 6 Comparación de metodologías ágiles .....	31
Tabla 7 Requerimientos funcionales .....	40
Tabla 8 Historia de usuario “ Añadir libro” .....	54
Tabla 9 Estimación de tiempo .....	57
Tabla 10 Estimación del tiempo de HU .....	57
Tabla 11 Plan de iteraciones.....	59
Tabla 12 Plan de entrega .....	62
Tabla 13 Reuniones de seguimiento .....	63
Tabla 14 Tarjeta CRC LibroView.....	65
Tabla 15 Técnica del camino básico .....	78
Tabla 16 Técnica del camino básico caso de prueba No. 1 .....	78
Tabla 17 Técnica del camino básico caso de prueba No. 2 .....	78
Tabla 18 Técnica del camino básico caso de prueba No. 3 .....	79
Tabla 19 Lista de chequeo para pruebas de usabilidad .....	80
Tabla 20 Clases de equivalencia.....	87
Tabla 21 Caso de prueba " Añadir libro" .....	88
Tabla 22 Descripción de los roles y responsabilidades.....	101
Tabla 23 Historia de usuario: Autenticar usuario.....	101
Tabla 24 Historia de usuario: Modificar usuario .....	103
Tabla 25 Historia de usuario: Eliminar usuario .....	104
Tabla 26 Historia de usuario: Listar usuario .....	104
Tabla 27 Historia de usuario: Filtrar usuario.....	105
Tabla 28 Historia de usuario “Modificar libro” .....	106
Tabla 29 Historia de usuario “Eliminar libro”.....	108
Tabla 30 Historia de usuario “Listar/Mostrar Libro” .....	109

Tabla 31 Historia de usuario: Filtrar libro.....	110
Tabla 32 Historia de usuario: Añadir tarjeta de préstamo .....	111
Tabla 33 Historia de usuario: Modificar tarjeta de préstamo .....	112
Tabla 34 Historia de usuario: Eliminar tarjeta de préstamo.....	113
Tabla 35 Historia de usuario: Listar tarjeta de préstamo .....	114
Tabla 36 Historia de usuario: Filtrar tarjeta de préstamo.....	115
Tabla 37 Historia de usuario: Visualizar estadísticas de almacén.....	116
Tabla 38 Historia de usuario: Exportar reporte de estadísticas de almacén ..	117
Tabla 39 Historia de usuario: Visualizar tarjeta personal de préstamo.....	117
Tabla 40 Matriz de trazabilidad: Requisito funcional vs Historia de usuario...	118

## INTRODUCCIÓN

A lo largo de la historia el hombre ha necesitado almacenar y transmitir datos e información, para esto se ha valido de distintos herramientas y mecanismos. Estos mecanismos se han ido perfeccionando con los avances de la ciencia y la tecnología, y evolucionan hacia un desarrollo tecnológico, que puso de manifiesto la necesidad, el interés y el ingenio del hombre en el camino evolutivo.

En la actualidad, es necesario integrar las tecnologías de la información y los sistemas informáticos, a los departamentos donde se trabaja con recursos humanos, para gestionar de la mejor manera la información que fluye a través de sus procesos operativos. Tomando en cuenta que, al no hacerlo, ocasionará que los servicios tengan pérdidas de tiempo y de la trazabilidad de sus procesos, afectando la atención a sus usuarios(Demestre,2019).

El mundo de la educación, no puede dejar de reconocer la realidad tecnológica de hoy como instrumento del que valerse. Poco a poco los centros educativos han entendido la importancia de apoyarse en la tecnología para ayudar a enriquecer, transformar y completar la trayectoria académica del estudiantado, a través de softwares interactivos y de gestión, tabletas, páginas web, video conferencias, correo electrónico, chats, entre otros(Salmerón Navarro,2022).

Los sistemas de gestión de la información no solo han sido empleados para mejorar el proceso de enseñanza, además, son utilizados para realizar actividades administrativas como, controlar y gestionar la información los estudiantes y administrativos. El uso de los sistemas de gestión de información es cada vez más común gracias a las ventajas que brindan estos para automatizar tareas. Estos permiten obtener información más precisa y ayudan a diseñar estrategias que beneficien a las universidades. Además, al confiar ciertas tareas a estos, los empleados pueden tener más tiempo para enfocarse en tareas más importantes que ameritan su atención(Pereda,2021).

La Universidad de Ciencias Informáticas (UCI) cuenta con varios sistemas que tributan a la informatización de diferentes procesos entre ellos: el Sistema de Gestión Universitaria (Akademos), que apoya el proceso de formación de pregrado y posgrado y la Suite de Gestión de Proyectos (GESPRO), la cual es

una Suite orientada a la web que permite la planificación, seguimiento y control de proyectos, entre otros.

Actualmente, en la universidad aún existen procesos que no se realizan con la calidad requerida, como son la entrega y devolución de libros al almacén y la distribución de los módulos de materiales docentes, oficina e insumos de limpieza, así como el control de su destino final. A pesar de que en todas las facultades realizan un control interno propio garantizando que estos recursos lleguen a su destino final, estos procesos suelen ser lentos, pues la información referente a los mismos, se encuentra en grandes armarios llenos de documentación en formato duro.

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

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

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

En el caso del proceso de distribución de los módulos de materiales docentes, oficina, e insumos de limpieza, así como el control de su destino final:

- El histórico de la cantidad de recursos entregados a la facultad y asignados a los trabajadores y estudiantes se encuentran en archivos

impresos por años, lo que dificulta la toma de decisiones y la revisión del destino final.

- El proceso de búsqueda y actualización suele ser lento debido, a las grandes cantidades de información, la cual en ocasiones se encuentra duplicada.

Teniendo en cuenta la situación antes descrita se identifica como **problema de investigación**: ¿Cómo contribuir a la gestión de los recursos materiales y libros en la Facultad 4 de la Universidad de las Ciencias Informáticas? Se define como **objeto de estudio**: Los procesos de gestión de recursos materiales y libros.

Para dar solución al problema de investigación planteado, se define como **objetivo general**: Desarrollar un sistema informático que contribuya a la gestión de los recursos materiales y libros en la facultad 4 de la Universidad de las Ciencias Informáticas.

Dentro del objeto de estudio de la investigación se precisa como **campo de acción**: La gestión de los recursos materiales y libros en la facultad 4 de la Universidad de las Ciencias Informáticas.

Para lograr el cumplimiento del **objetivo general**, se derivan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación mediante un estudio, análisis e interpretación, del estado del arte de los procesos relacionados con los sistemas de control de medios materiales y las tecnologías existentes para el desarrollo de los mismos.
- Realizar el análisis y diseño de la aplicación, correspondiente con los requisitos identificados.
- Realizar la implementación del sistema mediante las herramientas y tecnologías seleccionadas.
- Validar la solución propuesta mediante pruebas de software y el cumplimiento de los objetivos de la investigación.

Los cuales, se cumplirán a través de las siguientes **tareas de la investigación** por **objetivo específico**:

### **Sobre objetivo 1:**

- Análisis de los procesos de negocio.
- Valoración de los sistemas de gestión información de medios materiales y libros, tanto en Cuba como en el mundo.
- Definición de las tecnologías a utilizar en el desarrollo de la solución de software propuesta.
- Selección de la metodología de desarrollo de software que guiará el proceso de implementación de la propuesta de solución.

### **Sobre objetivo 2:**

- Realización del levantamiento de requisitos del sistema a desarrollar.
- Realización del diseño del sistema a desarrollar.

### **Sobre objetivo 3:**

- Realización de la implementación del sistema a desarrollar.

### **Sobre objetivo 4:**

- Elaboración de los diseños de casos de pruebas del sistema desarrollado.
- Realización de las pruebas de unitarias al sistema desarrollado.
- Realización de las pruebas de aceptación al sistema desarrollado.

Para darle solución a los objetivos anteriormente mencionadas se utilizarán los siguientes **Métodos Científicos**:

#### **Método teórico:**

- Analítico – Sintético: En la realización del análisis de la información empleada para la investigación, identificando así, conceptos, definiciones y avances acerca de los sistemas de gestión de información existentes.
- Modelación: Se utiliza en la modelación de los diagramas dentro de la metodología de desarrollo de software seleccionada para llevar a cabo la solución.

- Histórico – Lógico: Para el estudio del desarrollo y evolución de los diferentes sistemas de gestión de información similares, nacionales e internacionales, así como las herramientas y tecnologías para el desarrollo del software, entre ellos los lenguajes de programación, *framework* de desarrollo, metodologías y herramientas CASE.

### **Método empírico:**

- Observación: se utilizó para estudiar más de cerca el objeto de la investigación ya que mediante este se puede observar el comportamiento de las herramientas existentes y extraer información de ellas. Identificar funcionalidades, ventajas y desventajas en las herramientas similares. [Anexo 6 Guía de observación]
- Entrevista: Permitió conocer las necesidades del cliente y obtener los requisitos del sistema a desarrollar. Se usó al inicio y durante todo el desarrollo del trabajo, pues a medida que se desarrolló el sistema, se analizó si el mismo, cumplía con las expectativas requeridas. [Anexo 5 Entrevista]

El presente trabajo de diploma cuenta con la siguiente estructura: Introducción, tres capítulos, conclusiones, recomendaciones, referencias las bibliográficas y por último los anexos. A continuación, se presenta un resumen de las diferentes temáticas que se abordan en los capítulos.

### **Capítulo 1: Fundamentación teórica**

En este capítulo se expone la fundamentación teórica del trabajo, incluyéndose en el mismo, el estudio y análisis de sistemas homólogos existentes en la actualidad. Además, se define la metodología de software a seguir durante el proceso de desarrollo de software, y las herramientas y tecnologías a utilizar para la implementación de la propuesta solución.

### **Capítulo 2: Características del sistema y propuesta solución**

Se realiza una descripción de la propuesta solución y se exponen los requisitos funcionales y no funcionales identificados. Teniendo en cuenta la metodología

de software seleccionada, se realiza el análisis y diseño del sistema además, de presentar los artefactos ingenieriles que documentan el proceso de desarrollo de software.

### **Capítulo 3: Validación de la propuesta solución**

En este capítulo se define la estrategia de pruebas a realizar para verificar el cumplimiento de los requisitos de software. Además, se presentan los resultados de los casos de prueba, obteniendo una valoración integral de la solución propuesta que permita determinar los puntos débiles del sistema.

## **CAPÍTULO 1 Fundamentación teórica**

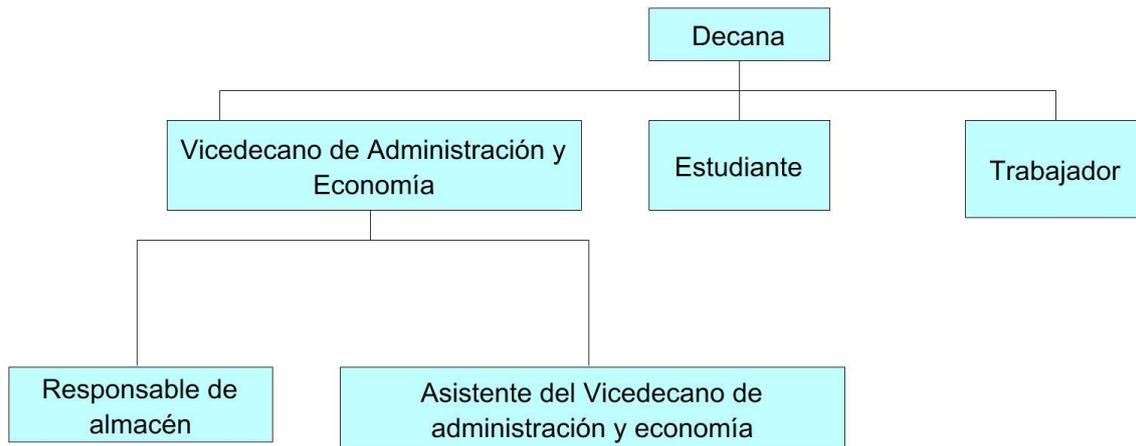
### Introducción

En el presente capítulo se describen los principales conceptos asociados al marco teórico para un mejor entendimiento del tema a tratar. Además, se realiza un estudio y análisis de sistemas similares con el objetivo determinar las características esenciales de los mismos. También se realiza una caracterización de las tecnologías, herramientas y lenguajes que fueron definidos como propuesta solución.

#### 1.1 Análisis de los procesos de negocio vinculados a la distribución y el control de los recursos materiales en la Facultad 4

Actualmente en la facultad aún existen procesos que no se realizan con la calidad requerida como son la entrega y devolución de libros al almacén, y la entrega de módulos de materiales docentes, oficina, e insumos de limpieza, así como el control del destino final de los mismos. Estos procesos son supervisados por el Vicedecanato de administración y economía, el cual tiene entre sus objetivos asignar, controlar, proteger y conservar los materiales que posee la facultad.

Las personas que intervienen en estos procesos son: decana de la facultad, el vicedecano de administración y economía, el asistente del vicedecano, el responsable de almacén, y los estudiantes y trabajadores de la facultad. En la figura que se muestra a continuación, se realiza un organigrama en el que se representa la estructura jerárquica de los roles que intervienen en la ejecución de estos procesos en la Facultad 4 de la Universidad de las Ciencias Informáticas. La descripción de los roles se encuentra en el [Anexo 1](#).



*Figura 1 Estructura jerárquica de la facultad teniendo en cuenta los roles que intervienen en los procesos de distribución y control de los recursos materiales y libros en la Facultad 4 (Elaboración propia)*

A continuación, se explicará el flujo de actividades de estos dos procesos:

A nivel UCI se envía una distribución de medios materiales a cada facultad, con los materiales disponibles que le corresponden a la misma. Estos materiales pueden ser material de limpieza, oficina, tecnología o mantenimiento. El vicedecano/a de administración y economía es el encargado de realizar un vale de solicitud de materiales en correspondencia de la cantidad de estudiantes y profesores que tiene su área.

El vale de solicitud de materiales debe ser firmado y autorizado por el decano/a de la facultad. Luego el vicedecano/a o una persona autorizada por el mismo debe dirigirse al almacén correspondiente, a sacar los materiales donde se le entregará un vale de salida, el cual debe ser firmado por el almacenero/a y la persona que recibe los materiales. Las copias de estos documentos son archivadas por el vicedecano/a como constancia de la transacción.

### **Entrega de medios materiales (módulos de materiales de docente, oficina e insumos de limpieza)**

Una vez recibido los medios materiales, el vicedecano/a junto con la decana/o de la facultad, en el consejo de dirección, conforman los módulos que se le entregarán a estudiantes o trabajadores. La persona responsable de entregar estos módulos es el asistente del vicedecano, el cual debe dejar una constancia de la entrega, en acta de destino final. Esta acta debe ser debidamente firmada

por la persona a la que se le entrega el módulo de materiales, y archivada por el asistente del vicedecano.

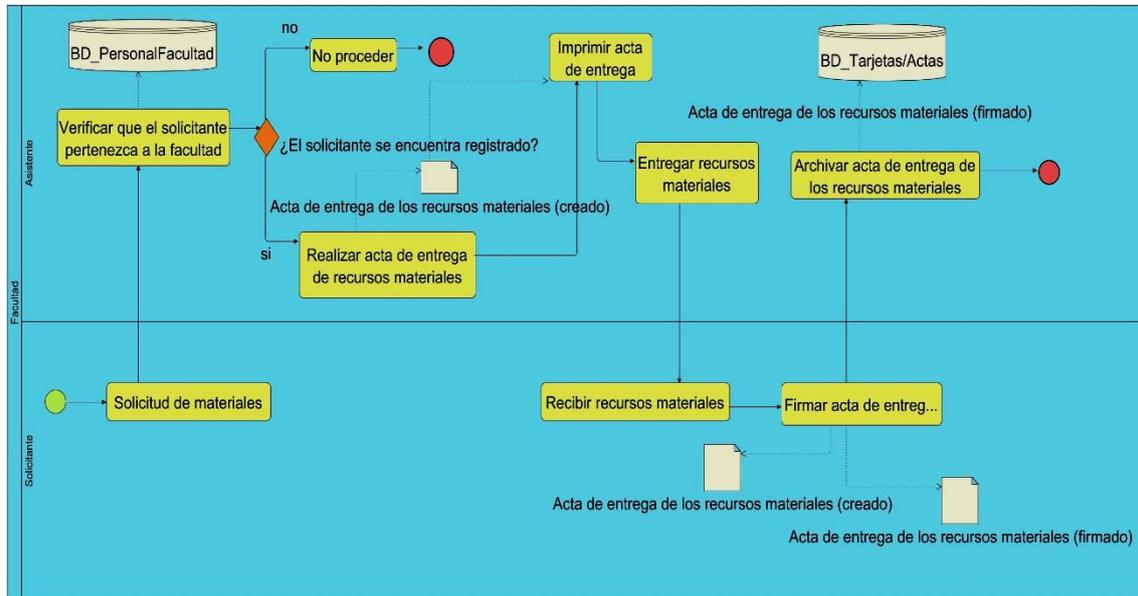


Figura 2 Descripción del subproceso: Entrega medios materiales (Elaboración propia)

### Entrega de libros a estudiantes y trabajadores

Para la entrega de libros, los solicitantes (estudiantes o trabajadores) se presentan en el almacén de libros de la facultad y solicitan los libros que desean. Una vez solicitados, el responsable del almacén debe verificar si la persona pertenece a la facultad o no. En caso de no pertenecer a la facultad se le notifica que no se puede proceder con la entrega de los libros. En caso contrario, se busca si el solicitante tiene creada una tarjeta de préstamo, si ya tiene una tarjeta, esta se actualiza, en caso contrario, se crea una nueva en la que se registran los siguientes datos: nombre y apellidos, facultad, año, número de solapín, título de los libros, fecha de entrega, firma de entrega, fecha de recibo, firma de recibo y en caso de ser trabajador, se le añade el cargo y el área al que pertenece. Una vez entregados los libros, el solicitante debe firmar la entrega. Estas tarjetas son archivadas en el almacén teniendo en cuenta el año y grupo de los estudiantes, y en caso de ser trabajador el área al que pertenece.

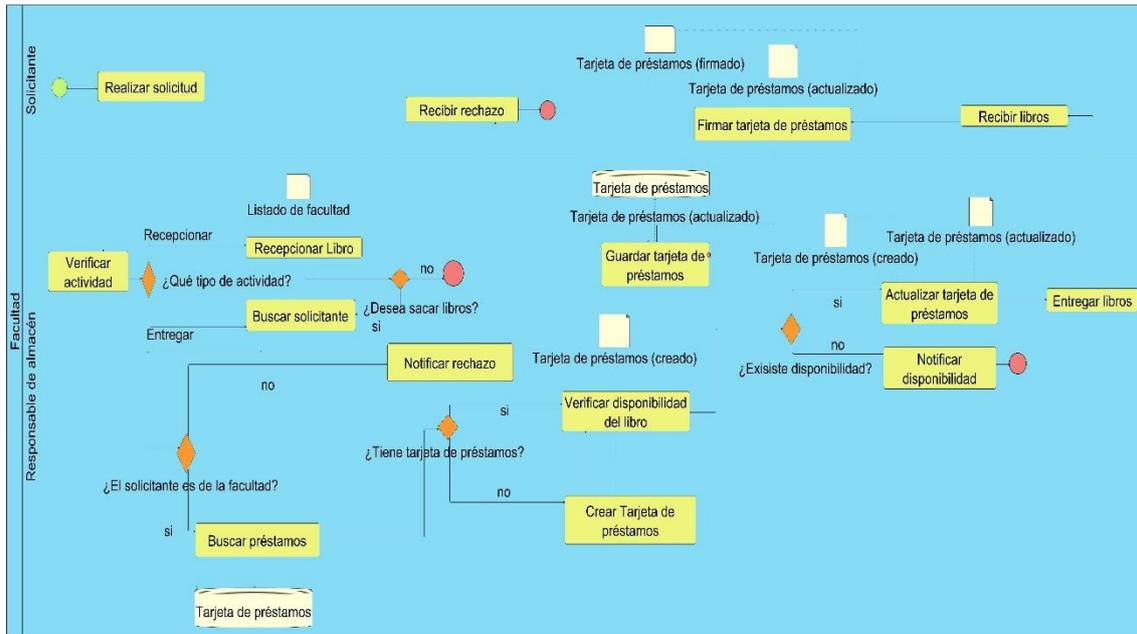


Figura 3 Descripción del subproceso entregar libros a estudiantes o trabajadores (Elaboración propia)

Hay que tener presente que también los estudiantes o trabajadores pueden hacer entrega de los libros que poseen. Por lo tanto, una misma persona puede solicitar libros y a su vez puede entregar. A continuación, se describe el proceso recepcionar libros.

### Recepción de libros a estudiantes y trabajadores

Para hacer una devolución de libros al almacén el estudiante o trabajador de presentarse al mismo con los libros que desea entregar. El responsable del almacén debe recepcionar los libros, actualizar la tarjeta de préstamo y el estudiante o trabajador debe firmar el recibo.

Si el estudiante o trabajador se encuentra en proceso de baja y les faltan libros entonces el responsable del almacén deberá llenarle el modelo de débito en el que se incluyen los siguientes datos: nombre y apellidos, número de solapín, grupo en el caso de que sea estudiante, o área si es un trabajador. Por cada libro no entregado, se deben registrar los siguientes datos: título del libro, precio, importe en cup y el importe total a cobrar en cup.

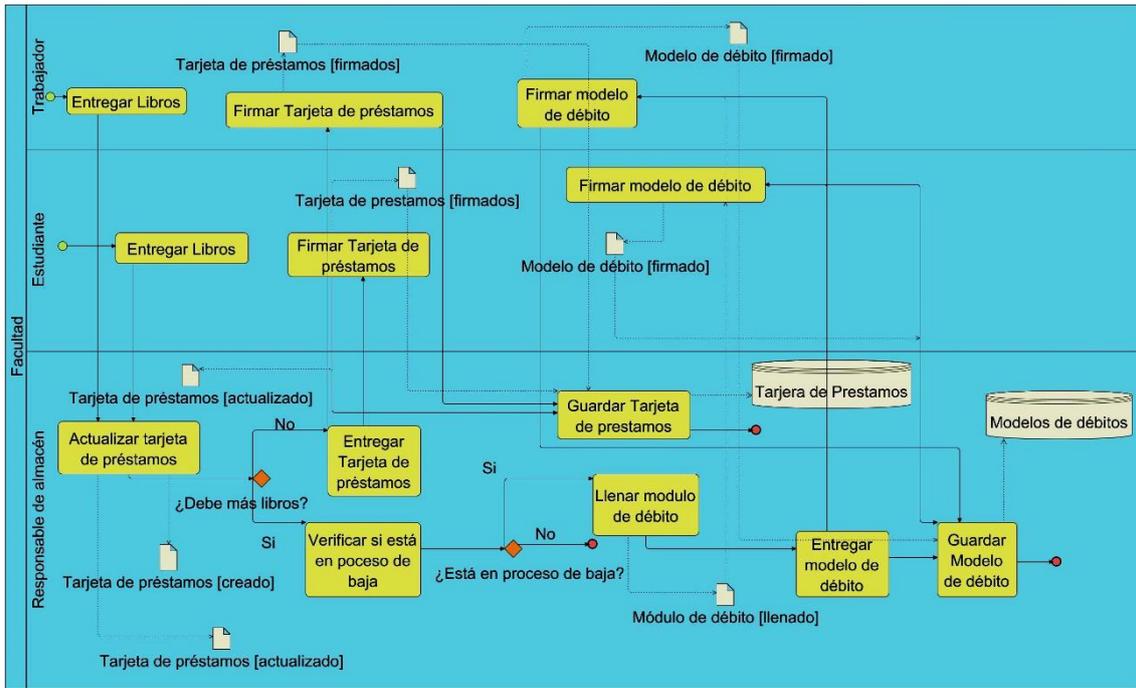


Figura 4 Descripción del subproceso: Recepcionar libros entregados a estudiantes y trabajadores (Elaboración propia)

### Control de libros en el almacén

El responsable del almacén debe comprobar las existencias reales (físicas) de los libros contra el listado de libros que se encuentran en el almacén. En este listado se encuentra una relación de la cantidad real de libros existentes en el almacén, la cantidad de libros prestados a estudiantes y trabajadores, y la cantidad de libros que hay actualmente en el almacén. En caso de existir diferencias, debe investigar las causas para corregir errores de conteo.

