

Aplicación que permita gestionar un banco de proyectos del grupo investigación LIDIS

Andrés F. Bastidas Mazuera, afbastidas@correo.usbcali.edu.co

Reporte final de trabajo de grado para optar al título de Ingeniero de Sistemas

Asesor: Rocío Segovia, Ingeniero de sistemas (Ms)



**UNIVERSIDAD DE
SAN BUENAVENTURA
SECCIONAL CALI**

Universidad de San Buenaventura Colombia

**Facultad de Ingeniería
Programa Académico Ingeniería de Sistemas
Santiago de Cali, Colombia
Abril, 2022**

Cómo citar/referencia:

(Bastidas, 2022)

Bastidas, Andrés F, (2022) “Aplicación que permita gestionar un banco de proyectos del grupo investigación LIDIS”. Trabajo de grado Ingeniería de Sistemas, Universidad de San Buenaventura Cali, Facultad de Ingeniería, 2022



Bibliotecas Universidad de San Buenaventura



Biblioteca Digital (Repositorio)
<http://bibliotecadigital.usb.edu.co>

- Biblioteca Fray Alberto Montealegre OFM - Bogotá.
- Biblioteca Fray Arturo Calle Restrepo OFM - Medellín, Bello, Armenia, Ibagué.
- Departamento de Biblioteca - Cali.
- Biblioteca Central Fray Antonio de Marchena – Cartagena.

Universidad de San Buenaventura Colombia

Universidad de San Buenaventura Colombia - <http://www.usb.edu.co/>

Bogotá - <http://www.usbbog.edu.co>

Medellín - <http://www.usbmed.edu.co>

Cali - <http://www.usbcali.edu.co>

Cartagena - <http://www.usbctg.edu.co>

Editorial Bonaventuriana - <http://www.editorialbonaventuriana.usb.edu.co/>

Revistas - <http://revistas.usb.edu.co/>

Dedicatoria

Este trabajo de grado es dedicado a mi familia por su acompañamiento emocional y su apoyo constante en mi proceso académico.

Agradecimientos

Agradezco primeramente a la profesora y directora de este proyecto, Rocío Segovia por el acompañamiento en el proceso, por las retroalimentaciones y la atención sobre los avances. Agradezco a mi mamá Jackeline Mazuera Hernández que ya no se encuentra en este plano terrenal, pero que fue un gran apoyo y motivación para lograr concluir con esta etapa, agradezco a mi tía María Eugenia Bastidas y a mi abuela Berta Ligia Sánchez por el apoyo emocional y económico. Agradezco a mi novia Alexandra Castaño por el apoyo y retroalimentación, y al resto de familiares que también estuvieron presentes en el proceso.

Resumen

En el ámbito de investigación, la documentación juega un papel importante para los investigadores y personas con interés en profundizar en las temáticas, puesto que serán posibles insumos para nuevas investigaciones. Además de ello se requiere llevar el registro de la cronología, debido a que las publicaciones se ven afectadas por la antigüedad de referencias. Dentro de lo que se requiere documentar se encuentran los hallazgos, teorías o supuestos, resultados, trabajos futuros, entre otros.

En la Universidad de San Buenaventura Cali se cuenta con un proceso manual para la inscripción de proyectos, separado por directores de acuerdo a su línea de investigación, cumpliendo con los registros de documentos obligatorios de la universidad y gestionando al grupo de estudiantes inscritos de manera independiente, generando posibles fugas de conocimiento y demoras en las tareas administrativas cuando sea solicitada la trazabilidad de los proyectos del grupo de investigación LIDIS o se deba realizar la delegación de tareas que requieren en el semillero para los estudiantes, además de la dependencia sobre las personas implicadas en los procesos de gestión y seguimiento que puedan dar una actualización del estado del proyecto.

Existen múltiples herramientas para la gestión de proyectos, así como para la implementación de desarrollos y herramientas para la gestión de la configuración o documentación. Sin embargo el proceso de investigación cuenta con factores diferentes a los planteados en herramientas ya existentes.

El siguiente trabajo de grado se enfoca en implementar una herramienta para la gestión y centralización de la información de los proyectos de investigación del grupo de investigación LIDIS para la universidad de San Buenaventura Cali, lo cual permitirá ser una base para mejorar el proceso de los grupos de investigación y facilitar el seguimiento de los proyectos, así como la búsqueda de referencias de producción académica del programa en la universidad.

Abstract

In the field of research, documentation plays an important role for researchers and people interested in delving into the issues, since they will be possible inputs for new research. In addition to this, it is necessary to keep a record of the chronology, since the publications are affected by the age of the references. Among what is required to document are the findings, theories or assumptions, results, future work, among others.

At the University of San Buenaventura Cali there is a manual process separated by directors according to their line of research, complying with the mandatory document records of the university and managing the group of students enrolled independently, generating possible leaks of knowledge and delays in administrative tasks when the traceability of the projects is requested or the delegation of tasks that they require in the seedbed for the students must be carried out, in addition to the dependence on the people involved in the management and monitoring processes that they can give an update on the status of the project.

There are multiple tools for project management, as well as for the implementation of developments and tools for configuration or documentation management. However, the research process has factors different from those proposed in existing tools.

The following degree work focuses on implementing a tool for the management and centralization of the information of the research projects of the LIDIS research group for the University of San Buenaventura Cali, which will allow to be a base to improve the process of the groups. of research and facilitate the monitoring of projects, as well as the search for references of academic production of the program at the university.

GLOSARIO

- **NCs:** No conformidades.
- **Backend:** Proyecto que corresponde a toda la lógica del sistema que interactúa directamente con el servidor
- **Frontend:** Proyecto que corresponde a la usabilidad e interacción con el cliente.
- **Principios SOLID:** Se refiere a cinco principios para desarrollar software de calidad, sus siglas traducen: S - Single Responsibility Principle (SRP), O - Open/Closed Principle (OCP), L - Liskov Substitution Principle (LSP), I - Interface Segregation Principle (ISP), D - Dependency Inversión Principle (DIP)
- **UML:** Lenguaje unificado de modelado para la elaboración de diagramas.
- **AWS:** Amazon Web Services.
- **CRUD:** Create, Read, Update, Delete.

TABLA DE CONTENIDO

1. INTRODUCCIÓN	10
1.1 Contexto	10
1.2 Planteamiento del Problema	10
1.3 Justificación del Problema	11
1.4 Descripción y análisis de la complejidad del problema	11
1.5 Objetivo General	12
1.6 Objetivos Específicos	12
1.7 Descripción de la Solución	12
1.7.1 Proceso de Ingeniería	12
1.7.2 Productos de Software	13
1.7.3 Productos No Software	13
2.2 MARCO TEÓRICO	15
i. Gestión de proyectos	15
iv. Herramientas de gestión de proyectos	17
3. PROCESO DE INGENIERÍA DE SOFTWARE	20
3.1 Modelo de proceso de desarrollo de software.	20
3.1.1 Descripción de Iteraciones realizadas	20
3.2 Educción de Requisitos	25
3.2.1 Técnica utilizada:	25
3.2.2 Requisitos funcionales documentados	25
3.3 Análisis y Diseño	26
3.3.1 Atributos y restricciones Arquitecturales.	26
3.3.2 Requisitos No Funcionales.	26
3.4 Diseño de la Arquitectura.	27
3.4.1 Vista de Alto Nivel.	27
3.4.2 Vista de Procesos.	28
3.4.3 Vista Lógica.	30
3.4.4 Vistas de Desarrollo.	37
3.5 Patrones de Diseño Utilizados	38
3.6 Justificación de Diseño	38
3.7 Desarrollo	39
3.7.1 Tecnologías utilizadas	39
3.7.2 Tecnologías utilizadas y su Relación con los Atributos de Calidad.	40
3.8 Prototipos del Software	41
3.8.1 Desarrollo de prototipos Versión 1.0	41
3.8.2 Desarrollo de prototipos Versión 2.0	44
3.9 Pruebas y Evaluación de Software	45
3.9.1 Parámetros de Pruebas	45
3.9.2 Casos de Prueba	46
3.9.3 Resumen Pruebas Funcionales	54
3.9.4 Pruebas Piloto del Proyecto	55
3.9.4.1.1 Escenario de la prueba piloto	55
3.9.4.1.2 Resultados obtenidos de la prueba piloto	55
3.10 Acceso a la aplicación	56
3.11 Estándares de ingeniería empleados durante el proyecto	56
4. ANÁLISIS DE RESULTADOS	58
4.1 Análisis de Impactos	58
4.2 Utilización de herramientas	58

4.3	Análisis de diseño de componentes	58
4.4	Cumplimiento de Objetivos	59
4.5	Conclusiones	59
4.6	Experiencia de diseño en ingeniería de software	59
4.7	Trabajos Futuros	60
5.	REFERENCIAS	62
6.	ANEXOS	64

CONTENIDO DE TABLAS

Tabla 1:	Análisis de herramientas para gestión de proyectos	18
Tabla 2:	Iteración del proceso de desarrollo de software	25
Tabla 3:	Especificación y priorización de requisitos funcionales	26
Tabla 4:	Matriz de especificación y calcificación requisitos no funcionales	27
Tabla 5:	Ficha técnica del producto para la ejecución de pruebas	45
Tabla 6:	Tipos de pruebas	46
Tabla 7:	Matriz de casos de pruebas	52
Tabla 8:	Tabla diseño de pruebas de rendimiento	53
Tabla 9:	Tabla resultados prueba de rendimiento 1	53
Tabla 10:	Tabla resultados prueba de rendimiento 2	53
Tabla 11:	Tabla resultados prueba de rendimiento 3	53
Tabla 12:	Tabla resultados prueba de rendimiento 4	54
Tabla 13:	Tabla resultados pruebas funcionales	54
Tabla 14:	Tabla de no conformidades	55
Tabla 15:	Tabla funcionalidades prueba piloto	55
Tabla 16:	Tabla resultados prueba piloto	56

CONTENIDO DE ILUSTRACIONES

Ilustración 1:	Diagrama de contexto	27
Ilustración 2:	Diagrama de actividad de crear proyecto	28
Ilustración 3:	Diagrama de actividad iniciar sesión	29
Ilustración 4:	Diagrama de actividad Aprobar proyectos	29
Ilustración 5:	Diagrama de actividad Aprobar proyectos	30
Ilustración 6:	Diagrama de clases entidades	31
Ilustración 7:	Diagrama de clases objetos request	31
Ilustración 8:	Diagrama de clases objetos response	32
Ilustración 9:	Diagrama de clases para los objetos de fcd y constantes	33
Ilustración 10:	Diagrama de clases objetos repositorys	34
Ilustración 11:	Diagrama de clases interfaces de servicios	34
Ilustración 12:	Diagrama de clases implementación de servicios	35
Ilustración 13:	Diagrama de clases paquete de seguridad	36
Ilustración 14:	Diagrama de clases paquete de controller	36
Ilustración 15:	Diagrama de Componentes	38



1. INTRODUCCIÓN

- 1.1 Contexto
- 1.2 Planteamiento del problema
- 1.3 Justificación del problema
- 1.4 Descripción y análisis de la complejidad del problema
- 1.5 Objetivo general
- 1.6 Objetivos específicos
- 1.7 Descripción de la solución

1. INTRODUCCIÓN

1.1 Contexto

Los grupos de investigación de las universidades tienen como principal objetivo la investigación de hipótesis que permitan demostrar con alta calidad que los conocimientos adquiridos en la universidad permiten al estudiante desenvolverse en el mundo real generando aportes desde una perspectiva social que viva el país o aplicando nuevas tendencias para lograr innovación y cercanía de conocimientos a nivel nacional e internacional[1], [2].

Para la Universidad de San Buenaventura el objetivo principal es:

“Realizar investigación aplicada de alta calidad en el área de ingeniería de Software, con el propósito de constituir un centro de investigación reconocido a nivel regional e internacional.” [3]

El grupo de investigación LIDIS de la USB Cali, brinda soporte a los programas de ingeniería de sistemas, ingeniería multimedia y especializaciones por medio de semilleros que adoptan los proyectos del grupo de investigación.

LIDIS, tiene como principal objetivo apoyar la disciplina de ingeniería de software mediante proyectos de investigación, de desarrollo y formación. Aquellos semilleros se enfocan en líneas de investigación tales como desarrollo de aplicaciones empresariales, inteligencia artificial y ciencias cognitivas, gamificación, gestión del conocimiento entre otras.

Dicho soporte busca el ser referente académico a partir de artículos, implementaciones que solventan necesidades de la vida real, comprobaciones de hipótesis, ideas convertidas en proyectos de grado, entre otras. El proceso de la creación de cada idea o proyecto es iterativo en tiempo semestral, y, en muchas ocasiones requiere para ello que un proyecto tenga continuidad por periodos prolongados debido a su nivel de complejidad y amplio alcance.

En aquellos proyectos intervienen diferentes estudiantes que se encuentran en el mismo o en diferentes periodos académicos lo que implica al profesor mantener una trazabilidad de los proyectos del semillero para controlar y gestionar la información y el estado del proyecto, de acuerdo con su necesidad (reporte de estado, indicadores de producción académica, estudiantes del semilleros, publicación de temáticas) sin tener que solicitar la intervención de un tercero como lo es la biblioteca de la universidad para gestionar la información que por el momento solo la recibe cuando se ha publicado el documento de investigación.

Del mismo modo cada director, requiere cumplir con procedimientos y aplicación de plantillas que son requeridas por la universidad para la inscripción de la idea o proyecto.

