



**FACULTAD 4**

# **Sistema de gestión para el registro de las lecciones aprendidas en la Universidad de las Ciencias Informáticas**

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

**Autor:**

Yeni Carmen Tamayo Morales

Maribel Kissis Valdés Barrios

**Tutor:**

M.Sc. Wisleidys Campos Wright

Dra.C. Yaimí Trujillo Casañola

**La Habana, noviembre del 2023**

**Año 65 de la Revolución**

## DECLARACIÓN DE AUTORÍA

Los autores del trabajo de diploma con título “**Sistema de gestión para el registro de las lecciones aprendidas en la Universidad de Ciencias Informáticas**” concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declaran como únicos autores de su contenido. Para que así conste firma la presente al día 24 de noviembre del año 2023.



---

Yeni Carmen Tamayo Morales  
**Firma del Autor**



---

Maribel Kissis Valdés Barrios  
**Firma del Autor**



---

MSc. Wisleidys Campos Wright  
**Firma del Tutor**

---

Dra. C PT Yaimí Trujillo Casañola

**Firma del Tutor**

## **DATOS DE CONTACTO**

Wisleidys Campos Wright, graduada de Ingeniería en Ciencias Informáticas en el año 2010, Máster en Calidad de Software desde el 2021. Especialista Superior y Sub-directora de la Dirección de Calidad de software. Miembro del proyecto Mejora de procesos de software con el Modelo de Capacidad y Madurez Integrado e ISO 9001:2015 basado en la experiencia cubana.

correo electrónico: [wcampos@uci.cu](mailto:wcampos@uci.cu)

Yaimí Trujillo Casañola, Ingeniera informática del Instituto Superior Politécnico José Antonio Echeverría desde julio del 2004 y Máster en Gestión de Proyectos Informáticos de la Universidad de las Ciencias Informáticas desde julio del 2007. Se desempeña como Especialista General de la Dirección de Calidad de la Infraestructura Productiva, miembro del grupo de Auditoría y Revisiones, ha participado como ponente en eventos científicos nacionales e internacionales de la rama. Ha ejercido como tutora de varias tesis de grado, así como ponente y miembro de tribunales. Ha trabajado en proyectos productivos como Programadora, Analista, Diseñadora y Líder de proyecto obteniendo software utilizados en empresas del país.

Correo electrónico: [yaimi@uci.cu](mailto:yaimi@uci.cu)

## AGRADECIMIENTOS

Yeni Carmen Tamayo Morales

*A dios por permitirme cumplir esta meta tan importante en mi vida.*

*A mis padres Yaumara y Sergio, por todo el sacrificio y amor incondicional en toda mi vida y en especial en mi etapa universitaria, que a pesar de la distancia no han dejado de apoyarme. Todos mis logros son por y para ustedes.*

*A mis hermanas Jessica y Karina que son lo más importante para mí, gracias por estar siempre conmigo. A mi novio por apoyarme en esos momentos donde la frustración y el desespero hacía de las suyas.*

*A todos los profesores que han estado enseñándonos a lo largo de estos 5 años, todo lo necesario para poder cumplir este objetivo, gracias por tanta paciencia y dedicación.*

*A mi tutora Wisleidys, gracias por todo el tiempo invertido, por todo el apoyo, por toda la paciencia y por cada mensaje de ánimo que nos sirvió de mucho.*

*A mi amiga, compañera de aula y compañera de tesis Kisis, gracias por todo el ánimo y por todo el apoyo para juntas lograr esta importantísima meta.*

*A mis mejores amigos Yeily, Didi y Alejandro Molina, son lo más sincero y más lindo que me pudo regalar la universidad.*

*A mis amigas y compañeras de apartamento Yanitza y mi Yoyi Daniela, casi 5 años compartiendo buenos y malos momentos, gracias por todo.*

*A mis amigos Drake, Néstor, René, José Eduardo, Ceylán, Naila, Daniela y a mi grupo que ya al final quedamos poco, pero para mí fue el mejor grupo de toda mi carrera FI023, a todos ustedes gracias por tantos lindos momentos.*

## AGRADECIMIENTOS

Maribel Kissis Valdés Barrios

*A dios por permitirme lograr mi sueño de ser ingeniera.*

*A mis padres Maribel y Roly por todo el apoyo incondicional estos 5 años de carrera.*

*A mis abuelas Elisa y Eloisa por regalarme su gran amor y sus consejos imprescindibles.*

*A mi hermano que ha sido siempre un gran apoyo en los momentos difíciles de mi carrera.*

*A mis tías Yahi y Yaniris por sus regaños constructivos que han hecho que me convierta en la profesional que soy hoy.*

*A mi novio por apoyarme en los momentos donde la frustración y el desespero se apoderaban de mí.*

*A mi tutora Wisleidys por todo su apoyo y sabiduría brindados hacia mí, y sus mensajes de si se puede que fueron clave para mí.*

*A todos los profesores que de una forma u otra he tenido el placer de aprender de ellos todos estos años. A mi amiga, compañera de aula y compañera de tesis Yeni, gracias por todo el ánimo y el apoyo que juntas nos brindamos para lograr nuestra meta.*

*A mis mejores amigas Daniela, Melisa, Patricia y Claudia por todas las noches de risas y cariño incondicional.*

*A mi grupo por todas esas noches de estudios que se convertían en ocio.*

*A todos gracias, sin ustedes nada hubiese sido posible.*

## DEDICATORIA

Yeni Carmen Tamayo Morales

*A las estrellas más bonitas que tiene el cielo, mis abuelos, Jesús y Leonila espero que se encuentren en el lugar más bonito, espero que estén orgullosos de mí.*

Maribel Kissis Valdés Barrios

*A mi bisabuela Dulce que sé que me guía y me protege y espero que esté orgullosa de mí.*

## **RESUMEN**

La Universidad de las Ciencias Informáticas fue creada con el objetivo de fortalecer la industria cubana del software. Hoy en día tiene como objetivo esencial certificar el Modelo de Calidad para el Desarrollo de Aplicaciones Informáticas (MCDAI) el cual enuncia entre sus buenas prácticas la documentación de las lecciones aprendidas. En la presente investigación se analiza la importancia de la recopilación, análisis y gestión de las lecciones aprendidas para elevar la madurez de los procesos en las organizaciones. Para ello, se realiza una revisión exhaustiva de la literatura existente sobre el tema, así como un estudio de los principales sistemas homólogos existentes. Se tiene como objetivo principal crear un sistema que permita la centralización, gestión y organización de todas las lecciones aprendidas recopiladas de la ejecución de los procesos en la Red de Centros de la Universidad. Se realizaron pruebas para la validación del correcto funcionamiento del sistema obteniéndose resultados satisfactorios en cuanto a su utilidad, obteniendo como resultado final un sistema que constituyó un avance en la mejora de procesos de software de la actividad productiva de la universidad.

**Palabras clave:** conocimiento, lecciones aprendidas, mejora de procesos, sistema de gestión del conocimiento, toma de decisiones

## **ABSTRACT**

*The University of Computer Sciences was created with the objective of strengthening the Cuban software industry. Today, its essential objective is to certify the Quality Model for the Development of Computer Applications (MCDAI), which lists among its good practices the documentation of lessons learned. This research analyzes the importance of collecting, analyzing and managing lessons learned to increase the maturity of processes in organizations. To this end, an exhaustive review of the existing literature on the subject is carried out, as well as a study of the main existing homologous systems. The main objective is to create a system that allows the centralization, management and organization of all the lessons learned compiled from the execution of the processes in the Network of University Centers. Tests were carried out to validate the correct functioning of the system, obtaining satisfactory results in terms of its usefulness, obtaining as a final result a system that constituted an advance in the improvement of software processes of the productive activity of the university.*

**Keywords:** *knowledge, lessons learned, process improvement, knowledge management system, decision making*

## Índice

RESUMEN .....	VII
INTRODUCCIÓN .....	1
CAPÍTULO 1: MARCO TEÓRICO CONCEPTUAL PARA LA GESTIÓN DE LECCIONES	
APRENDIDAS .....	6
1.1 Conocimiento y gestión del conocimiento .....	6
1.1.1 Modelos de gestión del conocimiento: .....	8
1.2 Lecciones aprendidas .....	11
1.2.1 Sistemas para registrar o documentar las LA. ....	13
1.2.2 Comparación de los sistemas estudiados .....	16
1.3 Notas del conocimiento .....	18
1.4 Mejora de procesos de software .....	19
1.5 Metodologías de desarrollo de software .....	21
1.6 Tecnología, lenguaje de programación y herramientas. ....	23
1.6.1 Lenguaje de programación .....	23
1.6.2 Sistema gestor de base de datos .....	25
1.6.3 Framework de desarrollo .....	26
1.6.4 Otras herramientas .....	26
Conclusiones del capítulo .....	27
CAPITULO 2 DISEÑO DE SOLUCIÓN PARA EL SISTEMA DE GESTIÓN DE LECCIONES	
APRENDIDAS .....	28
2.1 Propuesta de solución .....	28
2.2 Especificación de requisitos .....	30
2.2.1 Product Backlog .....	30
2.2.2 Requisitos no funcionales .....	36
2.3 Historias de Usuario .....	38
2.4 Planificación de los Sprint .....	39

2.5 Diseño de la arquitectura de software .....	40
2.5.1 Patrón arquitectónico .....	41
2.6 Modelo de datos .....	42
2.7 Patrones de Diseño .....	44
2.8 Estándares de codificación .....	46
Conclusiones del capítulo .....	48
<b>CAPÍTULO 3 VALIDACIÓN DEL SISTEMA DE GESTIÓN PARA EL REGISTRO DE LECCIONES</b>	
<b>APRENDIDAS.....</b>	<b>49</b>
3.1 Pruebas de Software .....	49
3.2 Plan de prueba .....	49
3.3 Pruebas funcionales .....	50
3.4 Pruebas no funcionales .....	54
3.5 Técnica para validar la satisfacción del usuario .....	57
3.6 Prueba de Aceptación.....	61
Conclusiones del capítulo .....	61
<b>CONCLUSIONES .....</b>	<b>62</b>
<b>RECOMENDACIONES .....</b>	<b>63</b>
<b>REFERENCIAS .....</b>	<b>63</b>
<b>ANEXOS .....</b>	<b>69</b>
Anexo 1 Entrevista para la obtención de la situación problemática .....	69
Anexo 2: Artefactos que evidencian la realización de pruebas funcionales y no funcionales .....	69
Anexo 3: Encuesta para evaluar la satisfacción de los usuarios .....	69
Anexo 4 Acta de aceptación.....	70
Anexo 5 Historias de usuario .....	71

## Índice de figuras

<b>Figura 1.</b> Procesos del modelo Wiig. Elaboración propia a partir de (Wiig, 1993). .....	9
<b>Figura 2.</b> Procesos de modelo KMAT. Elaboración propia, a partir de (Pulgarin Orozco, 2022) .....	9
<b>Figura 3.</b> Fases del modelo Nonaka-Takeuchi. Elaboración propia, a partir de (Aguilera-Luque, 2020). .....	11
<b>Figura 4.</b> Diseño de la propuesta de solución. Elaboración propia. ....	29
<b>Figura 5.</b> Representación del patrón arquitectónico Modelo- Vista -Template. Elaboración propia ....	42
<b>Figura 6.</b> Modelo de datos. Elaboración propia. ....	43
<b>Figura 7.</b> Evidencia del patrón Singleton. ....	44
<b>Figura 8.</b> Evidencia del patrón experto. ....	45
<b>Figura 9.</b> Ejemplo de declaración por línea. ....	47
<b>Figura 10.</b> Ejemplo de inicialización de variables. ....	47
<b>Figura 11.</b> Ejemplo de separación de línea entre métodos. ....	47
<b>Figura 12.</b> Ejemplo de importaciones. ....	47
<b>Figura 13.</b> Niveles de satisfacción de los usuarios .....	60

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Comparación de los sistemas homólogos estudiados. Elaboración propia. ....	16
<b>Tabla 2.</b> Comparación de Metodologías. Elaboración propia a partir de (Montero et al., 2018) .....	22
<b>Tabla 3.</b> Requisitos Funcionales. Elaboración propia. ....	36
<b>Tabla 4.</b> Prototipo de Historia de Usuario Registrar lección aprendida. Elaboración propia. .	38
<b>Tabla 5.</b> Prototipo de Historia de Usuario Editar lección aprendida. Elaboración propia. ....	39
<b>Tabla 6.</b> Planificación del Sprint 1. Elaboración propia. ....	39
<b>Tabla 7.</b> Planificación del Sprint 1. Elaboración propia. ....	40
<b>Tabla 8.</b> Plan de prueba basado en la estrategia de prueba de la UCI. Elaboración propia. ...	50
<b>Tabla 9.</b> Diseño de Caso de prueba. Registrar lección aprendida. Elaboración propia. ....	52
<b>Tabla 10.</b> Resultados de las pruebas de usabilidad. Elaboración propia. ....	55
<b>Tabla 11.</b> Resultados de las pruebas de portabilidad. Elaboración propia. ....	56
<b>Tabla 12.</b> Resultados de las pruebas de seguridad. Elaboración propia. ....	57
<b>Tabla 13.</b> Cuadro lógico de ladov. Elaboración propia. ....	58
<b>Tabla 14.</b> Nivel de satisfacción de los usuarios. Elaboración propia .....	59
<b>Tabla 15.</b> Escala numérica para niveles de satisfacción. Elaboración propia. ....	60

## AVAL DEL CLIENTE

En cumplimiento del **Convenio de colaboración** Dirección de Calidad de Software y en función de la ejecución del proyecto: Sistema de gestión para el registro de las lecciones aprendidas en la Universidad de las Ciencias Informáticas, se hace entrega de los productos que se relacionan a continuación:

- El documento final de tesis
- Especificación de Requisitos de Software
- Historias de usuarios
- El sistema implementado

La Parte Cliente, luego de haber revisado los productos de trabajo determina que los productos son aceptados.

---

**Entrega:**

**Recibe**

---

**Nombre y apellidos:** Yeni Carmen Tamayo Morales  
Maribel Kissis Valdés Barrios.

**Nombre y apellidos:** Wisleidys Campos Wright

---

**Cargo:**

**Cargo:** subdirectora de la DCSW

---



## INTRODUCCIÓN

Las sociedades han evolucionado de acuerdo con los factores estratégicos de desarrollo que cada momento histórico ha determinado. Dichos cambios se han dado ante la imperante necesidad de transformaciones que el hombre mismo ha generado, especialmente por su relación con el entorno. En pleno siglo XXI, se asiste a la Sociedad del Conocimiento, período en el cual se observa que la capacidad de aprendizaje es la puerta de entrada a un desarrollo sostenible e integral (Ceballos & Arias, 2018).

La gestión del conocimiento es un elemento fundamental en todo tipo de organizaciones, en donde las nuevas tecnologías conducen a la era del conocimiento y de la valoración del mismo como uno de los activos más importantes. El progreso del hombre ha sido marcado por el conocimiento, y en la actualidad se considera un elemento importante que establece la capacidad en las instituciones, aunque no se tiene un enfoque sobre una definición de lo que es la gestión del conocimiento, si existe la descripción de los impactos positivos de esta, tanto para las personas como para las organizaciones (Vásquez, 2020). Los beneficios que se consiguen de la gestión del conocimiento propician el mejoramiento de los servicios y productos de la institución, en la medida en que estos son el resultado del conocimiento existente tanto en el entorno como en el plano interno organizativo. Se identifican dentro de sus procesos fundamentales identificar, adquirir, desarrollar, compartir, utilizar y retener el conocimiento (Triana-Velázquez et al., 2018). Cada organización debe tener la capacidad de saber qué tipo de conocimientos posee y su gestión al mismo tiempo.

Los seres humanos se desarrollan en la sociedad en la que la incertidumbre y el cambio son comunes denominadores. En este contexto, es una necesidad de primer orden que todos los integrantes de una organización en mención apliquen la gestión del conocimiento como una disciplina para procesar, transferir, interpretar, aplicar y reutilizarlo; esto permitirá desarrollar una cultura que fomente la creatividad y el buen desempeño, lo que a su vez favorecerá la creación de nuevos y mejores servicios (Quispe Alcarraz, 2021). El conocimiento juega un papel relevante en la sociedad, si bien, su rol determina en gran medida la capacidad de acción tanto de individuos como de organizaciones, por sí solo no tiene implicancia, pero en manos de la supervisión de una persona que lo gestione, jugará un papel importante y protagónico. (Agudelo Ceballos et al., 2018).

En Cuba un antecedente importante en la implementación de la gestión del conocimiento fue el VI Seminario Iberoamericano para el intercambio y actualización de la ciencia y la innovación, Red Iberoamericana de indicadores de Ciencia y Tecnología (IBERGECYT) celebrado en el 2001, que

consideró principalmente sobre la gestión del conocimiento y el capital intelectual en el mundo empresarial y en la comunidad académica (Triana-Velázquez et al., 2018). Se proyecta que es una técnica que no se pueda implantar aisladamente, pues se requiere, además, de la gestión de la información, de la gestión documental, del uso de las tecnologías de información y de un eficiente manejo de los recursos humanos.

El propósito esencial en el país es contribuir a la creación de organizaciones inteligentes, aquellas capaces de recibir y procesar información, crear conocimiento a partir de la información procesada y usar el conocimiento para la toma de decisiones de manera eficaz, a fin de cooperar de modo creativo al desarrollo (Triana-Velázquez et al., 2018). El avance de las Tecnologías de la Informática y las Comunicaciones (TICs) han impulsado el uso de los productos informáticos en todas las esferas de la sociedad. El uso acelerado y masivo de las TICs, exige que los productos desarrollados sean más seguros, usables, adaptables y que evolucionen continuamente de acuerdo a su utilidad. Las aplicaciones para mantenerse en el mercado deben ser cada vez más competitivas y necesitan lograr satisfacer las expectativas de sus clientes, quienes demandan elevados niveles de calidad.

La Universidad de las Ciencias Informáticas (UCI) fue creada para fortalecer la industria cubana del software, su actividad productiva está conformada por 6 centros de desarrollo los cuales tienen como objetivo realizar actividades de Investigación, Desarrollo e Innovación (I+D+I), encargados de las aplicaciones informáticas, desarrollo tecnológico y las investigaciones asociadas. De conjunto conforman una red de trabajo colaborativo que opera bajo normas y procedimientos comunes, posibilitando la reutilización de componentes y eficiencia industrial (*Universidad de las Ciencias Informáticas*, 2018) se utiliza como modelo de calidad en su desarrollo de software el Modelo Integrado de Madurez de la Capacidad (CMMI) que al no poder certificar el nivel 3 por restricciones del bloqueo tiene como objetivo esencial certificar el Modelo de Calidad para el Desarrollo de Aplicaciones Informáticas (MCDAI). Este es el modelo y norma que se utilizan para mejorar las experiencias del desarrollo de las aplicaciones basado en las buenas prácticas (Oduardo & Carmona, 2021). MCDAI propone analizar cómo satisfacer las brechas identificadas mediante la adquisición de nuevos conocimientos, internamente o a partir de fuentes externas, define el plan de acción para obtener, mantener y compartir los conocimientos con el personal y se diseñan los mecanismos necesarios para identificar y compartir las lecciones aprendidas (Departamento de Gestión e Informatización ONN, 2021).

Una de las buenas prácticas (BP) que ofrece el modelo MCDAI, es la documentación de las Lecciones Aprendidas. Permite al equipo de desarrollo la identificación de factores de éxitos, deficiencias en estrategias, procesos y técnicas con el objetivo de no repetir resultados que fomenten al fracaso. Las lecciones aprendidas se recopilan a partir de las actividades de calidad que se desarrollan a lo largo del ciclo de vida de un proyecto o servicio. Para obtener el comportamiento de este proceso se realizó un estudio mediante entrevistas a especialistas de diferentes centros de desarrollo de software de la UCI. Dentro de los centros se encontraban: el Centro de Informática Médica (CESIM), el Centro de Tecnologías Interactivas (VERTEX), el proyecto Z17 (Todus, Picta, Apklis) y Centro de Informatización de la Gestión Documental (CIGED). Los años de experiencias de los especialistas oscilaban entre 3 y 10 años, los roles fueron: jefe de proyectos, analista, desarrolladores y administradores de la calidad. Los resultados de las entrevistas arrojaron que el registro de las LA en la universidad es un proceso institucionalizado donde cada proyecto es responsable de su registro, provocando que esta gestión no permita el aprovechamiento de nuevas experiencias en toda la organización. Se pudo constatar que la reutilización de códigos de proyectos anteriores es mediante la herramienta *GitHub* y solo puede ser accedido por proyectos del centro de desarrollo. En la herramienta para la Gestión de proyectos (GESPRO) de los centros se pueden obtener datos históricos relacionados con la estimación y el cronograma, pero no permite acceder a la información desde otros centros. En el Excriba se guarda el registro de las LA relacionados a los procesos propios del proyecto lo que evidencia la dificultad para el intercambio de experiencias.