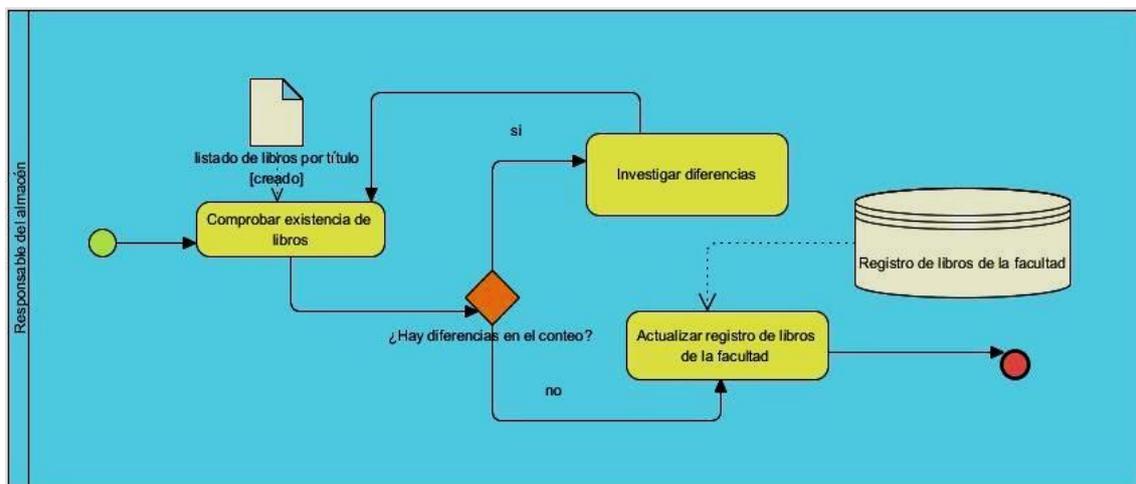


Figura 5 Descripción del subproceso: Controlar los libros en el almacén (Elaboración propia)

A partir del análisis de los procesos del negocio, se pudo determinar la existencia de varias deficiencias que afectan su ejecución en la facultad. A continuación, se hace mención de estas deficiencias:

- El histórico de la cantidad de recursos entregados a la facultad y asignados a los trabajadores y estudiantes se encuentran en archivos impresos por años, lo que dificulta la toma de decisiones y la revisión del destino final.
- Para el préstamo de libros, se realizan tarjetas de préstamos, que son archivadas de forma física. Esto puede provocar que exista duplicidad de la información, extravío de tarjetas de préstamos y pérdida de libros.
- Las actividades de creación y actualización de las tarjetas de préstamos, modelos de débitos y el registro de libros de la facultad se realizan de forma manual provocando demora en el proceso.
- La gestión de cantidades se realiza de forma manual, lo que dificulta la entrega de información. Por ejemplo, saber de un título en específico la cantidad total (existencia en almacén + préstamos), conocer cuántos estudiantes de un grupo o un año no han entregado los libros, o cuántos no han solicitado préstamos al almacén, requiere un tiempo considerable al encargado de almacén.
- Actualmente, el vicedecanato de administración y economía no utiliza ninguna herramienta capaz de gestionar y agilizar estos procesos.

Ante las deficiencias encontradas, es necesario realizar el estudio y análisis, de herramientas informáticas que se asemejen o den solución a los problemas anteriormente planteados. Durante el proceso de búsqueda se identificó que existe un tipo de sistema en particular que podría ser una posible solución a la situación anteriormente descrita.

Los sistemas de planificación de recursos empresariales por sus siglas ERP, se utilizan en las empresas para integrar y manejar procesos de diferentes áreas de una misma organización. Evidenciándose así, la informatización de procesos relacionados con la logística, distribución, inventario, envíos, facturas y la contabilidad (Alpert, 2022).

Destacar que la posible solución informática a desarrollar no es considerada un ERP, debido a que la misma estará enfocada a la mejora de algunos de los procesos que se realizan en el Vicedecanato de administración y economía. Es decir, no permitirá la integración e informatización de otros procesos que se realizan en diferentes áreas de la facultad, sin embargo, por las características y funcionalidades que tienen los módulos que la integran, estos tipos de sistemas se tienen en cuenta en el estudio y análisis de herramientas similares. A continuación, se exponen algunas de las soluciones existentes que facilitan la gestión de los recursos materiales y libros en el ámbito nacional e internacional, las cuales permitirán conocer las características de estos sistemas y el posible aporte a la propuesta de solución.

## 1.2 Análisis de los sistemas homólogos

El desarrollo de las tecnologías de la información y las comunicaciones ha permitido la incorporación de herramientas que apoyan los procesos de gestión administrativos. Para determinar las características que debe cumplir la aplicación que se desea desarrollar es necesario realizar un estudio y posterior análisis de los sistemas existentes tanto a nivel nacional e internacional.

Dicha investigación tuvo como resultado, la localización de varios sistemas existentes en universidades, tanto de Cuba como en el extranjero, y que tienen puntos en común con el sistema a desarrollar. Es necesario acotar que, en la investigación realizada no se encontró sistemas que se especialicen en la gestión de materiales y libros. A continuación, se realiza una descripción de estos referentes aportando las razones que avalan la necesidad de crear un nuevo sistema adaptado a las características y requisitos exigidos.

### 1.2.1 Soluciones internacionales

#### **KOHA**

Es un software libre y de código abierto de automatización de bibliotecas, desarrollado en arquitectura de cliente-servidor multiplataforma. Es un sistema basado en tecnologías web, flexible, que se adapta a requerimientos específicos de cada institución y además, no exige recursos avanzados de tecnología, por lo que se puede instalar en equipos con características mínimas de hardware, lo que hace de éste un sistema económico de acceso libre(Parkes,2011).

KOHA permite administrar la biblioteca en sus diferentes funciones como son: catalogación, autoridades, préstamo, devolución y renovación del material bibliográfico, procesamiento técnico, entre otras actividades diarias y necesarias para la biblioteca. A pesar de ello no constituye una solución al problema, porque es un software destinado al control de bibliotecas solamente, sin embargo, estas funcionalidades como catalogación de libros, préstamos y devolución de libros, son de gran relevancia para el negocio y por consiguiente van a ser tenidas en cuenta en la definición de los requisitos funcionales.

### **Integrado de Gestión Administrativa (SIGA)**

Es uno de los programas más recomendables para la gestión de almacenes e inventarios. Fue diseñada en España y ofrece un gran número de funcionalidades, como controlar el inventario, entradas y salidas de datos manuales y programadas, reubicaciones, entre otras. También dispone de módulos para comunicación con clientes o lector de códigos de barras.

Entre las ventajas de SIGA está el ahorro de tiempo, costes, y control de *stock*. Además, de ser un programa adaptable y personalizable a las necesidades de cada negocio. También es fácil de implantar y de actualizar, mejora la satisfacción de los operarios, ya que facilita sus tareas, minimiza los errores humanos y optimiza la reubicación de productos(Almacenes,2022).

Sin embargo, SIGA es un sistema *Desktop* de pago, que solo puede utilizarse en *Windows*. Es por ello que no constituye una solución, al problema de investigación. A pesar de esto, hay que tener en cuenta que SIGA presenta funcionalidades relacionadas con el control de recursos en los almacenes, control el inventario, las entradas y salidas de datos de manera manual y programadas, las cuales pueden ser definidos como requisitos funcionales en la propuesta de solución.

### **SoftExpert Almacén**

Es un software enfocado en la gestión de stock, depósitos e inventario. Posibilita a los usuarios el rápido y preciso acceso a las informaciones del historial de los eventos realizados en los locales de almacenamiento de materiales, sus entradas y salidas de datos, y alertas de niveles por debajo de lo ideal. La

solución fue desarrollada especialmente para agregar valor a los procesos dirigidos a la atención de las demandas de mantenimiento y manufactura, haciendo que los materiales y piezas correctas estén disponibles en los locales correctos, siempre que sea necesario(Brandao,2022).

Los informes del sistema permiten una visión detallada y precisa de los niveles de *stock*, tasas de uso y costos de transporte. El sistema cuenta con herramientas de monitorización y control de los niveles de *stock* y de automatización de las órdenes de compra, a fin de optimizar la disponibilidad de *ítems* para las tareas futuras, reducir los saldos de *stock* en exceso y eventuales costos relacionados a los procesos(Brandao,2022).

A pesar de no constituir una solución al problema porque es un software de pago y privativo, que solo puede desplegarse en la nube o en un Servidor Cliente (On Premise), SoftExpert Almacén cuenta con características importantes como: acceso seguro al contenido corporativo, integración con otras aplicaciones, aumento de colaboración entre los miembros de la empresa, acceso al contenido en tiempo real, entre otras características importantes, las cuales serán tenidas en cuenta a la hora de definir los requisitos funcionales.

### 1.2.2 Soluciones nacionales

#### **Biblioteca Visual**

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

El sistema informático organiza el trabajo atendiendo a las diferentes áreas que cuenta la biblioteca, facilitando así un mayor control para acceder a la información, permite un preciso control estadístico, cuenta con facilidades de búsquedas y una interfaz fácil de utilizar por cualquier usuario. Es importante destacar que gracias a este sistema se definen correctamente algunas áreas que no existían como es el caso del procesamiento estadístico(Iglesias,2010).

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

### **Versat-Sarasola**

Es desarrollado por la casa de software TEICO Villa Clara. Está programado en Delphi y SQL Server 2000. Ofrece al usuario la posibilidad de contar con un instrumento seguro, rápido, eficaz y de fácil manejo para la Planificación, Control y el Análisis de Gestión Económica(Hartman,2021).

Su nombre viene de la palabra Versatilidad (Versat) y Sarasola del apellido de un eminente contador cubano. Está implantado en más de 2500 unidades presupuestadas del país, entre las que figuran organismos de la Administración Central del Estado, las direcciones municipales de finanzas, tesorerías, la ONAT y otros.

El Versat cuenta con los módulos de Configuración, Contabilidad general y Costos, como elementos básicos e imprescindibles del paquete. Además, contiene los módulos adicionales de Finanzas, Inventario, Activos fijos y Facturación. Proporciona reportes y mensajes que alertan sobre problemas y otros aspectos de las actividades financieras, de inventarios, activos fijos, etc.

En resumen, VERSAT-Sarasola es un sistema concebido para su instalación en red, por las posibilidades que esta tecnología brinda para el trabajo en un entorno multiusuarios, aún más en sistemas económicos. Garantiza seguridad y confiabilidad en el procesamiento de los datos en tiempo real, de tal forma que las administraciones puedan utilizar la información con fines gerenciales.

A pesar de ello no constituye una solución al problema debido a que es un sistema de pago dedicado a análisis de gestión económica, sin embargo, cuenta con características importantes como son: entorno multiusuario, generación de reportes, seguridad y correcto manejo de la información; las cuales deben ser tenidas en cuenta en la definición de los requisitos funcionales.

## **ARCE**

Proyecto ARCE (Aplicación de Red para Casos de Emergencia), es una aplicación dirigida a la protección civil que tipifica el intercambio de los recursos y medios, consultas, apoyos logísticos, entre otras gestiones. Este es un proyecto de colaboración iberoamericana a la que pertenecen Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Cuba, Ecuador, El salvador, España, Guatemala, Honduras, México, Nicaragua, Venezuela, entre otros países(Escárez San Martín, 2008).

El objetivo del catálogo es conocer, organizar y localizar en forma rápida y eficaz los medios y recursos. La codificación de la catalogación está definida por ARCE, esto permite obtener informes, controlar y gestionar la base de datos. Entre otros datos de los recursos y medios se controlan titularidad, localización y sector. El mayor inconveniente de esta aplicación es que se debe realizar una inversión de más de \$ 200.000 dólares para poder adquirirla.

Arce permite una adecuada catalogación, gestión de los inventarios de medios y recursos y organiza la información de forma tal que el acceso a ella sea rápido. Sin embargo, no constituye una solución al problema, porque es un software de pago destinado a el intercambio de recursos y medios, consultas, apoyos logísticos, y otras gestiones provocadas por una situación de emergencia, y no cuenta con funcionalidades destinadas a la gestión, pero cuenta con características importantes como: gestión de inventarios de medios materiales y catalogación de los mismos y envío de modo automatizado correos electrónicos a cada uno de los titulares seleccionados; las cuales deben ser tenidas en cuenta en la definición de los requisitos funcionales.

Las soluciones informáticas analizadas y estudiadas en este capítulo, responden a las necesidades de las entidades para las que fueron desarrolladas, por lo que no son lo suficientemente adaptables para satisfacer las necesidades planteadas en la problemática de esta investigación, además, algunas de ellas son privativas o emplean tecnologías privadas y la forma de controlar los medios no se ajusta a las necesidades de la facultad, por lo que para poder implementar una propuesta solución es necesario el estudio de varias herramientas dentro de las que se fueron seleccionadas las siguientes:

### 1.3 Herramientas y tecnologías a utilizar

#### 1.3.1 Lenguaje de programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones, y es utilizado para controlar el comportamiento físico y lógico de una máquina. Existen diferentes tipos de lenguajes de programación como son: los lenguajes de bajo nivel (máquina y ensamblador) y los lenguajes de alto nivel (C++, C#, Java, *JavaScript*, Python), los cuales están diseñados dependiendo del bien informático que se pretenda elaborar y de las funciones que se quieran establecer(Olarte,2018). A continuación, se hace un estudio de tres lenguajes: *Python*, *Java* y *JavaScript*, con el objetivo de obtener el que más se ajusta a los requerimientos del desarrollo del sistema.

Python es uno de los lenguajes de programación más rápidos, ya que requiere muy pocas líneas de código. Su énfasis está en la legibilidad y la simplicidad, lo que lo convierte en una excelente opción para principiantes. Ofrece soporte para módulos y paquetes, lo que permite la modularidad del sistema y la reutilización de código. Además, contiene estructuras de datos integradas, combinadas junto a enlaces dinámicos y tipeo, lo que lo convierte en una opción ideal para el desarrollo rápido de aplicaciones(Corredera,2020).

Java es un lenguaje de programación multiplataforma, orientado a objetos y centrado en la red. Es uno de los lenguajes de programación más utilizados para crear aplicaciones del lado del cliente y del lado del servidor. Ofrece una gran cantidad de documentación detallada disponible, un código robusto y un manejo automático de la memoria, lo que reduce el número de errores(Coursera,2022).

*JavaScript* es un robusto lenguaje de programación que se puede aplicar a un documento HTML y usarse para crear interactividad dinámica en los sitios web. Es un lenguaje bastante compacto, aunque muy flexible, con una curva de aprendizaje alta. Es empleado principalmente para el desarrollo de *front end*, para ser ejecutado por el usuario, es decir, quien está navegando e interactuando con una página web determinada(Academia,2022).

Tabla 1 Comparación de los lenguajes de programación Python, Java y JavaScript

Parámetro	Java	Python	JavaScript
-----------	------	--------	------------

<b>Compilación</b>	Es un lenguaje compilado.	Es un lenguaje interpretado	Es un lenguaje interpretado
<b>Tipado (Estático o dinámico)</b>	Estático.	Dinámico.	Dinámico.
<b>Curva de aprendizaje</b>	Curva de aprendizaje compleja	Fácil de aprender.	Fácil de aprender.
<b>Soporte de base datos</b>	Conectividad estable	Conectividad débil	Conectividad débil
<b>Portabilidad</b>	Cualquier computadora o dispositivo móvil que pueda ejecutar la máquina virtual Java.	Los programas python necesitan un intérprete instalado en la máquina de destino para traducir el código. En comparación con Java, es menos portable.	Cualquier computadora o dispositivo móvil que tenga un navegador.
<b>Mejores características</b>	Grandes bibliotecas. Gran cantidad de documentación disponible. Excelentes herramientas	Código legible, Desarrollo rápido.	Desarrollo rápido, Resultados atractivos con poco código y esfuerzo.
<b>El mejor uso para</b>	Es el mejor para aplicaciones GUI de escritorio, sistemas de incrustación, servicios de	Es excelente para conmutación científica y numérica, aplicaciones de	Está pensado para hacer aplicaciones menos pesadas que java, principalmente

	aplicaciones web, etc.	aprendizaje automático.	orientadas a la web.
<b>Frameworks</b>	Spring Hibernate Vaadin JavaServer Faces(JSF) Grails	<i>Django</i> <i>Pyramid</i> <i>Pylons</i> <i>Flask</i> <i>Bottle</i>	<i>Angular</i> <i>Ember</i> <i>Backbone</i> <i>React</i> <i>Vue.js</i> <i>Express.js</i>

Al concluir con el estudio de algunos de los lenguajes de programación, se seleccionó el lenguaje Java para la implementación de la propuesta solución, debido a que es un lenguaje de alto nivel, gratuito, con un gran número de librerías de código abierto, con una amplia comunidad de desarrolladores ayudan tanto a los principiantes con sus primeras dudas, como a los expertos basándose en experiencias anteriores de otros desarrolladores *seniors*.

Puede ejecutarse en cualquier computadora o dispositivo móvil que pueda ejecutar la máquina virtual Java, ofrece conectividad estable, código robusto, y una amplia documentación detallada disponible. El hecho de que las aplicaciones se desplieguen, por lo general, en servidores dedicados hace que puedan ser fácilmente mantenidas y supervisadas por administradores de servidores experimentados. Las brechas de seguridad de las aplicaciones Java son menores. A continuación, se realiza una comparación entre los principales *frameworks* que utilizan este lenguaje.

### 1.3.2 Frameworks

Un *framework* es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos. Sirve para acometer un proyecto en menos tiempo, haciendo uso de herramientas y módulos. (Noa,2021).

**Vaadin** es un framework de desarrollo web Java, diseñado para hacer fácil la creación y el mantenimiento de interfaces web de usuario de alta calidad. Reduce los tiempos de desarrollo y el número de errores en aplicaciones web. Es una

tecnología de código abierto ya madura y contrastada. Muchas empresas ya lo utilizan en el desarrollo de sus aplicaciones web por las ventajas que proporciona tanto en costes de programación como en la experiencia de uso. Logra una interfaz de usuario de escritorio tradicional, con buena usabilidad.(Abellán,2019)

**Spring** es un *framework* de código abierto para la plataforma Java, es seguramente el más popular y utilizado del mundo. Se basa en la arquitectura MVC (Modelo-Vista-Controlador), además, tiene buenas características de seguridad que puede llamar como funciones ya escritas. Esto hace que procesos como la autenticación, la verificación y la validación sean mucho más fáciles de incluir (adecuadamente) en cualquier proyecto(Muradas,2018).

**JSF** (Java Server Faces) es el marco estándar de interfaz de usuario orientada a componentes para la plataforma Java EE (*Enterprise Edition*). Es un *framework* de desarrollo web basado en Java que establece el estándar para la construcción de interfaces de usuario del lado del servidor, establecidas en el patrón MVC (Modelo, Vista, Controlador). JSF permite desarrollar rápidamente aplicaciones de negocio dinámicas en las que toda la lógica de negocio se implementa en Java, o es llamada desde Java, creando páginas para las vistas muy sencillas(Dailyrazor, 2016).

Tabla 2 Comparación entre los frameworks de desarrollo

	Vaadin	JSF
<b>Habilidades requeridas</b>	Java	Java
<b>Documentación</b>	Cuenta con una amplia documentación actualizada.	Cuenta con una amplia documentación.
<b>Arquitectura</b>	Presenta una arquitectura centrada en el servidor, lo que implica que la mayoría de la lógica es ejecutada en los sevrvidores remotos.	Se basa en la arquitectura MVC (Modelo-Vista-Controlador)

<b>Herramientas y bibliotecas</b>	Cuenta con un <i>modelo basado en componentes</i> por el que se pueden seguir añadiendo de manera rápida más características o funciones al producto	Dos bibliotecas de etiquetas personalizadas para Java Server Pages que permiten expresar una interfaz Java Server Faces dentro de una página JSP.
<b>Aplicaciones</b>	Logra una interfaz de usuario de escritorio tradicional, con buena usabilidad.	Permite construir aplicaciones Web que implementan una separación entre el comportamiento y la presentación tradicionalmente ofrecidas por arquitectura UI del lado del cliente. JSF se hace fácil de usar al aislar al desarrollador del API de Servlet.

Teniendo en cuenta el análisis realizado anteriormente se seleccionó como marco de trabajo para el desarrollo de la propuesta solución, Vaadin el cual se utilizará para el desarrollo de las interfaces de usuario. Vaadin soporta la comunicación entre el cliente y el servidor, y gestiona las sesiones del navegador con lo que el programador se abstrae de esa parte y se centra en lo fundamental, la lógica. Además, cuenta con un modelo basado en componentes por el que se pueden seguir añadiendo de manera rápida más características o funciones al producto. También se utilizará el marco de trabajo del lado del servidor Spring, debido a que se encuentra estrechamente vinculado con el *framework* Vaadin.

### 1.3.3 Servidores de aplicación web

Un servidor web permite procesar aplicaciones de este tipo del lado del servidor, donde se establece una conexión bidireccional entre el cliente que realiza peticiones y el servidor que se encarga de dar respuestas o ejecutarlas. Para Java existen varios servidores, tales como: GlassFish, JBoss y Apache Tomcat.

**JBoss:** Este servidor actualmente es el que lidera el mercado y al igual que GlassFish posee un certificado de compatibilidad de Java EE, permite el desarrollo de aplicaciones web para sistemas empresariales y servicios de SOA (Arquitectura Orientada a Servicios) y está pensado para la obtención del máximo rendimiento y productividad(Jethva, 2022).

**GlassFish:** Es un servidor de aplicaciones web basado en código abierto y asistencia gratuita, fue creado inicialmente por Sun Microsystems y luego adquirido por ORACLE. Posee arquitectura modular basada en OSGi (Open Services Gateway Initiative), es multiplataforma y posee un certificado de compatibilidad con Java EE(Eclipse, 2022).

**Apache Tomcat 9.0.58:** Es un servidor de aplicaciones web basado en código abierto para tecnologías Java Servlet 3.0 y Java Server Pages 2.2 (JSP). Está publicado bajo licencia Apache versión 2. Posee una comunidad de desarrollo muy selectiva, utiliza Catalina como contenedor de servlets a partir de la versión 4. Es multiplataforma, siendo compatible con todo sistema operativo que posea instalada la máquina virtual de Java. Permite el monitoreo y la administración remota de las aplicaciones mediante una conexión segura, servicios de virtual hosting, clustering y equilibrado de cargas en las aplicaciones(Apache, 2022).

JBoss y Glassfish son básicamente servidores de aplicaciones Java EE completos, mientras que Tomcat es solo un contenedor de Servlet. La principal diferencia entre JBoss, Glassfish pero también WebSphere, WebLogic, etc. con respecto a Tomcat pero también a Jetty, estaba en la funcionalidad que ofrece un servidor de aplicaciones completo. Se decide optar por Apache Tomcat debido a que está recomendado para Vaadin.