1.2 Planteamiento del Problema

Los proyectos que desarrollan los semilleros de investigación del grupo LIDIS en la universidad de San Buenaventura Cali deben presentarse con una plantilla específica y cumplir unos procesos requeridos para ser inscrito, evaluado y aprobado por la universidad.

Toda la información que requiere ser gestionada y consultada en muchas ocasiones no se encuentra centralizada por lo que cada profesor debe llevar su inventario de documentos, gestiones e ideas por separado y posteriormente unirlos a través del canal de

comunicación definido. Del mismo modo, no se cuenta con un canal de comunicación donde los estudiantes puedan visualizar las ideas de proyectos del grupo de investigación LIDIS.

Lo anterior impacta en los profesores aumentando el tiempo que dedican a tareas administrativas y puede provocar fugas de conocimiento que no fueron plasmadas en ningún lugar en su momento por la cantidad de ideas no formales que se tienen día a día.

Por último, al no tener una herramienta que permita plasmar el proyecto, por consiguiente, no es posible generar tareas claras que puedan darse seguimiento y visualización del asesor y el estudiante para recibir retroalimentación activa.

1.3 Justificación del Problema

Existen dos factores principales que intervienen en la gestión del proyecto de investigación: las competencias profesionales de los investigadores y la gestión logística de los proyectos [4]

En relación con la gestión logística de proyectos la cual es de nuestro interés en esta temática, incluye el facilitar herramientas para dicha gestión. Por esta razón han creado herramientas que faciliten la visualización del proyecto para llegar a cumplir los objetivos en los tiempos estipulados con mayor orden [5]

En la institución universitaria tecnológico de Antioquia se implementó una aplicación web que permitiera la mejora en la gestión de los proyectos y trabajos de grados en sus diferentes etapas aplicando la metodología SCRUM, trabajo manual anteriormente realizado por el coordinador de proyectos. El resultado de la implementación fue la sistematización del banco de proyectos de la facultad que les permitió una mejor gestión.[6]

La problemática mencionada en este proyecto de grado se vio reflejada en la USB de la sede de Bogotá en la que se visualizó el inconveniente del tiempo de respuesta para las gestiones relacionadas con proyectos y emprendimientos. Utilizando tecnologías web para su solución (ver detalle en el marco teórico).[7]

El grupo de investigación LIDIS representa a la universidad generando productos que aportan al ranking de las universidades con alta calidad. Del mismo modo la representan por medio de congresos, exposiciones, publicaciones, entre otras.

Se considera necesario el desarrollo de la herramienta tecnológica que permita a docentes y estudiantes optimizar el proceso de consulta, asignación, elaboración y seguimiento de los proyectos.

1.4 Descripción y análisis de la complejidad del problema

La gran cantidad de herramientas que existen en el mercado sobre gestión de proyectos actualmente se enfocan en todo tipo de metodologías, sin embargo, se piensa que para obtener un producto que facilite al profesor la gestión completa de los proyectos es requerido que sea escalable y que pueda modificarse desde el código fuente de forma iterativa permitiendo el control de la herramienta para su propia gestión.

Puesto que los proyectos de investigación se comportan de una forma distinta a la que un proyecto de una empresa.

Uno de los puntos importantes a tener en cuenta en el problema es el control del estado del proyecto y cuando un estudiante puede conocer si su proyecto o idea se encuentra

aprobada o rechazada. El desarrollo de una aplicación para la fundación RASA tiene como objetivo ayudar a los usuarios del sector salud a recibir información sobre la protección de sus derechos. Sin embargo, el sistema no presentaba de una manera fácil para los usuarios dicha información. De lo anterior, se diseñó una herramienta modular que facilitará la implementación de banco de proyectos para que se adapte a cualquier proyecto permitiendo ampliarlos o disminuirlos en módulos fijos.

A partir de la implementación, se logró que las solicitudes de donación y los proyectos se puedan revisar y aprobar por medio de la herramienta para su financiamiento [8]

1.5 Objetivo General

Desarrollar una aplicación web que permita gestionar un banco de proyectos del grupo investigación LIDIS.

1.6 Objetivos Específicos

- Especificar los requisitos que definan el alcance de la aplicación del banco de proyectos
- Diseñar una solución acorde con la especificación de requisitos.
- Implementar la solución basada en el diseño.
- Validar la herramienta con el grupo de investigación LIDIS.

1.7 Descripción de la Solución

1.7.1 Proceso de Ingeniería

Para la consecución de las fases del proceso de desarrollo se consideró un modelo de desarrollo iterativo incremental con las siguientes fases:

Fase Planeación

Para la planeación se realizó la educación de requisitos dentro la Universidad de San Buenaventura Cali a partir de reuniones con la profesora Rocío Segovia, donde se hicieron lluvia de ideas y adicionalmente se construyeron prototipos de pantalla.

Se realizó un diagrama de casos de uso para extraer las funcionalidades y crear el documento de especificación de requisitos (SRS).

Fase Diseño:

Se diseñaron los diagramas requeridos para la implementación del software teniendo en cuenta las vistas 4+1 y el modelo C4.

Se realizó el diagrama de entidad relación para un mayor entendimiento de la lógica de almacenamiento y gestión de datos.

Se actualizó el documento de especificación de requisitos (SRS).

Fase de Desarrollo:

Se implementó lo plasmado en los diseños entregando producto por funcionalidad. Creación, eliminación, edición para las diferentes secciones del software. Proyecto, Usuarios, y funciones de la gestión.

Se actualizó el documento de especificación de requisitos (SRS) y el modelo de entidad relación, así como los prototipos de pantalla.



Fase de Pruebas

Se realizará e implementará el plan de pruebas.

Se definirán las métricas para que el software sea aceptado.

1.7.2 Productos de Software

- Entregables de proceso de software
- Documentación
- Código Fuente

1.7.3 Productos No Software

- Diagramas
- Prueba piloto con un profesor de la universidad de San Buenaventura.



2. MARCO TEÓRICO

- 2.1 Gestión de proyectos.
- 2.2 Herramientas de gestión de proyectos.

2.2 MARCO TEÓRICO

i. Gestión de proyectos

De acuerdo con la guía de gestión de proyectos de la Universidad EAFIT, un proceso debe tener para cumplir su concepto, los siguientes puntos[9]:

- Inicio: Formulación aprobada del proyecto.
- Entradas: Gestión de insumos necesarios para el proyecto
- Actividades: Tareas que se deben cumplir para lograr el objetivo del proyecto.
- Salidas: El resultado tangible del proyecto en documentación, pruebas de concepto, entrega del objetivo y sus resultados.
- Tiempo de las actividades: Plazos mínimos y máximos para desarrollar una actividad o agregar un insumo.

a. Gestión de proyectos en investigación

La ciencia universitaria es valorada por su capacidad para influir en la transformación del entorno y en que la sociedad disponga de mayor bienestar. El sector universitario, se ha distinguido, sobre todo en las naciones latinoamericanas, por ser el eje y el motor diseminador del conocimiento de la sociedad. Por consiguiente, la aplicación de herramientas para gestionar la ciencia y la tecnología en estas instituciones se convierte en un factor determinante para impulsar la producción científica y su gestión en el resto de las instituciones enmarcadas en la propia región [10]

a. Importancia de los semilleros de investigación:

En el proceso académico universitario los estudiantes tienen la oportunidad de pertenecer a los semilleros de investigación no solo para aprender sobre temas de su interés sino también para participar en congresos académicos, presentación de artículos de investigación e incluso en convocatorias de proyectos científicos, tecnológicos y de innovación de Minciencias (Ministerio de ciencia) donde de ser aprobados por la entidad gubernamental podrán recibir presupuesto para la gestión del proyecto. Dichos proyectos de Minciencias pueden ser de tipo investigación básica, aplicada, desarrollo tecnológico, de innovación y apoyo a la producción competitiva.

ii. Proceso de la universidad de San Buenaventura para semilleros de investigación.

Prerrequisitos:

- El estudiante debe inscribirse en un grupo de semillero existente llenando un formulario.
- El estudiante puede tener varios roles:
 - Un grupo de semillero tiene el rol de director, estudiante, administrador.

El proceso para realizar un proyecto como parte de un semillero de investigación es:

1. Generar una idea de proyecto: Se debe registrar en un formulario la idea del proyecto y ser enviado a los profesores que hacen parte del grupo de investigación.
2. Posterior a la aprobación de la idea de proyecto, se registra toda la información básica del proyecto tales como título, resumen, objetivos, introducción, justificación, estado del arte, y son enviados en un formulario para ser aprobados.

3. En adelante se comienza con toda la implementación requerida para alimentar el documento del proyecto.
- El proyecto pasará por diferentes estados tales como: solicitud inicial, disponible, en progreso, rechazado, terminado.

iii. Aplicaciones

1. Nacional

En la institución universitaria tecnológico de Antioquia se implementó una aplicación web que permitiera la mejora en la gestión de los proyectos y trabajos de grados en sus diferentes etapas aplicando la metodología SCRUM, trabajo manual anteriormente realizado por el coordinador de proyectos. El resultado de la implementación fue la sistematización del banco de proyectos de la facultad que les permitió una mejor gestión en el área.

Tecnologías utilizadas: Laravel Framework 5 con php 7, bootstrap, mysql y vagrant.

Arquitectura: Modelo – Vista – Control (MVC)

[6]

En la Universidad de San Buenaventura de Bogotá se implementó una página web para la gestión de proyectos de grado, proyectos integradores y prácticas empresariales en el programa de ingeniería de sistemas debido a que el programa no contaba con una aplicación que almacenara de forma sistematizada los datos referentes a la inscripción y seguimiento de los proyectos y se realizaba de forma manual generando mayores tiempos de respuesta al estudiante o consulta de información. El resultado obtenido fue el correcto proceso del almacenamiento de la información logrando en las consultas de los estudiantes, la visualización de un proceso más real de los docentes y estudiantes del programa.

Tecnologías utilizadas: php 4.0, mysql 4.0 y servidor apache 1.3.[7]

La fundación RASA tiene como objetivo ayudar a la comunidad de usuarios del sector salud a recibir de forma gratuita conocimiento sobre la protección de sus derechos relacionados con la salud.

Sin embargo, el sistema no presentaba de una manera fácil para los usuarios dicha información. De lo anterior se diseñó una herramienta modular que facilitará la implementación de banco de proyectos donde se presentan cualquier proyecto permitiendo ampliarlos o disminuirlos en módulos fijos.

Su objetivo era que las entidades y empresas del sector público aportaran a la ejecución de los proyectos de la Fundación RASA por lo que al final de la implementación, se logró que las solicitudes de donación y los proyectos se pudieran revisar y aprobar por medio de la herramienta para su financiamiento [8].

Se realizó un estudio en la Universidad de Cartagena para aplicar un sistema de información de producción científica por medio de un aplicativo web para que las instituciones pudieran informarse de la productividad científica de manera oportuna y generar indicadores con diferentes clasificaciones con la intención de dar mayor visibilidad a los grupos de investigación e investigadores. De lo anterior se pudo entregar utilizando la metodología RUP la sistematización de colecciones en sus áreas de conocimiento, circulación de la información, reportes sobre las publicaciones de revistas indexadas, libros de investigación, libros de texto, patentes, intervención en capítulos,

software, videos, proyectos entre otros. Cuenta con una sesión de administración y una de usuario visualizador.[11]

Tecnologías utilizadas: XHTML, XML, CSS y AJAX, framework Java server faces, mysql y servidor Glassfish 4.1, JPA ,ORM, EJB.

2. Internacional

El departamento de investigación CENI de la universidad técnica de Ambato, Ecuador, tuvo la necesidad de sistematizar la presentación de informes, controlar a los investigadores y de la información, así como las asignaciones presupuestarias. Para ello, implementaron un sistema web interno en la universidad que permitiera llevar el control de los proyectos de investigación para el ingreso de usuarios, emisión de Informes de proyectos, envío de archivos para revisión, control en tiempo de presentación de proyectos, búsqueda de investigadores, presentación de informes, modificación de fechas, modificación de información de (investigadores, coordinadores, coordinadores de unidades operativas de facultades) y búsqueda de proyectos.

Tecnologías utilizadas: php, yum, mysql.

Arquitectura: Modelo – Vista – Control (MVC)[12]

En la universidad de ciencias médicas Cienfuegos, Cuba. Se buscaba el garantizar que la gestión de la producción científica se hiciera de forma óptima, la razón principal fue la universidad contaba con más de 400 profesores los cuales deben producir material científico ya que la universidad es medida por un indicador de producción científica. Debido a lo anterior, iniciaron implementando un sistema de información web que generara currículos digitales de los profesores universitarios con la intención de resaltar la experiencia y recorrido de los profesores y pudiera ser consultada desde cualquier institución de salud, adicionando indicadores totales e individuales basados en criterios de la Junta de Acreditación Nacional del Ministerio de Educación Superior de Cuba

Tecnologías utilizadas: php, mysql y servidor apache.

Metodología RUP, modelado UML.[13]

El objetivo del centro de investigación y desarrollo de la PUCESE desde el 2009 no contaba con una herramienta para visibilizar el trabajo desarrollado el cual está enfocado en la implementación de metodologías de investigación aplicada en proyectos de investigación que aporten para la economía y la sociedad de la provincia de esmeraldas, Ecuador. Las funcionalidades que se implementaron fueron enfocadas en las participaciones y eventos de investigación tal como la publicación de información institucional, difusión de eventos de investigación y proyectos realizados, registrar propuestas, reportes de las propuestas y de los resultados de las participaciones, publicaciones de los mejores proyectos presentados.