El Sistema de Gestión de Calidad tiene definido que es de obligatorio cumplimiento el registro de estas experiencias, pero en ocasiones no es tomado en cuenta. Generalmente esta práctica no se realiza durante todo el ciclo de desarrollo, sino más bien, al cierre de cada proyecto, lo que trae como consecuencia la poca obtención de información de nuevos especialistas que se incorporan. En ocasiones estas informaciones son gestionadas por cada especialista desde su propia agenda de trabajo. Como se encuentra disperso en diferentes centros de desarrollo y proyectos, mucha información se pierde por la fluctuación de personal. Por el número de proyecto y la cantidad de actividades de calidad que se realizan en el proceso de gestión de las lecciones este proceso se hace engorroso y lento. Esto trae como consecuencia la repetición de errores, el desconocimiento de las causas de las experiencias exitosas y el retraso en la mejora de los procesos de la organización, debido a que no se guardan los cambios en el comportamiento de los mismos y se produce un aprendizaje natural que resulta más lento y costoso.

Dada la situación anterior esta investigación está dirigida a solucionar el siguiente **problema de investigación**:

¿Cómo gestionar las lecciones aprendidas en el proceso de desarrollo de software de la UCI para que pueda ser accedida por toda organización?

Para dar solución al problema de investigación se propone como **objetivo general** desarrollar un sistema de gestión que garantice la centralización y organización de las lecciones aprendidas para la reutilización del conocimiento en los proyectos de desarrollo de software.

Para dar cumplimiento al objetivo planteado se proponen las siguientes tareas de la investigación:

1. Identificación de las variantes de solución existentes y tendencias a seguir en la solución del problema de la investigación planteado, a partir del análisis de los referentes teóricos que preceden la realización de la presente investigación.
2. Evaluación del estado actual del proceso de gestión de las lecciones aprendidas
3. Identificar las tecnologías, metodologías y herramientas a utilizar para el desarrollo del sistema de gestión de lecciones aprendidas.
4. Elaboración de la planificación y diseño de la propuesta de solución para satisfacer las necesidades del cliente y el tiempo de entrega estimado.
5. Implementación de la propuesta de solución.
6. Validación del sistema propuesto.

Definiendo como **objeto de estudio**: sistemas de gestión de la información.

El **campo de acción**: Gestión de la información de las lecciones aprendidas.

**Idea a defender**: Con el desarrollo de un sistema de gestión que registra las lecciones aprendidas se podrá centralizar toda la información para que pueda ser reutilizada en los proyectos de desarrollo de software en la UCI.

Durante el proceso de investigación se emplearon **métodos científicos** los cuales se exponen a continuación:

**Métodos teóricos:**

*Analítico - Sintético*: permitió analizar de manera detallada la bibliografía para dar cumplimiento al objetivo general mediante sus componentes elementales, así como sus relaciones entre ellos.

*Modelación:* se utilizó en el diseño de la propuesta y sus componentes.

### **Métodos empíricos:**

*Entrevista:* permitió la obtención de información a través de la realización de entrevistas a un grupo de expertos para saber cómo se ejecuta el proceso de la toma de las lecciones aprendidas en los proyectos en la universidad. (Ver Anexo 1).

*Observación:* permitió sacar conclusiones relevantes sobre como los proyectos de la universidad realizan actualmente el proceso del registro de las lecciones aprendidas en los proyectos para extraer información referente al objeto de estudio y realizar la confrontación de los resultados de la investigación.

La presente investigación está compuesta por la introducción, tres capítulos, conclusiones finales, recomendaciones, referencias bibliográficas y anexos. A continuación, una breve descripción del contenido de los capítulos:

### **Capítulo 1: Marco Teórico Conceptual para la Gestión de Lecciones Aprendidas**

Se realiza un análisis de los principales conceptos relacionados con la gestión del conocimiento, las lecciones aprendidas y las notas del conocimiento, así como las metodologías, lenguajes, herramientas y técnicas para el desarrollo del sistema necesarios para el correcto entendimiento de la solución propuesta.

### **Capítulo 2: Diseño de Solución para Sistema de gestión para las Lecciones Aprendidas**

En este capítulo se describe una propuesta de solución para el problema identificado a partir del estudio realizado. Se definen y realizan los artefactos de la ingeniería de software que propone la metodología seleccionada para el desarrollo del sistema. También se define la arquitectura utilizada, los patrones de diseño identificados y el modelo de datos.

### **Capítulo 3: Validación del Sistema de Gestión para el registro de Lecciones Aprendidas**

En este capítulo queda documentado el proceso de pruebas llevado a cabo para garantizar la calidad del producto final. Se realiza una revisión final de los requisitos del sistema para verificar que se le dio cumplimiento a cada uno de ellos.

## **CAPÍTULO 1: MARCO TEÓRICO CONCEPTUAL PARA LA GESTIÓN DE LECCIONES APRENDIDAS**

En el presente capítulo se agrupan los principales elementos para crear una base teórica sólida que respalde y justifique la investigación. Son abordados los principales conceptos fundamentales relacionados con el objetivo general. Se identifican, comparan y analizan los sistemas homólogos. Se describen y seleccionan las tecnologías, herramientas y metodologías para el desarrollo de la propuesta de solución.

### **1.1 Conocimiento y gestión del conocimiento**

El conocimiento es el acto humano de querer saber (Rodríguez-Ponce et al., 2022) se entiende como un activo que tiene la capacidad de generar valor y producir riqueza, por lo cual requiere ser protegido para preservar la fuente de la ventaja competitiva de las organizaciones (Quispe Alcarraz, 2021) Incluye un sentido más amplio y profundo que los datos o la información. Cuando una persona racionaliza la información obtenida del mundo circundante la transforma en conocimiento, por tanto, se considera como el procesamiento de la información externa e interna para conocer el mundo desde una perspectiva individual, socializada e integrada(Mera López, 2021). Vale la pena destacar la necesidad de ofrecer conocimiento explícitos y tácitos (Hvorecký & Korenova, 2023). Estos tipos de conocimiento se pueden definir como:

- *Conocimiento tácito*: Se utiliza de forma intuitiva e inconsciente, y que se adquiere mediante la propia experiencia, caracterizándose por ser personal y contextual(Pérez-Fuillerat et al., 2019). Es aquel conocimiento que la persona tiene incorporado sin tenerlo permanentemente accesible a la conciencia, pero del cual el individuo hace uso cuando las circunstancias lo requieren y que es utilizado como instrumento para manipular el fenómeno que se está analizando(Piva et al., 2021).
- *Conocimiento explícito*: Es un conocimiento objetivo y racional que es registrado o codificado en documentos, en bases de datos, herramientas digitales, en procesos y en procedimientos, formulado en palabras o en una manera visual, se puede encontrar e incluir en la organización de manera formal y sistemática, disponible en patentes, marcas y procesos, en los elementos tangibles de la organización bajo estándares establecidos, se basa en evidencias y es

transmisible entre los colaboradores para hacerlo útil en el mejoramiento continuo de procesos (SARMIENTO, 2018).

En el ámbito del desarrollo de software se ha planteado que la gestión del conocimiento(GC) juega un rol importante en el éxito, debido principalmente a la planificación, organización, monitoreo y coordinación efectiva de los activos de este proceso (Rodríguez-Ponce et al., 2022). La tasa de éxito de la GC en una organización se relaciona con la posibilidad de utilizar diferentes conocimientos para adquirir nuevas fases, que pueden mejorar las acciones organizacionales o sentar las bases para nuevas acciones. Promueven el desempeño de la innovación y guían el sistema de conocimiento de la organización para generar nuevas ideas, mejorando así las capacidades de innovación(Villasana Arreguín et al., 2021). Gestionar el conocimiento contribuye a incrementar la productividad y la eficiencia de las organizaciones, ante la competencia de un mercado mundial que exige cada vez más productos de mejor calidad. La GC busca igualar conocimientos esenciales y ponerlos a disposición de aquellos especialistas o colaboradores que lo necesiten con el fin de lograr las metas de la organización(Alvarado Rodríguez, 2022a), apoyándose en un conjunto de subprocesos y actividades que ayuden a las instituciones a generar, adquirir, descubrir y organizar el conocimiento posibilitando su uso y difusión en los diferentes niveles de organización teniendo como objetivo principal implementar sistemas que puedan compartir el conocimiento adquirido a través de las Lecciones Aprendidas(LA). La buena planificación, organización, un buen monitoreo y una excelente coordinación son de gran importancia para el desarrollo de este análisis debido a que pueden aportarle muchos beneficios a la hora de desarrollar el sistema. Este estudio sostiene que la incorporación de los conocimientos tácitos y explícitos en las actividades de diseño son de gran significado porque se explica que tener las ideas claras, permite comunicarse abiertamente y a la vez da paso que las mismas puedan quedar escritas en documentos. Con el uso del conocimiento tácito, el cual se va adquiriendo de manera inconsciente, se van sumando experiencias de manera personal.

Existen diversos modelos de GC que representan un enfoque integral de cómo maniobrar el mismo en las organizaciones. Constituyen una herramienta valiosa para ser implementadas de forma conjunta, creando pequeñas comunidades del saber constituidas por grupos de personas que interactúan, intercambian experiencias e informaciones constituyendo un factor clave para la mejora continua de los procesos de desarrollo de software en una organización.

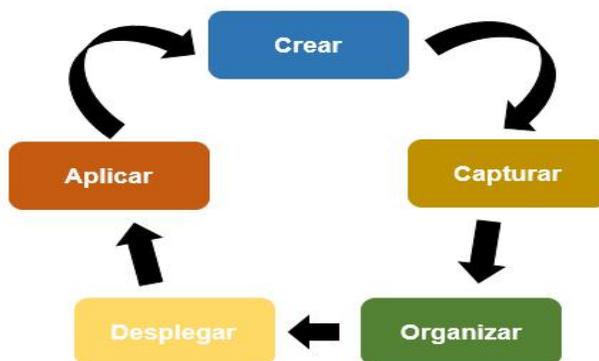
### 1.1.1 Modelos de gestión del conocimiento:

Para desarrollar la gestión del conocimiento se han creado diferentes modelos, cuyos autores han concedido mayor o menor importancia al uso de las tecnologías de la información y la comunicación (Rodríguez et al., 2023). Algunos de estos modelos son:

#### **Modelo de Wiig:**

Wiig realiza un modelo universal sobre la gestión del conocimiento encaminado a la solución de problemas, resaltando el proceso de elaboración, simbolización y utilización del conocimiento construido desde su experiencia deseando fortalecer su aplicación en las empresa a través de una estructura organizacional apropiada detectando en la empresa el conocimiento actual y su posible insuficiencia, para lograr estructurar el léxico del conocimiento y su enciclopedia (Wiig, 1993). Este modelo se caracteriza por emplearlo para la solución de problemas utilizando las experiencias prácticas adquiridas. El principal aporte radica en el uso del conocimiento y en los procesos de creación, codificación y aplicación, especificándose su contenido, localización y distribución. Su limitación consiste en que no distingue entre la dimensión epistemológica y la ontológica, lo cual constituye la esencia para comprender las necesidades de transformación del conocimiento entre las taxonomías tácito-explícito y la evolución individual hacia el colectivo entre diversas organizaciones (interorganizacional) (Rodríguez, 2018). Entre las aportaciones de este modelo se encuentra que su principal objetivo es reforzar el uso del conocimiento, describiendo su contenido, su localización, su distribución y su utilización (Casañola et al., 2021).

El modelo Wiig comprende 5 procesos para la gestión del conocimiento:



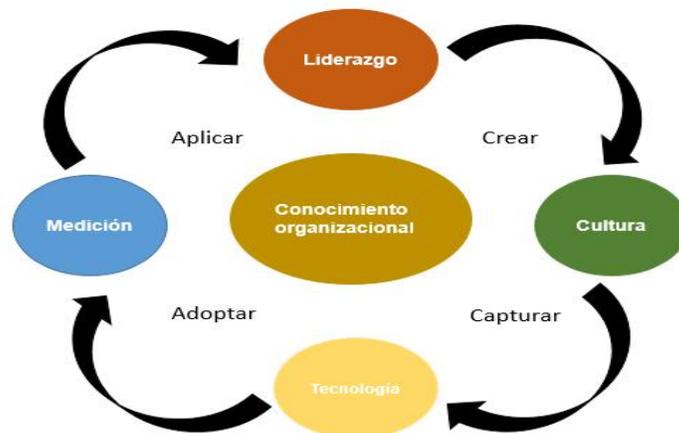
**Figura 1.** Procesos del modelo Wiig. Elaboración propia a partir de (Wiig, 1993).

El primer paso es crear, donde son utilizados todos los conocimientos previos e importantes que pueda generar la organización, el segundo es capturar el conocimiento para reutilizarlo, construirlos y aprovecharlos de otras maneras posibles en futuros proyectos, el tercero es organizar el conocimiento, para que este ampliamente disponible y tener un mejor control sobre él, el cuarto es desplegar el conocimiento dirigido al personal y el ultimo, aplicar el conocimiento para lograr la viabilidad y el éxito de la organización (ver figura 1) (Leiva Morantes & Álvarez Villamil, 2020).

La principal desventaja que posee este modelo es que no distingue el conocimiento cognitivo y el real. Sin embargo, se considera muy útil la característica que propone de desarrollar funciones de favorecimiento al aprendizaje e implementar la creación de un sistema capaz de capturar las buenas prácticas, experiencias y lecciones aprendidas que serán tangibles para la organización con aras de que no se comentan errores ni se dupliquen procesos.

**Modelo de KMAT:**

Es un modelo de evaluación y diagnóstico creado por Arthur Andersen, este modelo destaca cuatro variables: liderazgo, cultura, tecnología y medición (Valencia-Cruzaty et al., 2021). Es una valiosa herramienta de evaluación y diagnóstico que se enfoca en el nivel de compromiso individual y organizacional para trasladar conocimiento que agregue valor a los colaboradores de la organización y así maximizar los resultados del uso de la información (Salones & Aular, 2022)



**Figura 2.** Procesos de modelo KMAT. Elaboración propia, a partir de (Pulgarin Orozco, 2022)

Es un proceso cíclico y continuo, tanto de las personas hacia la organización como de esta hacia las personas (Andersen, 1999). Plantea que para facilitar la GC, los esfuerzos en las organizaciones tiene que ser compartidos y coordinados, prestándole una mayor importancia a las dimensiones de modelo (Figura 2) (*Repository Universidad Tecnológica de Pereira :: Search*, 2021). Lo nuevo e interesante de su invención está en que desde la individualidad hay responsabilidad personal por difundir y hacer explícito el saber para la organización y desde el punto de vista organizacional también envuelve una responsabilidad con la generación de la infraestructura de base para que sea efectiva la perspectiva individual, se desplieguen los procesos y procedimientos, la tecnología, la cultura y los sistemas que ayudan a capturar, analizar, sintetizar, aplicar, valorar y distribuir el conocimiento (Caceres Pieschacon & Romero, 2022). La desventaja principal que trae consigo el uso de este modelo es que al igual que modelo Wiig no distinguen entre el conocimiento cognitivo y el conocimiento real.

### ***Modelo Nonaka-Takeuchi:***

Nonaka-Takeuchi afirma que las empresas generan conocimiento nuevo a partir de experiencias (Pazmin, 2021), donde se visualiza el conocimiento tanto individual como colectivo y su gestión con una proyección hacia la empresa, el aprendizaje tanto formal como de persona a persona, el conocimiento explícitamente documentado como tácito y proveniente tanto de los integrantes de la entidad como de su entorno (Pérez & Urbáes, 2016). Este modelo se basa en que el conocimiento se crea y se expande a través de la interacción social entre el conocimiento tácito y explícito (Alvarado Rodríguez, 2022b). Enfatiza el que crear conocimiento no es crear información, sino procesarla, hacerla útil, fusionarla con ideas subjetivas e intuiciones de los individuos que pertenecen a la institución y hacerlas disponibles para todos (Pérez & Urbáez, 2016) Nonaka y Takeuchi presentan el proceso de creación de conocimiento compuesto por cinco fases: (ver figura 3).



**Figura 3.** Fases del modelo Nonaka-Takeuchi. Elaboración propia, a partir de (Aguilera-Luque, 2020).

El modelo Nonaka-Takeuchi es el más destacado y representativo entre los modelos, se refiere a la gestión del conocimiento como producto de movimientos cíclicos en espiral, en el cual interactúan conocimientos tácitos-empíricos y explícitos-formalizados en las dimensiones de una organización (Valencia-Cruzaty et al., 2021). Es un modelo dinámico sobre la creación del conocimiento que contextualiza tal proceso en un entorno social, donde se produce el cambio de unos tipos de conocimientos a otros y que, por tanto, necesita de los individuos, pero sin caer en la individualidad, sino como un producto de la interacción entre personas.

El modelo a seguir para la implementación del sistema es el propuesto por Nonaka-Takeuchi, el cual es utilizado por MCDAI para la implementación del proceso base de Gestión del Conocimiento en plan de garantizar acciones para la socialización, externalización, asociación e interiorización del conocimiento, según sea pertinente para alcanzar las metas planteadas en el proceso de desarrollo de software a través del registro de LA. Es el modelo que además está diseñado para la certificación de la producción de *software* en las pequeñas y medianas empresas en Cuba.

## 1.2 Lecciones aprendidas

Las LA pueden definirse como el conocimiento adquirido sobre un proceso a través de las causas que permiten que se obtengan o no los resultados planeados (Contreras, 2022). El PMBOK afirma que las LA de una fase o proyecto es el conocimiento adquirido durante un proyecto que muestra cómo se abordaron o deberían abordarse los eventos del plan con el propósito de mejorar el desempeño futuro

(*PMBOK GUIDE*, 2021). Esta práctica permiten adquirir el conocimiento, resultado de las experiencias evidenciadas dentro del desarrollo de un proyecto, donde se pueden aprovechar casos de éxitos (impacto positivo), mitigar las oportunidades de mejora para que no ocurran en futuros proyectos y para que se repliquen o amplifiquen según su impacto identificado sea una oportunidad o una amenaza (Santos Correa et al., 2020). Además ayudan a mejorar los procesos organizacionales y los patrones para la toma de decisiones con el fin de evitar que los equipos de proyectos vuelvan a cometer los mismos errores del pasado (Villa & Visbal, 2020). Son un conjunto de buenas prácticas adoptadas por organizaciones que no desean extender los errores de un proyecto a otro y también aquellas que desean continuar con los éxitos (Mattia et al., 2021).

El registro de LA es creado como una salida del proceso de gestión del conocimiento y se hace lo más tempranamente posible, debido a que su uso constituye una entrada para la actualización de las salidas de los procesos en los proyectos (Gómez, 2020). En consecuencia, si las personas realizan un análisis de los sucesos ocurridos positivos o negativos, al término de un evento pueden lograr la identificación de BP, barreras y riesgos en los procesos, así como las razones o el fracaso obtenido para posibles experiencias futuras.

Las empresas de software las utilizan para recopilar los conocimientos que van adquiriendo los integrantes de los equipos. El objetivo es dejar constancia de aquellos aspectos aprendidos que han contribuido tanto al éxito como al fracaso, con el fin de poder mejorar el rendimiento en trabajos futuros. La gestión de las mismas debe tener relación entre las personas que las poseen y las que se puedan beneficiar de ellas.

Gestionar las LA en una organización constituye una herramienta fundamental para fomentar la mejora continua de procesos, promueven dentro de la misma el esfuerzo de documentar todas las actividades que se realizan en pro de satisfacer las necesidades de futuros proyectos. Permiten mejorar el rendimiento de los proyectos evitando la repetición de errores e identificando problemas comunes, ayudan a obtener respuestas efectivas para variaciones y acciones correctivas. Proporcionan información sobre técnicas eficaces que a la vez permite mejorar la eficiencia y eficacia de la validación de los productos brindando información sobre los problemas encontrados y sobre cómo podrían haberse evitado. Permite concebir los errores pasados y mejorar el trabajo futuro. Con esta práctica las organizaciones minimizan los riesgos asociados a los procesos, evitan problemas ya identificados y reutilizan conocimiento para evitar proyectos fracasados.