#### 1.3.4 Entorno de desarrollo integrado (IDE)

El entorno de desarrollo integrado (IDE, *Integrated Development Environment*) son los programas o interfaces visuales de programación que se utilizan para interactuar con el compilador y depurador de un lenguaje de programación. A continuación, se realiza una comparación de algunos entornos de desarrollo integrado que han sido desarrollados para *Java*:

IDE	Visual Studio Code	IntelliJ IDEA	NetBeans
<b>Características</b>	<p>Tiene una amplia variedad de themes, para configurar el entorno de trabajo.</p> <p>Soporte para múltiples lenguajes de programación.</p> <p>Soporte multiplataforma.</p> <p>Tiene un rico ecosistema de extensiones.</p> <p>Multiproyectos.</p>	<p>Análisis de código</p> <p>Editor potente</p> <p>Herramientas Android integradas</p> <p>Productividad</p> <p>Brinda soporte para Maven y Gradle.</p> <p>Posee herramientas integradas para pruebas unitarias y de cobertura.</p> <p>Soporta muchos lenguajes.</p>	<p>Orientado principalmente a las apps de Java, ofrece diferentes herramientas digitales como editor de texto, código, compilador, interfaz gráfica de usuario; además, de un depurador.</p>
<b>Ventajas</b>	<p>Viene con soporte web incorporado.</p> <p>Mejora del código.</p> <p>Soporte de terminal.</p> <p>Cuenta con extensiones que te ayudaran a trabajar aún más rápido el lenguaje de programación de tu preferencia.</p>	<p>Integración con sistemas de control de versiones.</p> <p>Sensación de fiabilidad y robustez muy superior a otros entornos</p> <p>Herramientas de refactorización de extremadamente inteligentes.</p>	<p>Lenguaje Multiplataforma</p> <p>Manejo automático de la memoria</p> <p>Es gratis</p> <p>Desarrolla aplicaciones web dinámicas</p> <p>Desarrollar aplicaciones de servidor para foros en línea, almacenes,</p>

			encuestas, procesamiento de formularios HTML, etc.
<b>Desventajas</b>	Su interfaz de usuario es intimidante para los nuevos usuarios. Alto consumo de memoria RAM, debido a la cantidad de extensiones utilizadas.	Es de pago. Consume gran cantidad de memoria y CPU mientras se ejecuta. Sus archivos de instalación son más grandes que los de otros IDE.	Lentitud a la hora de ejecutar las aplicaciones. Requiere un intérprete. Algunas implementaciones y librerías pueden tener código rebuscado. Algunas herramientas tienen un costo adicional.

Se decide usar el editor de texto **Visual Studio Code (VS)**, debido a que es un editor de código libre (gratuito) y de código abierto. Es ligero pero potente. Ayuda a depurar y corregir el código utilizando el método *intelli-sense*. Además, viene con soporte web incorporado, soporte multiplataforma y multiproyectos.

### 1.3.5 Sistema gestor de base de datos

Un sistema de gestión de base de datos, constituye un conjunto de programas a través de los cuales se puede acceder, administrar y gestionar la información almacenada en una base de datos, que no es más que un conjunto de información que pertenecen a un mismo contexto que se encuentran almacenadas de forma sistemática y que pueden ser utilizadas posteriormente (Acero, 2018). A continuación, se realiza un análisis de las ventajas y desventajas de algunos sistemas de gestión de base de datos:

Tabla 3 Comparación de sistemas gestores de base de datos

	<i>PostgreSQL</i>	<i>DB2</i>	<i>MySQL</i>	<i>Oracle</i>
<b>Ventajas</b>	Fácil de administrar. Multiplataforma. Escalable. Realiza copias de seguridad en línea.	Permite tablas replicadas. Soporta todo tipos de datos.	<i>Open Source</i> . Velocidad al realizar las transacciones. Usa la licencia GPL.	Multiplataforma Monitorea en vivo los procesos. Presenta varias opciones al realizar un <i>BackUp</i> . Fácil interacción con el sistema.
<b>Desventajas</b>	Es vulnerable sin protección adecuada.	Lentitud al crear y ejecutar transacciones. Demasiado consumo de memoria RAM. Costo elevado en licencias personales.	Varias utilidades no están documentadas.	Sintaxis más difícil de entender. Inhabilidad de implementar el procesamiento recursivo.

Analizadas estas ventajas y desventajas se escoge a PostgreSQL, debido a su facilidad de administración y aprendizaje. Presenta sintaxis intuitiva y la capacidad de realizar copias de seguridad en línea en comparación con otros sistemas de gestión de bases de datos. Además, es multiplataforma y escalable, lo cual es ideal para el desarrollo web.

### 1.3.6 Control de versiones

El control de versiones, también conocido como “control de código fuente”, es la práctica de rastrear y gestionar los cambios en el código de software. Los sistemas de control de versiones, son herramientas que ayudan a los equipos de desarrollo de software a gestionar los cambios en el código fuente. Si se comete un error, los desarrolladores pueden ir hacia atrás en el tiempo y comprobar las versiones anteriores del código para ayudar a resolver el error(Atlassian,2022).

Tabla 4 Sistemas de control de versiones más populares - ventajas y desventajas

Sistema	Ventajas	Desventajas
<b>CVS-Sistema de versiones concurrentes</b>	Vigente durante mucho tiempo por lo que se considera una tecnología madura.	No se puede mover ni renombrar archivos en una nueva actualización. Existen riesgos de seguridad de enlaces simbólicos a archivos. No existe soporte para operaciones atómicas. No está diseñado para trabajos distribuidos.
<b>SVN-Apache Subversión</b>	Mejoras basadas de CVS. Soporta operaciones atómicas. Operaciones distribuidas. Amplia gama de plugins para IDEs. No utiliza modelo de igual a igual.	Aún existen errores cuando se renombra archivos. Insuficientes comandos de administración de repositorios. Es lenta en comparación a otras herramientas.
<b>Git/GitHub</b>	<i>OpenSource.</i> Trabaja íntegramente con muchos IDEs. Incremento de velocidad de operación. Operaciones de sucursales baratas ( <i>Cheap Branch operatios</i> ).	Es muy difícil de aprender para quienes trabajan con SVN. No está hecho para desarrolladores individuales. Soporte limitado para <i>Windows</i> .

	<p>Árbol de historial completo disponible sin conexión.</p> <p>Modelo distribuido de igual a igual.</p>	
<b>Mercurial</b>	<p>Más sencillo de aprender que Git.</p> <p>Tiene mejor documentación.</p> <p>Maneja un modelo distribuido.</p>	<p>Basado en extensiones en lugar de scriptability.</p> <p>Menos potencia fuera de la caja.</p>

A partir de la comparación de la información obtenida en la tabla anterior, se selecciona como herramienta de control de versiones, para controlar el proceso de implementación de la propuesta solución las herramientas Git/GitHub. Debido a que Git es el software de control de versiones a nivel local más utilizado en la actualidad. Además, cuenta con un alto rendimiento y eficiencia, y los cambios de código se pueden rastrear de manera muy fácil y clara.

Ofrece una increíble utilidad de línea de comandos conocida como git bash. También ofrece GUI GIT donde puede volver a escanear, cambiar de estado, cerrar sesión, confirmar y enviar el código rápidamente con solo unos pocos clics(Atlassian, 2022). Entre otras muchas cosas, Git te permite subir y actualizar tu código a la nube de GitHub. De esta forma siempre puedes disponer de él cuando lo necesites. Pero, además, se puede: realizar comparaciones entre versiones de una aplicación, observar la evolución del proyecto con el paso del tiempo, contar con una copia del código fuente para poder volver atrás ante cualquier imprevisto en la página web, tener una copia de seguridad del proyecto al completo, y disponer de un historial en el que se detallen las modificaciones realizadas en el código del sistema a desarrollar.

## 1.4 Metodologías de desarrollo de software y herramientas para el modelado

### 1.4.1 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. Se utilizan en el ámbito de la programación, entre otros, con el objetivo de trabajar en equipo de manera organizada. Estas metodologías han ido evolucionando a lo largo del tiempo, pasando de ser un mero trámite de organización a ser una base importantísima a la hora de desarrollar software de una manera productiva y eficaz(Maida,2018).

En la actualidad existen una gran cantidad de metodologías para el desarrollo de software, separadas en dos grandes grupos; las metodologías tradicionales o pesadas y las metodologías ágiles. Las metodologías tradicionales se basan en las buenas prácticas dentro de la ingeniería del software, siguiendo un marco de disciplina estricto y un riguroso proceso de aplicación. Las metodologías ágiles, en cambio, representan una solución a los problemas que requieren una respuesta rápida en un ambiente flexible y con cambios constantes, haciendo caso omiso de la documentación rigurosa y los métodos formales.

*Tabla 5 Comparación entre las metodologías*

Metodología Tradicional	Metodología Ágil
El trabajo se desarrolla en grupos numerosos de personas en ocasiones de forma distribuida.	El trabajo se enfoca a grupos de pocas personas con experiencia en los roles que desempeñan.
El cliente no forma parte del equipo de desarrollo, solo está presente a través de reuniones planificadas.	El cliente si forma parte del equipo y del proceso de desarrollo con reuniones frecuentes e incluso informales.
Cierta resistencia a los cambios	Pueden aparecer cambios en el proceso de desarrollo y se toman si es necesario.
Los proyectos presentan gran cantidad de documentación y de forma detallada	La documentación se genera a medida que se hace necesaria y de

	manera general no es amplia ni rigurosa en su estructura.
--	---

Se seguirá una metodología ágil, con el fin de satisfacer al cliente, mediante la entrega temprana de proyectos que tengan valor, aceptando que los requisitos técnicos del cliente cambien, incluso en etapas de desarrollo más avanzadas. Al trabajar con plazos y entregas parciales, es más sencillo disminuir los tiempos para entregar una mejor versión del producto. Además, las interacciones continuas entre los desarrolladores y el cliente, van a generar una retroalimentación bastante valiosa para trabajar en la calidad de los productos (da Silva, 2020).

En la actualidad existen un sin número de metodologías ágiles, unas más empleadas que otras, cada una aportando al desarrollo ágil distintos métodos que ayudan a mejorar de una manera eficaz la calidad del software. Entre las metodologías ágiles más utilizadas se encuentran las siguientes:

La metodología de **Programación Extrema** (*Extreme Programming*) o XP como también se le conoce, es una metodología centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo (González, 2022).

Se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas, y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (González, 2022).

**SCRUM.** Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son

las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración(Pressman, 2000).

**Dynamic Systems Development Method (DSDM).** Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además, de existir realimentación a todas las fases(Pressman, 2000).

**Adaptive Software Development (ASD).** Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo(Schwaber & Beedle, 2001).

**Feature-Driven Development (FDD).** Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad(Coad & Lefebvre, 1999).

Tabla 6 Comparación de metodologías ágiles

Características	XP	SCRUM	DSDM	FDD	ASD
<b>Enfoque</b>	Iterativo, incremental.	Iterativo, incremental.	Iterativo.	Iterativo.	Iterativo, incremental.
<b>Período del ciclo de iteración</b>	1 - 6 semanas.	2 - 4 semanas.	En 20% por ciento del tiempo total	2 días - 2 semanas.	4 - 8 semanas.

			80% de producto.		
<b>Tamaño adecuado del proyecto y complejidad</b>	Proyecto pequeño y sencillo.	Para proyectos grandes y complejos.	Todo tipo de proyectos	Proyectos a gran escala	Proyectos más pequeños y sencillos
<b>Involucramiento del usuario</b>	Activamente involucrado.	A través del propietario del producto.	A través de lanzamientos frecuentes	A través de informes	A través de lanzamientos frecuentes
<b>Documentación</b>	Documentación básica.	Documentación básica.	Más que XP y SCRUM	Más alto entre todos	Documentación básica
<b>Principales prácticas</b>	Simplicidad, Programación en pares.	Reuniones de SCRUM.	Time boxing, MoSCoW, Prototipado	Modelado de objetos, desarrollo por característica, uso del diagrama UML	Time boxing, Risk Driven, basado en características
<b>Desarrollo de características concurrentes</b>	Posible	Posible	Posible	Posible	Posible

En el momento de seleccionar una metodología para aplicar en la construcción de un sistema, es necesario tener en cuenta las características del proyecto y del equipo de desarrollo, así como la capacidad de adaptación de la misma a las necesidades del proyecto. Teniendo en cuenta todos los elementos de las comparaciones anteriores, se define como propuesta de solución enmarcarse en una metodología ágil, específicamente en la metodología de Programación Extrema.

XP al igual que otras metodologías ágiles, tiene un ciclo de vida dinámico, para ello se desarrolla en ciclos cortos o iteraciones. En cada una de las iteraciones se crea un ciclo completo de diseño de pruebas, análisis y desarrollo. Facilita los

cambios, origina una programación muy organizada y la satisfacción del programador.

Es muy eficiente durante el proceso de pruebas y planificación, su tasa de error es muy pequeña. También se puede aplicar a cualquier lenguaje de programación, y el cliente siempre tiene el control sobre las prioridades. Durante todo el proyecto se pueden realizar diversas pruebas y, sobre todo, permite ahorrar mucho tiempo y dinero.

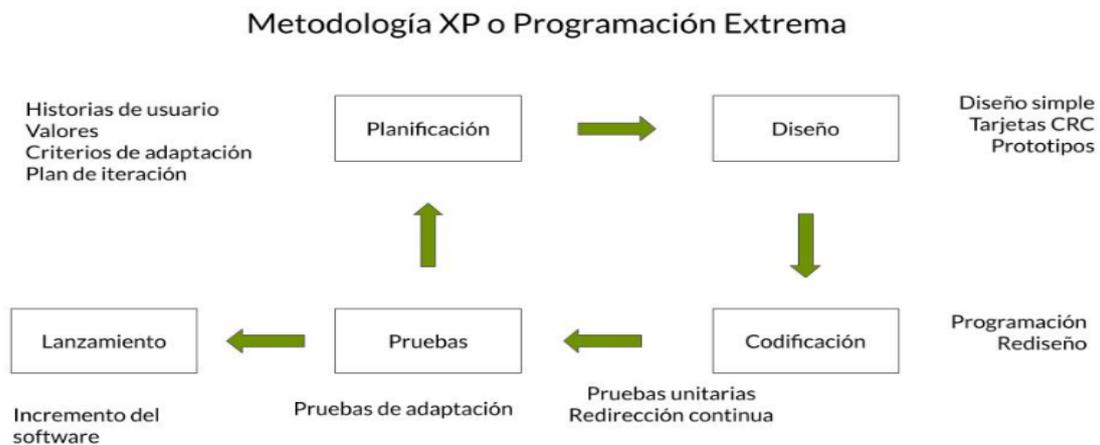


Figura 6 Metodología XP Fases (Sinnaps, 2020)

### Fase 1: Planificación

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.

Según la identificación de las historias de usuario, se priorizan y se descomponen en mini versiones. La planificación se va a ir revisando cada dos semanas aproximadamente de iteración, se debe obtener un software útil, funcional, listo para probar y lanzar( González,2022).

### Fase 2: Diseño

En este paso se intentará trabajar con un código sencillo, haciendo lo mínimo imprescindible para que funcione. Se obtendrá el prototipo. Además, para el diseño del software orientado a objetos, se crearán tarjetas CRC (Clase-Responsabilidad-Colaboración)(González,2022).

### Fase 3: Codificación

La programación aquí se hace a dos manos, en parejas en frente del mismo ordenador. Incluso a veces se intercambian las parejas. De esta forma, nos aseguramos que se realice un código más universal, con el que cualquier otro programador podría trabajar y entender. Y es que debe parecer que ha sido realizado por una única persona. Así se conseguirá una programación organizada y planificada(González,2022).

### Fase 4: Pruebas

Se deben realizar pruebas automáticas continuamente. Al tratarse de proyectos a corto plazo, este testeo automatizado y constante es clave, además, el propio cliente puede hacer pruebas, proponer nuevas pruebas e ir valorando las mini versiones(González,2022).

### Fase 5: Lanzamiento

Si hemos llegado hasta este punto, significa que hemos probado todas las historias de usuario o mini versiones con éxito. Ajustándonos a los requerimientos del cliente. Tenemos un software útil y podemos incorporarlo en el producto(González,2022).

Como se puede apreciar la metodología XP solo puede emplear para proyectos a corto plazo, es muy eficiente durante el proceso de pruebas y planificación, su tasa de error es muy pequeña. La planificación es flexible, y el flujo de trabajo sin presiones, garantizando así que el equipo de trabajo se encuentre dedicado y entusiasmado con el desarrollo del proyecto.

Además, esta metodología facilita los cambios, origina una programación muy organizada y la satisfacción del programador, además, de fomentar la comunicación entre los desarrolladores y los clientes. También se puede aplicar cualquier lenguaje de programación, el cliente tiene el control sobre las prioridades, durante todo el proyecto se pueden realizar diversas pruebas y sobre todo permite ahorrar tiempo y dinero.

#### 1.4.2 Lenguaje de modelado

Un lenguaje de modelado es un conjunto de elementos o notificaciones que están disponibles para el apoyar el modelado de una parte de una aplicación.

Suelen estar ligados a metodologías de desarrollo. A continuación, se muestra algunos de los tipos de lenguajes de modelado existentes:

El Lenguaje Unificado de Modelado (**UML**, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos, etc(Figuerola,2014).

El principal objetivo de UML es "proporcionar a los arquitectos de sistemas, ingenieros de software y desarrolladores de software" herramientas para el análisis, diseño e implementación de sistemas basados en software, así como para el modelado de negocios y procesos similares.

**BPNM** (*Business Process Modeling Notation*) es un lenguaje de notación gráfica, ampliamente aceptado para el modelado de procesos de negocio. Permite la creación de procesos de negocio "de extremo a extremo", está diseñado para cubrir muchos tipos de modelos, de modo que puedan comunicar una amplia variedad de información para una amplia variedad de audiencias(Figuerola, 2014).

El objetivo principal de BPMN es "proporcionar una notación que sea fácilmente comprensible por todos los usuarios de negocios, desde los analistas de negocio que crean los borradores iniciales de los procesos, hasta los desarrolladores técnicos responsables de la aplicación.

UML es un lenguaje que facilita a los desarrolladores la especificación, visualización y documentación de modelos de sistemas de software. Está dirigido en líneas generales a los arquitectos de software e ingenieros de software. Fue desarrollado como un medio para mejorar el proceso de desarrollo de software, desde el diseño de la arquitectura hasta la implementación de la aplicación, para ser utilizado por personas con conocimientos técnicos (analistas de sistemas y programadores)(Figuerola,2014).

BPMN está dirigido a los analistas de negocio, arquitectos de sistemas e ingenieros de software. Fue desarrollado para mejorar el ciclo de vida del desarrollo de procesos desde el diseño de los mismos. BPMN está emparentado con UML por el hecho que ambos definen una notación gráfica para los procesos de negocio, sin embargo, BPMN y UML usan enfoques diferentes para modelarlos. UML en general ofrece un enfoque orientado a objetos para modelar aplicaciones, mientras que BPMN toma un enfoque centrado en los procesos (Figuerola, 2014).

En base a la experiencia de los expertos que modelan negocios, se ha constatado que UML definitivamente aporta elementos muy valiosos como la identificación inmediata de las responsabilidades de los trabajadores del negocio y el comportamiento dependiente del estado de las entidades del negocio que en BPMN, si bien es posible, resulta impráctico

#### 1.4.3 Herramienta de modelado

Para modelar los artefactos de la propuesta de solución se utilizan las potencialidades que brinda Visual Paradigm. Esta es una herramienta CASE de Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Pressman, 2000).

Brinda grandes facilidades para la creación de los diferentes diagramas y cuenta con la posibilidad de realizar un control de versiones durante todo el ciclo de trabajo. Admite exportar e importar ficheros de proyectos realizados, y los artefactos que genera. También, facilita el desarrollo de ingeniería inversa y la generación automática de informes en formato PDF, Word o HTML (Pressman, 2000).

#### Conclusiones parciales

En este capítulo se abordaron los elementos teóricos que fundamentan la propuesta de solución de la problemática planteada, por lo que se puede concluir que:

- Se realizó un estudio de los procesos asociados a la gestión de los medios materiales y libros, permitiendo lograr una mejor comprensión de los procesos asociados al sistema.
- Con el estudio de las herramientas similares se pudo apreciar, que ninguna de las herramientas estudiadas puede dar solución al problema planteado, por lo que debe de implementarse una solución propia que responda a las necesidades de la facultad.
- Se realizó una investigación, estudio y análisis de herramientas y tecnologías y de las metodologías de desarrollo de software, la cual permitió una selección de las más indicadas para el desarrollo del sistema.

## **CAPÍTULO 2 Características del sistema y propuesta solución**

### Introducción

En este capítulo se aborda la propuesta solución a la problemática planteada. Se describen los requisitos funcionales y no funcionales, para dar cumplimiento a los objetivos planteados. Se realizan las historias de usuario, plan de iteraciones, tarjetas CRC y demás artefactos exigidos por la Metodología XP y sus fases.

### 2.1 Fase 1 Exploración

Durante la fase de exploración el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología(Montero, 2018).

Como se puede apreciar en el capítulo anterior se realizó un análisis de la temática a partir de la investigación realizada mediante la entrevista a los vicedecanos de administración de cada una de las facultades de la Universidad. Además, se realizó un estudio de sistemas homólogos con el fin de recopilar información de características necesarias y buenas prácticas para la confección de la aplicación web, para lo cual fueron seleccionadas las herramientas estudiadas durante el transcurso de la carrera.

#### 2.1.1 Educción de requisitos

Los requerimientos del software para un sistema, son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información(Sommerville, 2005).

La educación de requisitos es una actividad dentro del proceso de la ingeniería de requisitos que consiste en hallar e identificar información relevante acerca del dominio del problema y de las necesidades de las partes interesadas con el fin

de conformar los requisitos funcionales y no funcionales del software a desarrollar(Carrizo & Ortiz, 2016).

### **Técnicas de educación de requisitos:**

Durante el proceso de desarrollo del software los analistas emplean varias técnicas para obtener los requisitos del cliente lo cual contribuye a una mejor realización del sistema a desarrollar y por ende la satisfacción del mismo. A continuación, se muestran algunas de estas técnicas:

- **Entrevista:** Las entrevistas son un método común. Por lo general no se entrevista a todas las personas que se relacionarán con el sistema, sino a una selección de ellas que represente a todos los sectores críticos de la organización, con el énfasis puesto en los sectores más afectados o que harán uso más frecuente del nuevo sistema.
- **Lluvia de Ideas:** Todos los participantes pueden aportar distintas ideas en un ambiente libre de prejuicios. Ningún participante debe juzgar la propuesta de otro, sino que anotan todas las ideas en una pizarra y serán evaluadas al final de la sesión. El principio básico es no descartar de manera apresurada ningún planteamiento, de modo que existe la posibilidad de que surjan otras ideas derivadas de la idea original y se generen varios puntos de vista del problema(Guerra, 2021).
- **Diseño de aplicación conjunta:** Se trabaja directamente sobre los documentos a generar, las temáticas que se tratan durante las reuniones siguen un esquema y se busca que la misma sea ordenada y racional. Se define una agenda con los puntos principales a tratar durante la jornada. Este tipo de taller tiene el inconveniente de que es muy difícil poder reunir a todos los participantes, es costoso y generalmente es necesario más de una reunión para establecer requisitos del sistema(Guerra, 2021).
- **Forma de contrato:** En lugar de una entrevista, se pueden llenar formularios o contratos indicando los requisitos. En sistemas muy complejos
- **Objetivos medibles:** Los requisitos formulados por los formularios se toman como objetivos generales, a largo plazo, y en cambio se los debe analizar una y otra vez desde el punto de vista del sistema hasta determinar los objetivos críticos del funcionamiento interno que luego

darán forma a los comportamientos apreciables por el usuario. Luego, se establecen formas de medir el progreso en la construcción, para evaluar en cualquier momento que tan avanzado se encuentra el proyecto(Guerra, 2021).