Tecnologías utilizadas: Para este proyecto se utilizó un servidor físico en la universidad, servidor apache, herramienta CodeIgniter.

Arquitectura: Modelo – Vista – Control (MVC)[14]

iv. Herramientas de gestión de proyectos

A continuación, se presenta un cuadro comparativo entre algunas de las herramientas más conocidas para gestionar proyectos y sus capacidades representativas.

Herramienta	Metodologías	Modo	Estados del proyecto	Estados de las actividades
JIRA y Confluence [15]	Scrum Kanban Proyectos	Pago si se quieren informes de seguimiento. Tiene formulas y automatizaciones. Tiene diferentes vistas.	Personalizado	Personalizado
Trello [16]	Scrum Kanban Proyectos	Pago posterior a 10 tableros. Tiene integraciones con otros sistemas.	Personalizado	Personalizado
Assana [17]	Scrum Kanban Proyectos	Permite sesiones un solo usuario y posteriormente pedirá pago para análisis de información. Tiene diferentes vistas.	Personalizado	Personalizado
Office con planner [18]	Proyectos	Es de uso en línea para office.	N/A	N/A
Notion [19]	Scrum Kanban Proyectos	Es gratis. Tiene integraciones. Tiene diferentes vistas.	Activo / Inactivo	Etiquetas
Monday [20]	Scrum Kanban Proyectos	Solo permite gratis dos usuarios. Tiene formulas y automatizaciones. Tiene diferentes vistas.	Personalizado	Personalizado

Tabla 1: Análisis de herramientas para gestión de proyectos

Fuente: Elaboración propia



2 PROCESO DE INGENIERÍA DE SOFTWARE

- 3.1 Modelo de proceso de desarrollo de software
- 3.2 Educción de requisitos
- 3.3 Análisis y diseño
- 3.4 Diseño de la arquitectura.
- 3.5 Patrones de diseño utilizados
- 3.6 Justificación de diseño
- 3.7 Desarrollo
- 3.8 Prototipos del software
- 3.9 Pruebas y evaluación de software
- 3.10 Acceso a la aplicación
- 3.11 Estándares de ingeniería empleados durante el proyecto

3. PROCESO DE INGENIERÍA DE SOFTWARE

Como proceso de ingeniería se plantearon cuatro fases que son la fase de análisis, diseño, implementación y pruebas. En la fase de análisis se construyó el documento de educación de requisitos, prototipos de pantallas, diagramas de casos de uso y diagrama de contexto. En la fase de diseño se hizo el diagrama de entidad relación, se modificaron prototipos de pantalla y se diseñó el diagrama de componentes. Durante la fase de implementación, se tuvieron en cuenta tres etapas:

- A. La primera etapa del desarrollo, se enfocó en el backend, donde se escogieron los requisitos funcionales que fueran de alto impacto, tales como:
 - Crear proyecto
 - Listar proyectos asignados
 - Aprobar proyectos
 - Rechazar proyectos
- B. En la segunda etapa de la implementación, se diseñó la maquetación para el frontend, empezando por las interfaces de:
 - Login
 - Crear proyectos
 - Listar proyectos asignados
- C. Como tercera etapa, se pasó a realizar el acople entre los servicios de backend y el frontend. Una vez finalizada esta tercera etapa, se tuvo una primera versión para recibir la retroalimentación inicial de la aplicación.

Para la fase de pruebas, se hicieron pruebas unitarias, pruebas de carga y pruebas de humo.

3.1 Modelo de proceso de desarrollo de software.

Para el proceso del desarrollo de software, la metodología usada fue la iterativa incremental y se tuvieron en cuenta las siguientes fases:

- Análisis
- Diseño
- Implementación
- Pruebas

3.1.1 Descripción de Iteraciones realizadas

En la tabla 2, se muestra el proceso de iteraciones realizadas:

Iteración No.1	Septiembre de 2021
Artefactos de Entrada:	Entendimiento del problema
Descripción:	Reunión con la profesora Rocío Segovia para la contextualización del problema y la propuesta solución.
Artefactos de Salida:	Diagrama de casos de uso

Iteración No.1	Octubre de 2021
Artefactos de Entrada:	Prototipos de pantalla
Descripción:	Construcción de interfaces
Artefactos de Salida:	Matriz de MER

Iteración No.2	Noviembre de 2021
Artefactos de Entrada:	Educción de requisitos
Descripción:	Se construye la versión 1.0 de documento SRS
Artefactos de Salida:	Documento SRS

Iteración No.3	Diciembre de 2021
Artefactos de Entrada:	Matriz MER
Descripción:	Construcción del modelo entidad relación
Artefactos de Salida:	Modelo entidad relación

Iteración No.3	Diciembre de 2021
Artefactos de Entrada:	Modelo entidad relación
Descripción:	Se realizó la creación de la base de datos basado en el modelo entidad relación.
Artefactos de Salida:	Creación de base de datos

Iteración No.3	Diciembre de 2021
Artefactos de Entrada:	Configuración del proyectos backend/frontend
Descripción:	Se procede a configurar el proyecto verificando la conexión a base de datos por el lado del backend.
Artefactos de Salida:	Repositorio en github

Iteración No.4	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Login
Descripción:	Se comienza el desarrollo para que un usuario pueda loguearse en la aplicación.
Artefactos de Salida:	Creación de clases de modelo y repositorios

Iteración No.4	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Login
Descripción:	Se añade el servicio y DAO para autenticar un usuario
Artefactos de Salida:	Creación de clases para los servicios de login

Iteración No.4	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Login
Descripción:	Se añade el controlador para la autenticación de usuario
Artefactos de Salida:	Creación de clase controlador para el login

Iteración No.4	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Login
Descripción:	Se crea la interfaz de usuario para loguearse en el frontend
Artefactos de Salida:	Implementación de login en el frontend

Iteración No.4	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Login
Descripción:	Se crean los modelos, servicios y componentes en el frontend
Artefactos de Salida:	Implementación de login en el frontend

Iteración No.4	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Login
Descripción:	Se realiza el acople entre el frontend y backend para la implementación de login
Artefactos de Salida:	Implementación de login en el frontend y backend

Iteración No.5	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Crear proyecto
Descripción:	Se comienza el desarrollo para el requisito funcional de crear proyecto.
Artefactos de Salida:	Creación de clases del modelo, repositorios y servicios

Iteración No.5	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Crear proyecto
Descripción:	Se crean las listas genéricas de directores, entregables, tipologías de investigación
Artefactos de Salida:	Clases de listas genéricas backend

Iteración No.5	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Crear proyecto
Descripción:	Se crea el controlador y el llamado al servicio de crear proyecto
Artefactos de Salida:	Implementación de crear proyecto en el backend

Iteración No.5	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Crear proyecto
Descripción:	Se hace el diseño de la interfaz en el frontend, se crea el grid para la distribución de filas y columnas
Artefactos de Salida:	Diseño de interfaz crear proyecto en el frontend

Iteración No.5	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Crear proyecto
Descripción:	Se crean los modelos, componentes y servicios para las listas genéricas
Artefactos de Salida:	Clases para listas genéricas en el frontend

Iteración No.5	Diciembre de 2021
Artefactos de Entrada:	Requisito funcional Crear proyecto
Descripción:	Se hace el acople de frontend y backend
Artefactos de Salida:	Implementación funcional crear proyecto backend y frontend

Iteración No.5	Diciembre de 2021
Artefactos de Entrada:	Implementación de crear proyecto
Descripción:	Se realizaron pruebas unitarias para las funcionalidades de crear proyectos y login.
Artefactos de Salida:	Pruebas unitarias crear proyecto y login

Iteración No.5	Diciembre de 2021
Artefactos de Entrada:	Implementación de crear proyecto
Descripción:	Reunión con la profesora Rocío Segovia para la retroalimentación de la primera versión del desarrollo. Se hace en la lista genérica de entregables donde se le añaden más opciones como tesis de doctorado y artículo de investigación
Artefactos de Salida:	Ajustes en la funcionalidad crear proyecto

Iteración No.5	Enero de 2022
Artefactos de Entrada:	Requisito funcional proyectos asignados
Descripción:	Se construye el query para consultar los proyectos que tiene asignado un usuario
Artefactos de Salida:	Implementación proyectos asignados

Iteración No.5	Enero de 2022
Artefactos de Entrada:	Implementación proyectos asignados
Descripción:	Se construye las clases de los servicios
Artefactos de Salida:	Implementación proyectos asignados en el backend

Iteración No.5	Enero de 2022
Artefactos de Entrada:	Implementación proyectos asignados
Descripción:	Se construye las clases de los servicios
Artefactos de Salida:	Implementación proyectos asignados en el backend

Iteración No.5	Enero de 2022
Artefactos de Entrada:	Implementación proyectos asignados
Descripción:	Se construye el frontend de proyectos asignados
Artefactos de Salida:	Implementación frontend proyectos asignados en el frontend

Iteración No.6	Enero de 2022
Artefactos de Entrada:	Requisitos funcionales Aprobar y rechazar proyectos
Descripción:	Se realiza la implementación para aprobar y rechazar proyectos creados por estudiantes y profesores generando los servicios y controladores
Artefactos de Salida:	Implementación Aprobar y rechazar proyectos en el backend

Iteración No.6	Enero de 2022
Artefactos de Entrada:	Requisitos funcionales Aprobar y rechazar proyectos
Descripción:	Se construye la interfaz para aprobar y rechazar proyectos
Artefactos de Salida:	Implementación Aprobar y rechazar proyectos en el frontend

Iteración No.6	Enero de 2022
Artefactos de Entrada:	Requisito funcional asociar usuario a un proyecto
Descripción:	Se realiza la implementación para asociar usuarios a un proyecto
Artefactos de Salida:	Implementación asociar usuario a un proyecto backend

Iteración No.6	Enero de 2022
Artefactos de Entrada:	Requisito funcional asociar usuario a un proyecto
Descripción:	Se realiza construye la interfaz para asociar un usuario a un proyecto
Artefactos de Salida:	Implementación asociar usuario a un proyecto en el frontend

Iteración No.6	Enero de 2022
Artefactos de Entrada:	Pruebas unitarias
Descripción:	Se realizan pruebas unitarias para las funcionalidades proyectos asignados, aprobar/rechazar proyectos y asociar un usuario.
Artefactos de Salida:	Documentación de hallazgos encontrados

Iteración No.7	Enero de 2022
Artefactos de Entrada:	Requisitos funcionales para el CRUD de Usuario
Descripción:	Se realiza la implementación para crear, editar, borrar y eliminar un usuario.
Artefactos de Salida:	Implementación de requisitos funcionales para el CRUD de Usuario en el backend

Iteración No.7	Enero de 2022
Artefactos de Entrada:	Requisitos funcionales para el CRUD de Usuario
Descripción:	Se realiza la implementación para crear, editar, borrar y eliminar un usuario.
Artefactos de Salida:	Implementación de requisitos funcionales para el CRUD de Usuario en el frontend

Iteración No.7	Enero de 2022
Artefactos de Entrada:	Requisitos funcionales para el CRUD de Actividad
Descripción:	Se realiza la implementación para crear, editar, borrar y eliminar un actividad.
Artefactos de Salida:	Implementación de requisitos funcionales para el CRUD de Actividad en el backend

Iteración No.7	Enero de 2022
Artefactos de Entrada:	Requisitos funcionales para el CRUD de Actividad
Descripción:	Se realiza la implementación para crear, editar, borrar y eliminar un actividad.
Artefactos de Salida:	Implementación de requisitos funcionales para el CRUD de Actividad en el frontend

Iteración No.7	Enero de 2022
Artefactos de Entrada:	Requisitos funcionales para el CRUD de Comentario
Descripción:	Se realiza la implementación para crear, editar, borrar y eliminar un comentario.
Artefactos de Salida:	Implementación de requisitos funcionales para el CRUD de Comentario en el backend

Iteración No.7	Enero de 2022
Artefactos de Entrada:	Requisitos funcionales para el CRUD de Comentario
Descripción:	Se realiza la implementación para crear, editar, borrar y eliminar un comentario.
Artefactos de Salida:	Implementación de requisitos funcionales para el CRUD de Comentario en el frontend

Iteración No.7	Enero de 2022
Artefactos de Entrada:	Pruebas unitarias
Descripción:	Se realizan pruebas unitarias para los módulos de la iteración
Artefactos de Salida:	Documentación de hallazgos encontrados

Iteración No.7	Enero de 2022
Artefactos de Entrada:	Pruebas de humo
Descripción:	Se realizan de prueba para los requisitos implementados
Artefactos de Salida:	Documentación de hallazgos encontrados

Iteración No.8	Febrero de 2022
Artefactos de Entrada:	Documento SRS
Descripción:	Se eliminaron requisitos funcionales del alcance como generar reportes y adjuntar enlaces
Artefactos de Salida:	Versión SRS 2.0

Tabla 2: Iteración del proceso de desarrollo de software

Fuente: Elaboración propia

3.2 Educción de Requisitos

Para la educación de requisitos se utilizaron técnicas, lluvia de ideas, prototipos de pantalla y reuniones donde a partir de esto, se seleccionaron los principales requisitos funcionales tales como:

- El sistema debe permitir crear un proyecto para estudiantes, profesores y directores.
- El sistema debe permitir iniciar sesión en la aplicación como rol de estudiante, profesor y director.
- El sistema debe permitir crear usuario.
- El sistema debe permitir listar los proyectos asignados a un usuario.