### **1.2.1 Sistemas para registrar o documentar las LA.**

La documentación de LA contribuye a explicitar un nuevo conocimiento, desarrollando los elementos claves y la reconstrucción de la lógica que llevó a la consecución de los resultados y las relaciones causales que los condicionaron, capturadas durante la fase de identificación (Luna & Bu, 2015). Las lecciones deben capturarse a lo largo del proyecto y cuando este termina, el conocimiento acumulado debe disponerse en un lugar apropiado para que pueda ser distribuido a toda la organización (Villa & Visbal, 2020)

Para gestionar las LA, las organizaciones suelen utilizar sistemas de gestión del conocimiento. Este proceso viene acompañado de dificultades relacionados con la falta de tiempo y la desaparición del mismo por la fluctuación del personal, a eso se le aumenta un esfuerzo superior en la identificación del conocimiento y la difusión del mismo. Está claro que el procesamiento de ellas no acaba cuando se capturan, sino que su utilidad está en aplicarlas en trabajos posteriores (Jaime et al., 2022). Las actividades relacionadas con las LA deben de ser planificadas; y realizar seguimiento y control de su ejecución. Entre las actividades asociadas a desarrollar está la identificación, la cual debe mantenerse durante todo el ciclo de vida del proyecto. Después de identificarlas es conveniente validarlas para tener la certeza de que realmente es útil el conocimiento adquirido, para ser tomada en cuenta posteriormente. Algunas organizaciones crean repositorios para su almacenamiento. No es suficiente solo guardarlas, deben ser compartidas y utilizadas como ayuda a la realización de las tareas del proyecto y a la toma de decisiones (Oduardo & Carmona, 2021).

Existen diversos sistemas para registrar lecciones aprendidas en el proceso de desarrollo de software, algunos de los más comunes son:

#### ***JIRA***

Es una herramienta de gestión de proyectos y seguimiento de problemas desarrollada por Atlassian. Es muy utilizada en el desarrollo de software para planificar, rastrear y gestionar las tareas y problemas del proyecto. Permite a los equipos crear proyectos, agregar tareas, asignarlas a miembros del equipo y hacer un seguimiento de problemas, que permite a los equipos registrar errores o problemas encontrados durante el desarrollo y hacer un seguimiento de su solución. Una de sus principales ventajas es su flexibilidad (Atlassian, 2023) .Los equipos pueden personalizar la herramienta para adaptarla a sus necesidades. Admite hacer gestión de proyectos y tareas a través de la creación de flujos de trabajo, que definen los procesos y realizar un seguimiento de las tareas.

#### ***Asana***

Asana es una herramienta de gestión de proyectos y tareas en línea, desarrollada por Asana Inc. Muy utilizada en el mundo empresarial para gestionar las tareas y proyectos de equipos de trabajo. Asana permite a los equipos crear proyectos, asignar tareas y hacer un seguimiento del progreso del proyecto. Una de las principales ventajas es su facilidad de uso. Una forma que tiene esta aplicación de gestionar las lecciones aprendidas es mediante la creación de una sección de “lecciones aprendidas”, en esta sección los miembros del equipo pueden agregar comentarios, notas o documentos sobre lo aprendido durante el desarrollo del proyecto. Esto permite que el equipo tenga un registro centralizado de estas lecciones y puedan acceder a ellas fácilmente. Otra forma es mediante la creación de una tarea dedicada a la reflexión del proyecto al final del mismo. En esta tarea los miembros del equipo pueden compartir sus experiencias y reflexiones incluyendo las lecciones adquiridas esto permite que el equipo tenga un espacio dedicado para la reflexión y la documentación de las mismas. Además, cuenta con una amplia gama de integraciones con otras herramientas de productividad, como Slack, Dropbox y Google Drive, lo que permite a los equipos automatizar el flujo de trabajo y aumentar la eficiencia. Es muy útil para la gestión de proyectos y tareas en línea, que permite a los equipos colaborar y hacer un seguimiento del progreso del proyecto de manera efectiva y eficiente (Asana, 2023)

En la universidad existen diferentes sistemas que también son utilizados para gestionar las lecciones aprendidas, pero cada uno de ellos tienen diferentes enfoques en el registro de las mismas:

### ***XEDRO-GESPRO***

Es una suite orientada a la web que permite la planificación, seguimiento y control de productos en forma de proyectos. Cuenta con herramientas para el apoyo a la toma de decisiones a nivel de proyecto, nivel de entidad ejecutora y nivel gerencial. Se presenta en un modelo de negocios basado en servicios que combinan el uso de una solución informática para la dirección integrada de proyectos y un sistema de formación especializada en gestión de proyectos. Esta combinación posibilita no sólo la informatización de las organizaciones sino también la mejora integral de los procesos de planificación, control y seguimiento de proyectos. Esta suite cuenta con distintos módulos entre los que se encuentran: módulo de configuración y seguridad, módulo de planificación y módulo de gestión de riesgos (Socarras Navarro & Acevedo Guevara, 2019). Una de las desventajas que se pudo observar es que el manual de usuario es muy extenso por lo que se hace difícil para el usuario la búsqueda de información.

## **XABAL EXCRIBA**

Es un sistema de gestión documental que propicia la generación de documentos, su revisión, administración, distribución, custodia y disposición. Cumple con las normas internacionales para esta función y mantiene la seguridad de los documentos en formato electrónico. Incluye, además, módulos para la gestión de: acuerdos, reportes, notificaciones, registros de entrada y salida de documentos, y del cuadro de clasificación de la entidad (*EXCRIBA | Universidad de las Ciencias Informáticas, 2022*). Es un software para la gestión documental, diseñado para tramitar los documentos administrativos que se generan o reciben dentro de las organizaciones a partir de sus funciones, por lo tanto, involucra todas las áreas de una organización, permitiéndole gestionar de forma correcta la documentación como prueba, testimonio y evidencia de las actividades organizacionales. La puesta en práctica del Excriba agiliza el trámite de los documentos, controlar el estado de los mismos, evitando la pérdida de información y tributando como herramienta de apoyo para la toma de decisiones en las entidades que lo utilizan (López, 2018). Algunas de las desventajas que presenta esta herramienta es que solo puede acceder a fuentes de datos como son los sistemas de gestión de bases de datos PostgreSQL y 2 Access. No reutiliza los archivos de configuración generados por la herramienta, no presenta edición de archivos de configuración y dentro de los modelos de contenido que maneja solo tiene acceso a propiedades básicas y tipos de datos (Oquendo Arencibia, 2018). Este sistema no se integra correctamente con otros sistemas utilizados por la empresa u organización, lo que puede provocar problemas de compatibilidad y dificultades para la transferencia de datos entre sistemas.

## **GITLAB**

GitLab es una herramienta que simplifica el desarrollo de software, sin tener que dar soluciones puntuales para cada paso del ciclo de vida de la información. Ayuda a las empresas a gestionar las crecientes complejidades del desarrollo, la seguridad y el despliegue de software. Se ofrece como un software libre que se puede instalar en cualquier servidor. Su código abierto y su flexibilidad permite usar esta herramienta en una empresa u organización en sus propios servidores, sin ningún coste. Además, cubre todas las etapas del ciclo de vida de los datos, monitorizando el rendimiento de estos, y te permite crear informes sobre flujo de datos.(GitLab, 2023). Este sistema ofrece muchas integraciones con otras herramientas y servicios por esta razón las configuraciones y el mantenimiento de estas integraciones pueden ser complicadas y consumir recursos adicionales.

### 1.2.2 Comparación de los sistemas estudiados

Para el análisis de las características de los sistemas anteriormente mencionados, se tuvo en cuenta los siguientes indicadores, mostrados en la **Tabla 1**.

Sistemas Indicadores /	Licencia privativa	Gestionan conocimiento	Registran lecciones aprendidas	Presentan soporte en la universidad	Accesibilidad de la información desde todas las áreas de la UCI
Jira	Si	Si	Si	No	No
Asana	Si	Si	Si	No	No
GITLAB	No	No	Si	No	No
XEDRO-GESPRO	No	Si	Si	No	No
XABAL EXCRIBA	No	Si	Si	Si	No

**Tabla 1.** Comparación de los sistemas homólogos estudiados. *Elaboración propia.*

Teniendo en cuenta el estudio realizado a los sistemas homólogos anteriormente descritos se puede concluir que:

-Los sistemas internacionales, a pesar de cumplir con los indicadores necesarios para la gestión del conocimiento, son privativos, lo cual descarta su uso como solución a la problemática planteada.

- Para el caso de XEDRO-GESPRO actualmente no cuenta con soporte estable en el desarrollo de software.

-En el caso del GitLab guarda lecciones aprendidas, pero no gestiona el conocimiento como se quiere lograr.

-XABAL EXCRIBA posee muchas desventajas en cuanto a la administración de los usuarios, desde el punto de vista de la usabilidad y la gestión de las lecciones aprendidas la propuesta de solución brinda una variante más flexible y entendible para los usuarios.

GITLAB, XEDRO-GESPRO, XABAL EXCRIBA son sistemas que se utilizan para gestionar las lecciones aprendidas, pero como se plantea en la problemática cada centro realiza esta gestión de

manera independiente y lo que se quiere lograr con el sistema que se va a desarrollar, es la unificación del registro de lecciones y que se pueda acceder a esta información de manera organizada.

Dado el contexto y las necesidades de los centros productivos de la UCI, es fundamental abordar las desventajas asociadas al uso de los métodos existentes y desarrollar una propuesta de solución que permita gestionar el conocimiento de manera centralizada en la universidad. Según (Departamento de Gestión e Informatización ONN, 2021) tiene como propósito asegurar el entorno en el que la información esté accesible, se comparta, se transforme en conocimiento nuevo y permita su uso en beneficio de la organización. La universidad espera certificar el nivel básico de MCDAI y la norma plantea como resultados esperados que: se identifiquen las áreas del conocimiento clave, los activos de información y conocimiento, los expertos, las necesidades futuras de conocimiento, las brechas del conocimiento y los conocimientos organizacionales que serán compartidos con personal externo (Departamento de Gestión e Informatización ONN, 2021).

Existen varios formatos para documentar lecciones aprendidas, dependiendo de los propósitos que se pretendan lograr, las audiencias a las que están dirigidos, el tiempo y los recursos disponibles. El departamento de conocimiento y aprendizaje promueve la elaboración de notas del conocimiento para documentar el conjunto de lecciones aprendidas derivadas de la gestión de un determinado proyecto y su impacto en el desarrollo de iniciativas o procesos corporativos debido a la utilización de las mismas, proporciona una forma estructurada y organizada de almacenar el conocimiento adquirido. Además, permiten que la información se almacene y se pueda acceder fácilmente en el futuro, lo que es útil para la resolución de problemas, la toma de decisiones y la mejora continua (*PMBOK GUIDE*, 2021).

En el marco de la investigación y para dar solución al problema planteado se hará uso de las notas del conocimiento para registrar las LA las cuales permitirán documentar el conocimiento de las lecciones derivadas de los procesos en uno o varios proyectos, capturar, almacenar y compartir información valiosa para mejorar la calidad de los productos y servicios, evitando errores proyectos futuros. Se podrá identificar patrones y tendencias en el desempeño de un proyecto o de una actividad, fomentar la colaboración y el aprendizaje en equipo aprovechando así la experiencia y los conocimientos de todos para mejorar el desempeño del equipo en general y la organización del mismo.

### 1.3 Notas del conocimiento

Las notas del conocimiento son una técnica de organización y gestión de información que consiste en tomar notas de lo que se aprende, se lee o se investiga, estructurarlas de manera eficiente en el futuro (Sönke Ahrens, 2017). Es un bloque de construcción de conocimiento, una unidad discreta de información interpretada a través de su perspectiva única y almacenada fuera de la cabeza (Forte, 2020). Son un sistema de organización de información que permiten a las personas capturar, almacenar y recuperar conocimientos y aprendizajes importantes en su vida personal o profesional. Resalta el contexto en el que se desarrollan y privilegia la narrativa de la experiencia que se está analizando (Luna & Bu, 2015). A diferencia de un documento largo y formal, una nota es discreta: hace un solo punto, proviene de una sola fuente o tiene un solo propósito, es un bloque de construcción de conocimiento que se sostiene por sí mismo, sin embargo, cada bloque también se puede combinar con otros en obras mayores: un informe, un ensayo, un sitio web o un video (Forte, 2020).

El objetivo principal de las notas del conocimiento es ayudar a las personas a crear una base de conocimientos sólida y accesible que pueda utilizarse para tomar mejores decisiones, resolver problemas y generar nuevas ideas. Pueden ser cualquier tipo de información que se considere importante, desde ideas y conceptos hasta artículos y reflexiones personales. Lo importante es que se capturen de manera sistemática y se organicen de tal manera que puedan ser fácilmente accesibles y utilizables en el futuro.(Sönke Ahrens, 2017).

La utilización de las notas del conocimiento resulta ser una herramienta útil para el aprendizaje y el desarrollo de cualquier profesional la misma permite acceder a conocimientos y experiencias previas y aplicarlas en su entorno laboral para elevar su capital intelectual como parte de las GC.

Los autores antes referenciados coinciden en que las notas del conocimiento deben estructurarse a través de las siguientes secciones:

*1- Antecedentes:* describe la experiencia que se está analizando, sus objetivos, el contexto en el cual ésta tiene lugar y los momentos y/o factores críticos que condujeron a alcanzar o no los resultados esperados. En ese sentido, es clave la inclusión de una descripción detallada de los productos y/o resultados alcanzados.

*2- Descripción de las lecciones aprendidas y recomendaciones:* una nota puede documentar más de una lección aprendida identificada como producto del análisis de una determinada experiencia, proceso y/o proyecto. Para cada lección aprendida se debe incluir:

Las evidencias y elementos actuales, que justifican la existencia probable de una relación causal entre los resultados de una intervención y los factores críticos o condiciones que contribuyeron o no a alcanzarlos.

Recomendaciones o propuestas concretas, prácticas y adicionales, basadas en la lección enunciada, que permitirán resolver problemas, mitigar riesgos, repetir o reforzar éxitos. Las recomendaciones son claves para asegurar el rehúso de las lecciones aprendidas

*3-Referencias:* Incluye datos de contacto del autor o autores y otros actores claves involucrados, bibliografía del proyecto, experiencias similares. Estos datos son fundamentales para facilitar los contactos entre pares, la disseminación y el rehúso de las lecciones aprendidas.

Se recomienda que la nota: no exceda las ocho páginas (3,000 palabras); utilizar un lenguaje claro y simple; escribir párrafos cortos de manera de asegurar una lectura amena; evitar el uso excesivo de acrónimos; limitar el uso de citas, referencias y notas a pie de página; dar crédito explícito y reconocer la participación y el esfuerzo de los individuos o equipos que hicieron posible el análisis.

#### **1.4 Mejora de procesos de software**

Según la Organización Internacional de Normalización (ISO) 9001:2000, la mejora continua significa la eficacia constante del sistema de gestión de la calidad mediante el uso de las políticas de calidad, los resultados de las auditorías, el análisis de los datos, las acciones correctivas y preventivas y la revisión de la dirección. Esta definición no realiza distinción entre la industria sobre la cual aplica la mejora continua, sin embargo, la mejora continua al desarrollo de software como un enfoque estructurado, permite mejorar continuamente sus capacidades para proporcionar servicios de calidad (García Rodríguez, 2018).

El éxito de un proyecto según (*PMBOK GUIDE, 2021*) depende de varios factores. Estos indican que es importante que los objetivos de dicho proyecto estén claramente definidos y sean alcanzables. Una buena planificación es crucial para asegurar que se puedan abordar los problemas de manera oportuna. Identificar y gestionar adecuadamente los riesgos del proyecto es esencial para disminuir los problemas y propiciar el éxito. Además, una gestión adecuada de los recursos, incluyendo personal, tiempo y presupuesto, es fundamental para asegurar que el proyecto se complete dentro

del plazo y el presupuesto establecido. Asegurar la calidad del trabajo y los productos entregados es fundamental para el éxito del trabajo a largo plazo, con la realización de pruebas tempranas para el control de calidad se pueden detectar defectos y fallas antes de la puesta en operación del sistema, con la identificación de esto el equipo de trabajo ahorra tiempo y presupuesto.

La MPS tiene como objetivo principal incrementar el rendimiento y la utilidad de los procesos. Su esencia radica en la aplicación de BP que beneficien la mejora continua de los procesos. Diversas investigaciones insisten en la necesidad de reutilizar las experiencias previas de las organizaciones, considerando los factores críticos que influyen en el éxito. Sin embargo, se refleja mucha dispersión respecto al cómo ejecutar estas buenas prácticas para una mejora exitosa. Además, se considera esencial la reutilización de experiencias para apoyar la toma de decisiones, es por ello que resulta pertinente la definición de recomendaciones que guíen la aplicación de las BP, en base a los resultados obtenidos por otras organizaciones que han iniciado previamente un proceso similar (Socarras Peña et al., 2022).

Desarrollar un proyecto exitoso por sí solo tiene pocas ventajas, la incorporación y utilización de las experiencias anteriores de manera sistemática en la definición e implementación de los procesos es el verdadero éxito, pues eleva la madurez y capacidad de los mismos. Aunque el resultado final de un proyecto de desarrollo de software depende de la intervención de muchos factores, constituye un aspecto esencial centrarse en el proceso, este compone un elemento clave para la mejora continua de los procesos (Febles Pérez et al., 2022). Estos proyectos se centran en mejorar el rendimiento, la utilidad y la efectividad de los procesos de una manera disciplinada, no obstante esta ventaja no siempre se observa a corto o mediano plazo, la implantación de BP es una labor cuyo resultado puede tardar años en obtenerse (Socarras Peña et al., 2022).

En la UCI el registro de las LA se recopilan a partir de las actividades de calidad que se desarrollan a lo largo del ciclo de vida de un producto o servicio, es un proceso que está institucionalizado porque el modelo por el cual se rigen los procesos de desarrollo de software lo abordan, pero cada centro de desarrollo lo realiza de manera independiente lo que trae como consecuencia la pérdida de la información relevante con valor notable. La documentación de las LA está estrechamente relacionada con la mejora de los procesos. El registro de las LA, permite que el conocimiento adquirido en base a las experiencias que se obtienen durante la realización de un proceso quede registrado de manera tal que en algún otro momento pueda ser reutilizado. También permite el análisis sobre los factores que pueden haber afectado positiva o negativamente a esa experiencia, pues así los errores y aciertos

quedan guardados para ser usados en futuras iniciativas y de esta manera la organización realiza una toma de decisiones para futuros planes en la que le permita beneficiarse de las buenas prácticas, aprendiendo y mejorando (Departamento de Gestión e Informatización ONN, 2021). La mejora de procesos es importante porque ayuda a las entidades a ser más eficientes, mejorar la calidad, ser más flexibles y adaptables. Permite fomentar la innovación y reducir los costos, lo que se a su vez puede mejorar la rentabilidad y la competitividad además de facilitar la toma de decisiones a la hora de emprender o finalizar cualquier tipo de proyecto.

### **1.5 Metodologías de desarrollo de software**

Una metodología de desarrollo de software es un enfoque estructurado que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos. Las metodologías han evolucionado de manera significativa en las últimas décadas, tanto así, que pueden permitir el éxito o el fracaso de muchos de los sistemas desarrollados para distintas áreas (Oquendo Arencibia, 2018). Son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible (Santander Universidades, 2020).

En la actualidad se pueden mencionar dos grandes grupos de metodologías de desarrollo de software (Massengo Nekaka, 2020):

*Metodología Tradicional:* son aquellas que ponen mayor énfasis en la planificación y control del proyecto y en la especificación precisa de requisitos y modelado.

*Metodología Ágil:* están más orientadas a la generación de código con ciclo muy corto de desarrollo, se dirigen a equipos pequeños, priorizan los recursos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

La metodología ágil es un enfoque para el desarrollo de software que se centra en entregar el máximo valor al cliente en un corto periodo de tiempo. Se basa en un proceso iterativo e incremental, en el que se desarrolla el software en pequeños ciclos llamados sprints. El objetivo de esta metodología es proporcionar un proceso más flexible y adaptable que los modelos tradicionales de desarrollo de software, permitiendo que los equipos de desarrollo respondan rápidamente a los cambios en los requisitos del cliente (*The Scrum Guide*, 2020).

<i>Metodología Tradicional</i>	<i>Metodología ágil</i>
Predictivo	Adaptativo
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños.
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa.	Poca documentación

**Tabla 2.** Comparación de Metodologías. Elaboración propia a partir de (Montero et al., 2018)

Según la comparativa realizada (**Tabla 2**), teniendo en cuenta que el equipo de trabajo está compuesto por dos integrantes y contando con tiempo aproximado de 5 meses se decide seleccionar la metodología ágil

Existen varias metodologías ágiles que se han vuelto cada vez más populares en los últimos años especialmente en proyectos de desarrollo de software a gran escala. Entre ellas se encuentra Scrum, esta metodología tiene sus propias prácticas, principios específicos y la importancia de la comunicación frecuente y la colaboración entre el equipo de desarrollo y el cliente (*Agile Alliance*, 2023).