Luego de analizadas las posibles técnicas a utilizar se decidió emplear la entrevista para tener un contacto más directo con los clientes y saber sus necesidades de manera más directa, además, se utilizó la consulta de tesis monográficas de alumnos egresados de la universidad, relacionadas con el proyecto, las cuales sirvieron como base y apoyo para la elaboración del sistema. También el uso de la internet como medio investigativo fue de gran importancia para la confección de la aplicación a desarrollar.

### 2.1.2 Requerimientos del sistema

Establecen con detalle las funciones, servicios y restricciones operativas del sistema. El documento de requerimientos del sistema (algunas veces denominado especificación funcional) debe ser preciso. Debe definir exactamente qué es lo que se va a implementar. Puede ser parte del contrato ente el comprador del sistema y los desarrolladores del software (Sommerville, 2005). Pueden ser clasificados en requisitos funcionales o requisitos no funcionales.

#### 2.1.2.1 *Requisitos Funcionales*

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste, como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema (Sommerville, 2005). A continuación, se presentan los requisitos funcionales del sistema:

*Tabla 7 Requerimientos funcionales*

Nº	Nombre	Descripción	Prioridad para el cliente	Complejidad
RF1	Autenticar usuario	El sistema debe permitir autenticar un usuario, solicitando los	Alta	Alta

		<p>siguientes datos:</p> <ul style="list-style-type: none"> <li>(*) Nombre de usuario</li> <li>(*) Contraseña</li> </ul> <p>El usuario puede autenticarse siempre que su nombre de usuario y contraseña estén registrados en el sistema y sean datos válidos.</p>		
RF2	Modificar datos de usuario	<p>El sistema debe permitir modificar todos los datos de un usuario, solicitando los siguientes datos:</p> <p>El sistema debe permitir modificar solo el rol de un usuario, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>(*) Rol</li> </ul> <p>El sistema debe permitir modificar solo uno de los datos de un usuario:</p> <ul style="list-style-type: none"> <li>Añadir nuevo rol y mantener el rol básico.</li> </ul>	Media	Media
RF3	Eliminar usuario	<p>El sistema debe permitir eliminar uno o varios usuarios.</p> <p>Al eliminar uno o varios usuarios, toda la información que tenga relación con ellos también desaparecerá.</p>	Baja	Baja
RF4	Listar usuario	<p><i>El sistema debe permitir listar los usuarios existentes en forma ascendente según el nombre de usuario. En el listado se deben mostrar los siguientes datos:</i></p> <ul style="list-style-type: none"> <li><i>Nombre</i></li> <li><i>Rol</i></li> </ul>	Alta	Baja
RF5	Filtrar listado de usuarios	<p><i>El sistema debe permitir filtrar el listado de usuarios, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li><i>Nombre</i></li> <li><i>Rol</i></li> </ul>	Baja	Media

RF6	Añadir libro	<p>El sistema debe permitir incluir un libro, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) Código</li> <li>• (*) Título</li> <li>• (*) Autor</li> <li>• (*) Año académico</li> <li>• (*) Asignatura</li> <li>• (*) Cantidad</li> <li>• (*) Precio</li> <li>• Imagen</li> <li>• Volumen</li> <li>• Tomo</li> <li>• Parte</li> </ul> <p>Además, de los datos anteriores se debe guardar el nombre de la asignatura tomado de la entidad "Asignatura".</p>	Alta	Alta
RF7	Modificar libro	<p>El sistema debe permitir modificar todos los datos de un libro, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) Código</li> <li>• (*) Título</li> <li>• (*) Autor</li> <li>• (*) Año académico</li> <li>• (*) Asignatura</li> <li>• (*) Cantidad</li> <li>• (*) Precio</li> <li>• Imagen</li> <li>• Volumen</li> <li>• Tomo</li> <li>• Parte</li> </ul>	Media	Media

RF8	Eliminar libro	<p>El sistema debe permitir eliminar uno o varios libros.</p> <p>Cuando se elimina uno o varios libros, toda la información referente a los mismos, es eliminada de las tarjetas de préstamo y de las estadísticas del sistema.</p>	Media	Baja
RF9	Listar libro	<p><i>El sistema debe permitir listar los libros existentes en forma ascendente según el nombre. En el listado se deben mostrar los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• Código</li> <li>• Título</li> <li>• Autor</li> <li>• Año académico</li> <li>• Asignatura</li> <li>• Cantidad</li> <li>• Precio</li> <li>• Imagen</li> <li>• Volumen</li> <li>• Tomo</li> <li>• Parte</li> </ul>	Alta	Baja
RF10	Filtrar listado de libros	<p><i>El sistema debe permitir filtrar el listado de libros, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• Código</li> <li>• Título</li> <li>• Autor</li> <li>• Año académico</li> <li>• Asignatura</li> </ul>	Baja	Baja

		<ul style="list-style-type: none"> <li>• Cantidad</li> <li>• Precio</li> <li>• Imagen</li> <li>• Volumen</li> <li>• Tomo</li> <li>• Parte</li> </ul>		
RF11	Añadir tarjeta de préstamo	<p>El sistema debe permitir incluir una tarjeta de préstamo, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) Estudiante / Trabajador</li> <li>• (*) Libro</li> <li>• (*) Fecha Entrega</li> <li>• Fecha Devolución</li> </ul> <p>Además, de los datos anteriores se debe guardar el nombre del estudiante, tomado de la entidad usuario.</p> <p>Además, de los datos anteriores se debe guardar el nombre del trabajador, tomado de la entidad usuario.</p> <p>Además, de los datos anteriores se debe guardar el nombre del libro, tomado de la entidad la libro.</p>	Alta	Alta
RF12	Modificar tarjeta de préstamo	<p>El sistema debe permitir modificar todos los datos de una tarjeta de préstamo, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) Estudiante / Trabajador</li> <li>• (*) Libro</li> <li>• (*) Fecha Entrega</li> <li>• Fecha Devolución</li> </ul>	Media	Media
RF13	Eliminar tarjeta de préstamo	<p>El sistema debe permitir eliminar una o varias tarjetas de préstamo.</p>	Media	Baja

		Cuando se elimina una o varias tarjetas de préstamo, toda la información referente a las mismas, es eliminada de las estadísticas del sistema.		
RF14	Listar tarjeta de préstamo	<p><i>El sistema debe permitir listar las tarjetas de préstamo existentes, por cada uno de los estudiantes o trabajadores, en forma ascendente según el nombre del libro. En el listado se deben mostrar los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• (*) Libro</li> <li>• (*) Fecha Entrega</li> <li>• Fecha Devolución</li> </ul>	Alta	Baja
RF15	Filtrar tarjeta de préstamo	<p><i>El sistema debe permitir filtrar el listado de tarjetas de préstamos, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• Estudiante / Trabajador</li> <li>• Libro</li> <li>• Fecha Entrega</li> <li>• Fecha Devolución</li> </ul>	Alta	Media
RF16	Visualizar estadísticas de almacén	<p><i>El sistema debe permitir ver las siguientes estadísticas del almacén:</i></p> <ul style="list-style-type: none"> <li>• Cantidad real de libros</li> <li>• Cantidad de libros prestados a estudiantes y trabajadores.</li> <li>• Cantidad real de libros en el almacén actualmente.</li> </ul>	Alta	Alta
RF17	Crear Reporte estadísticas de	<i>El sistema debe permitir exportar a PDF, un reporte con las</i>	Alta	Alta

	almacén	<p><i>siguientes estadísticas del almacén:</i></p> <ul style="list-style-type: none"> <li>• Nombre de la Facultad</li> <li>• <i>Cantidad real de libros</i></li> <li>• Cantidad de libros prestados a estudiantes y trabajadores.</li> <li>• Cantidad real de libros en el almacén actualmente.</li> </ul>		
RF18	Añadir recurso material	<p>El sistema debe permitir incluir un recurso material, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) código</li> <li>• (*) descripción</li> <li>• (*) tipo</li> </ul>	Alta	Alta
RF19	Modificar recurso material	<p>El sistema debe permitir modificar todos los datos de un recurso material, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) código</li> <li>• (*) descripción</li> <li>• (*) tipo</li> </ul>	Media	Media
RF20	Eliminar recurso material	<p>El sistema debe permitir eliminar uno o varios recursos materiales.</p> <p>Cuando se elimina uno varios recursos materiales, toda la información referente a estos, es eliminada de los módulos y las tarjetas de destino final.</p>	Baja	Baja
RF21	Listar recurso material	<p><i>El sistema debe permitir listar los recursos materiales existentes en forma ascendente según su descripción. En el listado se deben mostrar los siguientes datos:</i></p>	Alta	Baja

		<ul style="list-style-type: none"> <li>• código</li> <li>• descripción</li> <li>• tipo</li> </ul>		
RF22	Filtrar recurso material	<p><i>El sistema debe permitir filtrar el listado de recursos materiales, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• código</li> <li>• descripción</li> <li>• tipo</li> </ul>	Baja	Media
RF23	Añadir módulo	<p>El sistema debe permitir incluir un módulo, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) nombre</li> <li>• (*) Lista de materiales</li> </ul> <p>Además, de los datos anteriores se debe guardar el de nombre de los materiales, los cuales serán tomados de la entidad "Recurso Material".</p>	Alta	Alta
RF24	Modificar módulo	<p>El sistema debe permitir modificar todos los datos de un módulo, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) nombre</li> <li>• (*) Lista de materiales</li> </ul>	Media	Media
RF25	Eliminar módulo	<p>El sistema debe permitir eliminar uno o varios módulos.</p> <p>Cuando se elimina uno o varios módulos, toda la información referente a estos, es eliminada de las tarjetas de destino final.</p>	Baja	Baja
RF26	Listar módulo	<p><i>El sistema debe permitir listar los módulos existentes en forma ascendente según su nombre. En el listado se deben mostrar los</i></p>	Alta	Baja

		<p><i>siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• nombre</li> <li>• Lista de materiales</li> </ul>		
RF27	Filtrar módulo	<p><i>El sistema debe permitir filtrar el listado de módulos, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• nombre</li> <li>• Lista de materiales</li> </ul>	Baja	Media
RF28	Añadir tarjeta de destino final	<p>El sistema debe permitir incluir una tarjeta de destino final, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) Estudiante / Trabajador</li> <li>• (*) Módulo</li> <li>• (*) Fecha Entrega</li> </ul> <p>Además, de los datos anteriores se debe guardar el nombre del estudiante, tomado de la entidad la estudiante.</p> <p>Además, de los datos anteriores se debe guardar el nombre del trabajador, tomado de la entidad usuario.</p> <p>Además, de los datos anteriores se debe guardar el nombre del módulo, tomado de la entidad el módulo.</p>	Alta	Alta
RF29	Modificar tarjeta de destino final	<p>El sistema debe permitir modificar todos los datos de una tarjeta de destino final, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• (*) Estudiante / Trabajador</li> <li>• (*) Módulo</li> <li>• (*) Fecha Entrega</li> </ul>	Media	Media
RF30	Eliminar tarjeta de destino final	<p>El sistema debe permitir eliminar una o varias tarjetas de destino</p>	Baja	Baja

		final.		
RF31	Listar tarjeta de destino final	<p><i>El sistema debe permitir listar las tarjetas de destino final existentes, en forma ascendente según el nombre del módulo. En el listado se deben mostrar los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• Estudiante / Trabajador</li> <li>• Módulo</li> <li>• Fecha Entrega</li> </ul>	Alta	Baja
RF32	Filtrar tarjeta de destino final	<p><i>El sistema debe permitir filtrar el listado de tarjetas de destino final, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• Estudiante / Trabajador</li> <li>• Módulo</li> <li>• Fecha Entrega</li> </ul>	Baja	Media
RF33	Visualizar tarjeta personal de préstamo	<p><i>El sistema debe permitir ver a cada estudiante o trabajador los siguientes datos de su tarjeta de préstamo:</i></p> <ul style="list-style-type: none"> <li>• Libro</li> <li>• Fecha Entrega</li> <li>• Fecha Devolución</li> </ul>	Media	Media
RF34	Notificar recogida de módulo	<p><i>El sistema deberá notificar a estudiantes o trabajadores la disponibilidad del módulo para su posterior recogida, teniendo en cuenta los siguientes campos:</i></p> <ul style="list-style-type: none"> <li>• Módulo</li> <li>• Estudiante / Trabajador</li> </ul>	Media	Media
RF35	Filtrar estudiante	<p><i>El sistema debe permitir filtrar el listado de estudiantes, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• Nombre</li> </ul>	Baja	Media

		<ul style="list-style-type: none"> <li>• Año académico</li> <li>• Grupo</li> </ul>		
RF36	Filtrar trabajador	<p><i>El sistema debe permitir filtrar el listado de trabajadores, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Área</li> </ul>	Baja	Media
RF37	Visualizar catálogo de libros	<p><i>El sistema debe permitir ver todos los libros que se encuentren el almacén. En caso de existir su versión digital, los usuarios tendrán acceso a leerla.</i></p>	Baja	Alta
RF38	Visitar sitios de interés.	<p><i>El sistema debe permitir visitar sitios de interés de la universidad.</i></p>	Baja	Baja
RF39	Ver / Ocultar columnas libro	<p><i>El sistema debe permitir añadir o eliminar columnas de la tabla de libros, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• Código</li> <li>• Título</li> <li>• Autor</li> <li>• Año académico</li> <li>• Asignatura</li> <li>• Cantidad</li> <li>• Precio</li> <li>• Imagen</li> <li>• Volumen</li> <li>• Tomo</li> <li>• Parte</li> </ul>	Baja	Baja
RF40	Ver / Ocultar columnas recurso material	<p><i>El sistema debe permitir añadir o eliminar columnas de la tabla de materiales, teniendo en cuenta los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>• Código</li> </ul>	Baja	Baja

		<ul style="list-style-type: none"> <li>• Nombre</li> <li>• Tipo</li> </ul>		
--	--	--	--	--

### 2.1.2.2 *Requisitos no funcionales*

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema(Sommerville, 2005). A continuación, se presentan los requisitos no funcionales del sistema:

- Portabilidad:

#### Software

RNF 1. Se deberá utilizar como servidor de aplicación web Apache Tomcat en su versión 9.0.58. Además, deberá contar con la versión de java 11.

RNF 2. Las computadoras clientes deberán contar con navegadores web como Google Chrome versión 108.0.5359.71 o superior o Mozilla Firefox Developer versión 108.0b8 o superior.

RNF 3. El servidor de base de datos deberá tener instalado el gestor de base de datos PostgreSQL 14.0.

#### Hardware

RNF 4. Las computadoras clientes como los servidores deberán tener una tarjeta de red con velocidad de transmisión mínima de 100Mbps.

RNF 5. El servidor de procesamiento requiere memoria RAM de 4 GB DDR3 o superior, procesador de 3.0 GHZ o superior y capacidad de almacenamiento de 500 GB o superior.

RNF 6. Las computadoras clientes requieren memoria RAM de 4GB DDR3 o superior, procesador de 3.0 GHZ.

RNF 7. El servidor de base de datos requiere memoria RAM de 4 GB DDR3 o superior, procesador de 3.0 GHZ o superior y capacidad de almacenamiento de 500 GB o superior.

- Requisitos de Fiabilidad

RNF 8. El Sistema debe estar disponible para su utilización las 24 horas del día, durante los siete días de la semana, con el menor tiempo posible de recuperación ante fallos.

- Requisitos de seguridad

RNF 9. Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador del sistema.

RNF 10. El acceso a la información debe estar restringido según el rol que la utiliza.

RNF 11. No deben existir “puertas traseras” que permitan el manejo de información fuera del flujo lógico del sistema.

- Requisitos de usabilidad

RNF 12. Los formularios y demás herramientas de apoyo deben ser intuitivos al usuario.

RNF 13. El sistema debe revisar todos los campos de datos para comprobar que no se introduzcan datos erróneos y en caso de encontrar algún error, mostrar un mensaje que indique el error.

RNF 14. El sistema debe comprobar que todos los campos obligatorios del formulario se encuentren llenos.

RNF 15. El sistema mostrará un mensaje con el resultado de la acción que se realizó, ya sea añadir, modificar o eliminar.

RNF 15. El sistema debe utilizar imágenes que no superen 1MB de almacenamiento, y componentes de diseño que permitan mostrar la información de manera dinámica, ágil y estética.

- Requisitos de Mantenibilidad

RNF 16. El sistema deberá ser fácilmente escalable, con el fin de poder hacer crecer la aplicación al incorporar a futuro nuevas funcionalidades.

- Requisitos de Eficiencia

RNF 17. Los tiempos de respuesta relacionados con formularios de manejo de información, adición, modificación, eliminación, consulta de registros, autenticación y emisión de avisos y confirmaciones por parte del usuario, en forma general, no deben ser superiores a 3 segundos.

RNF 18. El sistema debe ser capaz de operar adecuadamente con hasta 50 usuarios con sesiones concurrentes.

RNF 19. Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

RNF 20. Cada transición entre las páginas del sistema debe ocurrir en un máximo de 3 segundos, excepto los reportes los cuales pueden tomar entre 3 y 10 segundos.

## 2.2 Fase 2 Planificación de entregas

Se cumple con la Fase II de la Metodología XP donde el cliente establece la prioridad de cada historia de usuario. En correspondencia, los programadores realizan una estimación del esfuerzo necesario para la realización de cada una de estas. Se toman en cuenta acuerdos sobre el contenido de la primera entrega determinada y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres semanas.

### 2.2.1 Historias de usuario

Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final o cliente. Su propósito es articular cómo proporcionará una función de software al cliente. Se trata de tarjetas de papel en las cuales el cliente describe las características que el sistema debe poseer, sean requisitos funcionales o no. El tratamiento de las historias de usuario es dinámico y flexible. Cada historia de usuario es lo

suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

A continuación, se muestra la historia de usuario definida para implementar el requisito funcional RF2.1 Añadir libro. Las restantes historias de usuario se encuentran en el [Anexo 2](#).

Tabla 8 Historia de usuario “Añadir libro”

Historia de Usuario	
<b>Número: 6</b>	<b>Nombre del requisito: Añadir libro</b>
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 2da
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> <i>Permitir añadir nuevos libros en el sistema.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para añadir un libro hay que:</p> <ul style="list-style-type: none"> <li>- Tener en cuenta los siguientes datos: código, título, autor, año académico, asignatura, cantidad, precio, imagen, tomo, volumen, parte.</li> <li>- <i>Estar autenticado en el sistema con el rol responsable de almacén.</i></li> </ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b> Los campos título, autor, año académico, asignatura, cantidad, precio son obligatorios.</p> <p>Código: campo de texto que solo admite números con un máximo de 10 caracteres.</p> <p>Título: campo de texto que admite caracteres alfabéticos, numéricos, y caracteres especiales. Tiene un mínimo de 2 caracteres y un máximo de hasta 255 caracteres.</p> <p>Autor: campo de texto que solo admite caracteres alfabéticos. Tiene un mínimo de 2 caracteres y un máximo de hasta 255 caracteres.</p> <p>Año académico: campo numérico, que solo admite valores entre 1 y 5.</p>	

Asignatura: campo de selección. El usuario podrá escribir una parte de la palabra o la palabra completa, referente a la asignatura que desea y esta, será filtrada.

Cantidad: campo numérico, que solo admite valores mayores a 0.

Imagen: campo de selección. El usuario podrá buscar una imagen en su PC, la cual no deberá pesar más de 1MB.

Tomo: campo numérico, que solo admite valores entre 1 y 10.

Volumen: campo numérico, que solo admite valores entre 1 y 10.

Parte: campo numérico, que solo admite valores entre 1 y 10.

#### **4- Flujo de la acción a realizar:**

- El sistema debe permitir incluir y/o seleccionar los datos para añadir un nuevo libro.
- Cuando el usuario incluye y/o selecciona correctamente los datos necesarios para añadir un nuevo libro y selecciona la opción Añadir, se crea un nuevo elemento y el sistema muestra un mensaje de información.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión.
- Si selecciona la opción Cancelar regresará a la vista previa.

**Observaciones:** N/A

**Prototipo de interfaz:**

The image shows a web form titled "Libro" with a light blue header and a light gray background. At the top, there is a button labeled "Seleccione una imagen.". Below this are several input fields, each with a label and an asterisk indicating it is required: "Código \*", "Título \*", "Autor \*", "Año Académico \*", "Asignatura \*", "Cantidad \*", "Precio \*", "Tomo", "Volumen", and "Parte". The "Año Académico" and "Asignatura" fields are dropdown menus. At the bottom of the form, there are two buttons: "Añadir" and "Cancelar".

### 2.2.2 Estimación de tiempo por historias de usuario

La programación utilizando la metodología XP basa sus procesos de planificación en estimaciones temporales de las historias de usuario, las cuales deben ser realizadas por los desarrolladores durante las diversas reuniones de planificación.

Una historia de usuario debe ser lo suficientemente pequeña como para que el equipo la desarrolle durante una entrega de una a tres semanas; más de tres semanas implica que debe ser señalar al cliente que debe dividir una historia de usuario y menos de una semana implica que la historia de es demasiado sencilla y por lo que se deben unir dos o más de ellas para su mejor interpretación.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto de estimación. Un punto de estimación, equivale a una semana ideal de programación. Las

historias generalmente valen de 1 a 3 puntos de estimación. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos de iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

En la siguiente tabla de estimación del tiempo de las historias de usuario, se usó la denotación de punto de estimación por semana y la denotación de puntos flotantes para indicar los días.

Tabla 9 Estimación de tiempo

Denotación de punto de estimación	Interpretación
1 punto	1 día (8 horas laborables)
1.5 puntos	1 día y 4 horas

Tabla 10 Estimación del tiempo de HU

Historia de usuario	Punto estimado
Autenticar usuario	3
Modificar usuario	2
Eliminar usuario	2
Listar usuario	1
Filtrar usuario	2
Añadir libro	3
Modificar libro	2
Eliminar libro	2
Listar libro	1
Filtrar libro	2
Añadir tarjeta de préstamo	3
Modificar tarjeta de préstamo	2
Eliminar tarjeta de préstamo	2
Listar tarjeta de préstamo	1
Filtrar tarjeta de préstamo	2
Visualizar estadísticas de almacén	3

<b>Exportar reporte de estadísticas de almacén</b>	3
<b>Visualizar tarjeta personal de préstamo</b>	2
<b>Añadir recurso material</b>	3
<b>Modificar recurso material</b>	2
<b>Eliminar recurso material</b>	2
<b>Listar recurso material</b>	1
<b>Filtrar recurso material</b>	2
<b>Añadir módulo</b>	3
<b>Modificar módulo</b>	2
<b>Eliminar módulo</b>	2
<b>Listar módulo</b>	1
<b>Filtrar módulo</b>	2
<b>Notificar recogida de módulo</b>	2
<b>Añadir Tarjeta destino final</b>	3
<b>Modificar Tarjeta destino final</b>	2
<b>Eliminar Tarjeta destino final</b>	2
<b>Listar Tarjeta destino final</b>	1
<b>Filtrar Tarjeta destino final</b>	2
<b>Filtrar estudiante</b>	2
<b>Filtrar trabajador</b>	2
<b>Visualizar catálogo de libros</b>	3
<b>Visitar sitios de interés.</b>	1
<b>Ver / Ocultar columnas libro</b>	1
<b>Ver / Ocultar columnas recurso material</b>	1

Las estimaciones permitieron confeccionar una evaluación puntual del tiempo de implementación de cada historia de usuario para la posterior elaboración del plan de iteración. Una vez realizadas las estimaciones fue preciso construir un plan de iteración donde se pueden agrupar estas historias y dar cumplimiento de manera paulatina.