3.2.1 Técnica utilizada:

Las técnicas utilizadas fueron lluvia de ideas y sesiones con la profesora Rocío Segovia para recopilar información de los requisitos funcionales del sistema, adicionalmente se tuvo en cuenta el proceso de Inscripción de un proyecto de investigación en la universidad para el grupo de investigación LIDIS.

3.2.2 Requisitos funcionales documentados

Ref.	Prioridad	Descripción Requisito	Fuente de Información
------	-----------	-----------------------	-----------------------

RF-01	Alta	El sistema permitirá a los usuarios estudiante, profesor y director crear proyectos	Lluvia de ideas y Reuniones
RF-02	Alta	El sistema permitirá a los usuarios iniciar sesión en la aplicación	Lluvia de ideas y Reuniones
RF-03	Alta	El sistema permitirá a los usuarios estudiante, director y profesor visualizar los proyectos que tiene asociados	Lluvia de ideas y Reuniones
RF-08	Alta	Visualizar Proyectos aprobados/rechazados	Lluvia de ideas y Reuniones
RF-09	Alta	El sistema permitirá al director aprobar proyectos en el sistema	Lluvia de ideas y Reuniones
RF-10	Alta	El sistema permitirá al director rechazar proyectos en el sistema	Lluvia de ideas y Reuniones
RF-12	Alta	El sistema permitirá al director crear usuarios	Lluvia de ideas y Reuniones
RF-20	Alta	Asociar participantes a un proyecto	Lluvia de ideas y Reuniones

Tabla 3: Especificación y priorización de requisitos funcionales

Fuente: Elaboración propia

3.3 Análisis y Diseño

3.3.1 Atributos y restricciones Arquitecturales.

En esta sección se describe los criterios de calidad del software que tienen un impacto significativo en el diseño de la arquitectura.

- **Mantenibilidad:** La aplicación está diseñada por módulos para incorporar o actualizar funcionalidades sin generar conflictos o errores.
- **Portabilidad:** El sistema está diseñado para funcionar en Google Chrome con sistema operativo Windows 7,8,10,11.

3.3.2 Requisitos No Funcionales.

Ref.	Descripción Requisito	Seguridad	Rendimiento	Usabilidad
------	-----------------------	-----------	-------------	------------

RNF-1	El sistema deberá permitir que más de 30 usuarios se conecten a la vez y puedan utilizar la aplicación sin generar fallos en la aplicación.		x	
RUF-1	La interfaz de usuario debe ser adaptable para cualquier resolución de pantalla que sea mayor a 1024x768			x
RSF-1	El sistema solo debe permitir el ingreso a los usuarios registrados	x		
RSF-2	El sistema valida la fuente de la cual se hace la petición para el consumo de la aplicación	x		

Tabla 4: Matriz de especificación y calcificación requisitos no funcionales

Fuente: Elaboración propia

3.4 Diseño de la Arquitectura.

Como arquitectura de la aplicación se usó el MVC (Modelo Vista Controlador) para separar los módulos de persistencia, negocio y controladores, Se utilizaron principios SOLID como son:

- Principio de única responsabilidad (SRP),
- Principio de abierto y cerrado (OCP).

Adicional se utilizaron técnicas de código limpio, en clases, interfaces, métodos y atributos, logrando cumplir con criterios tales como:

- Nombres con sentido que revelen la intención.
- Eviten la desinformación con la asignación de nombres para las clases, métodos y variables.

3.4.1 Vista de Alto Nivel.

La Ilustración 1 representa en un diagrama de contexto usando el modelo C4 de como los usuarios interactúan con la aplicación web de forma directa e indirecta:

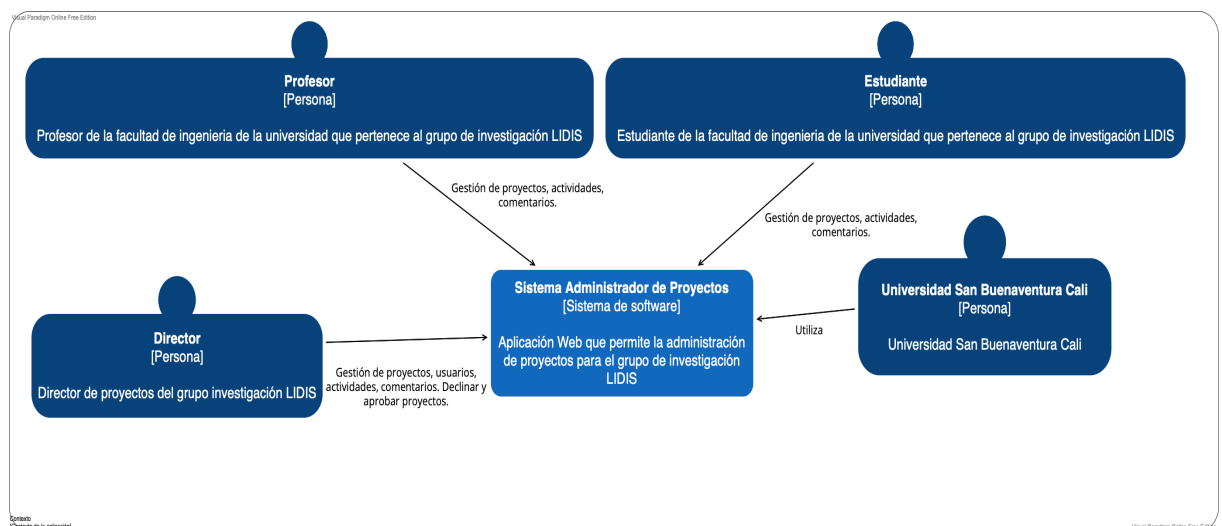


Ilustración 1: Diagrama de contexto

Fuente: Elaboración propia

En la figura se puede observar, los usuarios de estudiante, profesor y director, estos cuentan con diferentes permisos para acceder a la aplicación que se describen a continuación:

- Estudiante: Crear proyectos, gestionar proyectos (actualizar estado de proyecto, crear actividades, crear comentarios en la actividades), visualizar solicitudes de proyectos creados y visualizar todos los proyectos.
- Profesor: Crear proyectos, gestionar proyectos (actualizar estado de proyecto, crear actividades, crear comentarios en la actividades), visualizar solicitudes de proyectos creados y visualizar todos los proyectos.
- Director: Crear proyectos, gestionar proyectos (actualizar estado de proyecto, crear actividades, crear comentarios en la actividades), aprobar proyectos, rechazar proyectos, gestionar usuarios (crear, editar, eliminar usuarios), asociar usuario a un proyecto y visualizar todos los proyectos.

Por último, en la figura se puede ver a la Universidad San Buenaventura Cali, como beneficiado del uso de este aplicativo.

3.4.2 Vista de Procesos.

Para la vista de procesos, se representó en diagramas de actividades, tales como:

- Diagrama de crear proyecto.
- Diagrama iniciar sesión.
- Diagrama aprobación de proyectos.
- Diagrama de consultar proyectos aprobados.

La ilustración 2 representa todo el flujo para la creación de proyectos para cada perfil de usuario:

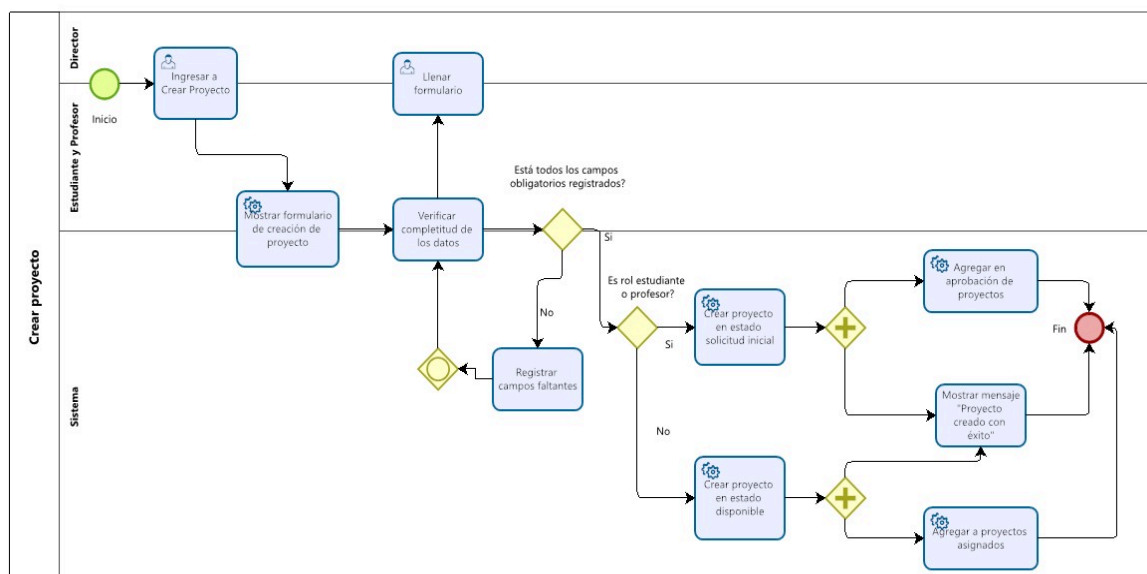
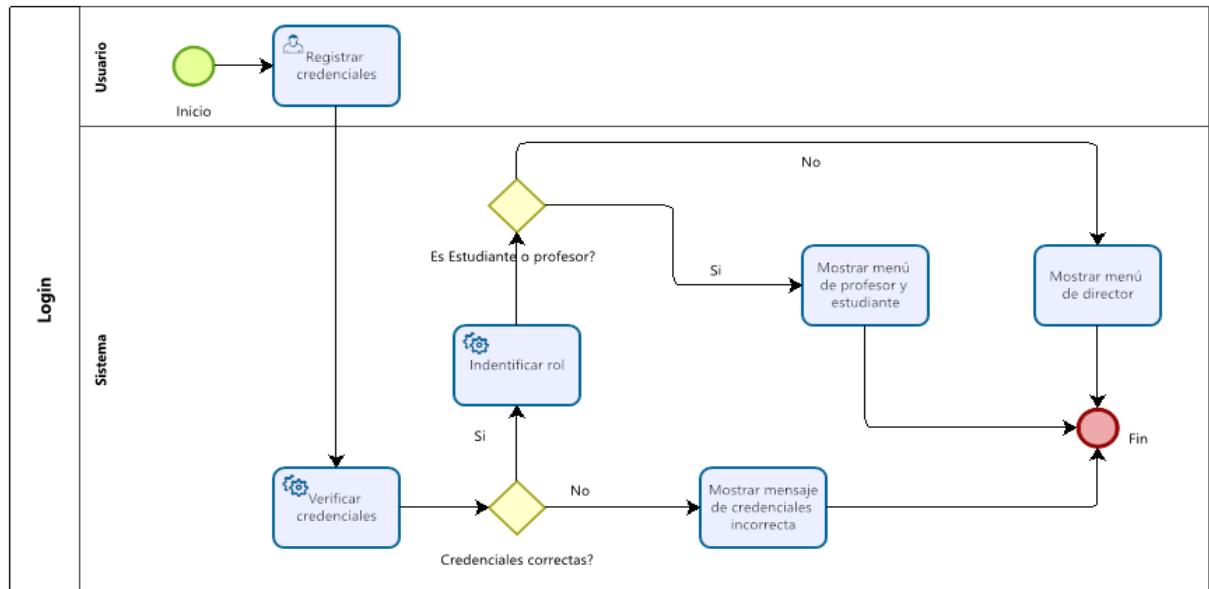


Ilustración 2: Diagrama de actividad de crear proyecto

Fuente: Elaboración propia

La Ilustración 3, representa el flujo de iniciar sesión:

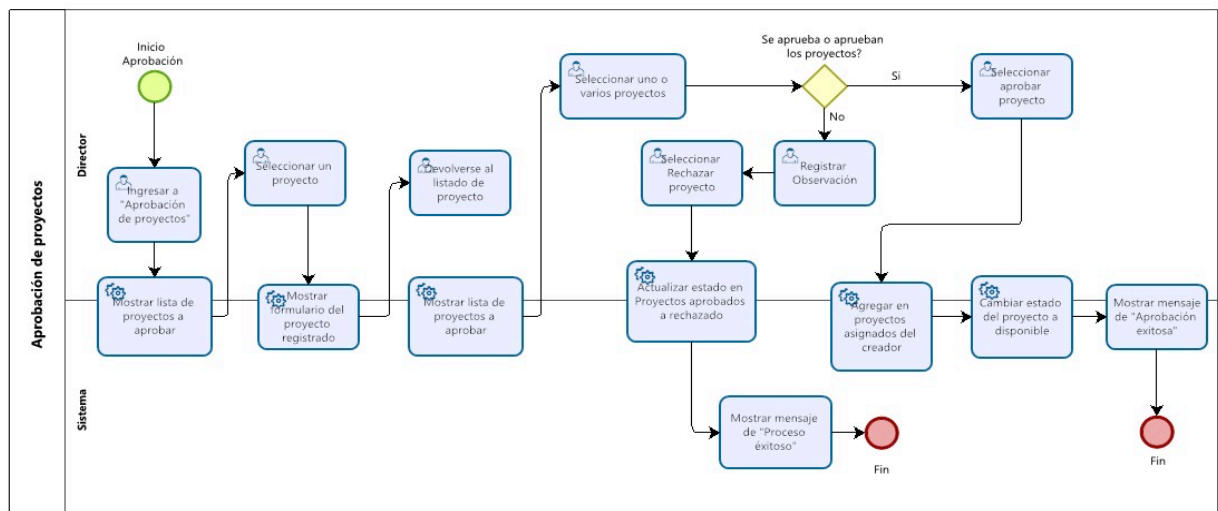


Powered by
bizagi
Modeler

Ilustración 3: Diagrama de actividad iniciar sesión

Fuente: Elaboración propia

La ilustración 4 representa el flujo para aprobar proyecto por parte del perfil de director:



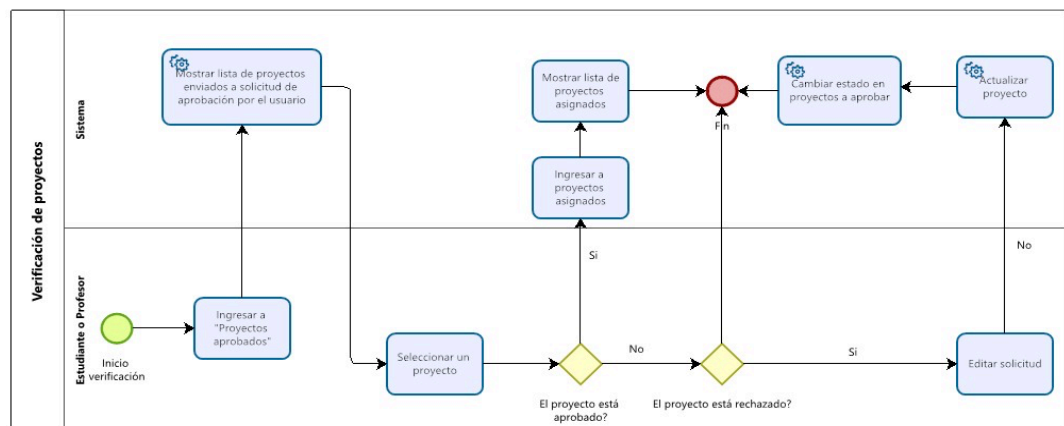
Powered by
bizagi
Modeler

Ilustración 4: Diagrama de actividad Aprobar proyectos

Fuente: Elaboración propia

La ilustración 5 representa el flujo de un estudiante o profesor para consultar los proyectos aprobados o rechazados:

No



Powered by
bizagi
Modeler

Ilustración 5: Diagrama de actividad consultar proyectos aprobados y rechazados

Fuente: Elaboración propia

3.4.3 Vista Lógica.

La vista lógica se separó en cuatro capas, la capa repository, business, model y controller, representadas en diagramas de clases que se explican a continuación.

En la Ilustración 6 representa uno de los paquetes de la capa model, en este paquete están todas las entidades y sus atributos del modelo de base de datos.

Las entidades creadas fueron:

- Project
- State
- Activity
- StateActivity
- Comment
- ProjectDelivery
- Delivery
- ResearchTypology
- LinkAttached
- ProjectRequest
- StateProjectRequest
- Userapp
- Profile
- ProjectUser

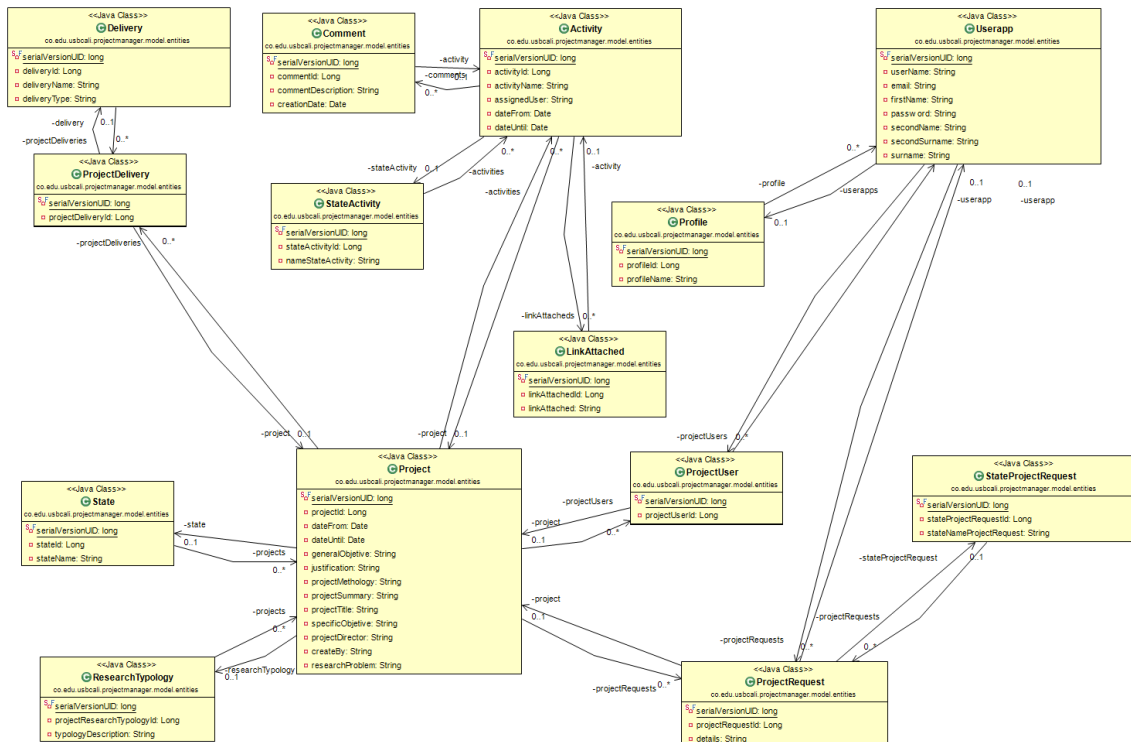


Ilustración 6: Diagrama de clases entidades

Fuente: Extraído de Eclipse

En la capa model, se crearon clases para las peticiones de los servicios que se muestran a continuación en la ilustración 7:

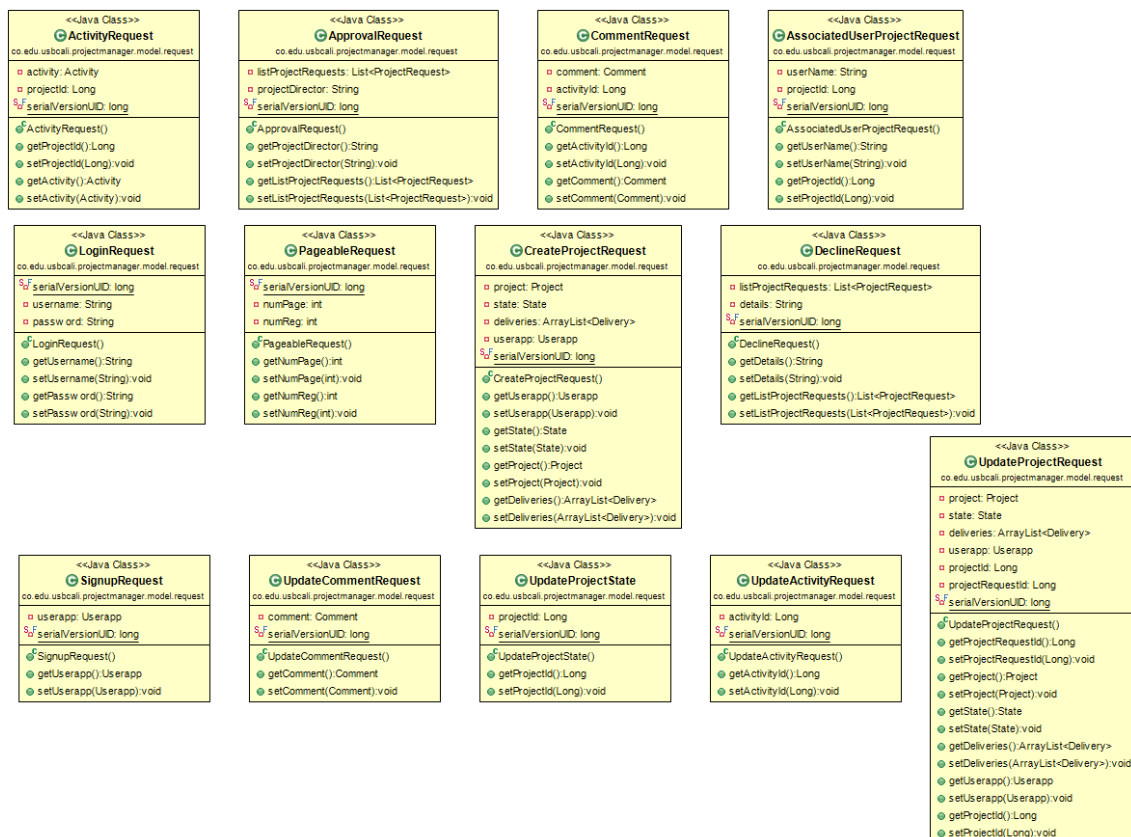


Ilustración 7: Diagrama de clases objetos request

Fuente: Extraído de Eclipse

En la ilustración 8, están las clases para las respuesta de los servicios:

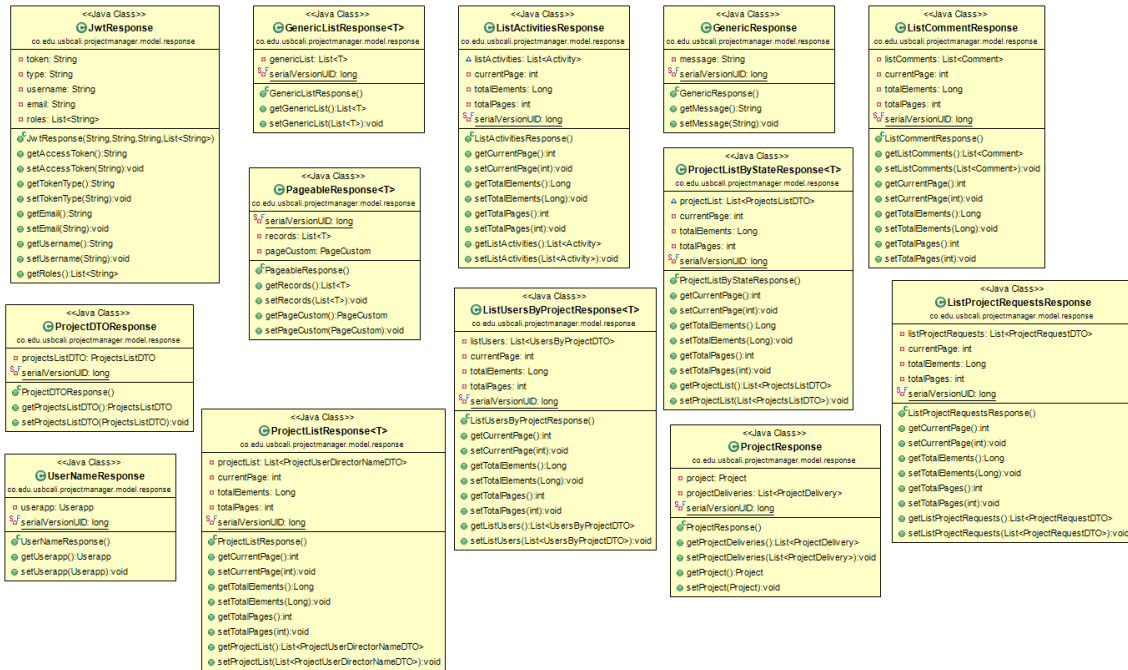


Ilustración 8: Diagrama de clases objetos response

Fuente: Extraído de Eclipse

En la Ilustración 9, en el lado derecho, se puede observar la clase KeyConstants, que se creó con el objetivo de mostrar los mensajes de éxito, los mensajes de error, los tipos de error y constantes usadas para validaciones de negocio.

En el lado izquierdo, se implementó la clase FcdConstants, fue usada para añadir las diferentes direcciones de cada servicio.