**SCRUM** es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo. Las actividades estructurales que incorporan son: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón

del proceso llamado sprint. El trabajo realizado dentro de un sprint (el número de estos que requiere cada actividad estructural varía en función de la complejidad y tamaño del producto) se adapta al problema en cuestión (Pressman & Maxim, 2021). El uso de esta metodología tiene como ventajas (Pérez Rodríguez, 2022):

- Satisfacción del cliente: El gran diferenciador de las metodologías ágiles, es que hacen al cliente parte del equipo de trabajo y lo comprometen con el resultado final. Esto es un gran cambio con respecto a metodologías tradicionales, en los que el cliente es una persona, o grupo de personas, con el que se realiza la tarea de levantar requerimientos o necesidades de tipo funcional y que posteriormente aprueba extensos documentos que no volverá a ver hasta que el producto se encuentre terminado.
- Simplicidad: los eventos manejados por Scrum están claramente identificados, indicando para cada uno: quiénes participan, su objetivo, el tiempo que debe tomar y cuál es el resultado esperado. Lo cual en esencia facilita a los integrantes del equipo la adopción de la metodología.
- Adaptación: la mejor parte de la metodología es la disposición que tienen al cambio las características del producto. Este es uno de los componentes que más la diferencia con el resto, el cambio puede ser efectuado en cualquier momento, incluso dentro del desarrollo de la ejecución de las diferentes iteraciones o sprints siempre y cuando no afecte la entrega pactada.

Para guiar el desarrollo de la propuesta de solución se selecciona la metodología Scrum como parte del cumplimiento de las exigencias del cliente, el cual planteó la necesidad de estar involucrado durante todo el proceso de desarrollo para validar la propuesta de solución a su problemática. Es una metodología que adopta buenas prácticas de otras metodologías ágiles para crear un marco de trabajo efectivo que se centre en la entrega de valor, la flexibilidad y la mejora continua.

## **1.6 Tecnología, lenguaje de programación y herramientas.**

Se realiza un estudio de las tecnologías, lenguajes de programación y herramientas seleccionadas para el desarrollo del sistema.

### **1.6.1 Lenguaje de programación**

A diferencia de las aplicaciones de escritorio, las aplicaciones web se ejecutan a través de un navegador web, lo que significa que los lenguajes de programación utilizados deben de ser compatibles con la tecnología web. Por esta razón para el desarrollo de este sistema solo se usarán lenguajes de programación web.

### **HTML (versión 5.0)**

El lenguaje de Marcado para Hipertexto (HTML por sus siglas en inglés) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia o presentación de una página web (CSS) o su funcionalidad (JavaScript) (*HTML*, 2023).

### **CSS (versión 3.0)**

Hojas de estilo en Cascada (Cascading Style Sheets) es el lenguaje de estilos utilizados para describir la presentación de documentos HTML o XML (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML). CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios (*CSS | MDN*, 2023).

### **JavaScript (versión ES 2021)**

JavaScript (JS) es un lenguaje de programación ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache CouchDB. Es un lenguaje script multiparadigma basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa (*JavaScript | MDN*, 2023).

### **Python (versión 3.11)**

Python, creado por Guido Van Rossum, es un lenguaje de programación diseñado para ser ejecutado por medio de un intérprete y diferentes plataformas. Es multiparadigma, permitiendo varios estilos de programación ya sea programación orientada a objetos, programación estructurada o programación funcional. Permite mantener de forma sencilla la interacción con el sistema operativo, y resulta muy adecuado para manipular archivos de texto. Su gestor de memoria, la gran cantidad de librerías disponibles y la potencia de este lenguaje, hacen que desarrollar una aplicación en Python sea sencillo, rápido y lo más importante legible para el desarrollador. Permite trabajar más rápido e integrar sus sistemas de manera más efectiva puede ser fácil de aprender tanto si es un programador principiante como si tiene experiencia con otros lenguajes (*Welcome to Python.Org*, 2023).

Python ofrece muchas ventajas (Marzal & Gracia, 2020):

- Suele ser bastante más corto que su equivalente en lenguajes como C.
- La sintaxis es muy elegante y permite la escritura de programas cuya comprensión resulta menos complicada que si se hiciera uso de otro lenguaje de programación.
- El entorno de ejecución de este lenguaje detecta muchos de los errores de programación que se escapan al control de los compiladores y proporciona información muy diversa para detectarlos y corregirlos.

Todos los lenguajes de programación analizados anteriormente serán utilizados para implementar la propuesta de solución. En JavaScript se utilizará la librería Sweetalert que va a permitir que el sistema muestre los mensajes de notificación cuando el usuario realice alguna acción. También se va a utilizar la librería Chart.js que permitirá al sistema mostrar los gráficos. El lenguaje CSS se va a utilizar para darle estilo a la página web. El lenguaje HTML es el lenguaje que se va a utilizar para estructurar la página web.

### **1.6.2 Sistema gestor de base de datos**

Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (Álvarez Medel & Cleger Despaigne, 2017).

#### **PostgreSQL (versión web)**

Es un poderoso sistema de base de datos relacional de objetos de código abierto con más de 35 años de desarrollo activo que le ha valido una sólida reputación por su confiabilidad, robustez de funciones y rendimiento. PostgreSQL viene con muchas funcionalidades destinadas a ayudar a los desarrolladores a crear aplicaciones, a los administradores a proteger la integridad de los datos y crear entornos tolerantes a fallas, y ayudarlo a administrar sus datos sin importar cuan grande o pequeño sea el conjunto de datos. Además de ser gratuito y de código abierto, es altamente extensible, es decir puede definir sus propios tipos de datos, crear funciones personalizadas e incluso escribir código de diferentes lenguajes de programación sin volver a compilar su base de datos (PostgreSQL, 2023).

### **1.6.3 Framework de desarrollo**

Un marco de trabajo o framework es un diseño abstracto orientado a objetos para un determinado tipo de aplicación, es un patrón arquitectónico que proporciona una plantilla extensible para un tipo específico de aplicaciones. Para el desarrollo de este sistema se hará uso del framework de desarrollo Django.

#### **Django (versión 4.1)**

Django es un marco web Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de gran parte de las molestias del desarrollo web. Es gratis y de código abierto. Fue diseñada para ayudar a los desarrolladores a llevar las aplicaciones desde el concepto hasta su finalización lo más rápido posible. Toma la seguridad en serio y ayuda a los desarrolladores a evitar muchos errores de seguridad comunes (*Django*, 2023).

El uso de Django ofrece importantes ventajas. El admin panel, un panel de administración que viene instalado por defecto. Este administrador implementa un CRUD a la base de datos de una manera sencilla. Cuenta con un sólido sistema de permisos para restringir el acceso a los datos como tú quieras. Abstrae la necesidad de escribir consultas SQL para crear tablas y consultar datos. Es bastante intuitivo de usar y tiene incluidas casi todas las consultas más comunes en su código. Desde filtrados, particionados, uniones e incluso hasta funciones. Django cuenta con muchísimos paquetes para resolver la mayoría de los problemas comunes, además son paquetes supervisados y mejorados por la comunidad, lo que garantiza una calidad impresionante.

### **1.6.4 Otras herramientas**

Para el modelado de los artefactos generados durante el desarrollo de la presente investigación se decide utilizar la herramienta Visual Paradigm. Además, como editor de código la herramienta Visual Code.

#### **Visual Paradigm (versión 8.0)**

Es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software, análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El lenguaje de modelado UML ayuda a una rápida construcción de aplicaciones de calidad. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. La

herramienta CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML (Visual Paradigm, 2023).

Entre las ventajas que brinda Visual Paradigm se destacan la navegación intuitiva entre el código y el modelo. Un poderoso generador de documentación y reportes UML PDF/HTML/MS Word. Soporte completo de notaciones UML. Diagramas de diseño automático sofisticado. Análisis de texto y soporte de tarjeta CRC. Proporciona soportes a varios lenguajes en generación de código e ingeniería inversa a través de plataformas Java, C++, entre otros lenguajes de programación.

### **Visual Studio Code (versión 1.78)**

Es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución como (C++, Java, Python, PHP, Go.Net) (Visual Studio Code, 2023).

### **Conclusiones del capítulo**

Una vez terminado el capítulo se arriban a las siguientes conclusiones parciales:

- El análisis de sistemas homólogos reveló que los sistemas internacionales son privativos y limitados en cuanto a su acceso y uso y los sistemas utilizados en la universidad no logran satisfacer plenamente la problemática definida en la investigación.
- Para el desarrollo del sistema de gestión para el registro de las lecciones aprendidas en la Universidad de las Ciencias informáticas se identifica a Django como framework de desarrollo, Python como lenguaje de programación y como sistema gestor de bases de datos PostgreSQL, Scrum como metodología para guiar el proceso de desarrollo de la solución.

## **CAPITULO 2 DISEÑO DE SOLUCIÓN PARA EL SISTEMA DE GESTIÓN DE LECCIONES APRENDIDAS**

El presente capítulo se identifican los requisitos funcionales y no funcionales con los que debe cumplir la solución propuesta, aborda los aspectos relacionados con las características de la propuesta de solución, así como el estilo arquitectónico para lograr buenas prácticas en el diseño y posterior implementación del sistema. Igualmente se muestran los principales artefactos de ingeniería de software propuestos por la metodología utilizada.

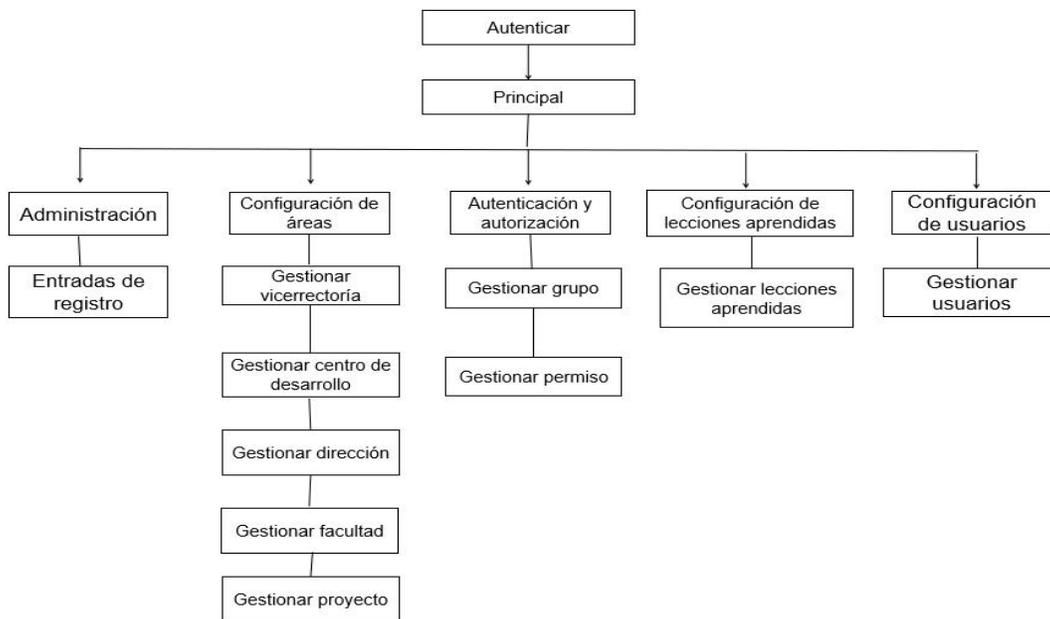
### **2.1 Propuesta de solución**

Para dar respuesta al objetivo de esta investigación, luego de analizar las necesidades del cliente y realizar la selección de las herramientas y metodología para la implementación, se definen los elementos a desarrollar. Se propone la implementación de un sistema de gestión para registrar las lecciones aprendidas en la universidad. Este sistema estará basado en una plataforma digital que permita a todos los usuarios autorizados documentar y compartir las lecciones aprendidas. El sistema presentará importantes funcionalidades.

Permitirá el registro de lecciones aprendidas, con esta funcionalidad los usuarios podrán ingresar información detallada sobre las lecciones aprendidas en sus proyectos y procesos. En este sistema las LA podrán ser categorizadas y etiquetadas según el área de proceso y al proyecto al que pertenecen facilitando así su búsqueda y consulta. Los usuarios podrán compartir las lecciones identificadas con otros miembros de la comunidad académica, fomentando la colaboración y el intercambio de conocimiento. El sistema permitirá realizar análisis de las lecciones aprendidas para identificar patrones, tendencias y áreas de mejora. Además, permitirá registrar todas las acciones que se realicen permitiéndole así al administrador tener el control del sistema y de todo lo que puedan realizar los usuarios. La implementación del mismo traerá consigo beneficios importantes para la organización ya que se podrá identificar oportunidades de mejora en los procesos y proyectos. Facilitará la captura y documentación del conocimiento generado en la institución. Fomentará la colaboración entre todos los miembros de la organización, promoviendo un ambiente de aprendizaje compartido y colaborativo. Proporcionará información valiosa para la tomade decisiones en la planificación y ejecución de proyectos académicos.

Se diseñó la propuesta de solución (Ver figura 4) que inicia con una interfaz de autenticar usuario donde el administrador del sistema es el encargado de registrar a cada usuario, asignándole un

usuario y una contraseña, después que el usuario tenga esta información podrá acceder al sistema cuando lo necesite. Seguidamente se encuentra la página principal que posee 5 módulos. En el módulo administración se tendrá el control de todas las acciones que se realizan en el sistema. El módulo configuración de áreas será el encargado de gestionar vicerrectorías, centros de desarrollo, direcciones, facultades y proyectos. El módulo autenticación y autorización será el encargado de asignar roles y permisos a los usuarios del sistema. El módulo de configuración de lecciones aprendidas será el encargado de gestionar toda la información necesaria sobre las lecciones aprendidas mediante el uso de las notas del conocimiento y los proyectos a los que pertenecen. Para finalizar el módulo de configuración de los usuarios tendrá como objetivo gestionar toda la información de los usuarios del sistema.



**Figura 4.** *Diseño de la propuesta de solución. Elaboración propia.*

Los usuarios del sistema podrán interactuar con el sistema según sus privilegios establecidos en niveles, donde:

*Nivel de usuario alta gerencia:* serán los que tienen acceso a todas las funcionalidades del sistema. Ejemplo de usuario de alta gerencia es el rector de la universidad.

*Nivel de usuario común:* serán los que solo tienen accesos permitidos para interactuar con algunas funcionalidades del sistema. Estos usuarios no tienen permisos para eliminar y modificar ninguna

información al no ser la suya personal, solo les es permitido añadir y ver toda la información de los demás usuarios.

## 2.2 Especificación de requisitos

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema (Requisitos funcionales) y sus restricciones operativas (Requisitos no funcionales). Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema (Sommerville, 2011).

### 2.2.1 Product Backlog

El Product Backlog es una lista emergente y ordenada de lo que se necesita para mejorar el producto. Es la única fuente del trabajo realizado por el equipo de Scrum. Los elementos del Product Backlog que el equipo de Scrum puede dar por terminados dentro de un sprint se consideran preparados para ser seleccionados en un Sprint. Suelen adquirir este grado de transparencia tras las actividades de refinamiento (Schwaber & Sutherland, 2020).

En la **Tabla 2** se muestran los requisitos funcionales de la propuesta de solución, a los cuales se les asignó una prioridad teniendo en cuenta la importancia fijada por el cliente a partir de sus necesidades.

<i>No</i>	<i>Nombre</i>	<i>Descripción</i>	<i>Prioridad</i>
RF 1	Autenticar Usuario	El sistema permite que el usuario introduzca sus datos para acceder al sitio web (usuario y contraseña).	Alta
RF 2	Registrar usuario	El sistema permite que el usuario se registre mediante el nombre de usuario, contraseña y publicaciones de este usuario en revistas científicas.	Alta
RF 3	Editar usuario	El sistema permite editar un usuario por el nombre, el nombre de usuario, los apellidos, el sexo, el teléfono, el correo, los títulos académicos, el	Media

		grado científico, el rol, los años de experiencia, las certificaciones académicas, los cursos de formación, el grupo y las publicaciones.	
RF 4	Eliminar usuario	El sistema permite eliminar los datos del usuario.	Media
RF 5	Listar usuario	El sistema muestra un listado con todos los usuarios del sistema y pueden ser filtrados por títulos académicos, categoría docente, grado científico o por rol.	Media
RF 6	Buscar usuario	El sistema permite filtrar por títulos académicos, categoría docente, grado científico o por rol.	Media
RF 7	Registrar vicerrectoría	Descripción: El sistema permite registrar una vicerrectoría por el nombre y por una descripción.	Alta
RF 8	Editar vicerrectoría	El sistema permite editar la vicerrectoría mediante el nombre y la descripción.	Media
RF 9	Eliminar vicerrectoría	El sistema permite al usuario eliminar una vicerrectoría.	Media
RF 10	Listar vicerrectoría	El sistema muestra un listado de todas las vicerrectorías registradas en el sistema.	Media
RF 11	Buscar vicerrectoría	El sistema permite buscar una vicerrectoría por el nombre.	Media
RF 12	Registrar dirección	El sistema permite registra una dirección por el nombre y por una descripción.	Alta
RF 13	Editar dirección	El sistema permite editar	Media

		una dirección por el nombre y la descripción.	
RF 14	Eliminar dirección	El sistema permite al usuario eliminar una dirección.	Media
RF 15	Listar dirección	El sistema muestra un listado de todas las direcciones registradas en el sistema.	Media
RF 16	Buscar dirección	El sistema permite buscar una dirección por el nombre.	Media
RF 17	Registrar centro de desarrollo	El sistema permite registrar un centro de desarrollo por el nombre y por una descripción.	Alta
RF 18	Editar centro de desarrollo	El sistema permite editar un centro de desarrollo por el nombre y la descripción.	Media
RF 19	Eliminar centro de desarrollo	El sistema permite al usuario eliminar un centro de desarrollo.	Alta
RF 20	Listar centro de desarrollo	El sistema muestra un listado de todos los centros de desarrollo registrados en el sistema.	Media
RF 21	Buscar centro de desarrollo	El sistema permite buscar un centro de desarrollo por el nombre.	Media
RF 22	Registrar lección aprendida	El sistema permite registrar una lección aprendida por el proyecto al que pertenecen, el área de proceso a la que pertenecen, antecedentes, descripción, recomendaciones, referencias y la fecha de registrado	Alta
RF 23	Editar lección aprendida	El sistema permite editar una lección aprendida por	Media

		el proyecto al que pertenecen, el área de proceso a la que pertenecen, antecedentes, descripción, recomendaciones, referencias y la fecha de editado.	
RF 24	Eliminar lección aprendida	El sistema permite al usuario eliminar una lección aprendida.	Alta
RF 25	Listar lección aprendida	El sistema muestra un listado de todas las lecciones aprendidas registradas en el sistema.	Alta
RF 26	Buscar lección aprendida	El sistema permite filtrar por los centros de desarrollo, la dirección, la facultad, rectoría y vicerrectoría.	Alta
RF 27	Registrar facultad	El sistema permite registrar una facultad por el nombre y por una descripción.	Alta
RF 28	Editar facultad	El sistema permite editar una facultad por el nombre y la descripción.	Media
RF 29	Eliminar facultad	El sistema permite al usuario eliminar una facultad	Media
RF 30	Listar facultad	El sistema muestra un listado de todas las facultades registradas en el sistema.	Media
RF 31	Buscar facultad	El sistema permite buscar la facultad por el nombre.	Media
RF 32	Registrar proyecto	El sistema permite registrar un proyecto por el nombre y por una descripción.	Alta
RF 33	Editar proyecto	El sistema permite editar un proyecto por el nombre	Media

		y la descripción.	
RF 34	Eliminar proyecto	El sistema permite al usuario eliminar una facultad	Alta
RF 35	Listar proyecto	El sistema muestra un listado de todos los proyectos registrados en el sistema.	Media
RF 36	Buscar proyecto	El sistema permite buscar el proyecto por el nombre.	Media
RF 37	Registrar certificaciones académicas	El sistema permite registrar una certificación académica por el nombre y por una descripción.	Alta
RF 38	Editar certificaciones académicas	El sistema permite editar una certificación académica por el nombre y por la descripción.	Media
RF 39	Eliminar certificaciones académicas	El sistema permite al usuario eliminar certificaciones académicas.	Media
RF 40	Listar certificaciones académicas	El sistema muestra un listado de todas las certificaciones académicas registradas en el sistema.	Media
RF 41	Buscar certificaciones académicas	El sistema permite buscar las certificaciones académicas por el nombre.	Media
RF 42	Registrar grupo de formación	El sistema permite registrar un grupo de formación por el nombre y por una descripción.	Media
RF 43	Editar grupo de formación	El sistema permite editar un grupo de formación por el nombre.	Media
RF 44	Eliminar grupo de formación	El sistema permite al usuario eliminar el grupo	Media

		de formación.	
RF 45	Listar grupo de formación	El sistema muestra un listado de todos los grupos de formación registrados en el sistema.	Media
RF 46	Buscar grupo de formación	El sistema permite buscar grupo de formación por el nombre.	Media
RF 47	Registrar publicación	El sistema permite registrar una publicación por el nombre, nivel al que pertenece y el nombre del usuario quien la realizó.	Media
RF 48	Editar publicación	El sistema permite editar una publicación por el nombre, nivel al que pertenece y el nombre del usuario quien la realizó	Media
RF 49	Eliminar publicación	El sistema permite al usuario eliminar una publicación.	Media
RF 50	Listar publicación	El sistema muestra un listado de todas las publicaciones registradas en el sistema.	Media
RF 51	Buscar publicación	El sistema permite buscar la publicación por el nombre.	Media
RF 52	Registrar grupo	El sistema permite registrar un grupo por el nombre el nombre y una descripción.	Alta
RF 53	Editar grupo	El sistema permite editar un grupo por el nombre y una descripción.	Media
RF 54	Eliminar grupo	El sistema permite al usuario eliminar un grupo.	Media
RF 55	Listar grupo	El sistema muestra un listado de todos los grupos registrados en el sistema.	Media

RF 56	Buscar grupo	El sistema permite buscar un grupo por el nombre.	Baja
RF 57	Cambiar contraseña	El sistema permite al usuario cambiar la contraseña.	Alta
RF 58	Mostrar cantidad de lecciones aprendidas por áreas de proceso	El sistema permite mostrar la cantidad de lecciones aprendidas por área de procesos en forma de gráfica.	Alta
RF 59	Mostrar cantidad de usuarios activos	El sistema permite mostrar la cantidad de usuarios activos e inactivos en forma de gráfica.	Media
RF 60	Mostrar cantidad de usuarios por roles	El sistema permite mostrar la cantidad de usuarios por roles en forma de gráfica.	Media
RF 61	Mostrar entradas de registros	El sistema permite el control de todas las acciones que se realicen en el sistema.	Alta

*Tabla 3. Requisitos Funcionales. Elaboración propia.*

## 2.2.2 Requisitos no funcionales

### RnF1. Usabilidad

**RnF1.1** El sistema debe contar con una función de búsqueda. Debe permitir búsquedas por palabras claves, filtros y otros criterios de búsquedas más específicos.