2.2.3 Plan de iteraciones

Es la práctica en donde el equipo establece el rumbo cada un par de semanas. Los equipos XP construyen software en iteraciones de dos semanas, y entregan software útil al finalizar cada iteración. Durante la Planificación de la Iteración, el Cliente presenta las características deseadas para las siguientes dos semanas. Los programadores las descomponen en tareas, y estiman su costo (a un nivel de detalle más fino que durante la Planificación de la Entrega). El equipo entonces se compromete a terminar ciertas características basándose en la cantidad de trabajo que pudieron terminar en la iteración anterior.

Un plan de iteración está constituido por un conjunto secuencial de actividades y tareas, cada una contiene recursos asignados y pueden depender a su vez de otras tareas. El plan de iteración se realiza una vez por cada interacción.

El contenido de la iteración puede variar, dependiendo de la posición del ciclo de vida de la naturaleza del proyecto. El plan de iteración incluye proporciones relevantes a todas las disciplinas para cada iteración en particular, la prioridad de implementación se evalúa en base al cliente y el equipo de desarrollo, el impacto del riesgo en base al juicio de experto. Los planes de iteración por lo general muestran un planeamiento detallado de quién va a realizar una tarea/actividad de acuerdo en conformidad a que criterios de evaluación.

Tabla 11 Plan de iteraciones

Nro. Iteración	Historia de usuario	Prioridad	Esfuerzo Estimado	
1	Autenticar usuario	Alta	3	20
	Modificar usuario	Media	2	
	Eliminar usuario	Baja	2	
	Listar usuario	Alta	1	
	Filtrar usuario	Baja	2	
	Añadir libro	Alta	3	
	Modificar libro	Media	2	

	Eliminar libro	Baja	2	
	Listar libro	Alta	1	
	Filtrar libro	Baja	2	
2	Añadir tarjeta de préstamo	Alta	3	18
	Modificar tarjeta de préstamo	Media	2	
	Eliminar tarjeta de préstamo	Baja	2	
	Listar tarjeta de préstamo	Alta	1	
	Filtrar tarjeta de préstamo	Baja	2	
	Visualizar estadísticas de almacén	Alta	3	
	Exportar reporte de estadísticas de almacén	Alta	3	
	Visualizar tarjeta personal de préstamo	Media	2	
	Añadir recurso material	Alta	3	
	Modificar recurso material	Media	2	
	Eliminar recurso material	Baja	2	

3	Listar recurso material	Alta	1	22
	Filtrar recurso material	Baja	2	
	Añadir módulo	Alta	3	
	Modificar módulo	Media	2	
	Eliminar módulo	Baja	2	
	Listar módulo	Alta	1	
	Filtrar módulo	Baja	2	
	Notificar recogida de módulo	Media	2	
4	Añadir Tarjeta destino final	Alta	3	20
	Modificar Tarjeta destino final	Media	2	
	Eliminar Tarjeta destino final	Baja	2	
	Listar Tarjeta destino final	Alta	1	
	Filtrar Tarjeta destino final	Baja	2	
	Filtrar estudiante	Baja	2	
	Filtrar trabajador	Baja	2	
Visualizar catálogo de libros	Baja	3		

Visitar sitios de interés.	Baja	1
Ver / Ocultar columnas libro	Baja	1
Ver / Ocultar columnas recurso material	Baja	1

Una vez realizado el plan de iteración se pudo agrupar las diferentes historias de usuario en 4 iteraciones teniendo en cuenta las características que rigen la metodología XP. Con este plan de iteraciones ya es posible realizar un plan de entrega que será entregado al cliente y que el grupo de desarrollo está obligado a hacer cumplir.

#### 2.2.4 Plan de entrega

Basándonos en las historias de usuario definidas para el desarrollo del sistema web, se ha elaborado el siguiente plan de entrega, el cual muestra las historias de usuario que se llevarán a cabo en cada iteración.

Tabla 12 Plan de entrega

	Iteración 1	Iteración 2	Iteración 3	Iteración 4
Cantidad de HU	10	8	11	11
Fecha de inicio	18/4/2022	8/05/2022	26/05/2022	17/06/2022
Fecha de entrega	8/05/2022	26/05/2022	17/06/2022	5/07/2022

Una vez realizado el plan de entrega se puede confirmar con el cliente que el proyecto durará 12 semanas aproximadamente, lo que significa que tendrá un tiempo de desarrollo de 1920 horas aproximadamente.

### 2.2.5 Reuniones diarias de seguimiento

El planeamiento es esencial para cualquier tipo de metodología, es por ello que XP requiere de una revisión continua del plan de trabajo. A pesar de ser una metodología que evita la documentación exagerada, es muy estricta en la organización del trabajo.

El equipo de desarrollo, define desde el inicio, aquellos espacios en los que el equipo y el cliente realizarían encuentros semanales para la evaluación del cumplimiento de los resultados de las diferentes iteraciones, así como el cumplimiento parcial, total, o nuevas redefiniciones a los conceptos establecidos. Las pequeñas entregas facilitarán el trabajo de los desarrolladores, haciendo posible suplir las necesidades del cliente .

Tabla 13 Reuniones de seguimiento

Reunión de/	Fecha	Descripción
<b>Plan de entregas</b>	1/4/2022	Se realiza entre el equipo de trabajo y los clientes y se define el marco temporal de la realización del sistema. El cliente expone las historias de usuario a los integrantes de grupo, quienes estimarán el grado de dificultad de la implementación de cada historia. A partir de las historias de usuario, el cliente plantea las pruebas de aceptación con las cuales se comprueba que cada una de estas ha sido correctamente implementada.
<b>Inicio de iteración</b>	18/4/2022	Esta reunión es realizada previo a iniciar una iteración donde se organizan las actividades de programación a realizar. Las historias de usuario son traducidas a tareas y asignadas a los desarrolladores.
		Estas reuniones se realizan al comenzar la jornada laboral, donde todo el equipo de

<p><b>Diarias</b></p>	<p>Al inicio de cada jornada de trabajo</p>	<p>desarrollo se reúne para exponer los problemas e ideas que se estén presentando. Es de vital importancia evitar las discusiones largas, ya que se está utilizando tiempo laboral que puede ser destinado a la construcción del sistema.</p>
<p><b>Fin de Iteración</b></p>	<p>8/05/2022 26/05/2022 17/06/2022 5/07/2022</p>	<p>Estas reuniones se realizan al finalizar cada iteración en conjunto con los clientes para presentar los avances en cada iteración y demostrar la aceptación por los mismos como muestra de una correcta implementación.</p>

Con el plan de reuniones, queda determinado la secuencia de las distintas actividades a realizar dentro del proceso de desarrollo de la solución, contribuyendo a la calidad del producto deseado. Una vez determinado los artefactos organizativos, se procede a la fase de diseño de la solución propuesta.

### 2.3 Fase 3 Etapa de diseño

Teniendo en cuenta lo que plantea la metodología XP, en esta fase se confeccionan las tarjetas Clase Responsabilidad-Colaborador (CRC) para la descripción de las principales clases de los módulos desarrollados. Se define la arquitectura del sistema y los estándares de codificación, así como los patrones de diseño utilizados en el desarrollo de la propuesta.

#### 2.3.1 Tarjeta CRC (Clase-Responsabilidad-Colaboradores)

Las Tarjetas CRC, permiten conocer que clases componen el sistema y cuales interactúan entre sí. Se dividen en tres secciones: Nombre de la clase, responsabilidades y colaboradores(Chiluisa Pallo & Loarte Cajamarca, 2014). Se elaboran durante la fase de diseño de la metodología XP para describir las entidades existentes en la aplicación. El objetivo es obtener un diseño simple, elegante y fácil de comprender por parte de los programadores.

Tabla 14 Tarjeta CRC LibroView

Nombre clase: LibroView	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> <li>• configureGrid() //mostrar lista de libros.</li> <li>• updateList() //actualizar lista de libros.</li> <li>• addBook() // Añadir libro.</li> <li>• editBook(Libro libro) //editar libro.</li> <li>• deleteBook(Set&lt;Libro&gt; libros) //eliminar uno o más libros</li> <li>• Filtros() //actualizar los elementos de la lista según el campo seleccionado.</li> </ul>	Libro.java  Asigantura.java

Con la realización de las tarjetas CRC se evidencia la interrelación existente entre cada clase de la solución y sus diferentes funcionalidades. Teniendo en cuenta la interrelación de las entidades se hace necesario realizar prototipos que evidencien al cliente y al equipo de desarrollo el resultado que se obtendría con la solución.

### 2.3.2 Arquitectura Modelo-Vista-Controlador

El este sistema será desarrollado utilizando la tecnología Spring-Boot la cual está desarrollada siguiendo el patrón clásico de arquitectura Modelo Vista Controlador (MVC), el cual separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

**Vista:** parte de la aplicación que contiene la interfaz gráfica. Cada elemento gráfico que interactúa con el usuario es la vista y su función es la de ser la capa que obtiene información sobre lo que quiere el usuario, a lo cual vamos a llamar eventos, la vista muestra siempre la información del modelo.

**Modelo:** tiene la responsabilidad de la relacionar los datos con los que la aplicación va a operar. A esto se le llama lógica de negocio, es decir, es la parte de la aplicación que se encarga de mapear las actividades del mundo real a la forma a la que se va a modificar la información.

**Controlador:** responde a eventos o acciones que realice el usuario a través de la vista para poder solicitar una operación de información, también es responsable de mostrarle al usuario la vista adecuada dependiendo de la solicitud recibida, por lo cual es el vínculo que une al modelo con la vista.

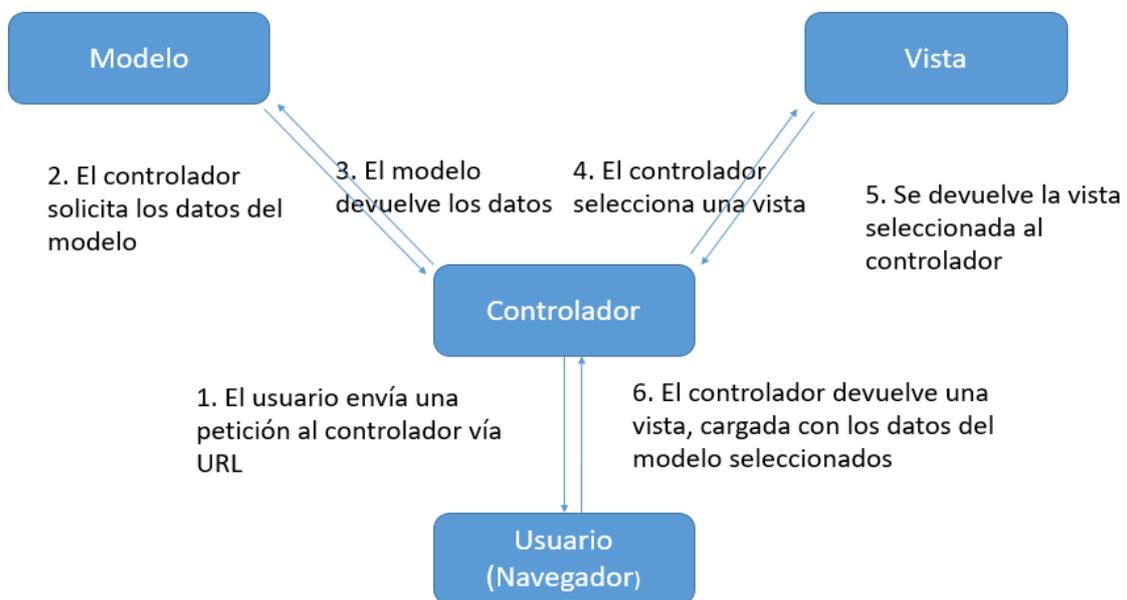


Figura 7 Funcionamiento del patrón Modelo Vista Controlador (Elaboración propia)

En resumen, la vista muestra el modelo al usuario, este activa eventos en las vistas los cuales son recibidos por el controlador. El controlador manipula al modelo dependiendo de la solicitud. El modelo devuelve un conjunto de datos acordes a la solicitud realizada por el controlador, para que finalmente el controlador muestre una nueva vista al usuario con la información actualizada del modelo.

Vaadin utiliza el constructor de las clases java como controlador de las vistas, este es el encargado de controlar todas las peticiones que se realizan sobre la vista y de mostrar en cada momento la vista adecuada al usuario. Además, gestiona la información que se encuentra en la base de datos, haciendo peticiones a la misma de creación, modificación, eliminación, entre otras.

## 2.4 Fase 4 Implementación

El desarrollo o la codificación es un proceso que se realiza en forma paralela con el diseño y la cual está sujeta a varias observaciones por parte de XP. Dentro de esta fase se describen los estándares de codificación, así como los patrones de diseño que se emplearon dentro del sistema.

### 2.4.1 Estándares de codificación

Los estándares de código, son parte de las llamadas buenas prácticas o mejores prácticas, estas son un conjunto no formal de reglas, que han ido surgiendo en las distintas comunidades de desarrolladores con el paso del tiempo y las cuales, bien aplicadas pueden incrementar la calidad de tu código, notablemente (Merkury, 2017).

Entendemos como estándar de código a un conjunto de convenciones establecidas de ante mano (denominaciones, formatos, etc.) para la escritura de código. Estos estándares varían dependiendo del lenguaje de programación elegido y, además, varían en cobertura, algunos son más extensos que otros.

En el proceso de desarrollo de un software siguiendo la metodología XP, es necesario que exista una adecuada comunicación entre los programadores del equipo de desarrollo. Para lograr esto, se establecen un conjunto de estándares definidos por las convenciones de escritura de código Java abarcando el marco de trabajo Spring-Boot.

Cada archivo java contiene una única clase o interfaz pública. Cuando algunas clases o interfaces privadas están asociadas a una clase pública, pueden ponerse en el mismo archivo que la clase pública. La clase o interfaz pública debe ser la primera clase o interfaz del archivo.

Los archivos fuente java tienen el siguiente orden:

- Comentarios de inicio

Todos los ficheros Fuente deben comenzar con un comentario en el que se lista el nombre de la clase, información de la versión, fecha y copyright:

```
/*  
* Nombre de la clase  
*
```

\* Información de la versión

\*

\* Fecha

\*

\* Copyright

\*/

- Sentencias package e import

La primera línea no-comentario de los archivos fuente Java es la sentencia package. Después de esta, pueden seguir varias sentencias import.

Por ejemplo:

```
package trabajodediploma.views.login;
```

```
import com.vaadin.flow.component.button.Button;
```

- Declaraciones de clases e interfaces

Primero las variables de clase public, después las protected, después las de nivel de paquete (sin modificador de acceso) y por último las privadas.

Los métodos se deben agrupar por funcionalidad más que por visión o accesibilidad.

Las variables y métodos utilizarán la notación CamelCase el cual es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello. El nombre CamelCase se podría traducir como Mayúsculas/Minúsculas Camello.

#### 2.4.2 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Son una descripción de un problema y su solución recibe un nombre. Muchos patrones proporcionan guías en el modo en que deberían asignarse las responsabilidades a los objetos, dada una categoría específica del problema (Larman, 2005).

##### 2.4.2.1 Patrones GRASP

Los patrones GRASP (*General Responsibility Assignment Software Patterns*), son patrones generales para la asignación de responsabilidades. Describen los

principios fundamentales de la asignación de responsabilidades a objetos, y sirven para asignar correctamente las responsabilidades en el diseño de la Programación Orientada a Objetos (POO). A continuación, se detalla el uso de estos patrones en la solución.

**Experto:** el objetivo de este patrón es asignar una responsabilidad a la clase que tiene la información necesaria para cumplir dicha responsabilidad. Permite conservar el encapsulamiento, ya que los objetos se valen de su propia información para realizar lo que se le oriente(Larman, 2005). En la presente solución se utiliza el mencionado patrón dentro de la capa de acceso a datos, se evidencia en las clases entidades, las cuales son expertas en su propia información, tales como Libro y Recurso Material.

**Creador:** Se emplea cuando se aplica la responsabilidad a una clase determinada de crear una o más instancias de otra. Esto sucede en caso de que la clase creadora contenga, agregué, registre, utilice o posea datos de inicialización de objetos de alguna clase determinada(Larman, 2005). El empleo de este patrón se ve reflejado en las clases LibroService, RecursoMaterial Service entre otras.

**Alta cohesión:** es la medida en la que un componente se dedica a realizar solo la tarea para la cual fue creado, delegando las tareas complementarias a otros componentes. Una clase debe hacer solo lo que respecta a ella y no hacer acciones que involucren a otra clase(Larman, 2005). Este patrón se aplica en los constructores de las clases LibroView.java, RecursoMaterialView.java, entre otras, las cuales se encargan de la implementación de las funcionalidades que le corresponden (dígase libros y recursos materiales en este caso).

**Bajo Acoplamiento:** Persigue tener las clases lo menos ligadas ente sí. Si se produce una modificación en alguna de ellas, lograr que la repercusión sea la mínima posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre clases(Larman, 2005). Se aplica en los constructores de las clases LibroView.java, RecursoMaterialView.java, entre otras.

#### 2.4.2.2 Patrones GOF

**Singleton:** (instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Spring utiliza este patrón mediante el uso de *beans* como `@Component`, `@Service`, `@Repository`, los cuales garantizan un punto de acceso a la clase que se le asigna. Mientras que con el uso del `@Autowired` se garantiza la creación de una instancia única de la clase. Un ejemplo de la utilización de este patrón en el proyecto, se encuentra evidenciado en el constructor la clase `LibroView`, en la cual se crean las instancias de los servicios que se utilizarán para conformar la información de la vista.

#### 2.4.3 Modelo de datos

Un modelo de base de datos es la estructura lógica que adopta la base de base de datos, incluyendo las relaciones y limitaciones que determinan cómo se almacenan y organizan y cómo se accede a los datos. Así mismo, un modelo de base de datos también define qué tipo de operaciones se pueden realizar con los datos.

Es decir, que también determina cómo se manipulan los mismos, proporcionando también la base sobre la que se diseña el lenguaje de consultas. En general, prácticamente todos los modelos de base de datos pueden representarse a través de un diagrama de base de datos.

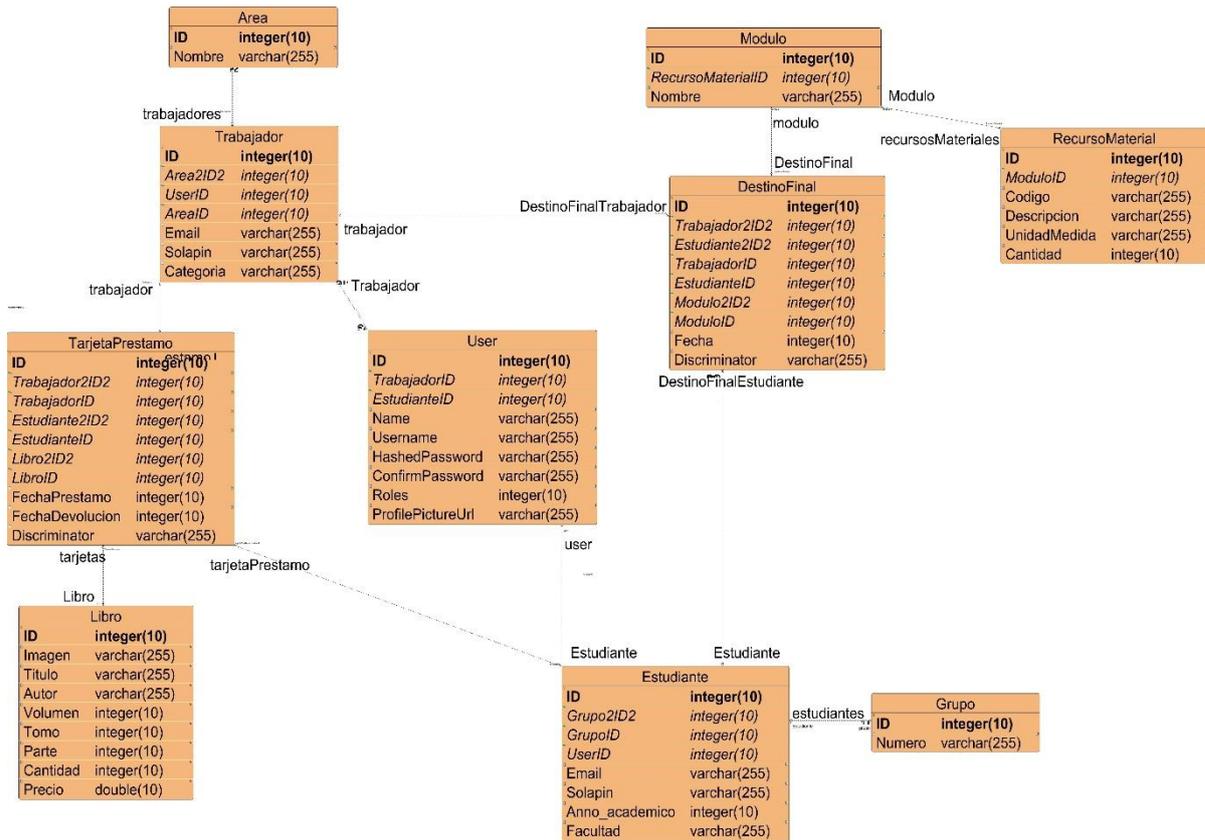


Figura 8 Diagrama Entidad-Relación (Elaboración propia)

## 2.5 Diagrama de despliegue

Un diagrama de despliegue muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Como los artefactos se ubican en los nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue. Los estereotipos definen la naturaleza del equipo ya sean dispositivos, procesadores o memoria (Creately, 2020).

### Descripción de los nodos

**Cliente:** representa una computadora desde la cual el usuario podrá acceder a la aplicación.

**Servidor Web:** representa una estación donde estará montado el servidor Apache sobre el cual se estará ejecutando la aplicación.

Servidor de BD: representa el servidor donde estará el sistema gestor de base de datos PostgreSQL que dará respuesta a las peticiones hechas por la aplicación.

Impresora: representa el dispositivo que imprimirá los reportes ofrecidos por la aplicación.

### Protocolos

- TCP/IP (Protocolo de control de transferencia): se utiliza en la comunicación entre el servidor y la base de datos para realizar operaciones sobre la información de las tablas.
- HTTP (Protocolo de transferencia de hipertexto): establece un esquema de comunicación cliente/servidor. El cliente es el navegador web que realiza las peticiones a las que el servidor se encarga de dar respuesta.
- USB (Bus Universal en Serie): permite conectar periféricos para atender peticiones, y proveer corriente si es necesario. El cliente envía una petición para imprimir un reporte.