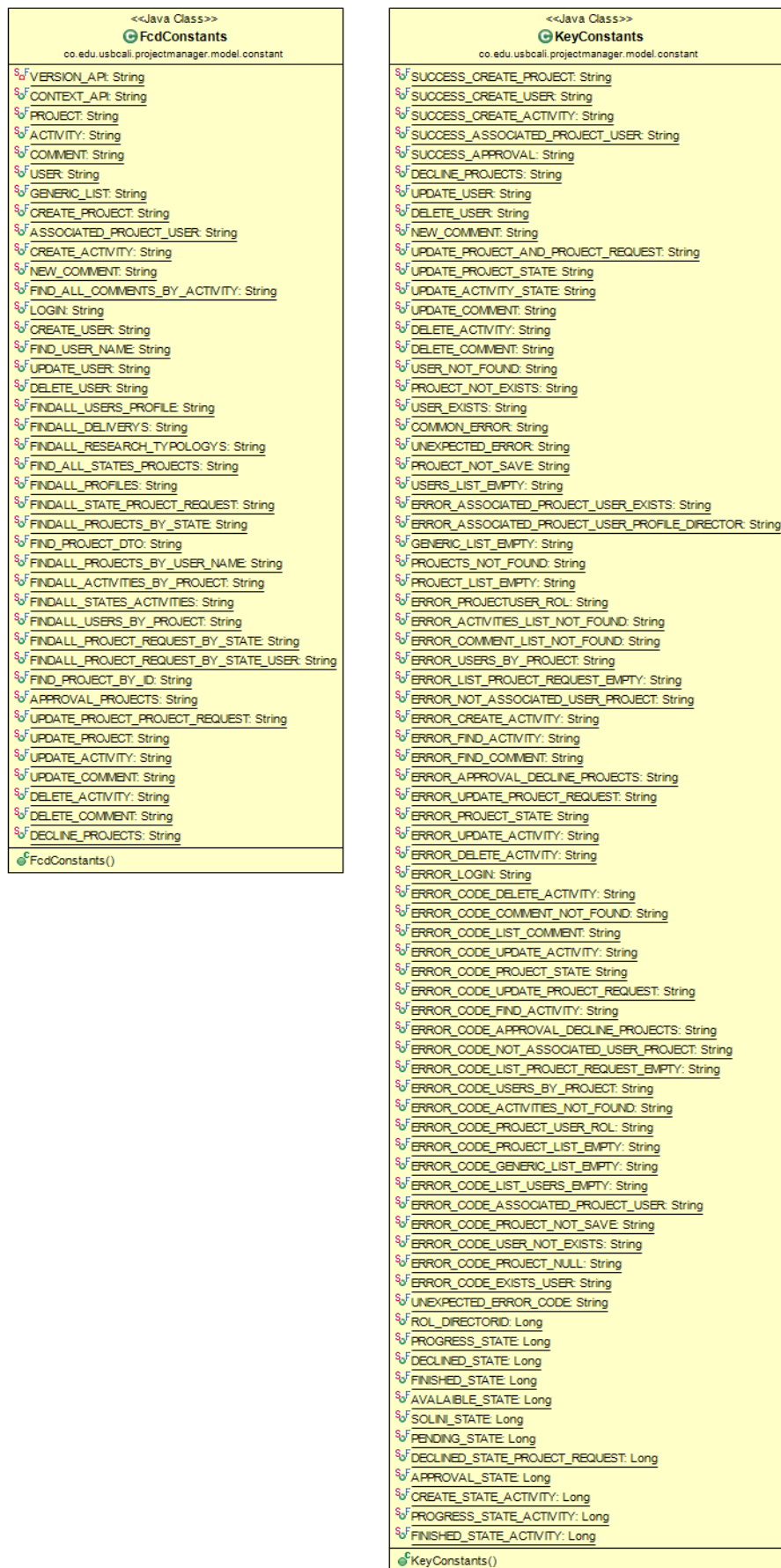


Ilustración 9: Diagrama de clases para los objetos de fcd y constantes.

Fuente: Extraído de Eclipse

Seguido de la capa de modelo, se implementó la capa repository, representada en la Ilustración 10, cuya función es consultar, guardar, modificar y eliminar, en las diferentes entidades de la base de datos. Cada entidad del modelo ,incluyendo los DTOs, tiene su representación en interfaces con el prefijo Repository, que utiliza la librería JpaRepository.

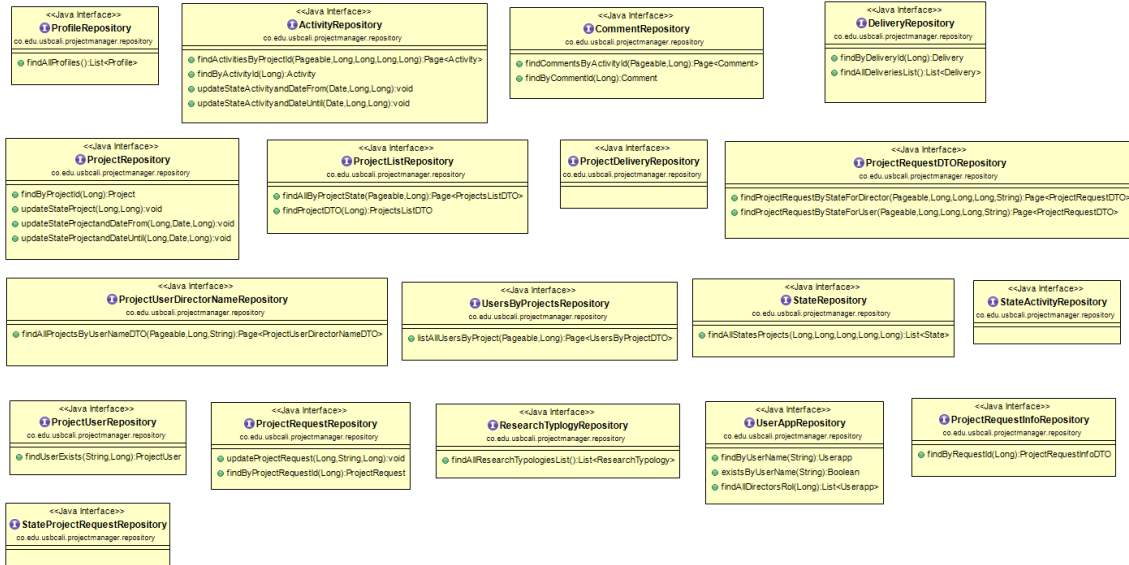


Ilustración 10: Diagrama de clases objetos repositorys
Fuente: Extraído de Eclipse

La capa de business, se representa en dos figuras, la Ilustración 11, representa las interfaces con la firma de los métodos.

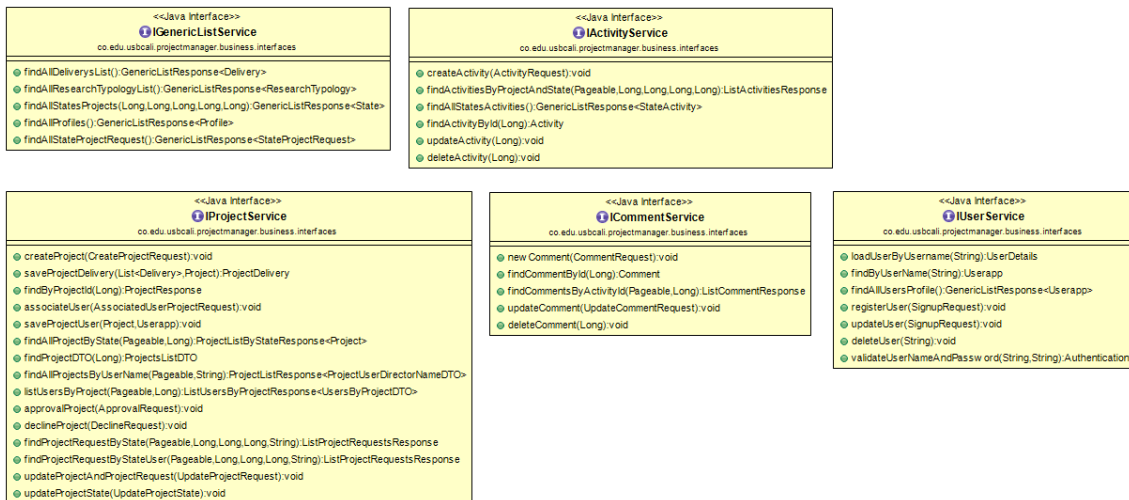


Ilustración 11: Diagrama de clases interfaces de servicios
Fuente: Extraído de Eclipse

La ilustración 12, representa las clases con los métodos e implementaciones.



Ilustración 12: Diagrama de clases implementación de servicios

Fuente: Extraído de Eclipse

En la capa de controller, para la implementación de seguridad, se crearon cuatro clases:

- AuthEntryPointJwt: intercepta las peticiones y valida la autorización con un token.
- WebSecurityConfig: Hace la configuración de CORS Origins y los métodos permitidos.
- AuthTokenFilter: Valida el usuario para la autenticación al sistema.

Ver Ilustración 13.

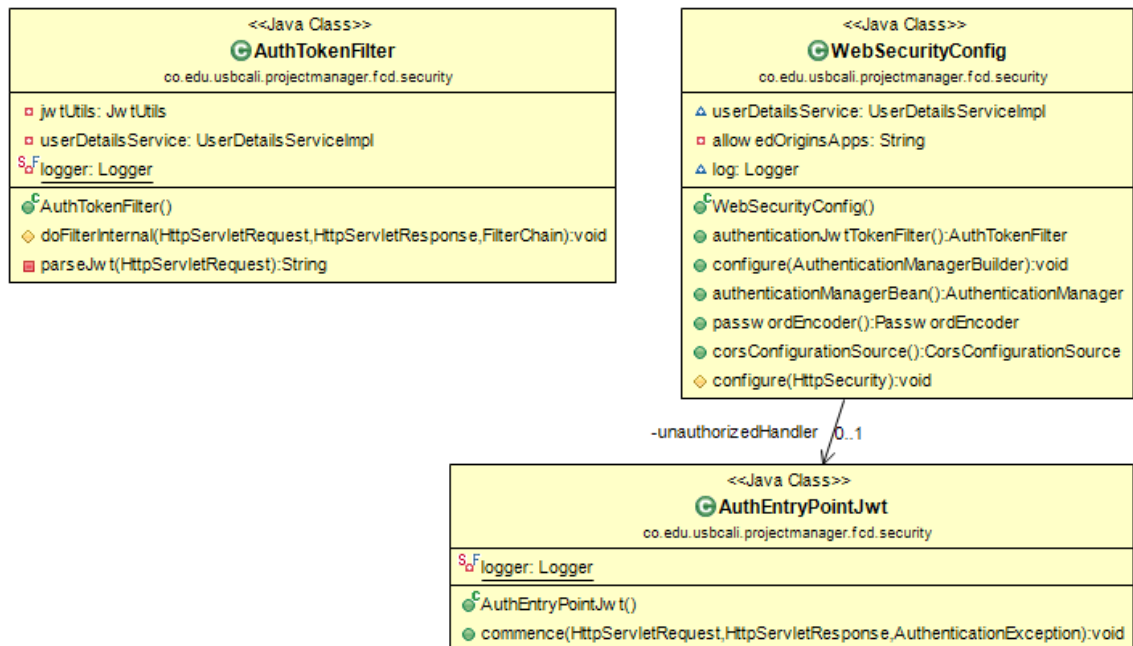


Ilustración 13: Diagrama de clases paquete de seguridad

Fuente: Extraído de Eclipse

Para finalizar con la capa del controller, se crearon las clases para exponer los servicios y que puedan ser consumidos. Ver Ilustración 14.

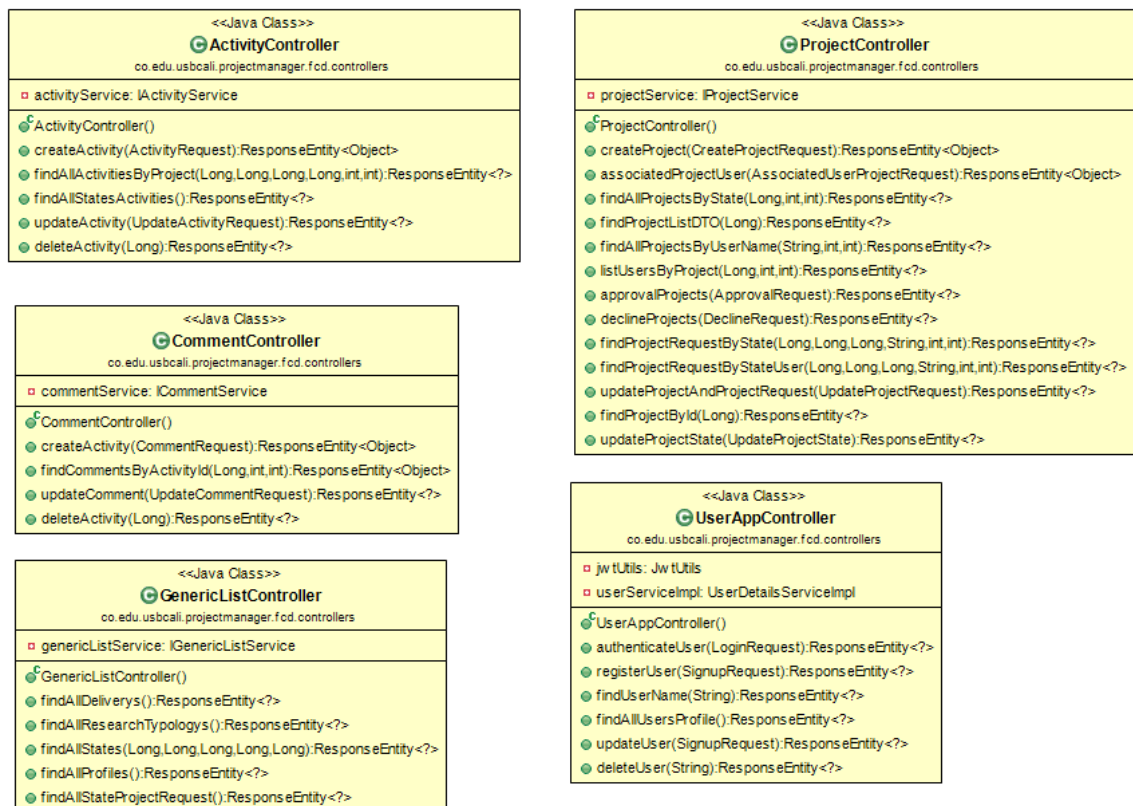


Ilustración 14: Diagrama de clases paquete de controller

Fuente: Extraído de Eclipse

3.4.4 Vistas de Desarrollo.

En la ilustración 15 que se muestra a continuación, representa en un diagrama de componentes, la forma en que cada módulo controlador interactúa con el sistema.