**RnF1.2** Los mensajes para interactuar con los usuarios y los de error deben ser lo suficientemente informativos y en idioma español para permitir la fácil comprensión del usuario y debe estar diseñado de forma que sea más fácil de utilizar para el usuario.

### RnF2. Requisitos de interfaz externa

**RnF2.1** El sistema debe tener una apariencia profesional y un diseño gráfico sencillo, con la utilización de tonalidades de colores correspondientes (Negro y Azul).

**RnF2.2** El sistema debe tener una estructura modular claramente definida, con interfaces bien documentadas, y fácilmente accesible para los usuarios externos.

### **RnF3. Hardware**

**RnF3.1** El servidor donde se instale el sistema debe tener una memoria *RAM* de 2GB o superior y un procesador a una velocidad 2.10 GHz o superior y para el servidor de base datos 10 GB de disco duro.

### **RnF4. Software**

**RnF4.1** Tener navegadores web instalados (Mozilla Firefox, Chrome, Microsoft Edge, Safari, Opera) que sean compatible con HTML5, CCS y Bootstrap5.

**RnF4.2** El sistema debe ser multiplataforma.

### **RnF5. Seguridad**

**RnF5.1** La información manejada en el sistema debe estar protegida del acceso no autorizado mediante los mecanismos de control de acceso que establece el marco de trabajo Django.

**RnF5.2** El sistema debe permitir el acceso a los usuarios registrados en la base de datos y que se encuentren correctamente autenticados.

**RnF5.3** El sistema debe contar con medidas de seguridad, como la autenticación de usuarios y el control de acceso.

### **RnF6. Mantenimiento y Soporte**

**RnF6.1** El sistema debe ser escalable, permitiendo incorporarle nuevas funcionalidades sin afectar las existentes.

### **RnF7. Portabilidad**

**RnF7.1** La aplicación debe ser compatible con otros sistemas operativos (Nova, Linux y otras versiones de Windows).

### 2.3 Historias de Usuario

El refinamiento del Product Backlog es el acto de dividir y definir aún más los elementos del Product Backlog en elementos más pequeños y precisos (Schwaber & Sutherland, 2020). Para el refinamiento del Product Backlog se harán uso de las historias de usuario.

Las historias de usuario se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal. Describen, en una o dos frases, una funcionalidad de software desde el punto de vista del usuario (Sánchez, 2022).

A continuación, se muestra dos ejemplos de las historias de usuario definidas para llevar a cabo el desarrollo del sistema. Para ver el resto visitar el **Anexo 5**.

<b>Historia de Usuario 22</b>	
Nombre: Registrar lección aprendida	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema permite registrar una lección aprendida por el proyecto al que pertenecen, el área de proceso a la que pertenecen, antecedentes, descripción, recomendaciones, referencias y la fecha de registrado	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

**Tabla 4.** Prototipo de Historia de Usuario Registrar lección aprendida. Elaboración propia.

<b>Historia de Usuario 23</b>	
Nombre: Editar lección aprendida	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema permite editar una lección aprendida por el proyecto al que pertenecen, el área de proceso a la que pertenecen, antecedentes, descripción, recomendaciones, referencias y la fecha de editado.	

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

**Tabla 5.** Prototipo de Historia de Usuario Editar lección aprendida. Elaboración propia.

## 2.4 Planificación de los Sprint

El trabajo pendiente de sprint es un plan por y para los desarrolladores. Es una imagen muy visible en tiempo real del trabajo que los desarrolladores planean realizar para lograr el objetivo. Se realiza durante la reunión de la planificación del sprint en la que el equipo de desarrollo y el propietario del producto seleccionan las tareas a realizar durante un sprint (Schwaber & Sutherland, 2020).

El sprint comienza con una reunión de planificación en la que el equipo de desarrollo y el propietario del producto seleccionan las tareas a realizar durante el sprint basándose en las prioridades que tengan los requisitos mencionados y descritos en el Product Backlog.

<b>ID</b>	<b>Tareas</b>	<b>Estado</b>	<b>Tiempo estimado</b>	<b>Tiempo Inicial</b>	<b>Tiempo final</b>
T1	Reunión de planificación del equipo de desarrollo y el propietario del producto	Resuelta	24h	3/04/2023	4/04/2023
T2	Instalación y configuración de la tecnología de desarrollo Django	Resuelta	24h	4/04/2023	5/04/2023
T3	Implementación de los requisitos funcionales 22, 23, 24, 25, 26.	Pendiente	72h	5/04/2023	8/04/2023
T4	Implementación de los requisitos funcionales 7,8,9,10,11,12,13,14,15,16,17,18,19,20,21-	Resuelta	120h	8/04/2023	13/04/2023
T5	Implementación de los requisitos funcionales 1,2,3,4,5,6.	Resuelta	72h	13/04/2023	16/04/2023
T6	Pruebas de validación	Resuelta	96h	16/04/2023	20/04/2023
T7	Revisión del sprint	Resuelta	72h	20/04/2023	23/04/2023

**Tabla 6.** Planificación del Sprint 1. Elaboración propia.

Durante cada sprint el equipo de desarrollo trabaja en las tareas seleccionadas. Se realiza una reunión diaria de seguimiento, donde cada miembro del equipo informa sobre su progreso, los obstáculos que enfrenta y las tareas que va a enfrentar durante el día. Al finalizar cada sprint se

realiza una revisión del mismo en la que el equipo muestra los avances, además llevan a cabo un proceso de pruebas para validar el funcionamiento de lo implementado hasta el momento o las tareas completadas al cliente y otros interesados, además muestra una retrospectiva del sprint realizado en la que el equipo reflexiona lo que salió bien, lo que salió mal y como se pueden mejorar en el próximo sprint.

<b>ID</b>	<b>Tareas</b>	<b>Estado</b>	<b>Tiempo estimado</b>	<b>Tiempo Inicial</b>	<b>Tiempo final</b>
T1	Reunión de planificación del equipo de desarrollo y el propietario del producto	Resuelta	24h	23/04/2023	24/04/2023
T2	Implementación de los requisitos funcionales 23, 24.	Resuelta	48h	24/04/2023	26/04/2023
T3	Implementación de los requisitos funcionales 32,33,34,35,36,47,48,49,50,51.	Resuelta	72h	26/04/2023	29/04/2023
T4	Implementación de los requisitos funcionales 37,38,39,40,41,42,43,44,45,52,53,54,55,56	Resuelta	72h	29/04/2023	2/05/2023
T5	Implementar los requisitos funcionales 57,58,59,60,61.				
T6	Pruebas de validación	Resuelta	96	2/05/2023	6/05/2023
T7	Revisión del sprint	Resuelta	72h	6/05/2023	9/05/2023

**Tabla 7.** Planificación del Sprint 1. Elaboración propia.

## 2.5 Diseño de la arquitectura de software

Para que un software sea desarrollado con calidad es necesario establecer desde el inicio una arquitectura que describa sus principios fundamentales, garantizando robustez y escalabilidad. La arquitectura de software define la estructura del sistema, constituida por componentes con funciones específicas que interactúan entre sí (Avila, 2022) .

### 2.5.1 Patrón arquitectónico

Un patrón arquitectónico brinda la descripción de un problema particular y recurrente de diseño, que aparece en contexto de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. El patrón arquitectónico es quien define la estructura básica de la aplicación (Avila, 2022). Para el diseño del sistema se hará uso de una versión del patrón Modelo-Vista-Controlador el cual garantiza la separación del código en tres capas, dependiendo de su naturaleza (Ennis, 2022).

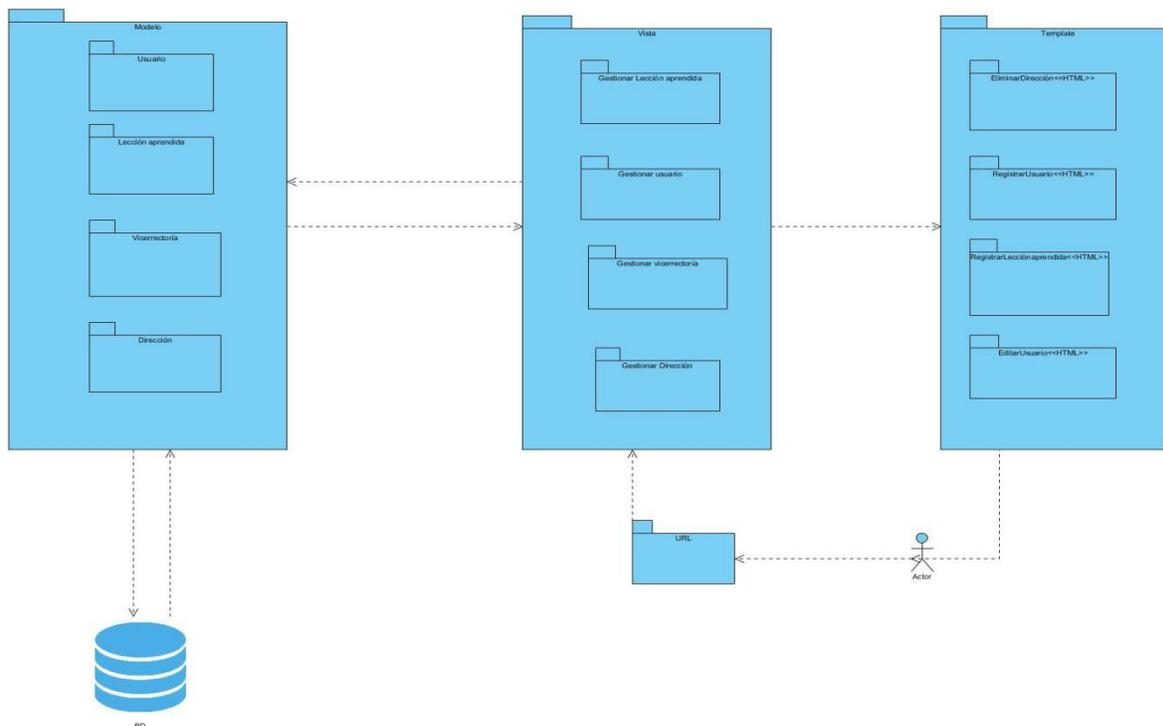
#### ***Modelo-vista-template***

Es un patrón de arquitectura de software utilizado por el framework del lenguaje de programación Python, Django. Es una variante del patrón arquitectónico Modelo- Vista – Controlador (Prezi, 2023)

**-Modelo:** la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos

**-Vista:** la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: se puede pensar en esto como un puente entre el modelo y las plantillas.

**-Template:** Llamada también Plantilla, es la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.



**Figura 5.** Representación del patrón arquitectónico Modelo- Vista -Template. Elaboración propia

### Descripción de la representación del patrón arquitectónico

-*Paquete Modelo:* en el modelo se agrupan los paquetes Facultad, Usuario, Dirección, Vicerrectoría, Centro, Lección aprendida, Publicación, Certificación académica, Grupo de formación, Grupo y Proyecto.

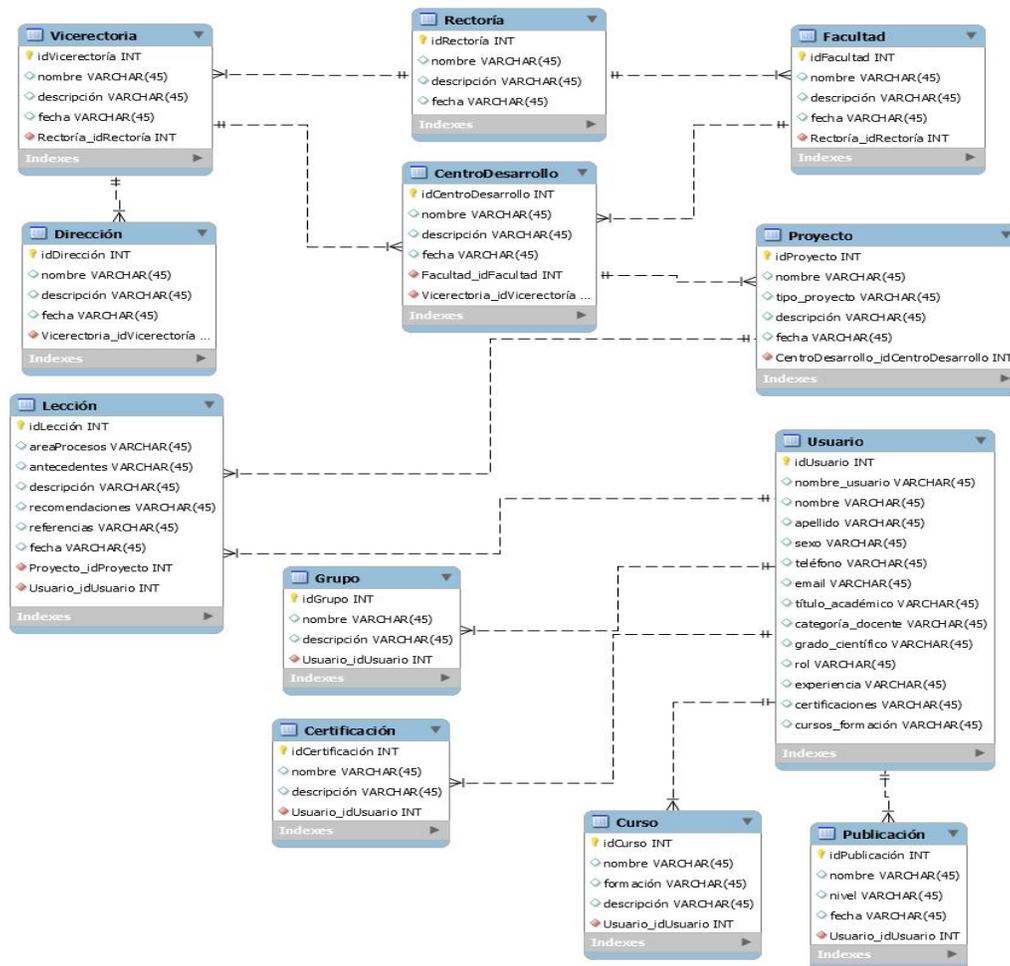
-*Paquete Vista:* en la vista se muestran cada una de las interfaces con las que interactúa el usuario. Se agrupan los paquetes que controlan las peticiones de una de las template que posee el sistema.

-*Paquete Template:* se encarga de enviar las peticiones al modelo que son recibidas desde la base de datos y mostrada mediante las interfaces guardadas en las vistas. Se agrupan los paquetes con las páginas HTML que contiene el diseño y la estructura del sistema.

## 2.6 Modelo de datos

Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos, la relación que existe entre sí, los

procesos que los transforman y cómo se accede a ellos. Se basa en la identificación de los objetos primarios que va a procesar el sistema, la composición y atributos de los mismos. En algunos casos, esta base de datos es independiente del sistema software (Sommerville, 2011).



**Figura 6.** Modelo de datos. Elaboración propia.

El modelo de datos obtenido para la propuesta de solución consta de 12 tablas, teniendo dos tablas principales: usuario y lección. La primera almacena la información requerida para los usuarios registrados en el sistema, mientras que la segunda guarda la información referente a las lecciones aprendidas. Las tablas (rectoría, vicerrectoría, dirección, centro, facultad, proyecto), constituyen áreas de la universidad, mientras que las tablas (grupo, certificación, curso, publicación) son características de los usuarios del sistema.

## 2.7 Patrones de Diseño

Los patrones de diseño son una técnica para resolver problemas comunes en el desarrollo web y otros ámbitos referentes al diseño de interacción o interfaces. Establece una relación entre un determinado contexto, un problema y una solución. Para el diseño de software, el contexto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. Un conjunto de requerimientos, incluidas limitaciones y restricciones, actúan como sistema de fuerzas que influyen en la manera en la que puede interpretarse el problema en este contexto y en cómo podría aplicarse con eficacia la solución (Pressman & Maxim, 2021).

### Patrones GoF

Para el desarrollo de aplicaciones web, se identificaron 8 patrones de diseño Gof de los 23 en los procesos de desarrollo software (Guerrero et al., 2013). A continuación, se muestra el que fue usado en el presente trabajo:

#### Singleton

Este patrón garantiza que una clase tenga sólo una instancia y proporciona un punto global de acceso a ella desde cualquier parte del programa. Esto se logra con un constructor privado y proporcionando un método estático para obtener la única instancia de la clase. Es útil para controlar estrictamente la creación y acceso a una única instancia de una clase en todo el programa. (Chávez, 2023).

```
class Rectoria(SingletonModel):
    nombre = models.CharField(max_length=254, verbose_name="Nombre")
    descripcion = RichTextField(blank=True, verbose_name="Descripción")
    fecha_registro = models.DateField(default=datetime.now)

    def __str__(self):
        return self.nombre

class Meta:
    verbose_name = 'Rectoría'
    verbose_name_plural = 'Rectorías'
    db_table = 'areas_app_rectoria'
```

*Figura 7. Evidencia del patrón Singleton.*

La clase rectoría hereda del SingletonModel, lo que significa que solo se puede crear una instancia de la clase rectoría. Esto se logra mediante la creación de un constructor privado que solo puede llamar desde dentro de la clase y un método público que devuelve la única instancia de la clase. En este caso, la única instancia de la clase rectoría se crea cuando se llama al método str.

## Patrones GRASP

GRASP es un acrónimo de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Una vez conocido esto se puede afirmar que los patrones GRASP describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones (Volodina, 1999). A continuación, se muestra los que fueron usados en el presente trabajo:

### Experto

Este patrón recomienda que las responsabilidades se deben asignar a los objetos que poseen la información para poder ejecutarlas, con el objetivo de preservar el encapsulamiento para de esta forma obtener una alta cohesión y un bajo acoplamiento lo cual es ideal en cualquier sistema de información (Menéndez Figueredo, 2022).

```
class UserManager(BaseUserManager):
    use_in_migrations = True

    def _create_user(self, username, email, password, is_staff, is_superuser, **extra_fields):
        if not email:
            raise ValueError('El usuario debe tener un correo electrónico!')
        email = self.normalize_email(email)

        user = self.model(
            username=username,
            email=email,
            is_staff=is_staff,
            is_superuser=is_superuser,
            **extra_fields,
        )

        user.set_password(password)
        user.save(using=self.db)
        return user

    def create_user(self, username, email, password=None, **extra_fields):
        return self._create_user(username, email, password, False, False, **extra_fields)

    def create_superuser(self, username, email, password=None, **extra_fields):
        return self._create_user(username, email, password, True, True, **extra_fields)
```

**Figura 8.** Evidencia del patrón experto.

El patrón experto se manifiesta en la clase UserManager. Esta clase es responsable de crear y administrar usuarios. Al hacerlo, se asegura de que la creación de usuarios se realice de manera eficiente y sin errores. Esto se logra mediante la creación de un método privado create\_user que se

encarga de crear un usuario con los parámetros necesarios. De esta manera, la clase UserManager se convierte en un experto en la creación y administración de usuarios.