A continuación, se muestra el ambiente donde debe ser desplegada la propuesta solución:

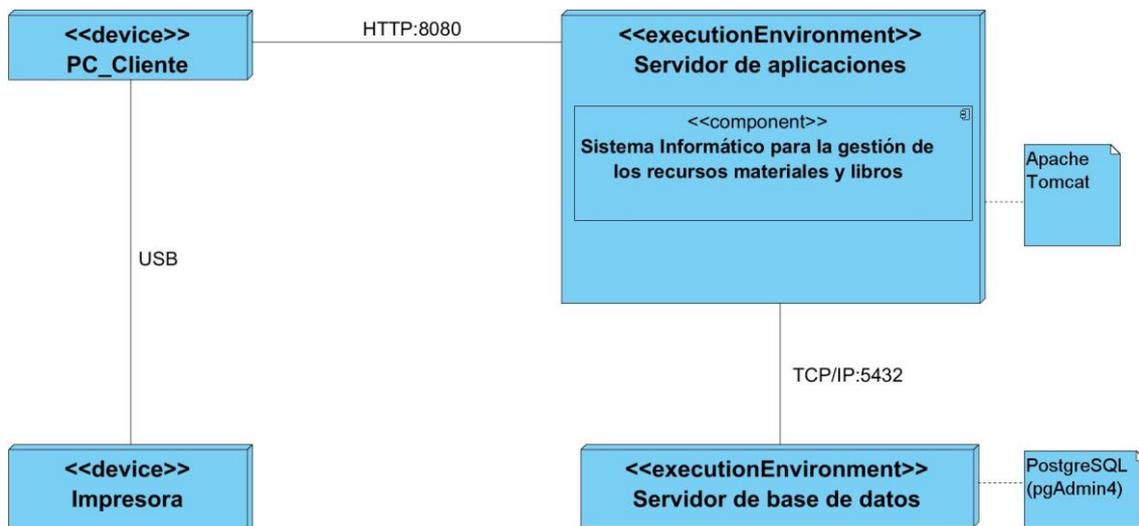


Figura 9 Diagrama de despliegue (Elaboración propia)

### Conclusiones parciales

Luego de haber descrito las características del sistema se puede llegar a las siguientes conclusiones:

- La estrategia de obtención de requisitos puesta en práctica, permitió identificar las funcionalidades que el sistema debe brindar, así como las restricciones sobre las que debe operar. A partir de la captura de los requisitos se especificaron los mismos en términos de HU.
- La identificación de los patrones de diseño y el estilo arquitectónico de la solución propuesta, permite disminuir el impacto de los cambios futuros en el código fuente de la misma.
- Partiendo de las funcionalidades del sistema se logró generar el modelo de datos con el que contará la aplicación.
- La generación de principales artefactos definidos por la metodología de desarrollo, documentan la solución propuesta, lo cual facilita su posterior mantenimiento (actualización o adición de funcionalidades).

## Capítulo 3 Validación de la propuesta solución

### Introducción

En este capítulo se validarán los requisitos y se abordarán las pruebas realizadas al software siguiendo las buenas prácticas de la metodología XP, para garantizar el correcto funcionamiento la calidad del sistema. Una vez concluida la fase de pruebas se podrá definir con exactitud, en qué grado de funcionamiento y rendimiento se encuentra el sistema, compararlo con las exigencias del cliente y determinar si se cumplieron sus expectativas.

#### 3.1 Matriz de trazabilidad

La trazabilidad de requisitos es clave para conseguir una exitosa gestión de los requisitos. Funciona tanto para demostrar que cumplió con los requisitos como para identificar lo que no. Si hay elementos arquitectónicos o código fuente que no se pueden rastrear hasta un requisito, entonces es un riesgo y no debería estar allí(Hamilton, 2020).

Se utiliza la matriz de trazabilidad para validar los requisitos que se identifican en el sistema. Su principal objetivo es comprobar que las Historias de Usuario (HU) obtenidas satisfacen los requisitos identificados en el sistema, verificando así que no existan inconsistencias en el proceso de incluir requisitos en las HU.

En el [Anexo 3](#) se muestra una matriz de trazabilidad donde todas las HU se muestran de forma horizontal y los requisitos funcionales están ubicados de forma vertical, de manera que puedan relacionarse entre ellos. Con la realización de esta tabla, queda evidenciado, que se implementarán todos los requisitos funcionales, ya que cada uno está relacionado con una HU, evitándose así la inconsistencia.

#### 3.2 Fase 5 Pruebas

Las pruebas de software forman parte de la última fase que propone la metodología XP, con el objetivo de lograr una herramienta que cumpla con los requisitos previamente identificados.

Por lo general, las pruebas involucran operaciones del sistema evaluando los resultados bajo condiciones controladas. El proceso de pruebas permite evaluar y verificar la calidad del producto, controlando su correcto funcionamiento según

los requisitos definidos por el cliente, evitando de esta manera el número de errores no deseados, reduciendo los costos para el desarrollo de la aplicación y garantizando una mejora en el rendimiento del sistema.

El proceso de pruebas se realiza en varios niveles, estos se enfocan a determinados objetivos y están estrechamente relacionados con los tipos de pruebas. Seguidamente se muestran algunos tipos de niveles de pruebas para valorar cuáles se van a utilizar:

**Prueba de sistema:** Son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados(Ruiz, 2010).

**Pruebas unitarias:** En este nivel se verifican por separados las piezas de software en un funcionamiento aislado, estas piezas pueden ser módulos individuales, subprogramas y componentes. Pueden llevarse a cabo con herramientas de depuración, acceso al código fuente y pueden participar en esta de forma opcional los programadores(Juristo, Moreno, & Vegas, 2006).

**Pruebas de Aceptación:** Son realizadas por el usuario final en lugar del responsable del desarrollo del sistema. Una prueba de aceptación puede ir desde un informal paso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas. Además, se llevan a cabo con el objetivo de que el cliente valide todos los requisitos que debe poseer el sistema(digite, 2021).

### 3.2.1 Pruebas unitarias

Las pruebas unitarias son la verificación del código y son ejecutadas por los desarrolladores, pues son los encargados de revisar el código directamente. El objetivo de estas pruebas es obtener un código limpio y conciso, que cumpla con cada una de las funcionalidades a las que tributan.

Como técnica de prueba, se utilizó la técnica de camino básico, la cual se realizó mediante el diseño de casos de prueba los cuales se centraron en obtener una medida de la complejidad lógica del diseño procedural a utilizar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

Para aplicar la técnica se debe introducir la notación para la representación del flujo de control, este puede representarse por un grafo de flujo en el cual:

- Cada nodo del grafo corresponde a una o más sentencias de código fuente.
- Todo segmento de código de cualquier programa se puede traducir a un grafo de flujo.
- Se calcula la complejidad ciclomática del grafo

Un grafo de flujo está formado por 3 componentes fundamentales que ayudan a su elaboración y comprensión:

**Nodos:** son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.

**Aristas:** son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aun cuando el nodo no representa la sentencia de un procedimiento.

**Regiones:** son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento.

Para realizar la técnica planteada, específicamente la técnica del camino básico es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar.

$$V(G) = (\text{Aristas} - \text{Nodos}) + 2$$

$$V(G) = (13-11)+2 = 4$$

```

366 private void saveLibro(LibroForm.SaveEvent event) {
367
368     List<Libro> listLibros = libroService.findAll();
369
370     listLibros = listLibros.parallelStream()
371         .filter(lib -> event.getLibro().getImagen().equals(lib.getImagen())
372             && event.getLibro().getTitulo().equals(lib.getTitulo())
373             && event.getLibro().getAutor().equals(lib.getAutor())
374             && (event.getLibro().getVolumen() != null && event.getLibro().getVolumen().equals(lib.getVolumen()))
375             && (event.getLibro().getTomo() != null && event.getLibro().getTomo().equals(lib.getTomo()))
376             && (event.getLibro().getParte() != null && event.getLibro().getParte().equals(lib.getParte()))
377             && event.getLibro().getCantidad().equals(lib.getCantidad())
378             && event.getLibro().getPrecio().equals(lib.getPrecio()))
379         .collect(Collectors.toList());
380
381     if (listLibros.size() != 0) {
382         Notification notification = Notification.show(
383             text: "El libro ya existe",
384             duration: 2000,
385             Notification.Position.MIDDLE);
386         notification.addThemeVariants(NotificationVariant.LUMO_ERROR);
387     } else {
388         if (event.getLibro().getId() == null) {
389             libroService.save(event.getLibro());
390             Notification notification = Notification.show(
391                 text: "Libro añadido",
392                 duration: 5000,
393                 Notification.Position.BOTTOM_START);
394             notification.addThemeVariants(NotificationVariant.LUMO_SUCCESS);
395         } else {
396             libroService.update(event.getLibro());
397             Notification notification = Notification.show(
398                 text: "Libro modificado",
399                 duration: 5000,
400                 Notification.Position.BOTTOM_START);
401             notification.addThemeVariants(NotificationVariant.LUMO_SUCCESS);
402         }
403         toolbar.remove(total);
404         total = new Html("<span>Total: <b>" + libroService.count() + "</b> libros</span>");
405         toolbar.addComponentAtIndex(index: 1, total);
406         toolbar.setFlexGrow(flexGrow: 1, buttons);
407         updateList();
408         closeEditor();
409     }
410 }
411

```

Figura 10 Código del método de prueba salvar libro

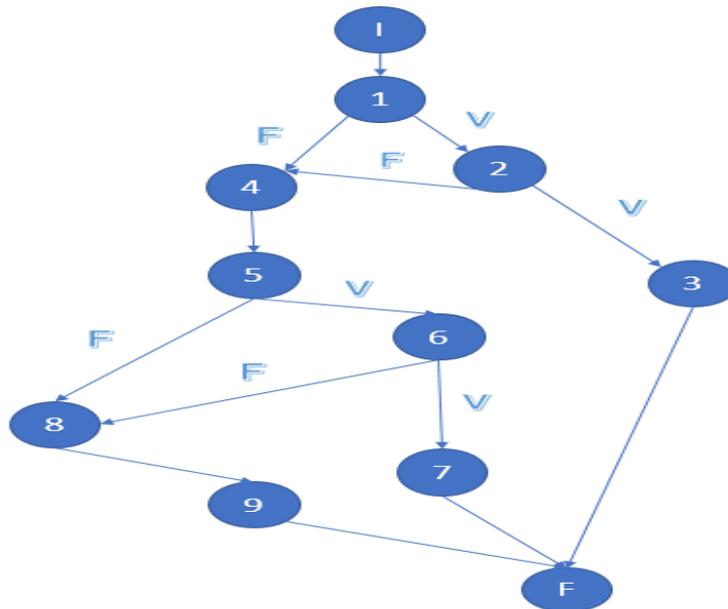


Figura 11 Grafo de flujo

A partir del resultado obtenido se determina que la funcionalidad función saveLibro(), presenta una complejidad ciclomática de 4, lo que deriva que existe a lo sumo 4 caminos lógicos por donde ejecutarse dicha funcionalidad, por lo que es de poco riesgo para el sistema. En la [Tabla 15](#) se muestran los caminos básicos.

Tabla 15 Técnica del camino básico

No.	Camino básico
1	I-1-2-3-F
2	I-1-2-4-5-6-7-F
3	I-1-2-4-5-6-8-9-F

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

Tabla 16 Técnica del camino básico caso de prueba No. 1

Caso de prueba	
Camino	I-1-2-3-F
Descripción:	Verificar si existe el libro
Condiciones de ejecución	Tiene que estar autenticado Tiene que tener los permisos pertinentes Debe verificar si el libro existe
Entrada	Se introducen todos los datos válidos, correspondientes a un elemento guardado anteriormente
Resultados Esperados	El sistema notifica al usuario, que el libro que desea añadir ya existe

Tabla 17 Técnica del camino básico caso de prueba No. 2

Caso de prueba	
Camino	I-1-2-4-5-6-7-F
Descripción:	Adicionar un nuevo libro
Condiciones de ejecución	Tiene que estar autenticado Tiene que tener los permisos pertinentes

	El libro no debe existir en la BD
Entrada	Se introducen todos los datos válidos
Resultados Esperados	Se adiciona el libro correctamente

Tabla 18 Técnica del camino básico caso de prueba No. 3

Caso de prueba	
Camino	I-1-2-4-5-8-9-F
Descripción:	Modificar un nuevo libro existente
Condiciones de ejecución	Tiene que estar autenticado Tiene que tener los permisos pertinentes El libro debe existir en la BD
Entrada	Se modifica al menos uno de los campos de datos, se introducen nuevos valores válidos
Resultados Esperados	Se adiciona el libro correctamente

### 3.2.2 Pruebas de sistema

La fase de pruebas del sistema tiene como objetivo verificar el sistema software para comprobar si este cumple sus requisitos. Dentro de esta fase pueden desarrollarse varios tipos distintos de pruebas en función de los objetivos de las mismas. Algunos tipos son pruebas funcionales, pruebas de usabilidad, pruebas de rendimiento, pruebas de seguridad, etc.

Para la realización de las pruebas de sistema, se utilizarán listas de chequeos, debido a que estas, constituyen una guía básica y única para el probador en la revisión de los artefactos. Además, constituyen un apoyo en la ejecución de evaluaciones estáticas garantizando una mayor calidad en los artefactos de apoyo a los sistemas desarrollados.

### Pruebas de Usabilidad

Las pruebas de usabilidad demuestran la capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones(ISO 25010/EIC, 2011). Estas brindan un examen imparcial, preciso y directo de la experiencia del usuario del producto o sitio web.

Tabla 19 Lista de chequeo para pruebas de usabilidad

<b>Elementos definidos por la metodología</b>				
<b>No</b>	<b>Indicador a evaluar</b>	<b>Evaluación</b>	<b>NP</b>	<b>Observación</b>
<b>Visibilidad del sistema</b>				
1.	¿La página refleja la identidad de la empresa logos, compañía...)?	<b>no</b>		
2.	¿Cada pantalla empieza con un título que describe su contenido?	<b>no</b>		
3.	¿El logo de la organización está ubicado en el mismo lugar en todas las páginas, y hacer click en el logo retorna al usuario a la página más lógica (Ejemplo: la página de inicio)?	<b>si</b>		
<b>Lenguaje común entre sistema y usuario</b>				
4.	¿Utiliza los conceptos establecidos para las funciones estándar? ("buscar" para las búsquedas, etc.)	<b>si</b>		

5.	¿Evita el lenguaje técnico: términos informáticos o propios de Internet?	si		
6.	¿Se utiliza siempre la misma nomenclatura para las mismas funciones?	si		
<b>Libertad y control por parte del usuario</b>				
7.	¿Existe una manera lógica de acceder a páginas relacionadas o a otras secciones?	si		
8.	¿Tras una acción relevante hay una opción de vuelta atrás?	si		
9.	¿Se pueden guardar las páginas web?	si		
10	¿La caja de búsqueda es suficientemente grande para manejar la longitud de las consultas más comunes?	no		<b>La caja de búsqueda no es suficientemente grande para manejar la longitud de las consultas más comunes, cuando se introduce una oración el sistema emite un error.</b>
11	¿Las búsquedas cubren todo el sitio, no una porción de él?	si		

12	¿La interfaz de búsqueda está ubicada donde los usuarios esperan encontrarla (en la parte superior derecha de la página)?	<b>no</b>		<b>La interfaz de búsqueda no está ubicada donde los usuarios esperan encontrarla (en la parte superior derecha de la página).En la página al inicio aparece a la izquierda y cuando te desplazas en la página hacia abajo aparece a la derecha.</b>
<b>Consistencia y estándares</b>				
13	¿Se usa la misma fuente para todos los navegadores?	<b>si</b>		
14	¿Añade una descripción en las imágenes?	<b>si</b>		
15	¿Señala claramente los vínculos a archivos PDF como tal?	<b>si</b>		
16	¿Especifica el tamaño de los archivos PDF?	<b>No</b>		<b>No se especifica el tamaño de los archivos pdf.</b>
17	¿El cambio de idioma en los textos está identificado?  El texto que se encuentre en otro idioma debe estar en cursiva o en un formato diferente al del resto del texto.	<b>si</b>		
18	¿Es posible la navegación sin ratón?	<b>si</b>		
<b>Estética y diseño minimalista</b>				

19	¿Los tipos y tamaños de letra son legibles y distinguibles?	si		
20	¿No Poseen las páginas animaciones innecesarias?	si		
21	¿El sitio puede ser usado sin desplazamiento horizontal?	si		
22	¿El uso de los colores es moderado?	si		
<b>Prevención de errores</b>				
23	¿Hay ausencia de enlaces rotos o que no lleven a ninguna página?	si		
24	¿Se dan indicaciones para completar campos complejos?	si		
25	¿El buscador (si existe) permite errores tipográficos y ortográficos (tildes)?	si		
<b>Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores</b>				
26	¿Ofrece área de Preguntas Frecuentes con datos de ayuda a usuarios?	si		

27	¿Ofrece páginas de ayuda que explican cómo usar el Sitio?	<b>no</b>		No ofrece páginas de ayuda que explican cómo usar el sitio.
<b>Ayuda y documentación</b>				
28	¿Existe un vínculo a los datos de contacto en un lugar bien visible en todas las páginas web del sitio?	<b>si</b>		
29	¿Información necesaria en página de contacto? (Atención al cliente, Consultas, Soporte técnico, Solicitudes de empleo.)	<b>si</b>		
30	¿Incluye un vínculo a los datos legales en todas las páginas?	<b>si</b>		
<b>Flexibilidad y eficiencia de uso</b>				
31	¿El cursor se desplaza adecuadamente en un formulario al presionar "tabulador"?	<b>si</b>		
32	¿Se implementen validaciones antes de que el usuario envíe información?			No se implementen validaciones antes de que el usuario envíe información.

### Pruebas de seguridad

Las pruebas de seguridad se realizan con la intención de explotar vulnerabilidades en la aplicación o dentro de su ambiente. Se centra en la identificación y evaluación de los peligros potenciales que podrían afectarla

negativamente y que podrían ocasionar que falle todo el sistema. Si los peligros se identifican al principio del proceso del software, las características de su diseño se especifican de modo que los eliminen o controlen (Pressman, 2000).

No	Pregunta	Respuesta
<b>Confidencialidad</b>		
1	¿Se utiliza el protocolo https para garantizar una transferencia de datos segura?	Si
2	¿Los errores del sistema muestran información que no debe ser vista por el usuario? (datos de la consulta a la base de datos, la versión del apache, del sistema operativo y demás software instalados en el servidor de aplicaciones)	No
3	¿La información que muestra el sistema luego de un acceso fallido con credencial incorrectas indica cuál está incorrecto?	No
<b>Integridad</b>		
4	¿El sistema permite inyecciones de código SQL en los formularios? (XXX') OR 1 = 1 --)]	No
<b>Autenticidad</b>		
5	¿El sistema tiene un mecanismo de autenticación para todos los usuarios?	Si
6	¿Los usuarios son las que trae por defecto el lenguaje de programación?	No
7	¿Las contraseñas y los usuarios son adivinables?	No
8	¿El sistema solo permite la creación de contraseñas robustas? (combinación de letras y números sin un significado evidente, con una longitud mínima de seis caracteres)	Si
9	¿El sistema permite a los usuarios cambiar su contraseña?	Si
10	¿El sistema controla el historial de contraseña para prever que no se repitan? (cuando haya transcurrido un tiempo o hasta que no se hayan utilizada otras X cantidad de contraseñas diferentes)	Si
11	¿El sistema tiene implementado un mecanismo que lo proteja ante X número de intentos fallidos por acceder? (bloqueo del IP, usuario, pc o la utilización de CAPTCHA)	Si
12	¿El sistema cierra sesión cuando transcurrió un lapso de tiempo X de inactividad?	Si
13	¿Al salir o cerrar el sistema se destruye el ID de sesión?	Si

14	¿El sistema cifra los identificadores de sesión?	Si
15	¿Un usuario estándar (no administrador) puede modificar sus privilegios en el sistema o los de otros usuarios?	No
16	¿El sistema actualiza inmediatamente la gestión que se realizó sobre los usuarios incluyendo los roles y permisos?	Si
17	¿Se puede acceder directamente a una página protegida a través de la barra de direcciones en tu navegador?	No
18	¿Después de una autenticación exitosa si se copia la URL y se cierra sesión es posible acceder nuevamente con esa URL a la aplicación?	No
19	¿Al cerrar sesión, si se presiona el botón "Atrás" del navegador es posible acceder nuevamente a la aplicación?	No
20	¿Existe un botón de cierre de sesión visible y que esté presente en todas las páginas que requieren autenticación?	Si
21	¿Es posible acceder a un recurso protegido, aunque el usuario no este autenticado?	No
22	¿Es posible acceder a funciones y recursos que deben ser accesibles solo a usuarios que tengan un perfil o privilegios diferentes?	No
23	¿Un usuario administrador es capaz de eliminar su propia cuenta?	No

### 3.2.3 Pruebas de aceptación

Las pruebas de Caja Negra deben su nombre a los elementos que estas revisan y las condiciones en que se hace la revisión. Estas se basan en los requerimientos funcionales del sistema y se llevan a cabo desde el exterior de la aplicación. Este tipo de prueba es importante a la hora de medir el grado de cumplimiento de los requerimientos solicitados por el cliente y se aplican sobre la interfaz de la aplicación, observando las respuestas del sistema ante determinadas acciones(Pressman, 2000).

Las pruebas de aceptación son un nivel de pruebas que emplean en mayor medida técnicas de caja negra, las cuales son definidas por el cliente para cada historia de usuario, estas se realizan antes finalizar cada iteración. El cliente es el encargado de realizar dichas pruebas verificando con distintas entradas de

datos, que el sistema responde a estas adecuadamente, y que cumple con las funcionalidades deseadas.

Para la correcta realización de estas pruebas, se empleó la técnica partición equivalente, la cual es una técnica de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. Un caso de prueba ideal de manejo simple descubre una clase de errores; como, el procesamiento incorrecto de todos los datos de caracteres; que, de otra forma, requeriría la ejecución de muchos casos antes de que se observe el error general. La partición equivalente se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse.

El diseño de casos de prueba para partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleana(Pressman, 2000).

Se realizaron varios casos de prueba usando la técnica mencionada anteriormente, a continuación, se muestra un ejemplo de los mismos:

Tabla 20 Clases de equivalencia

Identificador	Entrada(clase)	Clases válidas	Clases inválidas
<b>Foto</b>	Foto	Foto= .jpg, .png	Foto= otro formato
<b>Título</b>	Título	Título = a-z,0-9	Caracteres especiales
<b>Autor</b>	Autor	Autor = a-z	Autor = 0-9, caracteres especiales
<b>Año</b>	Año	Año = $1 \leq x \leq 5$	Volumen= $x < 1$ , $x > 5$
<b>Asignatura</b>	Asignatura	Asignatura =a-z	Asignatura = 0-9,

			caracteres especiales
<b>Cantidad</b>	Cantidad	Cantidad $x \geq 1$	Cantidad = $x < 1$
<b>Precio</b>	Precio	Precio= $x \geq 0.0$	Precio= $x < 0.0$
<b>Volumen</b>	Volumen	Volumen = $x \geq 1$	Volumen $x < 1$
<b>Tomo</b>	Tomo	Tomo = $x \geq 1$	Tomo $x < 1$
<b>Parte</b>	Parte	Parte = $x \geq 1$	Parte $x < 1$

Tabla 21 Caso de prueba " Añadir libro"

Id del escenario	Descripción	Título	Autor	Asignatura	Año	Precio	Respuesta del sistema	Flujo Central
EC 1.1 Operación Añadir libro	Selecciona la opción añadir un nuevo libro.						Brinda la posibilidad de introducir o seleccionar de manera obligatoria los siguientes datos del Libro: -Título -Autor -Año -Académico -Asignatura -Cantidad -Precio Y de forma opcional: -Imagen -Volumen -Tomo	Libro/Listado de Libros/ Añadir libro

							-Parte Guardar Datos Cancelar la operación en cualquier momento	
EC 1.2 Opción de Guardar los datos.	Introduce y/o selecciona los datos de un libro y selecciona la opción guardar los datos.	Programación orientada a objetos V	Leinier Caraballo V	Introducción a la Programación	1	10.0	Valida los datos. Crea un nuevo libro. Muestra el listado de libros y un mensaje de información	Libro/Listado de Libros/ Añadir libro
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar.						Elimina los datos creados. Regresa al listado de libros.	Libro/Listado de Libros/ Añadir libro
EC 1.4 Datos incompletos		Programación orientada a objetos V	I	Introducción a la Programación V	1 V	10.0 V	Muestra un mensaje de información. Muestra un indicador sobre los campos vacíos.	Libro/Listado de Libros/ Añadir libro
EC 1.5 Datos incorrectos		Programación orientada a objetos V	Leinier Caraballo 98 I	Introducción a la Programación V	0 I	10.0 V	Muestra un mensaje de información. Muestra un indicador sobre los campos incorrectos.	Libro/Listado de Libros/ Añadir libro

### 3.3 Resultados de las pruebas

Como parte de la ejecución de las pruebas, y teniendo en cuenta como objetivo esencial el de identificar en qué medida satisface la aplicación las funcionalidades implementadas, se realizó una primera iteración de pruebas, aplicadas a un total de 10 historias de usuario. Para una primera iteración se identificaron en total de 8 No Conformidades (NC), clasificadas en 3 no significativas y 5 significativas. Una vez corregidas se procedió a realizar una segunda iteración de pruebas donde se identificaron 3 nuevas no conformidades de tipo no significativas y 2 significativas.