La interfaz de usuario, fue representada en el contenedor página de administrador de proyectos, cuya función es enviar las peticiones generadas por el usuario al servicio del backend.

- Controlador gestor de proyectos: Se encarga de la creación, listado y actualización de proyectos.
- Controlador gestor de usuarios: Recibe la petición para realizar el CRUD de usuario.
- Controlador gestor de actividad: Recibe la petición para realizar el CRUD de actividad.
- Controlador gestor de comentarios: Recibe la petición para realizar el CRUD de comentario.
- Controlador gestor de listas genéricas: Consulta las listas como estados de proyecto, perfil de usuario, entregables, tipología de investigación, estados de solicitudes de proyectos y usuarios con rol de director.
- Controlador asociar usuarios a proyectos: Su objetivo es la gestión para consultar y asociar usuarios a un proyecto.
- Controlador asociar/declinar proyectos: Su objetivo es aprobar/declinar los proyectos creados por estudiantes y profesores que se encuentran en estado SOLICITUD INICIAL. Si es aprobado el proyecto, cambia de estado a DISPONIBLE. En caso contrario que se declinado, cambia el estado del proyecto a RECHAZADO, con lo cual, los usuarios con rol de estudiante o profesor, tienen la posibilidad de hacer correcciones según el comentario del director, y volver a enviar la solicitud del proyecto, quedando nuevamente el proyecto en estado DISPONIBLE.
- Controlador de seguridad: Sus objetivos son validar los datos de usuarios como nombre de usuario y contraseña para iniciar sesión. También, valida la dirección de la cual se hace la petición al servicio de backend.

Como contenedor final, se encuentra la base de datos con un motor SQL postgres.

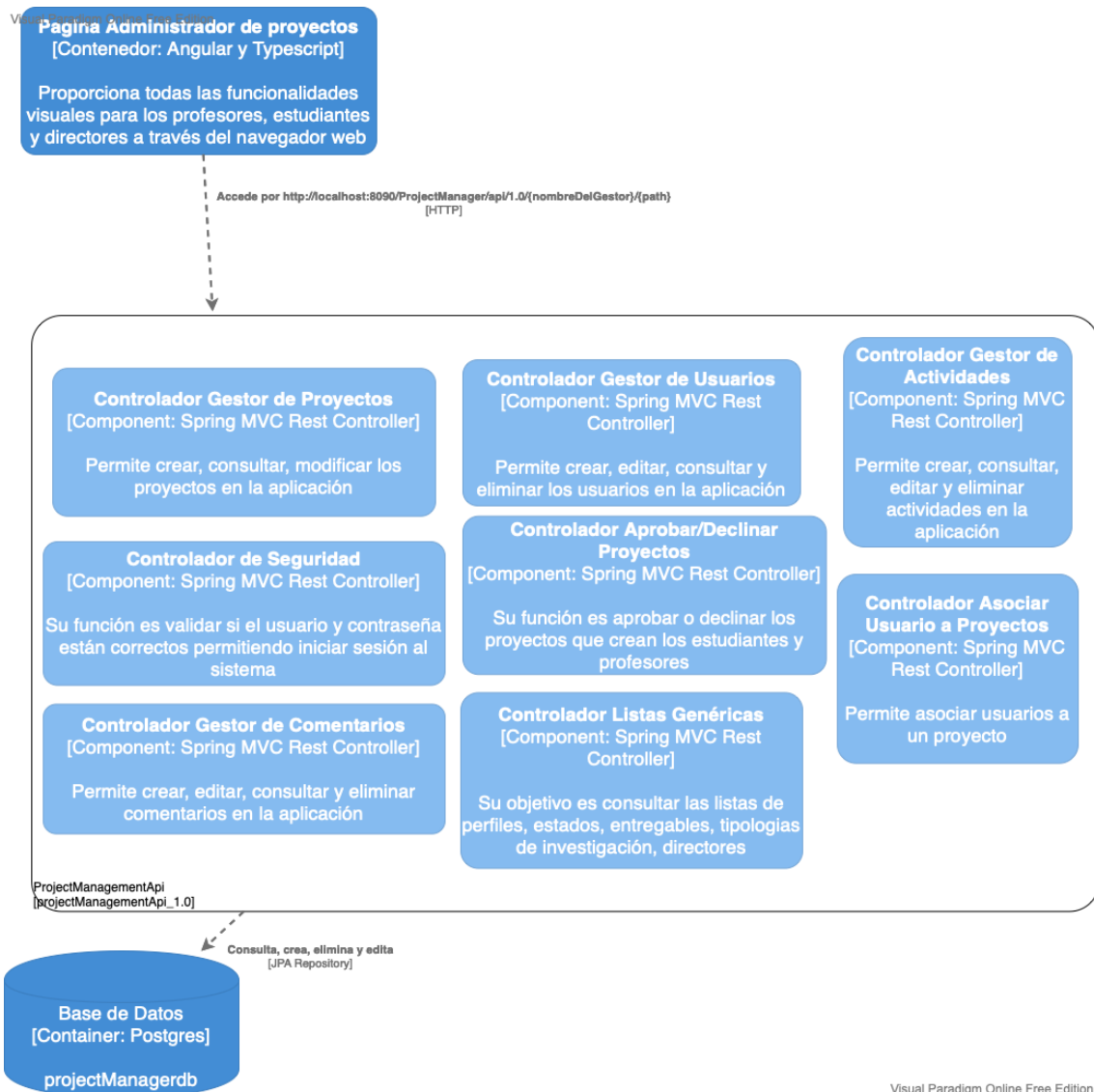


Ilustración 15: Diagrama de Componentes

Fuente: Elaboración propia

3.5 Patrones de Diseño Utilizados

Los patrones de diseño usados fueron:

- **Builder:** Se usó en el backend para la construcción de objetos en la clase project, activity y comments, ya que a través este patrón se reutiliza el código, evitando duplicidad en la implementación.
- **Observador:** Se utilizó en el frontend para hacer el llamado a los servicios mediante el subscribe permitiendo además, compartir datos entre los componentes.

3.6 Justificación de Diseño

El diseño implementado se adecuó a las tecnologías utilizadas con el fin de tener capas para cada nivel de desarrollo, dando como resultado la capa de controller, capa de business, capa model y la capa repository.

En la interfaz de usuario, el diseño se hizo basado en los prototipos de pantalla, donde lo que se quería lograr, es una interfaz que fuera usable e intuitiva para el usuario.

3.7 Desarrollo

3.7.1 Tecnologías utilizadas

A continuación se describen algunas de las tecnologías utilizadas:

- Java 11 como lenguaje de programación para el backend
- Spring boot 2.3.5
- Spring security 2.3.5
- Spring JPA 2.3.5
- Hibernate core
- Jwt 0.9
- Mockito 3.3.3
- Junit 4.13
- Bootstrap 5.1.3
- JQuery 3.6.0
- Primeicons 5.0.0
- Primeeng 13.0.0
- Ngx Pagination 5.1.1
- Ngidle
- Angular material 13.0.2
- Typescript como lenguaje de programación para el frontend
- Angular CLI 13.0.3
- Node: 16.13.0
- Git para el control de versiones.
- AWS CLI
- Postgres SQL
- Maven 3.8.1

3.7.2 Arquitectura tecnologías utilizadas

En la ilustración 16 que se muestra a continuación, se diseñó un diagrama de arquitectura para detallar cómo interactúan las tecnologías usadas en la implementación del frontend y el backend.

Cuando se accede al sitio del proyecto a través de un navegador web, invoca un servicio de AWS llamado EC2 (Elastic Compute Cloud) cuya función es gestionar las instancias y capacidad de la máquina en la nube. EC2 hace la petición al servicio de S3, donde se encuentra alojado el empaquetado del frontend y el backend. Para el despliegue del backend el servicio usado de AWS es Elastic Beanstalk, el cual almacena en el bucket de S3. Al ejecutar una acción en el project-management-frontend, por ejemplo, iniciar sesión, este hace el llamado al project-management-backend, el cual recibe la solicitud y valida en base de datos. La creación de base de datos se hizo en heroku, donde se usó Postgres SQL.

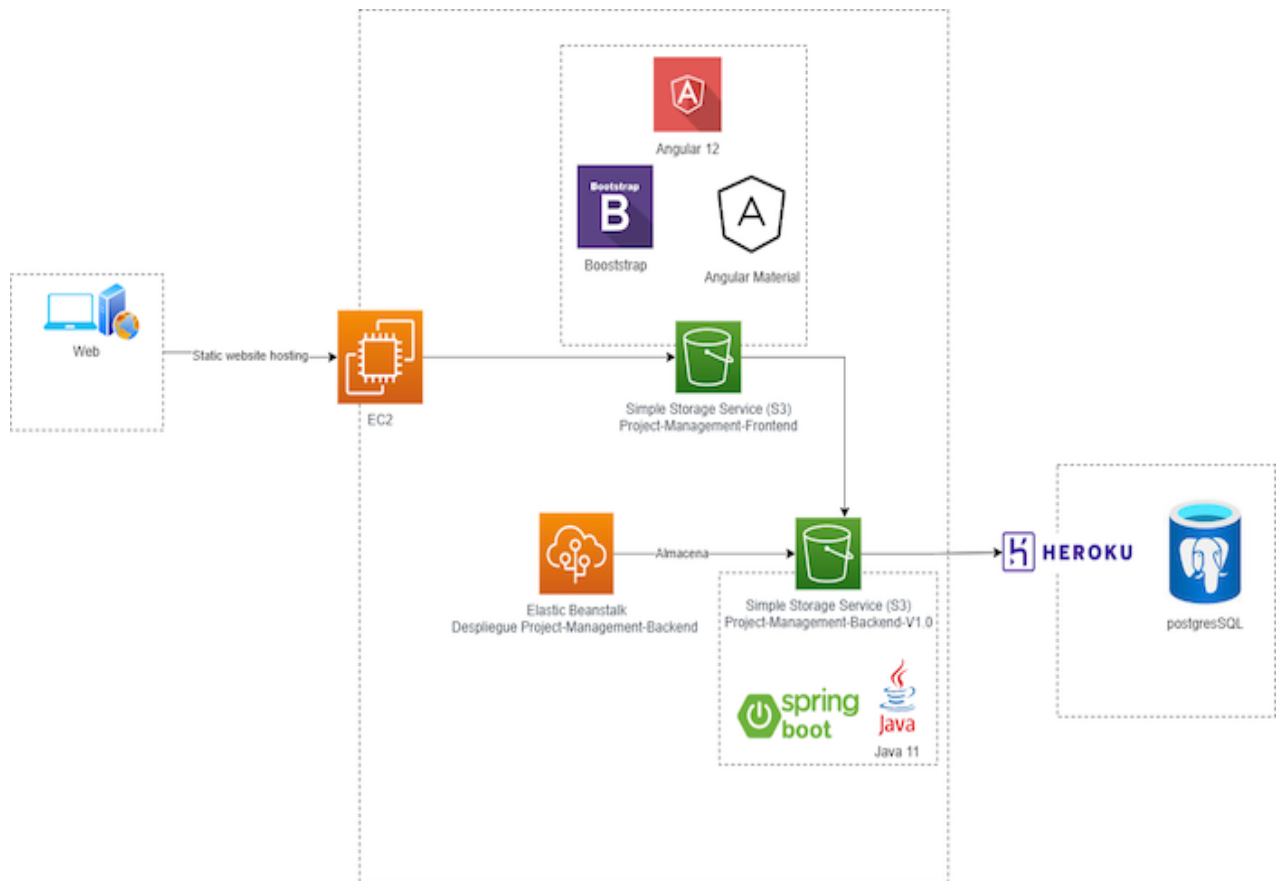


Ilustración 16: Diagrama de arquitectura tecnologías utilizadas

Fuente: Elaboración propia

3.7.3 Tecnologías utilizadas y su Relación con los Atributos de Calidad.

Para el aseguramiento de la calidad en lo que respecta a seguridad se implementó spring security y jwt, verificando los accesos a las funcionalidades con nombre de usuario y contraseña, verificando el tipo de peticiones, los tipos de métodos GET, PUT, POST, DELETE y verificando la dirección de la petición. Adicionalmente, se usaron interceptores que verifican un token para la comunicación entre backend y frontend. Esto asegura la calidad del sistema, brindando una aplicación confiable y segura.

En la usabilidad, se usaron tecnologías de estilos para las interfaces gráficas, tales como Bootstrap, primeicons para el uso de iconos que fueran dicientes. Como tecnología resaltada para la usabilidad, se usó ngidle lo cual cierra la sesión si la aplicación no se está usando.

En el rendimiento, la implementación de acceso de base de datos con Spring JPA, se llevó a cabo con la estrategia de realizar la menor cantidad de consultas a base de datos para Aumentar el rendimiento de la aplicación y concurrencia.

Las pruebas de unidad se aplicaron con junit y mockito, con lo que se asegura la independencia de las pruebas, logrando la ejecución de estas sin tener conexión a base de datos o servidores.

Para el despliegue, local se utilizó Docker y para el despliegue web AWS, con el objetivo de generar despliegues de baja complejidad.

3.8 Prototipos del Software

A continuación se evidencia los prototipos en cada iteración.