### **Bajo Acoplamiento**

Este patrón minimiza las dependencias entre las clases, asignando responsabilidades de manera que las clases sean independientes entre sí. En el desarrollo web con Django, este patrón se puede lograr mediante la separación clara de responsabilidades en el modelo, la vista y las plantillas (Castro Quiñones, 2021). Se manifiesta en la clase UserManager. Esta clase se encarga de crear y administrar usuarios, pero no tiene conocimiento directo de la implementación de la clase User. En su lugar, utiliza los métodos públicos de la clase User para crear y administrar usuarios. Esto significa que la clase UserManager está débilmente acoplada a la clase User.

### **Alta cohesión**

Este patrón busca asignar responsabilidades a las clases de manera que sean coherentes con su naturaleza y finalidad. En el desarrollo web con Django, esto se puede lograr siguiendo las convenciones y buenas prácticas del framework, como utilizar los modelos para representar los datos y la lógica de negocio, las vistas para manejar las solicitudes del usuario y las plantillas para la presentación de la información (Sánchez, 2022). Se manifiesta en la clase User. Esta clase es responsable de definir los atributos y métodos necesarios para representar a un usuario. Todos los atributos y métodos de la clase User están relacionados con la representación de un usuario. Esto significa que la clase User tiene una alta cohesión.

Ambos patrones buscan asignar responsabilidades de manera que las clases sean independientes entre sí y coherentes con su naturaleza y finalidad.

La clase UserManager es un experto en la creación y administración de usuarios, pero está débilmente acoplada a la clase User. La clase User tiene alta cohesión y es responsable de definir los atributos y métodos necesarios para representar a un usuario. En conjunto, estos patrones aseguran que el código sea fácil de entender, mantener y extender.

## **2.8 Estándares de codificación**

La estandarización del código es un aspecto fundamental en el desarrollo de software permitiendo establecer reglas y pautas comunes para la escritura y estructura del código fuente. Esta práctica se

basa en la creación de estándares y convenciones que facilitan la comprensión, mantenibilidad y colaboración en los proyectos de programación. En este epígrafe se mencionarán los estándares utilizados en la implementación del software (Rossum et al., 2013).

**Declaraciones:** Se realiza una declaración por línea.

```
def has_delete_permission(self, request, obj=None):
    if obj != None and request.POST.get('action') == 'delete_selected':
        messages.add_message(request, messages.ERROR, (
            "Realmente desea hacer esta accion?"
        ))
    return True
```

*Figura 9. Ejemplo de declaración por línea.*

**Inicialización:** Inicializar las variables locales donde se declaran.

```
class PublicacionInline(admin.TabularInline):
    model = Publicacion
    extra = 0
    exclude = ['fecha_registro',]
```

*Figura 10. Ejemplo de inicialización de variables.*

**Métodos:** Los métodos se separan con una línea en blanco.

```
def eliminacion_logica_usuarios(self, request, queryset):
    for user in queryset:
        user.is_active = False
        user.save()

def activacion_logica_usuarios(self, request, queryset):
    for user in queryset:
        user.is_active = True
        user.save()

def has_add_permission(self, request):
    return True

def has_change_permission(self, request, obj=None):
    return True
```

*Figura 11. Ejemplo de separación de línea entre métodos.*

**Importaciones:** Se deben realizar en líneas separadas.

```
from apps.indicadores_app.models import Publicacion
from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from .forms import FormUserAdmin
from .models import User
```

*Figura 12. Ejemplo de importaciones.*

## **Conclusiones del capítulo**

En este capítulo se ponen de manifiesto los elementos y características más importantes de la propuesta de solución, así como los diferentes requisitos funcionales y no funcionales.

-La identificación de los requisitos funcionales y no funcionales garantizó que la solución responda a las necesidades del cliente, sirviendo de guía para el desarrollo de las funcionalidades del sistema de gestión.

-El empleo de la metodología Scrum permitió generar los artefactos necesarios pertenecientes a la disciplina de análisis y diseño, así como proporcionar mayor organización en el desarrollo de la propuesta de solución.

-La arquitectura seleccionada Modelo-Vista-Template permitió definir la estructura que tendrá el software.

-El diseño del Modelo de datos ayudó a definir y a estructurar los datos en el contexto en el proceso de desarrollo de software.

-Los patrones de diseño y estándares de codificación definidos permitieron sentar las normas a seguir durante todo el ciclo de la implementación.

## CAPÍTULO 3 VALIDACIÓN DEL SISTEMA DE GESTIÓN PARA EL REGISTRO DE LECCIONES APRENDIDAS.

En este capítulo se emplearán las mejores prácticas para llevar a cabo el proceso de pruebas efectivas en el sistema. Se define la estrategia de pruebas. Se definen y realizan pruebas funcionales y pruebas no funcionales para validar el funcionamiento del software y se analizan los resultados de las pruebas realizadas que permiten la correcta validación del sistema.

### 3.1 Pruebas de Software

Las pruebas de software consisten en un proceso de evaluación y verificación hechas para aplicaciones con la intención de verificar que cada parte de sus funcionalidades cumplan con el objetivo deseado. Permiten evaluar las soluciones y determinar el nivel de calidad que poseen aportando beneficios como: prevenciones de errores, disminución de costos del desarrollo y mejoras en el rendimiento (Rivera Florez, 2023).

### 3.2 Plan de prueba

La **tabla 8** muestra el plan de prueba que se utiliza para validar el sistema:

<b>Tipo de prueba</b>	<b>Objetivo</b>	<b>Tecnologías y herramientas</b>
Pruebas funcionales	Verificar que todas las funcionalidades del software comporten correctamente	Método caja negra
Pruebas de usabilidad	Evaluar la facilidad de uso y la experiencia del usuario.	Lista de chequeo
Pruebas de portabilidad	Verificar que el software pueda ser fácilmente transferido o adaptado a diferentes entornos o plataformas sin perder su funcionalidad o rendimiento	Lista de chequeo
Pruebas de seguridad	Identificar, analizar y corregir las vulnerabilidades del software	Lista de chequeo
Técnica de ladov	Evaluar y medir la satisfacción del usuario.	Encuesta
Prueba de aceptación	Garantizar que el sistema cumpla con lo establecido por el cliente.	-

**Tabla 8.** Plan de prueba basado en la estrategia de prueba de la UCI. Elaboración propia.

### 3.3 Pruebas funcionales

Las pruebas funcionales de un sistema incluyen pruebas que evalúan las funciones que el sistema debe realizar. Los requisitos funcionales pueden estar descritos en productos de trabajo tales como especificaciones de requisitos de negocio, historias de usuario, casos de usos o especificaciones funcionales (Oslen, 2018). Este tipo de pruebas no solo aplican para validar estructuralmente el desarrollo del código si no el testeo de la funcionalidad del sistema o software desarrollado, el conjunto de estas pruebas es también llamadas pruebas de caja negra ya que cumplen como función el soporte o la validación de las funcionalidades del sistema y la interacción que tiene el cliente con el desarrollo (Quintero López & Páez Acuña, 2023).

La **tabla 9** muestra el diseño de caso de prueba para las funcionalidades registrar lección aprendida. Los diseños de casos de pruebas restantes se pueden observar en el **Anexo 2**:

#### Registrar lección aprendida

Escenario	Proyecto	Área del proceso	Descripción	Antecedentes	Recomendaciones	Respuesta del sistema	Resultado
EC 1. Entrada de datos válidos	V Proyecto de prueba sobre redes	V Gestión del conocimiento	V Durante el proyecto, se experimentaron dificultades debido a la falta de comunicación, lo que llevó a malentendidos, retrasos en la toma	V Los malentendidos y la falta de claridad en las instrucciones y expectativas generaron confusiones y retrasos en la ejecución de las tareas.	V Para futuros proyectos de gestión, se recomienda establecer desde el principio un plan de comunicación claro y efectivo.	El sistema registra la lección aprendida y la coloca en la primera posición del listado	Satisfactorio

			de decisiones y conflictos internos.				
EC 2. Entrada de datos inválidos	V Proyecto de prueba sobre redes	V Gestión del conocimiento	I Introducir un documento.jpg	V Los malentendidos y la falta de claridad en las instrucciones y expectativas generaron confusiones y retrasos en la ejecución de las tareas.	V Para futuros proyectos de gestión, se recomienda establecer desde el principio un plan de comunicación claro y efectivo.	El sistema registra la lección aprendida y la coloca en la primera posición del listado	Insatisfactorio
EC 3. Campos vacíos	NA	NA	V Durante el proyecto, se experimentaron dificultades debido a la falta de comunicación, lo que llevó a malentendidos, retrasos en la toma de	V Los malentendidos y la falta de claridad en las instrucciones y expectativas generaron confusiones y retrasos en la ejecución de las tareas.	V Para futuros proyectos de gestión, se recomienda establecer desde el principio un plan de comunicación claro y efectivo.	El sistema no registra la lección aprendida y devuelve los siguientes mensajes: "Por favor corrija el siguiente error" "Este campo es obligatorio"	Satisfactorio

			decisiones y conflictos internos.				
--	--	--	-----------------------------------	--	--	--	--

**Tabla 9.** *Diseño de Caso de prueba. Registrar lección aprendida. Elaboración propia.*

### **Resultado de la prueba**

Se encontraron varias No Conformidades (NC) al finalizar la realización de los diseños de casos de pruebas:

En el caso de prueba 7 Registrar vicerrectoría, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar una vicerrectoría, permite realizar el registro si el usuario rellena la información con algún campo inválido.

En el caso de prueba 8 Editar vicerrectoría, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar una vicerrectoría, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido.

En el caso de prueba 12 Registrar dirección, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar una dirección, permite realizar el registro si el usuario rellena la información con algún campo inválido.

En el caso de prueba 13 Editar dirección, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar una dirección, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido.

En el caso de prueba 17 registrar centro, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar un centro, permite realizar el registro si el usuario rellena la información con algún campo inválido.

En el caso de prueba 18 Editar centro, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar un centro, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido.

En el caso de prueba 22 Registrar lección aprendida, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar una lección, permite realizar el registro si el usuario rellena la información con algún campo inválido.

En el caso de prueba 23 Editar lección, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar una lección, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido.

En el caso de prueba 27 Registrar facultad, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar una facultad, permite realizar el registro si el usuario rellena la información con algún campo inválido.

En el caso de prueba 28 Editar facultad, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar una facultad, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido.

En el caso de prueba 32 Registrar proyecto, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar un proyecto, permite realizar el registro si el usuario rellena la información con algún campo inválido.

En el caso de prueba 33 Editar proyecto, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar un proyecto, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido

En el caso de prueba 37 Registrar certificación, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar una certificación, permite realizar el registro si el usuario rellena la información con algún campo inválido.

En el caso de prueba 38 Editar certificación, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar una certificación, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido.

En el caso de prueba 42 Registrar curso de formación, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar un curso de formación, permite realizar el registro si el usuario rellena la información con algún campo inválido.

En el caso de prueba 43 Editar curso de formación, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar un curso de formación, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido.

En el caso de prueba 47 Registrar publicación, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar una publicación, permite realizar el registro si el usuario rellena la

información con algún campo inválido.

En el caso de prueba 48 Editar publicación, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar una publicación, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido.

En el caso de prueba 52 Registrar grupo, en el escenario 2 se encontró una NC ya que el sistema a la hora de registrar un grupo, permite realizar el registro si el usuario rellena la información con algún campo inválido.

En el caso de prueba 53 Editar grupo, en el escenario 2 se encontró una NC ya que el sistema a la hora de editar un grupo, permite realizar la acción de editar si el usuario rellena la información con algún campo inválido.

Después de analizar los resultados obtenidos en las pruebas se obtuvo un total de 20 NC en el sistema. Se procedió a corregir las deficiencias encontradas. Para ello se llevaron a cabo un conjunto de validaciones en esos campos que permitió su corrección en la segunda iteración de pruebas

### 3.4 Pruebas no funcionales

Las pruebas no funcionales de un sistema evalúan las características del software como la usabilidad, portabilidad, eficiencia del desempeño y seguridad (Oslen, 2018).

**Usabilidad:** Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones (*Usabilidad*, 2022). Para garantizar la seguridad del sistema desarrollado se realizaron pruebas de usabilidad basada en la lista de chequeo (**Ver Anexo 2**), herramienta utilizada en la UCI en los laboratorios de calidad de software para agilizar los procesos de prueba, teniendo en cuenta que constituyen la guía básica y única para el probador en la revisión de los artefactos.

#### Resultados de la prueba

Categoría de vulnerabilidades	Cantidad de errores
Operabilidad	3
Cognoscibilidad	1

Recognoscibilidad	2
Estética de interfaz de usuario	1

**Tabla 10.** Resultados de las pruebas de usabilidad. Elaboración propia.

Después de analizar los resultados obtenidos en las pruebas de usabilidad en la primera iteración se obtuvo un total de 7 NC en el sistema.

#### *Operabilidad*

- No se muestra un cambio visible cuando el cursor está apuntando a un elemento en acción, excluyendo los cambios de cursor.
- No se definió como máximo entre 80 y 100 caracteres para la longitud de línea de los bloques de texto.
- No se implementaron validaciones antes de que el usuario envíe información.

#### *Cognoscibilidad*

- No se utilizaron aceleradores o accesos rápidos a operaciones frecuentes.
- No se estableció quienes serán los responsables del sistema.
- No se brinda información de contacto del equipo de soporte.

#### *Protección frente a errores de usuarios*

- No se personalizaron las páginas de error.

Se procedió a corregir las deficiencias encontradas. Para ello se llevaron a cabo un conjunto de acciones para corregir esos errores en una segunda iteración.

**Portabilidad:** Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro (*Usabilidad, 2022*). Para garantizar la seguridad del sistema desarrollado se realizaron pruebas de portabilidad basada en la lista de chequeo (**Ver Anexo 2**), herramienta utilizada en la UCI en los laboratorios de calidad de software para agilizar los procesos de prueba, teniendo en cuenta que constituyen la guía básica y única para el probador en la revisión de los artefactos.

#### **Resultados de la prueba**

<b>Categoría de vulnerabilidades</b>	<b>Observaciones</b>
Instabilidad	1- Se debe realizar una prueba de la aplicación en varias versiones de navegador. 2- Se debe realizar la prueba de la aplicación en varios sistemas operativos. 3- Se debe realizar la prueba de la aplicación en diferentes dispositivos (Apple, iOS, Google y Android)

**Tabla 11.** Resultados de las pruebas de portabilidad. Elaboración propia.

Después de analizar los resultados obtenidos en las pruebas de portabilidad se obtuvo un total de 3 observaciones para la categoría inestabilidad. Se procedió a realizar las acciones pertinentes para finalizar.

**Seguridad:** Las pruebas de seguridad se podrían definir como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en aplicaciones web, buscando disminuir el impacto de ataques a ellas y pérdida de información importante (Diaz, 2014). El objetivo de las Pruebas de Seguridad es medir la confidencialidad, integridad y disponibilidad de los datos, partiendo a identificar amenazas y riesgos desde el uso o interface de usuario final. Durante las pruebas de seguridad, el responsable de la prueba desempeña el papel de un individuo que desea entrar en el sistema. Debe intentar conseguir las claves de acceso por cualquier medio, debe bloquear el sistema, negando así el servicio a otras personas. Una vez ejecutadas las Pruebas de Seguridad es posible medir y cuantificar los riesgos a los cuales se ven expuestos los aplicativos tanto en la infraestructura interna como externa, así como la identificación de vulnerabilidades (Pressman & Maxim, 2021). Para garantizar la seguridad del sistema desarrollado se realizaron pruebas de portabilidad basada en la lista de chequeo, herramienta utilizada en la UCI en los laboratorios de calidad de software para agilizar los procesos de prueba, teniendo en cuenta que constituyen la guía básica y única para el probador en la revisión de los artefactos (**Ver Anexo 2**).

### Resultados de la prueba

Para realizar las pruebas de seguridad al sistema, se utiliza la lista de chequeo con el objetivo de validar la seguridad de la solución propuesta, la cual arrojó los siguientes resultados:

<b>Categoría de vulnerabilidades</b>	<b>Cantidad de errores</b>
--------------------------------------	----------------------------

Prueba de fuerza bruta	2
Pruebas de gestión del cache de navegación y de salida de sesión	2
Pruebas de escalación de privilegios	1

**Tabla 12.** Resultados de las pruebas de seguridad. Elaboración propia.

Después de analizar los resultados obtenidos en las pruebas de seguridad se obtuvo un total de 5 vulnerabilidades.

-Las contraseñas no están validadas para que sean alfanuméricas y que incluyan caracteres especiales y mayúsculas.

-No existen un formulario CAPCHAT o reCAPCHAT para los formularios de registro.

-El sistema cierra la sesión de un usuario automáticamente cuando este ha estado inactivo durante un cierto lapso de tiempo.

-Al cerrar sesión si presionas el botón "Atrás" del navegador es posible acceder nuevamente al sistema.

-No es posible activar, desactivar o bloquear cuentas de usuario ante requisitos del negocio o posibles ataques.

Para ello se llevaron a cabo un conjunto de acciones en la segunda iteración de pruebas que permitieron reforzar la seguridad del sistema.

### **3.5 Técnica para validar la satisfacción del usuario**

La técnica de ladov constituye una vía indirecta para el estudio de la satisfacción de los usuarios, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov (Rodríguez & Maura, 2002).

ladov ha demostrado su efectividad en diversas investigaciones fundamentalmente pedagógicas y académicas, orientadas a evaluar el nivel de satisfacción de las personas con respecto a propuestas

metodológicas, planes y programa de estudio, modelos, estrategias, sistemas, procesos y procedimientos (Lardoeyt Ferrer & Herrera Masó, 2023).

Para la aplicación de esta técnica se diseñó un cuestionario (**Ver Anexo 3**), compuesto por tres preguntas cerradas y tres abiertas. La relación entre las preguntas cerradas se establece en el cuadro lógico de ladov. En la **tabla 13** se muestra el cuadro lógico de ladov indicando la posición de cada persona en la escala de satisfacción modificado para la presente investigación.

Luego de conocer el resultado final del software, refleje en qué medida le gusta la solución.	¿Considera usted que la propuesta de solución satisfaga las necesidades del cliente?								
	No			No sé			Sí		
	¿Utilizaría usted en su proyecto las notas del conocimiento para la recopilación de lecciones aprendidas?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

**Tabla 13.** Cuadro lógico de ladov. *Elaboración propia.*

Los datos reflejados en el cuadro lógico forman parte del resultado de la aplicación de la encuesta realizada a 15 profesionales con experiencia en el desarrollo de software en la UCI. Los participantes pertenecen a los centros de desarrollo de la universidad, vinculados a la actividad productiva con más de 5 años de experiencias y han ocupado los roles de líder de proyectos, analistas, programadores y planificadores, en todos los casos cuentan con experiencia en más de un rol.

El número resultante de la interrelación de las tres preguntas cerradas indica la posición de cada cual en la escala de satisfacción siguiente:

- 1) Clara satisfacción
- 2) Más satisfecho que insatisfecho
- 3) No definida
- 4) Más insatisfecho que satisfecho
- 5) Clara insatisfacción
- 6) Contradictoria

El resultado de la satisfacción individual lo muestra la **tabla 14**

Nivel de satisfacción	Cantidad	Escala
Clara satisfacción	12	A
Más satisfecho que insatisfecho	2	B
No definida	0	C
Más insatisfecho que satisfecho	0	D
Clara insatisfacción	0	E
Contradictoria	1	C

**Tabla 14.** Nivel de satisfacción de los usuarios. *Elaboración propia*

El nivel de satisfacción se necesita para calcular el Índice de Satisfacción Grupal (ISG) mediante la fórmula:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N}$$

**Fórmula 1.** Índice de satisfacción Grupal

Donde N representa el número total de sujetos del grupo.

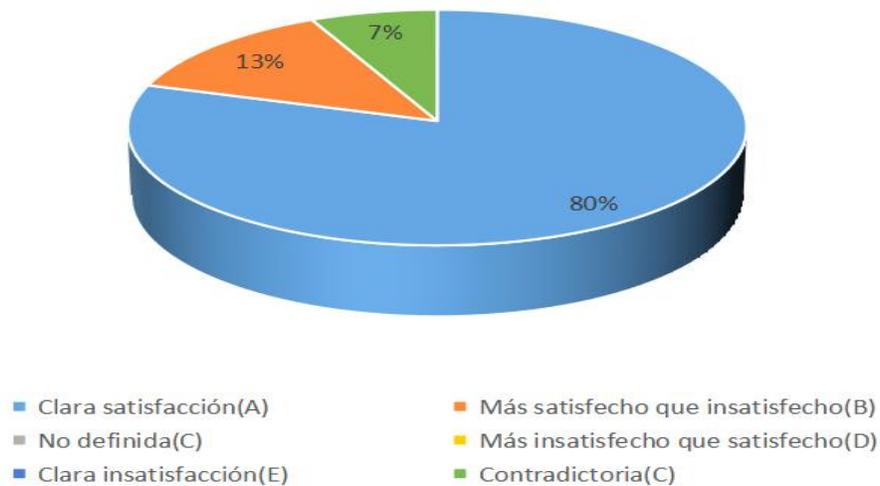
Para obtener el ISG se procesan los criterios de las personas de acuerdo a los niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Escala	Niveles de satisfacción
-1 hasta -0.5	Insatisfacción

-0.49 hasta +0.49	Contradictorio
+0.5 hasta +1	Satisfacción

**Tabla 15.** Escala numérica para niveles de satisfacción. Elaboración propia.

La Figura muestra los resultados de la aplicación de la técnica de ladov, donde el ISG tuvo un valor de 0.87, lo que indica satisfacción de los usuarios con el sistema desarrollado



**Figura 13.** Niveles de satisfacción de los usuarios

En la aplicación de la técnica de ladov fueron incluidas 3 preguntas abiertas que permiten ahondar y analizar las causas que originan los niveles de satisfacción. Las respuestas tributan a sugerencias y recomendaciones a tener en cuenta para mejorar la propuesta

En el caso de la pregunta 4 (**Ver anexo 3**), fueron resaltados los elementos positivos de la propuesta:

- Contribuye en la toma de decisiones para afrontar nuevos proyectos.
- Permite que el conocimiento se expanda.