Una vez corregidas se procedió a realizar una tercera iteración de pruebas donde se identificaron 1 nueva no conformidad de tipo no significativa y 1 significativa. Luego de corregidas estas deficiencias se procedió a la realización de una cuarta iteración de pruebas obteniendo resultados satisfactorios, por lo que se determina no realizar nuevas iteraciones de pruebas dando por concluidas las mismas. Los resultados obtenidos en las pruebas de aceptación para esta primera iteración pueden verse en la siguiente figura.

## Gráfico de no conformidades

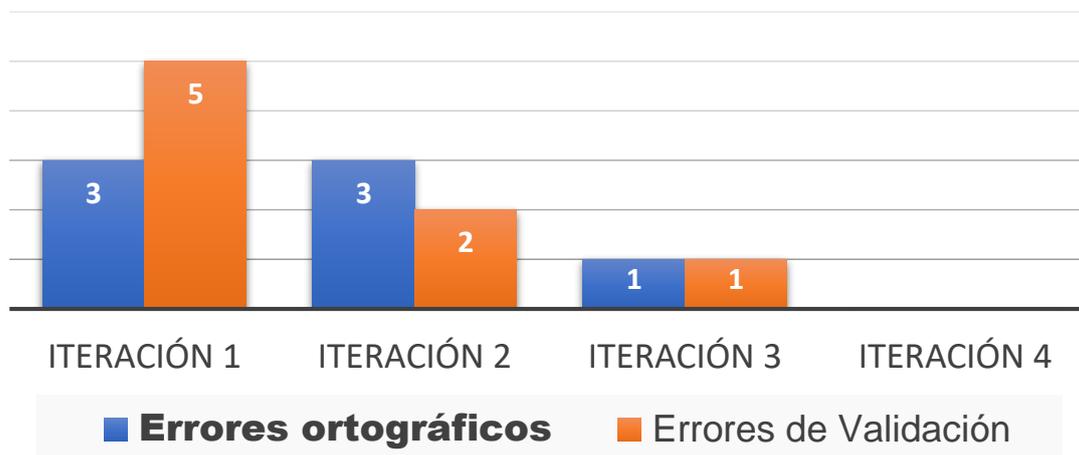


Figura 12 Gráfico de no conformidades

Las principales NC no significativas encontradas fueron errores ortográficos, omisiones de tildes y cambio de mayúsculas por minúsculas. Los elementos de la interfaz, no eran sugerentes para usuarios con poca experiencia. Las principales NC significativas encontradas fueron errores de validación, errores

en validaciones de los campos de los formularios. Después de concluida cada iteración se resolvieron las NC arrojadas quedando en un estado cerrado.

Las pruebas de aceptación fueron realizadas por el cliente, una vez recibido el producto final y su documentación, de conjunto con los desarrolladores del sistema (pruebas de tipo alfa). Este proceso se realizó en el Vicedecanato de administración y economía de la Facultad 4 en presencia de los desarrolladores y los especialistas del área quienes se encargaron de comprobar las funcionalidades y aprobar el producto. La realización de las pruebas de aceptación tuvo como salida la liberación de un acta de aceptación, la cual se encuentra en el [Anexo 4](#) del documento, con la firma de los clientes.

#### Conclusiones parciales

La definición y ejecución de una estrategia de pruebas permitieron concluir que:

- La ejecución de pruebas de software en diferentes niveles permitió comprobar correcta implementación del sistema de gestión de los recursos materiales y libros.
- La aplicación del método de caja blanca, empleando la técnica de camino básico, posibilitó que se identificaran errores existentes en la aplicación y que estos fueran corregidos tempranamente.
- La ejecución de las pruebas de caja negra, a través del método de partición de equivalencia con el diseño de los casos de prueba correspondientes, posibilitó la identificación y eliminación de no conformidades que afectaban moderadamente el funcionamiento de la solución de software construida.
- Los resultados de estas pruebas a la plataforma demostraron su efectividad para ser desplegada.

### **Conclusiones generales**

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

- El estudio del arte sobre los sistemas de gestión enmarcados en los sistemas para la gestión referente a libros y recursos materiales, arrojó que estas soluciones no satisfacen la problemática, lo cual evidencia la necesidad de esta investigación.
- Se desarrolló un sistema informático que contribuirá a la gestión de los recursos materiales y libros en la facultad 4 de la Universidad de las Ciencias Informáticas.
- Las pruebas realizadas, y la corrección de las deficiencias detectadas en cada una de las iteraciones, posibilitaron la terminación exitosa de un producto de calidad.

### **Recomendaciones**

A partir de los resultados obtenidos se recomienda:

- Implementación de las funcionalidades necesarias, para atender los procesos de pago por pérdida de libros, y asignación de materiales a la facultad.

### Bibliografía

- Abellán, P. (2019, mayo 24). Introducción a Vaadin Flow | Tribalyte Technologies. Recuperado 17 de octubre de 2022, de <https://tech.tribalyte.eu/blog-introduccion-vaadin-flow>
- Academia. (2022, febrero 21). ¿Para qué se utiliza JavaScript? Recuperado 16 de octubre de 2022, de Kodigo website: <https://kodigo.org/para-que-se-utiliza-javascript/>
- Acero Linares, J. C., & S.L, C. (2018, noviembre 11). Las Mejores Bases de Datos para tu empresa—Coregistros (octubre2022). Recuperado 17 de octubre de 2022, de Coregistros website: <https://www.coregistros.com/mejores-bases-de-datos/>
- Almacenes, S. S. I. de G. de. (2022). SIGA | Sistema Integral de Gestion de Almacenes. Recuperado 15 de octubre de 2022, de SIGA | Sistema Integral de Gestion de Almacenes website: <https://www.siga.com.es/>
- Alpert, G., & Halton, C. (2022). Enterprise Resource Planning (ERP): Meaning, Components, and Examples. Recuperado 14 de octubre de 2022, de Investopedia website: <https://www.investopedia.com/terms/e/erp.asp>
- Apache. (2022, octubre 3). Apache Tomcat 9 (9.0.68)—Documentation Index. Recuperado 4 de noviembre de 2022, de <https://tomcat.apache.org/tomcat-9.0-doc/index.html>
- Atlassian. (2022). Qué es el control de versiones | Atlassian Git Tutorial. Recuperado 17 de octubre de 2022, de Atlassian website: <https://www.atlassian.com/es/git/tutorials/what-is-version-control>

- Brandao de Andrade, C., Rossete, L., & Reis, G. (2022). Software para la Gestión de Almacenes | SoftExpert Almacén. Recuperado 5 de julio de 2022, de SoftExpert | Soluciones para Excelencia en la Gestión y Conformidad Empresarial website: <https://www.softexpert.com.es/todos-los-productos/gestion-almacenes/>
- Carrizo, D., & Ortiz, C. (2016). Modelos del proceso de educación de requisitos: Un mapeo sistemático. *Ingeniería y Desarrollo*, 34(1), 184-203.
- Chiluisa Pallo, A. P., & Loarte Cajamarca, B. G. (2014). *Desarrollo e implantación del sistema de control de inventarios y gestión de laboratorios para la de la facultad de Ciencias*. Recuperado de <http://bibdigital.epn.edu.ec/handle/15000/7732>
- Coad, P., & Lefebvre, E. (1999). *Java Modeling in Color with UML: Enterprise Components and Process*—Peter Coad, Eric Lefebvre, Jeff De Luca, Eric De Luca—Google Books. Recuperado 10 de noviembre de 2022, de [https://books.google.com.cu/books/about/Java\\_Modeling\\_in\\_Color\\_with\\_UML.html?id=fo0\\_AQAAIAAJ&redir\\_esc=y](https://books.google.com.cu/books/about/Java_Modeling_in_Color_with_UML.html?id=fo0_AQAAIAAJ&redir_esc=y)
- Corredera, P. Á. (2020, mayo 28). Java vs Python, ¿Cuáles son sus diferencias? Recuperado 16 de octubre de 2022, de CiberNinjaS website: <https://ciberninjas.com/java-vs-python-diferencias/>
- Coursera. (2022, junio 9). Java vs. JavaScript: What's the Difference? Recuperado 16 de octubre de 2022, de Coursera website: <https://www.coursera.org/articles/java-vs-javascript>
- Creately. (2020, octubre 26). Tutorial de Diagrama de Despliegue | ¿Qué es un Diagrama de Despliegue. Recuperado 21 de junio de 2022, de Blog

- de Creately website: <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>
- da Silva, D. (2020, noviembre 10). Metodologías Ágiles: Qué son + 4 ventajas de implementarlas. Recuperado 10 de noviembre de 2022, de Zendesk MX website: <https://www.zendesk.com.mx/blog/metodologias-agiles/>
  - Dailyrazor. (2016, junio 7). JSF vs Spring MVC - Comparison Spring MVC vs JSF Frameworks. Recuperado 18 de octubre de 2022, de <https://www.dailyrazor.com/blog/jsf-vs-spring-mvc/>
  - Demestre, D. H., & González, Y. H. (2019). Sistema de Gestión de Información de Recursos Humanos. *Serie Científica de la Universidad de las Ciencias Informáticas*, 12(2), 46-53.
  - digite. (2021, mayo 18). Pruebas De Aceptación: El Qué Y Porqué + Los Tipos Que Hay Que Conocer [2022]. Recuperado 1 de diciembre de 2022, de <https://www.digite.com/es/agile/pruebas-de-aceptacion/>
  - Eclipse, G. (2022). Eclipse GlassFish. Recuperado 4 de noviembre de 2022, de <https://glassfish.org/>
  - Escárcz San Martín, S. (2008). Inventario de Recursos y Medios – Protección Civil – Planes de Emergencias. Recuperado 30 de noviembre de 2022, de <https://proteccioncivil.info/inventario-de-recursos-y-medios/>
  - Figuerola, N. (2014). BPMN vs UML. Los Requerimientos y el Modelo del Negocio. Autor: Norberto Figuerola - PDF Free Download. Recuperado 17 de octubre de 2022, de <https://docplayer.es/1839170-Bpmn-vs-uml-los-requerimientos-y-el-modelo-del-negocio-autor-norberto-figuerola.html>

- González, A., & Cruz, N. (2022). ¿Tradicional o ágil? La metodología ágil como alternativa a la transformación | AXPE Consulting. Recuperado 11 de mayo de 2022, de <https://www.axpe.com/noticias/analisis-y-tendencias/tradicional-o-agil-la-metodologia-agil-como-alternativa-a-la-transformacion/>
- Guerra, C. A. (2021). Obtención de Requerimientos. Técnicas y Estrategia. Recuperado 18 de octubre de 2022, de SG Buzz website: <https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>
- Hamilton, D. (2020, abril 9). Plantillas y ejemplos de matrices de trazabilidad de requisitos. Recuperado 1 de diciembre de 2022, de Parasoft website: <https://es.parasoft.com/blog/requirements-management-and-the-traceability-matrix/>
- Hartman Díaz, Y. (2021, julio 16). Versat Sarasola – Datazucar. Recuperado 30 de junio de 2022, de [https://www.datazucar.cu/?featured\\_item=another-print-package](https://www.datazucar.cu/?featured_item=another-print-package)
- Iglesias, C., Zamora, F., Linares, A., Martínez, S., & Nuñez, C. (2010, marzo). LA BIBLIOTECA VIRTUAL DE LA UNIVERSIDAD DE CIENFUEGOS. Recuperado 28 de mayo de 2022, de <https://www.eumed.net/rev/cccss/07/ifasc.htm>
- ISO 25010/EIC. (2011). ISO 25010. Recuperado 2 de diciembre de 2022, de <https://iso25000.com/index.php/normas-iso-25000/iso-25010?start=3>
- Jethva, H. (2022, agosto 9). Tomcat vs JBoss—What’s the Difference? (Pros and Cons). Recuperado 4 de noviembre de 2022, de

<https://cloudinfrastructureservices.co.uk/tomcat-vs-jboss-whats-the-difference/>

- Juristo, N., Moreno, A. M., & Vegas, S. (2006, octubre 17). Técnicas de evaluación de software—Trabajos de investigación—19528 Palabras. Recuperado 1 de diciembre de 2022, de <https://www.buenastareas.com/ensayos/T%C3%A9cnicas-De-Evaluaci%C3%B3n-De-Software/55102386.html>
- Larman, C. (2005). UML y Patrones Craic Larman | PDF | Objeto (informática) | Programación orientada a objetos. Recuperado 18 de octubre de 2022, de Scribd website: <https://es.scribd.com/document/373179561/UML-y-Patrones-Craic-Larman>
- Maida, E. G., & Pacienza, J. (2018). Metodologías de desarrollo de software. *Tesis de Licenciatura en Sistemas y Computación, 2015. Universidad Católica Argentina (Rosario). Facultad de Química e Ingeniería «Fray Rogelio Bacon»*. Recuperado de <http://repositorioslatinoamericanos.uchile.cl/handle/2250/1657162>
- Montero, B. M. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software.
- Muradas, Y. (2018, junio 5). Qué es Spring Framework y por qué usarlo. Recuperado 18 de marzo de 2022, de OpenWebinars.net website: <https://openwebinars.net/blog/conoce-que-es-spring-framework-y-por-que-usarlo/>

- Noa, L. (2021, agosto 19). Framework: Qué es, para qué sirve y algunos ejemplos. Recuperado 17 de octubre de 2022, de Edix España website: <https://www.edix.com/es/instituto/framework/>
- Olarte Gervacio, L. (2018, febrero 6). Lenguaje de Programación. Recuperado 16 de octubre de 2022, de Conogasi website: <https://conogasi.org/articulos/lenguaje-de-programacion/>
- Parkes, D. (2011, septiembre). Koha—The World’s First Free and Open Source Integrated Library Management System | EIFL. Recuperado 12 de marzo de 2022, de <https://www.eifl.net/resources/koha-worlds-first-free-and-open-source-integrated-library-management-system>
- Pereda, T. (2021, enero 7). Ventajas de un sistema ERP para universidades—Fusionworks. Recuperado 28 de mayo de 2022, de <https://fwpr.com/es/ventajas-de-un-sistema-erp-para-universidades/>
- Pressman, R. S. (2000). *Ingeniería del Software. Un Enfoque Practico*. 810.
- Ruiz. (2010). Las pruebas de software y su importancia | Certus Blog. Recuperado 1 de diciembre de 2022, de <https://www.certus.edu.pe/blog/pruebas-de-software-importancia/>
- Salmerón Navarro, A. (2022). La importancia de las TIC en la educación—MEDAC. Recuperado 28 de mayo de 2022, de <https://medac.es/blogs/sociocultural/las-herramientas-tic-en-la-educacion>
- Schwaber, K., & Beedle, M. (2001, octubre 1). Agile Software Development with Scrum: | Guide books. Recuperado 10 de noviembre de 2022, de <https://dl.acm.org/doi/10.5555/559553>

- Sinnaps. (2020). Metodología XP o Programación Extrema: ¿Qué es y cómo aplicarla? Recuperado 21 de marzo de 2022, de Gestor de proyectos online website: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>
- Sommerville, I. (2005). *Ingeniería de Software—Ian Sommerville 7a Edición*. Recuperado de [https://www.academia.edu/15059886/Ingenieria\\_de\\_Software\\_Ian\\_Sommerville\\_7a\\_Edicion](https://www.academia.edu/15059886/Ingenieria_de_Software_Ian_Sommerville_7a_Edicion)

**Anexos**

**Anexo 1 Descripción de los roles y responsabilidades**

*Tabla 22 Descripción de los roles y responsabilidades*

Roles	Responsabilidades
Administrador	Responsable del control de los usuarios, así como de asignación de roles.
Vicedecano de Administración y Economía	Máximo responsable de controlar el destino final de todos los recursos de la facultad.
Asistente de Control	Responsable de entregar los recursos materiales a los estudiantes y trabajadores. Además, debe mantener actualizada y firmadas las actas de destino final.
Responsable del almacén	Encargado de la entrega, recepción y control de los libros en el almacén.
Usuario (Trabajadores y estudiantes)	Personal de la facultad al que le es entregado los medios materiales o libros.

**Anexo 2 Historias de usuario:**

*Tabla 23 Historia de usuario: Autenticar usuario*

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre del requisito:</b> Autenticar usuario
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> N/A
<b>Descripción:</b> <b>1- Objetivo:</b> <i>Permitir autenticación un usuario en el sistema.</i> <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para autenticarse hay que: - Tener en cuenta los siguientes datos: usuario, contraseña - <i>Estar registrado en la Base de Datos local y en la Base de Datos UCI</i>	

### 3- Comportamientos válidos y no válidos (flujo central y alternos):

Los campos usuario, contraseña son obligatorios.

Usuario: campo de texto que admite caracteres alfabéticos, numéricos, y caracteres especiales.

Contraseña: campo de texto que admite caracteres alfabéticos, numéricos, y caracteres especiales.

### 4- Flujo de la acción a realizar:

- El sistema debe permitir incluir y/o seleccionar los datos para que los usuarios puedan autenticarse.

- Cuando el usuario incluye y/o selecciona correctamente los datos necesarios para autenticarse y selecciona la opción "Iniciar sesión", este accede al sistema satisfactoriamente, y el mismo le muestra todas las opciones a las que tiene acceso según su rol.

- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión.

**Observaciones:** N/A

### Prototipo de interfaz:

**GENIUS**  
Sistema Informático para la gestión de  
información de los recursos materiales  
y libros en la facultad 4.

Usuario (\*)

Contraseña (\*)

Iniciar Sesión

Tabla 24 Historia de usuario: Modificar usuario

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre del requisito:</b> Modificar usuario
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 2 días
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> <i>Permitir modificar el rol del usuario en el sistema.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para modificar los datos de un usuario hay que:</p> <ul style="list-style-type: none"> <li>- Tener en cuenta: rol.</li> <li>- <i>Estar autenticado en el sistema con el rol administrador.</i></li> <li>- <i>Debe existir en el sistema al menos un usuario.</i></li> </ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b> El campo de rol es obligatorio. Rol: campo de selección en el cual solo se puede seleccionar un rol. Los roles del sistema son: administrador, vicedecano de administración y economía, responsable de almacén, asistente del vicedecano de administración y economía, y usuarios (estudiantes y trabajadores).</p> <p><b>4- Flujo de la acción a realizar:</b></p> <ul style="list-style-type: none"> <li>- El sistema debe permitir modificar un usuario, esta acción puede realizarse seleccionando la opción editar en el listado de usuarios.</li> <li>- Cuando el administrador modifica de forma correcta los datos necesarios y selecciona la opción añadir, se muestra un mensaje de información afirmando que el usuario fue modificado de forma correcta.</li> <li>- Si selecciona la opción Cancelar regresará a la vista previa.</li> </ul>	
<b>Observaciones:</b> N/A	
<b>Prototipo de interfaz:</b>	

Tabla 25 Historia de usuario: Eliminar usuario

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre del requisito:</b> Eliminar usuario
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 2 días
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> <i>Permitir eliminar un usuario en el sistema.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para eliminar un usuario hay que:</p> <ul style="list-style-type: none"> <li>- <i>Estar autenticado en el sistema con el rol administrador.</i></li> <li>- <i>Debe existir en el sistema al menos un usuario.</i></li> </ul> <p><b>3- Flujo de la acción a realizar:</b></p> <ul style="list-style-type: none"> <li>- <i>El sistema permite eliminar una o varios usuarios, para esto el sistema muestra un listado usuarios permitiendo marcar más de un elemento a eliminar, luego se debe seleccionar la opción eliminar de las acciones en lote.</i></li> <li>- <i>En caso de querer eliminar un elemento se puede realizar seleccionando la opción eliminar de las opciones que muestra el propio elemento o desde la vista previa del mismo. Para ambos casos el sistema muestra un mensaje de confirmación para Aceptar o no la acción que se está realizando.</i></li> <li>- <i>Si selecciona la opción eliminar, se actualiza el listado de usuarios y el sistema muestra un mensaje de información.</i></li> </ul>	
<b>Observaciones:</b> N/A	
<b>Prototipo de interfaz:</b>	

Tabla 26 Historia de usuario: Listar usuario

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre del requisito:</b> Listar usuario

<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1 día
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 1 día
<b>Descripción:</b> <b>1- Objetivo:</b> <i>Permitir listar los datos de los usuarios en el sistema.</i> <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para listar usuarios hay que: - <i>Estar autenticado en el sistema con el rol administrador.</i> - <i>Debe existir en el sistema al menos un usuario.</i> <b>3- Flujo de la acción a realizar:</b> <i>Cuando el administrador selecciona la vista de usuarios, aparecen todos los usuarios registrados en el sistema. El administrador tiene la posibilidad de modificarlos y eliminarlos.</i>	
<b>Observaciones:</b> N/A	
<b>Prototipo de interfaz:</b>	

Tabla 27 Historia de usuario: Filtrar usuario

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre del requisito:</b> Filtrar usuario
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 2 días
<b>Descripción:</b> <b>1- Objetivo:</b> <i>Permitir filtrar un listado de usuarios en el sistema.</i> <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para filtrar un listado de usuarios hay que: - Tener en cuenta los siguientes datos: nombre y rol.	

- *Estar autenticado en el sistema con el rol administrador.*
- *Debe existir en el sistema al menos un usuario.*

**3- Flujo de la acción a realizar:**

Cuando el administrador inserta un criterio de búsqueda en alguno de los campos de texto, el sistema muestra un listado de los usuarios que cumplen con los criterios establecidos.

**Observaciones:** N/A

**Prototipo de interfaz:**

Tabla 28 Historia de usuario "Modificar libro"

Historia de Usuario	
<b>Número: 7</b>	<b>Nombre del requisito:</b> Modificar libro
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 2 días
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> <i>Permitir modificar datos un libro en el sistema.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para modificar los datos de un libro hay que:</p> <ul style="list-style-type: none"> <li>- Tener en cuenta los siguientes datos: código, título, autor, año académico, asignatura, cantidad, precio, imagen, tomo, volumen, parte.</li> <li>- <i>Estar autenticado en el sistema con el rol responsable del almacén.</i></li> <li>- <i>Debe existir en el sistema al menos un libro.</i></li> </ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b> Los campos título, autor, año académico, asignatura, cantidad, precio son obligatorios. Código: campo de texto que solo admite números con un máximo de 10 caracteres. Título: campo de texto que admite caracteres alfabéticos, numéricos, y caracteres especiales. Tiene un mínimo de 2 caracteres y un máximo de hasta</p>	

255 caracteres.