En el caso de la pregunta 5 y 6 (**Ver anexo 3**), representan recomendaciones para mejorar la propuesta presentada:

- Mejorar la interfaz de usuario.
- Recopilar en el sistema todas las lecciones aprendidas detectadas para contribuir a la mejora de procesos.

### **3.6 Prueba de Aceptación.**

Cuando se construye un software a la medida para un cliente, se realiza una serie de pruebas de aceptación a fin de permitir al cliente validar todos los requisitos; estas son realizadas por el usuario final en lugar de los ingenieros de software (Pressman & Maxim, 2021).

Para la aceptación final del producto y validación de la solución, se le presentó al cliente una versión final operativa del sistema, el cliente comprobó que los requisitos acordados fueron implementados correctamente y que el resultado esperado es válido y acorde a los parámetros de pruebas utilizados. Como parte del proceso se elaboró el acta de aceptación, la cual se muestra en el **Anexo 4**.

Por tanto, se demuestra que el desarrollo del sistema para gestionar el registro de lecciones aprendidas, contribuyen a la mejora de los procesos de la UCI.

### **Conclusiones del capítulo**

Con la realización de este capítulo se concluye que:

- Las pruebas funcionales y no funcionales mostraron el correcto funcionamiento de la aplicación al validar la completitud de los requisitos y determinar la aceptación del cliente.
- Con el uso de la técnica de ladov se demostró la aceptación de los usuarios en su mayoría, obteniendo un valor de 80% en los niveles de satisfacción de los usuarios.
- A través de la prueba de aceptación se determinó que el cliente está totalmente de acuerdo con los resultados obtenidos.

## **CONCLUSIONES**

Una vez culminada la investigación se pudo llegar a las siguientes conclusiones:

-Se desarrolló un sistema que permite gestionar todas las lecciones aprendidas de forma unificada y centralizada, siendo esta propuesta de solución un aporte significativo para la universidad para la toma de decisiones futuras.

-Las funcionalidades propuestas por el cliente fueron implementadas y demostraron su utilidad y calidad.

-La arquitectura y tecnologías utilizadas fueron claves en el éxito del resultado final.

-La propuesta de solución influyó positivamente en la mejora de procesos del sistema de gestión de la calidad como parte de la gestión del conocimiento.

-Se obtuvo un acta de aceptación por parte del cliente que demuestra y confirma el éxito de la solución obtenida.

## **RECOMENDACIONES**

A partir de los resultados obtenidos se recomienda:

-Agregar a la solución propuesta, aprendizaje automático para mejorar la eficiencia y la toma de decisiones.



## REFERENCIAS

Agile Alliance. (2023). <https://www.agilealliance.org/>

Agudelo Ceballos, E. J., Valencia Arias, A., Agudelo Ceballos, E. J., & Valencia Arias, A. (2018). La gestión del conocimiento, una política organizacional para la empresa de hoy. *Ingeniare. Revista chilena de ingeniería*, 26(4), 673-684. <https://doi.org/10.4067/S0718-33052018000400673>

Aguilera-Luque, A. M. (2020). LA GESTIÓN DEL CONOCIMIENTO ORGANIZACIONAL. [https://www.academia.edu/30819149/LA\\_GESTI%C3%93N\\_DEL\\_CONOCIMIENTO\\_ORGANIZACIONAL](https://www.academia.edu/30819149/LA_GESTI%C3%93N_DEL_CONOCIMIENTO_ORGANIZACIONAL)

Alvarado Rodríguez, L. E. (2022a). Gestión del conocimiento y herramientas de tecnologías de la información de una ONG de alumnos asociados de Ingeniería, de la ciudad de Trujillo 2021. Universidad Privada Antenor Orrego. <https://repositorio.upao.edu.pe/handle/20.500.12759/8687>

Alvarado Rodríguez, L. E. (2022b). Gestión del conocimiento y herramientas de tecnologías de la información de una ONG de alumnos asociados de Ingeniería, de la ciudad de Trujillo 2021. Universidad Privada Antenor Orrego. <https://repositorio.upao.edu.pe/handle/20.500.12759/8687>

Álvarez Medel, A., & Cleger Despaigne, E. (2017). Desarrollo del módulo Administración y Configuración del Sistema para la Consulta Integrada de Información Estadística. <https://repositorio.uci.cu/jspui/handle/ident/8955>

Asana. (2023). Cómo usar Asana, tutoriales de Asana, consejos de implementación y casos de uso en equipos | Guía del producto • Guía del producto de Asana. Guía del producto de Asana. <https://asana.com/es/guide>

Atlassian. (2023). Atlassian Support Offerings | Atlassian Support | Atlassian Documentation. <https://confluence.atlassian.com/support/atlassian-support-offerings-193299636.html>

Avila, J. G. (2022). Módulo y Fototeca de la biblioteca virtual del centro de información para la prensa.

Caceres Pieschacon, I., & Romero Romero, J. C. (2022). Propuesta de modelo de gestión del conocimiento para la creación de nuevos productos a través de metodologías ágiles. <https://bdigital.uexternado.edu.co/entities/publication/cd229993-8959-459f-be48-6ef928e7480f>

Casañola, Y. T., Estrada, A. F., Rodríguez, Y. B., Rodríguez, A. M. G., Tarrero, G. C., Caballero, P. J. M., & Morales, L. L. (2021). Modelo de gestión del conocimiento en función de la calidad de los procesos. <https://repositorio.altecasociacion.org/handle/20.500.13048/560>

Castro Quiñones, E. A. (2021). Sistema de gestión de archivos para la Organización Nacional de Bufetes Colectivos [bachelorThesis, Universidad de la Ciencias Informática. Facultad 1]. <https://repositorio.uci.cu/jspui/handle/123456789/10538>

Ceballos, E. J. A., & Arias, A. V. (2018). La gestión del conocimiento, una política organizacional para la empresa de hoy. 26, 684.

Chávez, J. D. (2023). (PDF) Clase pg\_connect y PHP para acceder a PostgreSQL. [https://www.researchgate.net/publication/373338908\\_Clase\\_pg\\_connect\\_y\\_PHP\\_para\\_acceder\\_a\\_PostgreSQL](https://www.researchgate.net/publication/373338908_Clase_pg_connect_y_PHP_para_acceder_a_PostgreSQL)

Contreras, J. L. C. (2022). «Propuesta de una metodología para la gestión de lecciones aprendidas p» by Jessica Lorena Calderón Contreras. [https://ciencia.lasalle.edu.co/maest\\_administracion/517/](https://ciencia.lasalle.edu.co/maest_administracion/517/)

CSS | MDN. (2023, marzo 13). <https://developer.mozilla.org/es/docs/Web/CSS>

Departamento de Gestión e Informatización ONN. (2021). Modelo de la Calidad para el Desarrollo de Aplicaciones Informáticas (MCDAI)- Parte 2 Requisitos.: Vol. parte 2 (1.ª ed.). [www.ncnorma.cu](http://www.ncnorma.cu)

Díaz, S. M. D. (2014). Pruebas de seguridad en aplicaciones web como imperativo en la calidad de desarrollo del software. <https://repository.unab.edu.co/handle/20.500.12749/12308>

Django. (2023). Django Project. <https://www.djangoproject.com/>

Ennis, X. P. (2022). Sistema de gestión de contratos para el centro internacional de la Habana basado en la localización cubana de Odo.

EXCRIBA | Universidad de las Ciencias Informáticas. (2022). <https://www.uci.cu/investigacion-y-desarrollo/productos/xabal/excriba>

Febles Pérez, D., Trujillo Casañola, Y., Mendosa Garnache, A., Febles Pérez, D., Trujillo Casañola, Y., & Mendosa Garnache, A. (2022). Oportunidades de mejora al proceso de aseguramiento de la calidad del proceso y el producto. *Revista Cubana de Ciencias Informáticas*, 16(1), 46-61.

Forte, T. (2020, octubre 26). Knowledge Building Blocks: The New Meaning of Notes. Forte Labs. <https://fortelabs.com/blog/knowledge-building-blocks-the-new-meaning-of-notes/>

García Rodríguez, A. M. (2018). Modelo de recomendación de escenarios al iniciar la mejora de procesos de software [doctoralThesis, Universidad de las Ciencias Informáticas. Facultad-3.]. <https://repositorio.uci.cu/jspui/handle/123456789/7931>

GitLab. (2023). Why GitLab? <https://about.gitlab.com/why-gitlab>

Gómez, C. M. B. (2020). Contribuir a una mejor gestión de proyectos en organizaciones que no posean un.

Guerrero, C. A., Suárez, J. M., & Gutiérrez, L. E. (2013). Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. [https://www.scielo.cl/scielo.php?script=sci\\_arttext&pid=S0718-07642013000300012](https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642013000300012)

HTML: Lenguaje de etiquetas de hipertexto | MDN. (2023, marzo 13). <https://developer.mozilla.org/es/docs/Web/HTML>

Hvorecký, J., & Korenova, L. (2023). Calidad de los libros desde la perspectiva de la gestión del conocimiento.

Jaime, A., Blanco, J. M., Domínguez, C., Usandizaga, I., & Arruabarrena, R. (2022). RUA: Estimulando la creación y difusión de lecciones aprendidas en el contexto del aprendizaje basado en proyectos. <https://rua.ua.es/dspace/handle/10045/128581>

JavaScript | MDN. (2023, febrero 9). <https://developer.mozilla.org/es/docs/Web/JavaScript>

- Lardoeyt Ferrer, R., & Herrera Masó, J. R. (2023). Índice de satisfacción de profesionales con la formación científica metodológica en una institución académica de Cuba. <https://doi.org/10.6018/edumed.559701>
- Leiva Morantes, H. M., & Álvarez Villamil, H. A. (2020). Modelo de gestión del conocimiento para la Universidad de San Buenaventura, Bogotá [Master Thesis, Universidad EAN]. <https://repository.universidadean.edu.co/handle/10882/10125>
- Luna, E., & Bu, L. R. (2015). Cómo documentar lecciones aprendidas—Abierto al Público. <https://blogs.iadb.org/conocimiento-abierto/es/como-documentar-lecciones-aprendidas/>
- Marzal, A., & Gracia, I. (2020). Introducción a la programación con Python. Universitat Jaume I. <https://repositorio.uci.cu/jspui/handle/123456789/10208>
- Massengo Nekaka, P. (2020). Sistema de Gestión de Estudiantes para el Sector de Apoyo Estudiantil de Angola en Cuba [bachelorThesis, Universidad de las Ciencias Informáticas. Facultad 1]. <https://repositorio.uci.cu/jspui/handle/123456789/10357>
- Mattia, C. D., Schneider, A., Scoz, D., & Correa, V. (2021, julio). LECCIONES APRENDIDAS DE LA IMPLEMENTACIÓN DE QUINCE OFICINAS DE GESTIÓN DE PROYECTOS EN EL ESTADO DE SANTA CATARINA – BRASIL. 19.
- Menéndez Figueredo, M. de la C. (2022). Módulo de Gestión de Patrones de Diseño para EGPAT con soporte para interoperabilidad [bachelorThesis, Universidad de las Ciencias Informáticas. Facultad 4.]. <https://repositorio.uci.cu/jspui/handle/123456789/10617>
- Mera López, R. J. (2021). La gestión del conocimiento en el Programa de Fortalecimiento de Emprendimientos e Iniciativas Productivas Locales del Gobierno Provincial de Imbabura [masterThesis, Quito, EC: Universidad Andina Simón Bolívar, Sede Ecuador]. <http://repositorio.uasb.edu.ec/handle/10644/7992>
- Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiraes Revista Multidisciplinaria de investigación*, 2(17), Article 17. <https://doi.org/10.31876/re.v2i17.269>
- Oduardo, D. D., & Carmona, Y. P. (2021). Gestión de lecciones aprendidas en proyectos de la UCI: Fortalezas y oportunidades de mejora. *Revista Cubana de Transformación Digital*, 2(4), Article 4. <https://doi.org/10.5281/zenodo.5810578>
- Oquendo Arencibia, Y. (2018). Herramienta eXcriba Migration Tool versión 2.0 para el sistema gestor de documentos administrativos XABAL EXCRIBA 3.1 [bachelorThesis, Universidad de las Ciencias Informáticas . Facultad 2]. <https://repositorio.uci.cu/jspui/handle/123456789/10201>
- Oslen, K. (2018). Probador certificado de ISTQB.
- Pérez Rodríguez, A. (2022). Sistema de gestión de información web para los almacenes de suministros médicos distribuidos por EMSUME [bachelorThesis, Universidad de las Ciencias Informáticas. Facultad de Ciencias y Tecnologías Computacionales]. <https://repositorio.uci.cu/jspui/handle/123456789/10652>

Pérez, V. A., & Urbáes, M. F. (2016). Modelos teóricos de gestión del conocimiento: Descriptores, conceptualizaciones y enfoques. Universidad Nacional Autónoma de México. <https://www.redalyc.org/pdf/4576/457646537004.pdf>

Pérez, V. A., & Urbáez, M. F. (2016). Modelos teóricos de gestión del conocimiento: Descriptores, conceptualizaciones y enfoques. *Entreciencias: Diálogos en la Sociedad del Conocimiento*, 4(10), 201-227.

Pérez-Fuillerat, N., Solano-Ruiz, M. C., & Amezcua, M. (2019). Conocimiento tácito: Características en la práctica enfermera. *Gaceta Sanitaria*, 33(2), 191-196. <https://doi.org/10.1016/j.gaceta.2017.11.002>

Piva, R., Cattaneo, C., Pessoni, L., & Bertone, B. (2021). El conocimiento tácito en arquitectura, su reconocimiento y valoración en el ámbito de la construcción. *Arquitecto*, 18, Article 18. <https://doi.org/10.30972/arq.0185670>

PMBOK GUIDE (7ma ed.). (2021). [https://ibimone.com/PMBOK%207th%20Edition%20\(iBIMOne.com\).pdf](https://ibimone.com/PMBOK%207th%20Edition%20(iBIMOne.com).pdf)

PostgreSQL. (2023). PostgreSQL: About. <https://www.postgresql.org/about/>

Pressman, R. S., & Maxim, B. R. (2021). Ingebook—INGENIERÍA DE SOFTWARE 9ED - Un enfoque práctico. [https://www.ingebook.com/ib/NPcd/IB\\_BooksVis?cod\\_primaria=1000187&codigo\\_libro=11988](https://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=11988)

Prezi. (2023). Modelo—Vista—Template. Prezi.Com. <https://prezi.com/nd9ydrb01kqv/modelo-vista-template/>

Pulgarin Orozco, A. J. (2022). Modelo de gestión del conocimiento para la preservación de la memoria estratégica. Caso de estudio: Telemedellín. <https://repositorio.itm.edu.co/handle/20.500.12622/5708>

Quintero López, Y., & Páez Acuña, R. A. (2023). Diseñar de la metodología de pruebas funcionales para los desarrollos en Fidelity Marketing SAS [Master Thesis, Maestría en Gerencia de Proyectos]. <https://repository.universidadean.edu.co/handle/10882/12468>

Quispe Alcarraz, E. (2021). Gestión del conocimiento e innovación organizacional en los docentes de la Universidad Nacional Federico Villarreal. <http://repositorioslatinoamericanos.uchile.cl/handle/2250/4712959>

Repository Universidad Tecnológica de Pereira: Search. (2021). <https://repositorio.utp.edu.co/search?spc.page=1&query=congreso%20nacional%20de%20gerencia%20del%20deporte&spc.sf=score&spc.sd=DESC>

Rivera Florez, J. D. (2023). Diseño y cobertura de pruebas de software para el backend. <https://bibliotecadigital.udea.edu.co/handle/10495/34365>

Rodríguez, A. L., & Maura, V. G. (2002). La técnica de ladov. <https://www.efdeportes.com/efd47/iadov.htm>

Rodriguez, Z. G., Batista, D. O., & Jiménez, A. D. (2023). Auditorías del conocimiento: Un reto para la gestión del conocimiento en las universidades.

Rodríguez-Ponce, E. R., Pedraja-Rejas, L. M., Araneda-Guirriman, C. A., Muñoz-Fritis, C. P., Rodríguez-Ponce, E. R., Pedraja-Rejas, L. M., Araneda-Guirriman, C. A., & Muñoz-Fritis, C. P. (2022). Relaciones entre las fases del proceso de gestión del conocimiento en unidades académicas. *Información tecnológica*, 33(1), 49-56. <https://doi.org/10.4067/S0718-07642022000100049>

Rossum, G. V., Warsaw, B., & Coghlan, N. (2013). PEP 8 – Style Guide for Python Code | [peps.python.org](https://peps.python.org). <https://peps.python.org/pep-0008/>

Salones, D., & Aular, M. (2022). LA GESTIÓN DEL CONOCIMIENTO PARA EL DESARROLLO ESTRATÉGICO GERENCIAL EN ORGANIZACIONES FINANCIERAS. *Revista Gerens*, 9. <http://200.11.218.106/index.php/rgerens/article/view/1820>

Sánchez, A. J. S. (2022). Aplicación móvil para la gestión de inventario de Activos Fijos Tangibles en la Residencia Estudiantil #2 de la Universidad de las Ciencias Informáticas [Universidad de las Ciencias Informáticas]. [https://repositorio.uci.cu/jspui/bitstream/123456789/10485/1/TD\\_09907\\_22.pdf](https://repositorio.uci.cu/jspui/bitstream/123456789/10485/1/TD_09907_22.pdf)

Santander Universidades. (2020). Metodologías de desarrollo de software: ¿qué son? <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

Santos Correa, J. E., Triana Rodríguez, N., Zuluaga Castiblanco, A. P., & Aristizabal Valencia, D. (2020). Lecciones aprendidas, ¿Qué factores influyen en la incorrecta gestión de las lecciones aprendidas en los proyectos del sector eléctrico? [Bachelor Thesis, Universidad EAN]. <https://repository.universidadean.edu.co/handle/10882/9640>

SARMIENTO, R. G. (2018). Propuesta de un modelo de gestión del conocimiento (GC) para el fortalecimiento de las competencias laborales de los colaboradores de una organización educativa en Mosquera. <https://bdigital.uexternado.edu.co/handle/001/646>

Schwaber, K., & Sutherland, J. (2020). *La Guía de Scrum*.

Socarras Navarro, T., & Acevedo Guevara, A. (2019). Subsistema para la creación de sistemas conversacionales en la suite XEDRO-GESPRO. [bachelorThesis, Universidad de las Ciencias Informáticas. Facultad 3]. <https://repositorio.uci.cu/jspui/handle/123456789/10095>

Socarras Peña, R., Zamora Fernández, L. F., Garcia Rodríguez, A. M., Crespo Urrutia, Y., & Milanés Zamora, Y. (2022). Sistema de Recomendaciones en la Mejora de Procesos de Software aplicando técnicas de Inteligencia Artificial. *Serie Científica de la Universidad de las Ciencias Informáticas*, 15(6), 1-17.

Sommerville, I. (2011). *Ingeniería de Software* (9.<sup>a</sup> ed.). [https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9\\_compressed.pdf](https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf)

Sönke Ahrens. (2017). *How to Take Smart Notes* (2da edición).

The Scrum Guide. (2020). Scrum.Org. <https://www.scrum.org/resources/scrum-guide>

Triana-Velázquez, Y., Febles-Rodríguez, J. P., Mena-Mugica, M., González-Benítez, N., & Garcia-González, M. (2018). Diagnóstico de los sistemas de gestión documental para desarrollar la gestión del conocimiento. *Ingeniería Industrial*, 39(1), 46-55.

Universidad de las Ciencias Informáticas. (2018). <https://www.uci.cu/>

Usabilidad. (2022). <https://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad>

Valencia-Cruzaty, L. E., Ayala-Bolaños, M., & González-Vásquez, Á. E. (2021). Gestión del Conocimiento como estrategia para Asegurar la Calidad de la Educación Superior en Ecuador. *Maestro y Sociedad*, 18(3), Article 3.

Vásquez, Y. (2020). Diseño de un Modelo de Gestión de Conocimiento en INALTEC S.A.S, a partir de buenas prácticas encontradas, que contribuya a su competitividad. <https://bibliotecadigital.udea.edu.co/handle/10495/18616>

Villa, J. L., & Visbal, F. (2020). Diseño de un Sistema de Gestión de Lecciones Aprendidas para la Gerencia de Proyectos. *Proceedings of the 18th LACCEI International Multi-Conference for Engineering, Education, and Technology: Engineering, Integration, And Alliances for A Sustainable Development* "Hemispheric Cooperation for Competitiveness and Prosperity on A Knowledge-Based Economy". The 18th LACCEI International Multi-Conference for Engineering, Education, and Technology: Engineering, Integration, And Alliances for A Sustainable Development "Hemispheric Cooperation for Competitiveness and Prosperity on A Knowledge-Based Economy". <https://doi.org/10.18687/LACCEI2020.1.1.220>

Villasana Arreguín, L. M., Hernández García, P., Ramírez Flores, É., Villasana Arreguín, L. M., Hernández García, P., & Ramírez Flores, É. (2021). La gestión del conocimiento, pasado, presente y futuro. Una revisión de la literatura. *Trascender, contabilidad y gestión*, 6(18), 53-78. <https://doi.org/10.36791/tcg.v0i18.128>

Visual Paradigm. (2023). Visual Paradigm Guides. Visual Paradigm Guides. <https://guides.visual-paradigm.com/>

Visual Studio Code. (2023). Documentation for Visual Studio Code. <https://code.visualstudio.com/docs>

Volodina, L. (1999). (PDF) UML y patrones | Ludmila Volodina—Academia.edu. [https://www.academia.edu/3105204/UML\\_y\\_patrones](https://www.academia.edu/3105204/UML_y_patrones)

Welcome to Python.org. (2023, marzo 8). Python.Org. <https://www.python.org/>

Wiig, K. M. (1993). Knowledge Management Foundations: Thinking about Thinking : How People and Organizations Create, Represent and Use Knowledge / K.M. Wiig. [https://www.researchgate.net/publication/31672277\\_Knowledge\\_Management\\_Foundations\\_Thinking\\_about\\_Thinking\\_How\\_People\\_and\\_Organizations\\_Create\\_Represent\\_and\\_Use\\_Knowledge\\_KM\\_Wiig](https://www.researchgate.net/publication/31672277_Knowledge_Management_Foundations_Thinking_about_Thinking_How_People_and_Organizations_Create_Represent_and_Use_Knowledge_KM_Wiig)

## ANEXOS

### Anexo 1 Entrevista para la obtención de la situación problemática

- 1- ¿Cuál es su nombre y apellidos?
- 2- ¿Cuántos años de experiencia posee en el sector?
- 3- ¿A qué centro de desarrollo pertenece?
- 4- ¿Cuál es el rol que desempeña?
- 5- ¿En su centro de desarrollo se registran las lecciones aprendidas?
- 6- ¿Qué herramientas utilizan para el registro de las lecciones aprendidas?
- 7- ¿Utilizan el conocimiento registrado en antiguas lecciones aprendidas para la mejora de los procesos?
- 8- ¿Pueden hacer uso de las lecciones aprendidas de otros centros de desarrollo o proyectos de la universidad?
- 9- ¿Cree que sea útil la creación de un software capaz de para unificar el registro de las lecciones aprendidas en la universidad?
- 10- ¿Considera que sea importante el registro de las lecciones aprendidas para la mejora de los procesos?

### Anexo 2: Artefactos que evidencian la realización de pruebas funcionales y no funcionales

<https://drive.google.com/drive/folders/1a9UvDwUAZKnSF2kTusXBCiQpjfPErdRx>

### Anexo 3: Encuesta para evaluar la satisfacción de los usuarios

1- Luego de conocer el resultado final del software, refleje en qué medida le gusta la solución.  
Marque con una X donde considere.

\_\_\_ Me gusta mucho      \_\_\_ No me gusta tanto      \_\_\_ Me da lo mismo  
\_\_\_ Me disgusta más de lo que me gusta      \_\_\_ No me gusta nada      \_\_\_ No sé qué decir

2- ¿Considera usted que la propuesta de solución satisfaga las necesidades del cliente?

3- ¿Utilizaría usted en su proyecto las notas del conocimiento para la recopilación de lecciones aprendidas?

4- ¿Qué elementos considera positivos del software?

5- ¿Cuál es la importancia de recopilar más datos para introducir en el dataset y cómo puede ayudar a mejorar los resultados después del entrenamiento?

6- ¿Qué aspectos consideras que podrían mejorarse en nuestra aplicación de software para que sea más fácil de usar y satisfactoria para ti?

#### **Anexo 4 Acta de aceptación.**

En cumplimiento del **Convenio de colaboración** Dirección de Calidad de Software y en función de la ejecución del proyecto: Sistema de gestión para el registro de las lecciones aprendidas en la Universidad de las Ciencias Informáticas, se hace entrega de los productos que se relacionan a continuación:

- El documento final de tesis
- Especificación de Requisitos de Software
- Historias de usuarios
- El sistema implementado

La Parte Cliente, luego de haber revisado los productos de trabajo determina que los productos son aceptados.

---

**Entrega:**

**Recibe**

---

**Nombre y apellidos:** Yeni Carmen Tamayo Morales

**Nombre y apellidos:** Wisleidys Campos Wright

Maribel Kissis Valdés Barrios.

---

**Cargo:**

**Cargo:** subdirectora de la DCSW

---



## Anexo 5 Historias de usuario

Historia de Usuario 1	
Nombre: Autenticar usuario	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Media
Descripción: El sistema permite que el usuario introduzca sus datos para acceder al sitio web.	
Observación: Para poder autenticarse previamente, el administrador del sistema tiene que haber creado el usuario.	

*Historia de Usuario 1. Elaboración propia.*

Historia de Usuario 2	
Nombre: Registrar usuario	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El usuario se registra mediante el nombre de usuario, contraseña y publicaciones de este usuario en revistas científicas.	
Observación: El usuario no se puede registrar solo, sino que lo tiene que registrar el administrador del sistema.	

*Historia de Usuario 2. Elaboración propia.*

Historia de Usuario 3	
Nombre: Editar usuario	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta

Descripción: Se puede editar el nombre, el nombre de usuario, los apellidos, el sexo, el teléfono, el correo, los títulos académicos, el grado científico, el rol, los años de experiencia, las certificaciones académicas, los cursos de formación, el grupo y las publicaciones.

Observación: Quien puede editar toda esta información es el administrador del sistema.

*Historia de Usuario 3. Elaboración propia.*

#### **Historia de Usuario 4**

Nombre: Eliminar usuario

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema permite eliminar los datos del usuario.

Observación: Esta opción solo la puede realizar el administrador del sistema.

*Historia de Usuario 4. Elaboración propia.*

#### **Historia de Usuario 5**

Nombre: Listar usuario

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Media

Descripción:

Observación: Es el administrador del sistema es quien puede acceder a esta información.

*Historia de Usuario 5. Elaboración propia.*

### Historia de Usuario 6

Nombre: Buscar usuario

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema permite filtrar por títulos académicos, categoría docente, grado científico o por rol.

Observación: Es el administrador del sistema quien puede acceder a esta información.

*Historia de Usuario 6. Elaboración propia.*

### Historia de Usuario 7

Nombre: Registrar vicerrectoría

Iteración Asignada: 1

Prioridad en Negocio: Alta

Riesgos de Desarrollo: Alta

Descripción: El sistema permite registrar una vicerrectoría por el nombre y por una descripción.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 7. Elaboración propia.*

### Historia de Usuario 8

Nombre: Editar vicerrectoría

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema permite editar la vicerrectoría mediante el nombre y la descripción.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 8. Elaboración propia.*

### **Historia de Usuario 9**

Nombre: Eliminar vicerrectoría

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Media

Descripción: El sistema permite al usuario eliminar una vicerrectoría.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 9. Elaboración propia.*

### **Historia de Usuario 10**

Nombre: Listar vicerrectoría

Iteración Asignada: 1

Prioridad en Negocio: Baja

Riesgos de Desarrollo: Media

Descripción: El sistema muestra un listado de todas las vicerrectorías registradas en el sistema.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 10. Elaboración propia.*

### **Historia de Usuario 11**

Nombre: Buscar vicerrectoría

Iteración Asignada: 1

Prioridad en Negocio: Media	Riesgos de Desarrollo: Media
Descripción: El sistema permite buscar una vicerrectoría por el nombre.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 11. Elaboración propia.*

<b>Historia de Usuario 12</b>	
Nombre: Registrar dirección	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema permite registra una dirección por el nombre y por una descripción.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 12. Elaboración propia.*

<b>Historia de Usuario 13</b>	
Nombre: Editar dirección	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema permite editar una dirección por el nombre y la descripción.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 13. Elaboración propia.*

### Historia de Usuario 14

Nombre: Eliminar dirección

Iteración Asignada: 1

Prioridad en Negocio: Alta

Riesgos de Desarrollo: Alta

Descripción: El sistema permite al usuario eliminar una dirección.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 14. Elaboración propia.*

### Historia de Usuario 15

Nombre: Listar dirección

Iteración Asignada: 1

Prioridad en Negocio: Alta

Riesgos de Desarrollo: Alta

Descripción: El sistema muestra un listado de todas las direcciones registradas en el sistema.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 15. Elaboración propia.*

### Historia de Usuario 16

Nombre: Buscar dirección

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Media

Descripción: El sistema permite buscar una dirección por el nombre.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 16. Elaboración propia.*

### **Historia de Usuario 17**

Nombre: Registrar centro de desarrollo	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite registrar un centro de desarrollo por el nombre y por una descripción.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 17. Elaboración propia.*

### **Historia de Usuario 18**

Nombre: Editar centro de desarrollo	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite editar un centro de desarrollo por el nombre y la descripción.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 18. Elaboración propia.*

### **Historia de Usuario 19**

Nombre: Eliminar centro de desarrollo	Iteración Asignada: 1
---------------------------------------	-----------------------

Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema permite al usuario eliminar un centro de desarrollo.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 19. Elaboración propia.*

<b>Historia de Usuario 20</b>	
Nombre: Listar centro de desarrollo	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema muestra un listado de todos los centros de desarrollo registrados en el sistema.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 20. Elaboración propia.*

<b>Historia de Usuario 21</b>	
Nombre: Buscar centro de desarrollo	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite buscar un centro de desarrollo por el nombre.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 21. Elaboración propia.*

### Historia de Usuario 22

Nombre: Registrar lección aprendida	Iteración Asignada: 1
-------------------------------------	-----------------------

Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
----------------------------	-----------------------------

Descripción: El sistema permite registrar una lección aprendida por el proyecto al que pertenecen, el área de proceso a la que pertenecen, antecedentes, descripción, recomendaciones, referencias y la fecha de registrado

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 22. Elaboración propia.*

### Historia de Usuario 23

Nombre: Editar lección aprendida	Iteración Asignada: 1
----------------------------------	-----------------------

Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
----------------------------	-----------------------------

Descripción: El sistema permite editar una lección aprendida por el proyecto al que pertenecen, el área de proceso a la que pertenecen, antecedentes, descripción, recomendaciones, referencias y la fecha de editado.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 23. Elaboración propia.*

### Historia de Usuario 24

Nombre: Eliminar lección aprendida	Iteración Asignada: 1
------------------------------------	-----------------------

Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
----------------------------	-----------------------------

Descripción: El sistema permite al usuario eliminar una lección aprendida.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 24. Elaboración propia.*

### **Historia de Usuario 25**

Nombre: Listar lección aprendida

Iteración Asignada: 1

Prioridad en Negocio: Alta

Riesgos de Desarrollo: Alta

Descripción: El sistema muestra un listado de todas las lecciones aprendidas registradas en el sistema.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 25. Elaboración propia.*

### **Historia de Usuario 26**

Nombre: Buscar lección aprendida

Iteración Asignada: 1

Prioridad en Negocio: Alta

Riesgos de Desarrollo: Alta

Descripción: El sistema permite filtrar por los centros de desarrollo, la dirección, la facultad, rectoría y vicerrectoría.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 26. Elaboración propia.*

### **Historia de Usuario 27**

Nombre: Registrar facultad

Iteración Asignada: 1

Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite registrar una facultad por el nombre y por una descripción.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 27. Elaboración propia.*

<b>Historia de Usuario 28</b>	
Nombre: Editar facultad	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Media
Descripción: El sistema permite editar una facultad por el nombre y la descripción.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 28. Elaboración propia.*

<b>Historia de Usuario 29</b>	
Nombre: Eliminar facultad	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite al usuario eliminar una facultad.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 29. Elaboración propia.*

### Historia de Usuario 30

Nombre: Listar facultad

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema muestra un listado de todas las facultades registradas en el sistema.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 30. Elaboración propia.*

### Historia de Usuario 31

Nombre: Buscar facultad

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema permite buscar la facultad por el nombre.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 31. Elaboración propia.*

### Historia de Usuario 32

Nombre: Registrar proyecto

Iteración Asignada: 1

Prioridad en Negocio: Alta

Riesgos de Desarrollo: Alta

Descripción: El sistema permite registrar un proyecto por el nombre y por una descripción.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 32. Elaboración propia.*

### **Historia de Usuario 33**

Nombre: Editar proyecto

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema permite editar un proyecto por el nombre y la descripción.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 33. Elaboración propia.*

### **Historia de Usuario 34**

Nombre: Eliminar proyecto

Iteración Asignada: 1

Prioridad en Negocio: Alta

Riesgos de Desarrollo: Alta

Descripción: El sistema permite al usuario eliminar un proyecto.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 34. Elaboración propia.*

### **Historia de Usuario 35**

Nombre: Listar proyecto

Iteración Asignada: 1

Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema muestra un listado de todos los proyectos registrados en el sistema.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad	

*Historia de Usuario 35. Elaboración propia.*

<b>Historia de Usuario 36</b>	
Nombre: Buscar proyecto	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite buscar el proyecto por el nombre.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad	

*Historia de Usuario 36. Elaboración propia.*

<b>Historia de Usuario 37</b>	
Nombre: Registrar certificaciones académicas	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema permite registrar una certificación académica por el nombre y por una descripción.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad	

*Historia de Usuario 37. Elaboración propia.*

### Historia de Usuario 38

Nombre: Editar certificaciones académicas

Iteración Asignada: 1

Prioridad en Negocio: Alta

Riesgos de Desarrollo: Alta

Descripción: El sistema permite editar una certificación académica por el nombre y por la descripción.

Observación: Solamente lo pueden realizar usuarios con alta prioridad

*Historia de Usuario 38. Elaboración propia.*

### Historia de Usuario 39

Nombre: Eliminar certificaciones académicas

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema permite al usuario eliminar unas certificaciones académicas.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 39. Elaboración propia.*

### Historia de Usuario 40

Nombre: Listar certificaciones académicas

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema muestra un listado de todas las certificaciones académicas registradas en el sistema.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 40. Elaboración propia.*

### **Historia de Usuario 41**

Nombre: Buscar certificaciones académicas

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Baja

Descripción: El sistema permite buscar las certificaciones académicas por el nombre.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 41. Elaboración propia.*

### **Historia de Usuario 42**

Nombre: Registrar grupo de formación

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema permite registrar un grupo de formación por el nombre y por una descripción.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 42. Elaboración propia.*

### Historia de Usuario 43

Nombre: Editar grupo de formación	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite editar un grupo de formación por el nombre.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 43. Elaboración propia.*

### Historia de Usuario 44

Nombre: Eliminar grupo de formación	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite al usuario eliminar el grupo de formación.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 44. Elaboración propia.*

### Historia de Usuario 45

Nombre: Listar grupo de formación	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema muestra un listado de todos los grupos de formación registrados en el sistema.	

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 45. Elaboración propia.*

### **Historia de Usuario 46**

Nombre: Buscar grupo de formación	Iteración Asignada: 1
-----------------------------------	-----------------------

Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
-----------------------------	-----------------------------

Descripción: El sistema permite buscar grupo de formación por el nombre.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 46. Elaboración propia.*

### **Historia de Usuario 47**

Nombre: Registrar publicación	Iteración Asignada: 1
-------------------------------	-----------------------

Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
-----------------------------	-----------------------------

Descripción: El sistema permite registrar una publicación por el nombre, nivel al que pertenece y el nombre del usuario quien la realizó.

Observación: Solamente lo pueden realizar usuarios con alta prioridad.

*Historia de Usuario 47. Elaboración propia.*

### **Historia de Usuario 48**

Nombre: Editar publicación	Iteración Asignada: 1
----------------------------	-----------------------

Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite editar una publicación por el nombre, nivel al que pertenece y el nombre del usuario quien la realizó.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 48. Elaboración propia.*

<b>Historia de Usuario 49</b>	
Nombre: Eliminar publicación	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Descripción: El sistema permite al usuario eliminar una publicación.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad.	

*Historia de Usuario 49. Elaboración propia.*

<b>Historia de Usuario 50</b>	
Nombre: Listar publicación	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema muestra un listado de todas las publicaciones registradas en el sistema.	
Observación: Solamente lo pueden realizar usuarios con alta prioridad	

*Historia de Usuario 50. Elaboración propia.*

<b>Historia de Usuario 51</b>	
Nombre: Buscar publicación	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Media
Descripción: El sistema permite buscar la publicación por el nombre	
Observación: Solamente lo pueden realizar usuarios con alta prioridad	

*Historia de Usuario 51. Elaboración propia.*

<b>Historia de Usuario 52</b>	
Nombre: Registrar grupo	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: media
Descripción: El sistema permite registrar un grupo por el nombre el nombre y una descripción	
Observación: Solo los usuarios autorizados	

*Historia de Usuario 52. Elaboración propia.*

<b>Historia de Usuario 53</b>	
Nombre: Editar grupo	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema permite editar un grupo por el nombre y una descripción.	

Observación: Solo los usuarios autorizados

*Historia de Usuario 53. Elaboración propia.*

#### **Historia de Usuario 54**

Nombre: Eliminar grupo

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema permite al usuario eliminar un grupo.

Observación: Solo los usuarios autorizados

*Historia de Usuario 54. Elaboración propia.*

#### **Historia de Usuario 55**

Nombre: Listar grupo

Iteración Asignada: 1

Prioridad en Negocio: Media

Riesgos de Desarrollo: Alta

Descripción: El sistema muestra un listado de todos los grupos registrados en el sistema.

Observación: Solo los usuarios autorizados

*Historia de Usuario 55. Elaboración propia.*

#### **Historia de Usuario 56**

Nombre: Buscar grupo

Iteración Asignada: 1

Prioridad en Negocio: Baja	Riesgos de Desarrollo: Media
Descripción: El sistema permite buscar un grupo por el nombre.	
Observación: Solo los usuarios autorizados	

*Historia de Usuario 56. Elaboración propia*

<b>Historia de Usuario 57</b>	
Nombre: Cambiar contraseña	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema permite al usuario cambiar la contraseña.	
Observación: Solo lo pueden realizar los usuarios que estén registrados en el sistema.	

*Historia de Usuario 57. Elaboración propia.*

<b>Historia de Usuario 58</b>	
Nombre: Cantidad de lecciones aprendidas por áreas de proceso	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Descripción: El sistema permite mostrar la cantidad de lecciones aprendidas por área de procesos en forma de gráfica.	
Observación: Solo lo pueden realizar los usuarios que estén registrados en el sistema.	

*Historia de Usuario 58. Elaboración propia.*

### Historia de Usuario 59

Nombre: Cantidad de usuarios activos      Iteración Asignada: 1

Prioridad en Negocio: Media      Riesgos de Desarrollo: Alta

Descripción: El sistema permite mostrar la cantidad de usuarios activos e inactivos en forma de gráfica.

Observación: Solo lo pueden realizar los usuarios que estén registrados en el sistema.

*Historia de Usuario 59. Elaboración propia.*

### Historia de Usuario 60

Nombre: Cantidad de usuarios por roles.      Iteración Asignada: 1

Prioridad en Negocio: Alta      Riesgos de Desarrollo: Alta

Descripción: El sistema permite mostrar la cantidad de usuarios por roles en forma de gráfica.

Observación: Solo lo pueden realizar los usuarios que estén registrados en el sistema.

*Historia de Usuario 60. Elaboración propia.*

### Historia de Usuario 61

Nombre: Entrada de registro.      Iteración Asignada: 1

Prioridad en Negocio: Media      Riesgos de Desarrollo: Alta

Descripción: El sistema permite el control de todas las acciones que se realicen en el sistema

Observación: Solo lo pueden realizar los usuarios que estén registrados en el sistema.

*Historia de Usuario 61. Elaboración propia.*