Autor: campo de texto que solo admite caracteres alfabéticos. Tiene un mínimo de 2 caracteres y un máximo de hasta 255 caracteres.

Año académico: campo numérico, que solo admite valores entre 1 y 5.

Asignatura: campo de selección. El usuario podrá escribir una parte de la palabra o la palabra completa, referente a la asignatura que desea y esta, será filtrada.

Cantidad: campo numérico, que solo admite valores mayores a 0.

Imagen: campo de selección. El usuario podrá buscar una imagen en su PC, la cual no deberá pesar más de 1MB.

Tomo: campo numérico, que solo admite valores entre 1 y 10.

Volumen: campo numérico, que solo admite valores entre 1 y 10.

Parte: campo numérico, que solo admite valores entre 1 y 10.

#### **4- Flujo de la acción a realizar:**

- El sistema debe permitir modificar un libro, esta acción puede realizarse seleccionando la opción editar en el listado de libros.
- Cuando el usuario modifica de forma correcta los datos necesarios y selecciona la opción "Añadir", se muestra un mensaje de información notificando que el libro fue modificado de forma correcta.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión.
- Si selecciona la opción Cancelar regresará a la vista previa.

**Observaciones:** N/A

**Prototipo de interfaz:**

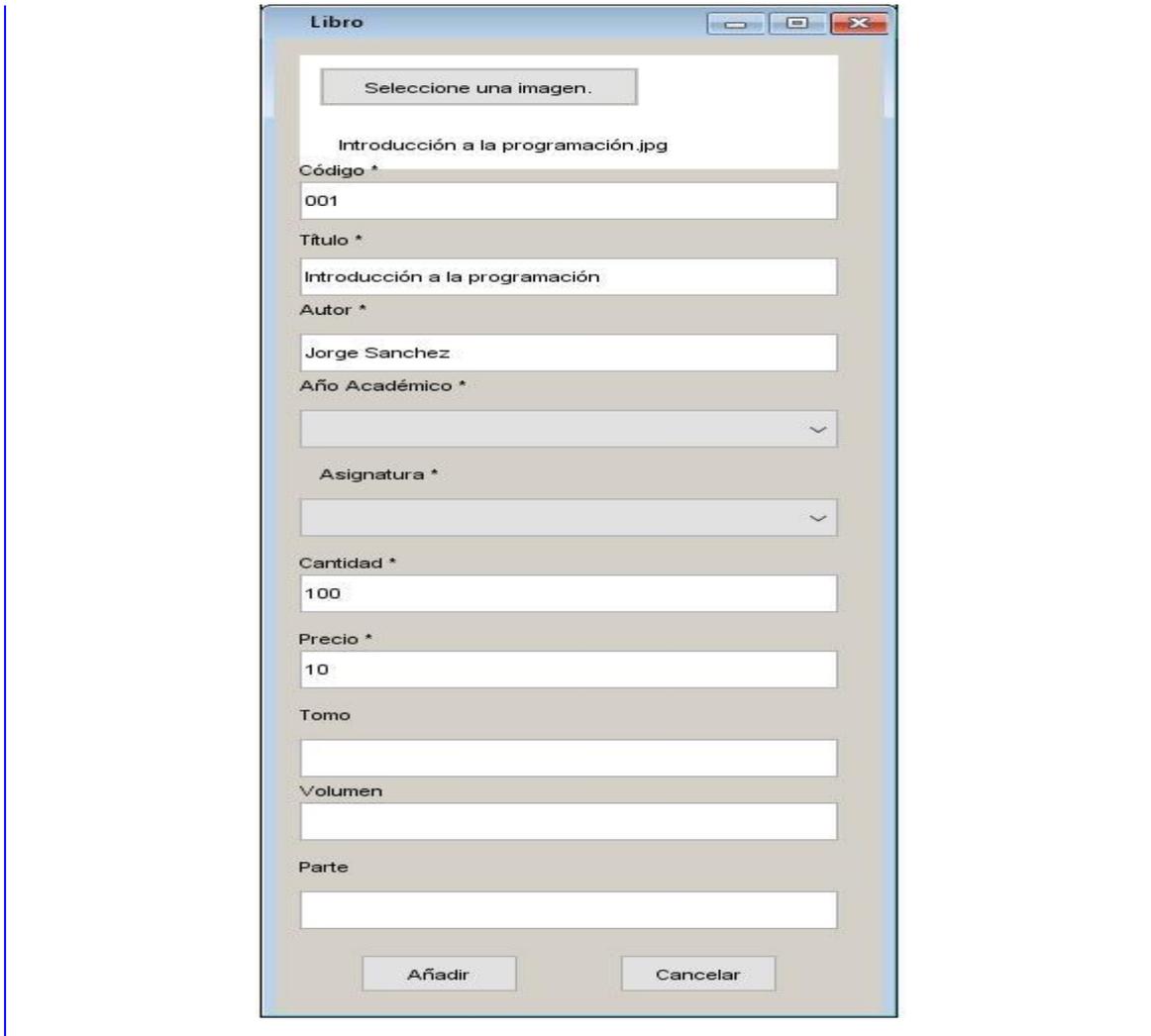


Tabla 29 Historia de usuario "Eliminar libro"

Historia de Usuario	
<b>Número: 8</b>	<b>Nombre del requisito:</b> Eliminar libro
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 2da
<b>Prioridad:</b> Libro	<b>Tiempo Estimado:</b> 2 día
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 1 día
<b>Descripción:</b> <b>1- Objetivo:</b> <i>Permitir eliminar un libro en el sistema.</i> <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para eliminar un libro hay que:	

- Estar autenticado en el sistema con el rol responsable de almacén.
- Debe existir en el sistema al menos un libro.

**3- Flujo de la acción a realizar:**

- El sistema permite eliminar uno o varios libros, para eliminar uno o varios libros el sistema se muestra un listado de los mismos permitiendo marcar más de un elemento a eliminar, y luego seleccionar la opción eliminar.
- Una vez eliminado el libro de sistema se actualiza el listado de libros y el sistema muestra un mensaje de información.

**Observaciones:** si se presiona la opción de eliminar sin haber seleccionado al menos un libro en el listado, el sistema muestra un mensaje de error notificando al usuario que debe seleccionar al menos uno.

**Prototipo de interfaz:**



Tabla 30 Historia de usuario "Listar/Mostrar Libro"

Historia de Usuario	
<b>Número:</b> 9	<b>Nombre del requisito:</b> Listar libro
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 2da
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 1 día
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 1 día
<b>Descripción:</b>	
<b>1- Objetivo:</b>	

Permitir listar los datos de los libros en el sistema.

**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para listar los libros hay que:

- *Estar autenticado en el sistema con el rol responsable de almacén.*
- *Debe existir en el sistema al menos un libro.*

**3- Flujo de la acción a realizar:**

*Cuando el usuario selecciona la opción Libro aparecen todos los libros registrados en el sistema. Además, el usuario tiene la posibilidad de ver detalles de un libro, modificarlo y eliminarlo.*

**Observaciones:** N/A

**Prototipo de interfaz:**



Tabla 31 Historia de usuario: Filtrar libro

Historia de Usuario	
<b>Número:</b> 10	<b>Nombre del requisito:</b> Filtrar libro
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 2 días
<b>Descripción:</b>	
<b>1- Objetivo:</b>	

*Permitir filtrar un listado de libros en el sistema.*

**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para filtrar un listado de libros hay que:

- Tener en cuenta los siguientes datos: *código, título, autor, año académico, asignatura, cantidad, precio, imagen, volumen, tomo, parte.*
- *Estar autenticado en el sistema con el rol responsable de almacén.*
- *Debe existir en el sistema al menos un libro.*

**3- Flujo de la acción a realizar:**

Cuando el responsable de almacén inserta un criterio de búsqueda en alguno de los campos de texto, el sistema muestra un listado de los libros que cumplen con los criterios establecidos.

**Observaciones:** N/A

**Prototipo de interfaz:**

Tabla 32 Historia de usuario: Añadir tarjeta de préstamo

Historia de Usuario	
<b>Número:</b> 11	<b>Nombre del requisito:</b> Añadir tarjeta de préstamo
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 2da
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 3 días
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> <i>Permitir incluir nuevas tarjetas de préstamos para controlar la entrada y salida de libros del almacén.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para incluir una tarjeta de préstamo hay que:</p> <ul style="list-style-type: none"> <li>- Tener en cuenta los siguientes datos: estudiante/trabajador, libro, fecha de entrega.</li> <li>- <i>Estar autenticado en el sistema con el rol responsable de almacén.</i></li> </ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b> Los campos estudiante/trabajador, libro, fecha de entrega son obligatorios.</p>	

Estudiante/Trabajador: campo de selección en el que se muestra los nombres de los estudiantes/trabajadores que se encuentran disponibles en la base de datos del sistema.

Libro: campo de selección en el que se muestra el nombre de los libros registrados en el sistema.

Fecha de entrega: campo de selección, el cual deberá iniciar en la fecha en la que fue seleccionado.

**4- Flujo de la acción a realizar:**

- El sistema debe permitir incluir y/o seleccionar los datos para incluir una nueva tarjeta de préstamo.
- Cuando el responsable de almacén incluye y/o selecciona correctamente los datos necesarios para incluir una tarjeta de préstamo y selecciona la opción Añadir, se crea un nuevo elemento y el sistema muestra un mensaje de información.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al responsable de almacén de realizar nuevamente la acción en cuestión.
- Si selecciona la opción Cancelar regresará a la vista previa.

**Observaciones:** N/A

**Prototipo de interfaz:**

Tabla 33 Historia de usuario: Modificar tarjeta de préstamo

Historia de Usuario	
<b>Número:</b> 12	<b>Nombre del requisito:</b> Modificar tarjeta de préstamo
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 2da
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 2 días
<b>Descripción:</b>	
<b>1- Objetivo:</b>	
<i>Permitir modificar datos de una tarjeta de préstamo en el sistema.</i>	
<b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>	

*Para modificar los datos de una tarjeta de préstamo hay que:*

- *Tener en cuenta los siguientes datos: estudiante/trabajador, libro, fecha de entrega, fecha de devolución.*
- *Estar autenticado en el sistema con el rol responsable de almacén.*
- *Debe existir en el sistema al menos una tarjeta de préstamo.*

**3- Comportamientos válidos y no válidos (flujo central y alternos):**

Los campos: estudiante/trabajador, libro, fecha de entrega, fecha de devolución son obligatorios.

Estudiante/Trabajador: campo de selección en el que se muestra los nombres de los estudiantes/trabajadores que se encuentran disponibles en la base de datos del sistema.

Libro: campo de selección en el que se muestra el nombre de los libros registrados en el sistema.

Fecha de entrega: campo de selección, el cual deberá iniciar en la fecha en la que fue seleccionado.

Fecha de devolución: campo de selección, el cual deberá iniciar en la fecha en la que fue seleccionado.

**4- Flujo de la acción a realizar:**

- El sistema debe permitir modificar una tarjeta de préstamo, esta acción puede realizarse seleccionando la opción editar en el listado de tarjeta de préstamo.
- Cuando el responsable de almacén modifica de forma correcta los datos necesarios y selecciona la opción Añadir, se muestra un mensaje de información confirmando que la tarjeta de préstamo fue modificada de forma correcta.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al responsable de almacén de realizar nuevamente la acción en cuestión.
- Si selecciona la opción Cancelar regresará a la vista previa.

**Observaciones: N/A**

**Prototipo de interfaz:**

Tabla 34 Historia de usuario: Eliminar tarjeta de préstamo

Historia de Usuario
---------------------

<b>Número:</b> 13	<b>Nombre del requisito:</b> Eliminar tarjeta de préstamo	
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 2da	
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 2 días	
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 2 días	
<b>Descripción:</b> <b>1- Objetivo:</b> <i>Permitir eliminar una tarjeta de préstamo en el sistema.</i> <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para eliminar una tarjeta de préstamo hay que: - <i>Estar autenticado en el sistema con el rol responsable.</i> - <i>Debe existir en el sistema al menos una tarjeta de préstamo.</i> <b>3- Flujo de la acción a realizar:</b> - <i>El sistema permite eliminar una o varias tarjetas de préstamo, para esto el sistema muestra un listado de las mismas permitiendo marcar más de un elemento a eliminar, luego se debe presionar la opción eliminar del lote las acciones.</i> - <i>Una vez eliminada la tarjeta de préstamo, el sistema muestra un mensaje de información confirmando la acción realizada.</i>		
<b>Observaciones:</b> N/A		
<b>Prototipo de interfaz:</b>		

Tabla 35 Historia de usuario: Listar tarjeta de préstamo

Historia de Usuario		
<b>Número:</b> 14	<b>Nombre del requisito:</b> Listar tarjeta de préstamo	
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 1era	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 1 día	
<b>Descripción:</b>		

**1- Objetivo:**

*Permitir listar los datos de las tarjetas de préstamo en el sistema.*

**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para listar las *tarjetas de préstamo* hay que:

- *Estar autenticado en el sistema con el rol responsable de almacén.*
- *Debe existir en el sistema al menos una tarjeta de préstamo.*

**3- Flujo de la acción a realizar:**

*Cuando el responsable de almacén selecciona la opción Tarjeta de préstamo aparece un listado con todos los estudiantes y trabajadores registrados en el sistema. Luego el usuario debe seleccionar el botón tarjeta, el cual le dará acceso a la tarjeta personal de préstamo del estudiante/trabajador, en cual se encuentran todos los préstamos recibidos a este. Además, el responsable de almacén tiene la posibilidad de modificarlo y eliminarlo.*

**Observaciones: N/A**

**Prototipo de interfaz:**

Tabla 36 Historia de usuario: Filtrar tarjeta de préstamo

<b>Número:</b> 15	<b>Nombre del requisito:</b> Filtrar tarjeta de préstamo
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 2da
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 2 días
<b>Descripción:</b>	
<b>1- Objetivo:</b>	
<i>Permitir filtrar un listado de tarjetas de préstamo en el sistema.</i>	
<b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>	
Para filtrar un listado de tarjetas de préstamo hay que:	
<ul style="list-style-type: none"> <li>- Tener en cuenta los siguientes datos: estudiante/trabajador, libro, fecha de entrega, fecha de devolución.</li> <li>- <i>Estar autenticado en el sistema con el rol responsable de almacén.</i></li> <li>- <i>Debe existir en el sistema al menos una tarjeta de préstamo.</i></li> </ul>	

**3- Flujo de la acción a realizar:**

Cuando el responsable de almacén inserta un criterio de búsqueda en alguno de los campos de texto, el sistema muestra un listado de las tarjetas de préstamo que cumplen con los criterios establecidos.

**Observaciones: N/A**

**Prototipo de interfaz:**

Tabla 37 Historia de usuario: Visualizar estadísticas de almacén

<b>Número:</b> 16	<b>Nombre del requisito:</b> Visualizar estadísticas de almacén
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 2da
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 3 días
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b>  <i>Permitir consultar informaciones estadísticas de los libros que se encuentran en el base de datos en el sistema.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>            Para ver las estadísticas de almacén hay que:  <i>- Estar autenticado en el sistema con el rol responsable de almacén.</i></p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b>            Debe haberse añadido un libro con anterioridad, en la vista podrán verse los siguientes datos: cantidad de real de libros, cantidad de libros actualmente en el almacén, cantidad de libros prestados a estudiantes y trabajadores.</p> <p><b>4- Flujo de la acción a realizar:</b>  <i>Una vez accedido a la página de estadísticas, se muestra al responsable de almacén, una gráfica de pastel en la que se representa cantidad de real de libros, cantidad de libros actualmente en el almacén, cantidad de libros prestados a estudiantes y trabajadores.</i></p>	
<b>Observaciones:</b> N/A	

**Prototipo de interfaz:**

Tabla 38 Historia de usuario: Exportar reporte de estadísticas de almacén

Historia de Usuario	
<b>Número:</b> 17	<b>Nombre del requisito:</b> Exportar reporte de estadísticas de almacén.
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 3ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> 2 días
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b>  <i>Permitir exportar un reporte con cantidad real de libros existentes en el almacén actualmente, la cantidad de libros prestados a estudiantes y trabajadores, y la cantidad de libros disponibles actualmente en el almacén.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>            Para generar el reporte de estadísticas de almacén hay que:            - <i>Estar autenticado en el sistema con el rol responsable de almacén.</i>            - <i>Debe existir en el sistema al menos un libro.</i></p> <p><b>3- Flujo de la acción a realizar:</b>  <i>Cuando el usuario selecciona la opción Generar reporte, se muestre un listado de informaciones con los siguientes datos:</i>            - <i>Nombre de la Facultad</i>            - <i>Cantidad de real de libros existentes en el almacén</i>            - <i>Cantidad de real de libros prestados a estudiantes y trabajadores</i>            - <i>Cantidad de real de libros actualmente en el almacén</i></p>	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tabla 39 Historia de usuario: Visualizar tarjeta personal de préstamo

Historia de Usuario
---------------------

<b>Número:</b> 18	<b>Nombre del requisito:</b> Visualizar tarjeta personal de préstamo		
<b>Programador:</b> Leinier Caraballo Yanes	<b>Iteración Asignada:</b> 2da		
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 2 días		
<b>Riesgo en Desarrollo:</b>	<b>Tiempo Real:</b> 2 días		
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> <i>Permitir consultar los datos de la tarjeta de préstamo del usuario autenticado.</i></p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para ver detalles de la tarjeta personal de préstamo hay que: - <i>Estar autenticado en el sistema con el rol usuario.</i></p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b> Debe haberse realizado algún préstamo de libros al usuario autenticado con anterioridad, de la tarjeta personal de préstamos de este usuario pueden verse los siguientes datos: libro, fecha de entrega, fecha de devolución.</p> <p><b>4- Flujo de la acción a realizar:</b> <i>Una vez accedido a la página de tarjeta personal, se muestra al usuario un listado con los préstamos que le han sido asignados en el sistema.</i></p>			
<b>Observaciones:</b> N/A			
<b>Prototipo de interfaz:</b>			

### Anexo 3 Matriz de trazabilidad de requisitos:

Tabla 40 Matriz de trazabilidad: Requisito funcional vs Historia de usuario

	HU Autenticar usuario	HU Modificar usuario	HU Eliminar usuario	HU Listar usuario
RF1 Autenticar usuario	X			

RF2 Modificar datos de usuario		X		
RF3 Eliminar usuario			X	
RF4 Listar usuario				X
	<b>HU Filtrar usuario</b>	<b>HU Añadir libro</b>	<b>HU Modificar libro</b>	<b>HU Eliminar libro</b>
RF5 Filtrar listado de usuarios	X			
RF6 Añadir libro		X		
RF7 Modificar libro			X	
RF8 Eliminar libro				X
	<b>HU Listar libro</b>	<b>HU Filtrar libro</b>	<b>HU Añadir tarjeta de préstamo</b>	<b>HU Modificar tarjeta de préstamo</b>
RF9 Listar libro	X			
RF10 Filtrar listado de libros		X		
RF11 Añadir tarjeta de préstamo			X	
RF12 Modificar				X

tarjeta de préstamo				
	<b>HU</b> Eliminar tarjeta de préstamo	<b>HU</b> Listar tarjeta de préstamo	<b>HU</b> Filtrar tarjeta de préstamo	<b>HU</b> Visualizar estadísticas de almacén
RF13 Eliminar tarjeta de préstamo	<b>X</b>			
RF14 Listar tarjeta de préstamo		<b>X</b>		
RF15 Filtrar tarjeta de préstamo			<b>X</b>	
RF16 Visualizar estadísticas de almacén				<b>X</b>
	<b>HU</b> Exportar reporte de estadísticas de almacén	<b>HU</b> Añadir recurso material	<b>HU</b> Modificar recurso material	<b>HU</b> Eliminar recurso material
RF17 Crear Reporte estadísticas de almacén	<b>X</b>			
RF18 Añadir recurso material		<b>X</b>		
RF19 Modificar			<b>X</b>	

recurso material				
RF20 Eliminar recurso material				X
	<b>HU</b> Listar recurso material	<b>HU</b> Filtrar recurso material	<b>HU</b> Añadir módulo	<b>HU</b> Modificar módulo
RF21 Listar recurso material	X			
RF22 Filtrar recurso material		X		
RF23 Añadir módulo			X	
RF25 Modificar módulo				X
	<b>HU</b> Eliminar módulo	<b>HU</b> Listar módulo	<b>HU</b> Filtrar módulo	<b>HU</b> Añadir tarjeta de destino final
RF25 Eliminar módulo	X			
RF26 Listar módulo		X		
RF27 Filtrar módulo			X	
RF28 Añadir tarjeta de destino final				X

	<b>HU</b> Modificar tarjeta de destino final	<b>HU</b> Eliminar tarjeta de destino final	<b>HU</b> Listar tarjeta de destino final	<b>HU</b> Filtrar tarjeta de destino final
<b>RF29</b> Modificar tarjeta de destino final	X			
<b>RF30</b> Eliminar tarjeta de destino final		X		
<b>RF31</b> Listar tarjeta de destino final			X	
<b>RF32</b> Filtrar tarjeta de destino final				X
	<b>HU</b> Visualizar tarjeta personal de préstamo	<b>HU</b> Notificar recogida de módulo	<b>HU</b> Filtrar estudiante	<b>HU</b> Filtrar trabajador
<b>RF33</b> Visualizar tarjeta personal de préstamo	X			
<b>RF34</b> Notificar recogida de módulo		X		
<b>RF35</b> Filtrar estudiante			X	

<b>RF36 Filtrar trabajador</b>				<b>X</b>
	<b>HU Visualizar catálogo de libros</b>	<b>HU Visitar sitios de interés</b>	<b>HU Ver / Ocultar columnas libro</b>	<b>HU Ver / Ocultar columnas recurso material</b>
<b>RF37 Visualizar catálogo de libros</b>	<b>X</b>			
<b>RF38 Visitar sitios de interés.</b>		<b>X</b>		
<b>RF39 Ver / Ocultar columnas libro</b>			<b>X</b>	
<b>RF40 Ver / Ocultar columnas recurso material</b>				<b>X</b>

**Anexo 4 Acta de aceptación**



**Anexo 5: Entrevista realizada a: Mayra Claribel Texidor Ferral: Responsable del almacén correspondiente a la Facultad 4. Fecha: 21/03/2022**

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

- 1- ¿Descripción del flujo de pasos de los distintos procesos de préstamos, devoluciones, pérdidas, modelo de no tenencia?
- 2- ¿Cuál es el tratamiento que tiene el proceso de pérdida de un libro y como se realiza el pago?

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

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

#### **Anexo 6:** Guía de observación del proceso

#### **Guía de Observación de los procesos del negocio**

Con esta guía de observación se pretende describir cómo se lleva a cabo los procesos de entrada y salida de libros del almacén, y los procesos de distribución y control de destino final de los recursos materiales en la Facultad 4.

Criterios de observación:

- Actividades que realizan.
- Relaciones que se establecen entre las actividades
- Productos de trabajo que se generan en cada actividad.
- Roles que intervienen.
- Responsabilidades asociadas a los roles.
- Tiempo de duración de cada actividad.
- Principales dificultades presentadas.