3.8.1 Desarrollo de prototipos Versión 1.0

Iteración 1 - Sketch No.1 – Login



UNIVERSIDAD DE
SAN BUENAVENTURA
CALI



APP PROYECTOS

Ingresar

Iniciar sesión

Usuario

Contraseña

Continuar

@ 2021 powered by USB CALI. All Rights Reserved

Escena que representa:	Interfaz inicio de sesión
Funcionalidades:	Login
Tipo Usuario:	Estudiante, profesor y director.

Iteración 1 - Sketch No.2 – Aprobar/declinar proyectos



UNIVERSIDAD DE
SAN BUENAVENTURA
CALI



APP PROYECTOS

Salir

Aprobación de proyectos

Gestión de Proyectos
Crear proyecto
Proyectos en curso
Proyectos disponibles
Mis proyectos

Gestión de estados
Aprobación de proyectos
Aceptación de participantes

Gestión de usuarios
Crear usuario
Editar usuario
Borrar usuario

Nombre del proyecto	Resumen de la propuesta	Director del proyecto	Creado por:	Objetivo general	Seleccionar ☐	
Proyecto 1	Proyecto que gestiona un banco de proyectos del grupo de investigación LIDIS	Rocio Segovia	Andrés Bastidas	Objetivo general	☐	Aprobar Declinar

Escena que representa:

Interfaz para aprobar o declinar proyectos

Funcionalidades:

Aprobar/declinar proyectos

Tipo Usuario:

Director

Iteración 1 - Sketch No.3 – Asociar usuario a proyecto



UNIVERSIDAD DE
SAN BUENAVENTURA
CALI



APP PROYECTOS

Salir

Asociar usuarios

Gestión de Proyectos
Crear proyecto
Proyectos en curso
Proyectos disponibles
Mis proyectos

Gestión de estados
Aprobación de proyectos
Aceptación de participantes
Asociar participantes

Gestión de usuarios
Crear usuario
Editar usuario
Borrar usuario

Buscar usuario

Buscar

Listado de proyectos

Proyecto 1

Asociar participante

Escena que representa:

Interfaz para asociar usuarios al proyecto

Funcionalidades:

Asociar usuarios a un proyecto

Tipo Usuario:	Director
----------------------	----------

Iteración 1 - Sketch No.4 – Crear proyecto



Crear Proyecto

Gestión de Proyectos
 Crear proyecto
 Proyectos en curso
 Proyectos disponibles

Gestión de estados
 Aprobación de proyectos

Gestión de usuarios
 Crear usuario
 Editar usuario
 Borrar usuario

Formulario creación de Proyecto

Titulo del proyecto* Director del proyecto

Problema u objeto de investigación* Justificación

Objetivo general* Fecha inicio

Objetivo específicos* Fecha fin

Resumen de la propuesta* Duración

Metodología del proyecto*

Entregables

@ 2021 powered by USB CALI. All Rights Reserved

Escena que representa:	Interfaz para crear proyectos
Funcionalidades:	Crear proyecto
Tipo Usuario:	Estudiante, Profesor y Director

Iteración 1 - Sketch No.5 – Proyectos asignados



UNIVERSIDAD DE
SAN BUENAVENTURA
CALI



APP PROYECTOS

Salir

Mis proyectos

Gestión de Proyectos

Mis proyectos
Proyectos disponibles
Proyectos aprobados

Nombre del proyecto	Estado	Fecha inicio	Fecha fin	Actividades	Director del proyecto	Actualizar proyecto	Generar informe
Proyecto 1	EN PROGRESO ▾	2021-09-04	2022-02-04	Gestionar actividades	Rocio Segovia	Actualizar proyecto	Generar informe

Escena que representa:	Interfaz para visualizar los proyectos asignados
Funcionalidades:	Proyectos asignados
Tipo Usuario:	Estudiante, Profesor y Director

3.8.2 Desarrollo de prototipos Versión 2.0

Iteración 2- Sketch No.5 – Proyectos asignados



UNIVERSIDAD DE
SAN BUENAVENTURA
CALI



APP PROYECTOS

Salir

Mis proyectos

Gestión de Proyectos

Mis proyectos
Proyectos disponibles
Proyectos aprobados

Nombre del proyecto	Estado	Fecha inicio	Fecha fin	Actividades	Director del proyecto	Actualizar proyecto
Proyecto 1	EN PROGRESO ▾	2021-09-04	2022-02-04	Gestionar actividades	Rocio Segovia	Actualizar proyecto

Escena que representa:	Interfaz para visualizar los proyectos asignados
Funcionalidades:	Proyectos asignados
Tipo Usuario:	Estudiante, Profesor y Director

3.9 Pruebas y Evaluación de Software

Describir el proceso de pruebas realizado, especificando técnicas y herramientas utilizadas. Las pruebas realizadas fueron pruebas funcionales en las que se probaron las diferentes funcionalidades del software.

3.9.1 Parámetros de Pruebas

- a. Alcance técnico – Ficha técnica del producto

Browser:	Chrome 100.0.4896.60 (Official Build) (x86_64) en Adelante, Safari 15.2 (15612.3.6.1.8, 15612) en Adelante.
Sistema Operativo	Windows 10, macOS Catalina 10.15
Servidor Web	Nginx
Lenguaje de desarrollo	Java 11
Procesador	Intel Core i5 dual core 2.5 GHz
Disco Duro	SSD 500gb – 175gb disponibles
Memoria Ram	16GB 1600 mhz

Tabla 5: Ficha técnica del producto para la ejecución de pruebas

Fuente: Elaboración propia

b. Estrategia de pruebas

Tipos de Pruebas	Descripción
Prueba de humo	Prueba que identifica si es posible comenzar a probar
Pruebas funcionales	Casos positivos y negativos que indicarán si la aplicación cumple con las funcionalidades.
Pruebas de rendimiento	Se realizarán pruebas de rendimiento con Jmeter para evaluar el número de usuario y peticiones que soporta la aplicación.
Prueba piloto	Se realizarán pruebas piloto con el director del LIDIS

Tabla 6: Tipos de pruebas

Fuente: Elaboración propia

3.9.2 Casos de Prueba

i. Pruebas funcionales

En la tabla 7 se muestra el diseño de casos de prueba.

ID	Descripción	Descripción Caso Prueba	Guion	Resultados Esperados	Tipo de Prueba
1.1	Crear proyecto por un DIRECTOR	Realizar la creación del proyecto como director Cuando: Funcionalidad: Crear proyecto. Restricciones: El usuario debe estar logueado en el sistema como DIRECTOR.	- Ingresar a la funcionalidad Crear proyecto - Llenar todos los campos obligatorios para habilitar botón. - Clic en crear proyecto	Resultado esperado: 1- El sistema debe mostrar el mensaje: Proyecto creado con éxito. Nota: Si su perfil es estudiante o profesor, el proyecto deberá ser aprobado por su director y limpiar el formulario	+
1.2	Crear proyecto por un ESTUDIANTE o PROFESOR	Realizar la creación del proyecto como Profesor o estudiante Cuando: Funcionalidad: Crear proyecto Valores de variables en el archivo: alfanumérico. Restricciones: El usuario debe estar logueado en el sistema como PROFESOR o	- Ingresar a la funcionalidad Crear proyecto - Llenar todos los campos obligatorios para habilitar botón. - Clic en crear proyecto	Resultado esperado: 1- El sistema debe mostrar el mensaje: Proyecto creado con éxito. Nota: Si su perfil es estudiante o profesor, el proyecto deberá ser aprobado por su director y limpiar el formulario	+



		ESTUDIANTE.			
1.3	Listar todos los proyectos	Listar todos los proyectos Cuando: Funcionalidad: Listar todos los proyectos. Restricciones: El usuario debe estar logueado en el sistema como PROFESOR, ESTUDIANTE o DIRECTOR.	1- Ingresar a la funcionalidad Listado de todos los proyectos	Resultado esperado: 1- El sistema debe mostrar el listado de todos los proyectos	+
1.4	Listado de proyectos asignados	Listar todos los proyectos asignados al usuario Cuando: Funcionalidad: Proyectos asignados. Restricciones: - El usuario debe estar logueado en el sistema como PROFESOR, ESTUDIANTE o DIRECTOR. - El usuario debe tener proyectos asignados	1- Ingresar a la funcionalidad proyectos asignados	Resultado esperado: 1- El sistema debe mostrar todos los proyectos asignados que tiene el usuario.	+
1.5	Listado sin proyectos asignados	Cargar la funcionalidad de proyectos asignados Cuando: Funcionalidad: Proyectos asignados. Restricciones: - El usuario debe estar logueado en el sistema como PROFESOR, ESTUDIANTE o DIRECTOR. - El usuario no debe tener proyectos asignados	1- Ingresar a la funcionalidad proyectos asignados	Resultado esperado: 1- El sistema debe mostrar el mensaje No se encontraron proyectos	-



2.1	Listar solicitudes de proyectos para un ESTUDIANTE o PROFESOR	Cuando: Funcionalidad: Listar solicitudes de proyectos Restricciones: - El usuario debe estar logueado en el sistema como PROFESOR o ESTUDIANTE, - El usuario debe tener solicitudes de proyectos	1- Ingresar a la funcionalidad Mis Proyectos.	Resultado esperado: 1- El sistema debe mostrar todas las solicitudes de proyecto, adicionalmente seleccionar el estado de la solicitud ya sea APROBADA, RECHAZADA o PENDIENTE.	+
2.2	Aprobar proyecto	Cuando: Funcionalidad: Aprobar proyectos Restricciones: - El usuario debe estar logueado en el sistema como DIRECTOR, - El usuario debe tener solicitudes de proyectos	1- Ingresar a la funcionalidad Aprobar proyectos 2- Seleccionar uno o más proyectos en la lista 3- Clic en la funcionalidad Aprobar	Resultado esperado: 1- El sistema debe mostrar el mensaje Proyectos Aprobados 2- El sistema debe asociar el proyecto aprobado al usuario que realizó la solicitud.	+
2.3	Rechazar proyectos	Cuando: Funcionalidad: Rechazar proyectos. Restricciones: - El usuario debe estar logueado en el sistema como DIRECTOR, - El usuario debe tener solicitudes de proyectos	1- Ingresar a la funcionalidad Aprobar proyectos. 2- Seleccionar uno o más proyectos de la lista. 3- Clic en la funcionalidad rechazar	Resultado esperado: 1- El sistema debe mostrar el mensaje Proyectos rechazados 2- El sistema no debe mostrar el proyecto en proyectos asignados.	+
3.1	Iniciar sesión	Cuando: Funcionalidad: Login Restricciones: El usuario debe estar	1- Digitar usuario y contraseña.	Resultado esperado: 1- El sistema debe dirigirlo a la funcionalidad de crear proyecto.	+



		registrado en el sistema			
3.2	Crear usuario en el sistema	Cuando: Funcionalidad: Crear usuario Restricciones: - El usuario no debe estar registrado en el sistema - El usuario debe estar logueado como DIRECTOR	- Ingresar a la funcionalidad Crear Usuario - Digitar todos los campos obligatorios - Seleccionar la funcionalidad Crear usuario	Resultado esperado: El sistema debe mostrar el mensaje Usuario Creado.	+
3.3	Actualizar usuario	Cuando: Funcionalidad: Actualizar usuario Restricciones: El usuario debe estar logueado como DIRECTOR	- Ingresar a la funcionalidad Editar/Borrar Usuario - Buscar por nombre de usuario - Digitar el campo que se desea cambiar - Presionar en la funcionalidad Actualizar	Resultado esperado: El sistema debe mostrar el mensaje Usuario Actualizado.	+
3.4	Eliminar Usuario	Cuando: Funcionalidad: Eliminar usuario Restricciones: El usuario debe estar logueado como DIRECTOR	- Ingresar a la funcionalidad Editar/Borrar Usuario - Buscar por nombre de usuario - Digitar el campo que se desea cambiar - Presionar en la funcionalidad Eliminar	Resultado esperado: El sistema debe mostrar el mensaje Usuario Eliminado.	+
4.1	Crear actividad	Cuando: Funcionalidad: Eliminar usuario Restricciones: - El usuario debe estar logueado como DIRECTOR, PROFESOR o ESTUDIANTE - El usuario debe tener	- Ingresar a la funcionalidad Proyectos asignados. - Seleccionar el proyecto - Seleccionar la funcionalidad Gestionar Actividades - Selecciona funcionalidad crea actividad - Llenar todos los campos obligatorios - Seleccionar la funcionalidad Crear Actividad	Resultado esperado: El sistema debe mostrar el mensaje Actividad creada.	+

4.2		proyectos asociados			
	Listar todas las actividades de un proyecto	Cuando: Funcionalidad: Listar Actividades Restricciones: - El usuario debe estar logueado como DIRECTOR, PROFESOR o ESTUDIANTE - El usuario debe tener proyectos asociados	- Ingresar a la funcionalidad Proyectos asignados. - Seleccionar el proyecto - Seleccionar la funcionalidad Gestionar Actividades	Resultado esperado: El sistema debe listar todas las actividades del proyecto	+
	Editar actividad	Cuando: Funcionalidad: Actualizar actividad Restricciones: - El usuario debe estar logueado como DIRECTOR, PROFESOR o ESTUDIANTE - El usuario debe tener proyectos asociados	- Ingresar a la funcionalidad Proyectos asignados. - Seleccionar el proyecto - Seleccionar la funcionalidad Gestionar Actividades - Seleccionar la funcionalidad actualizar actividad en una de las actividades.	Resultado esperado: - El sistema debe mostrar el mensaje actividad actualizada. - El sistema de actualizar el estado de la actividad. Si el estado de la actividad es CREADA debe cambiarlo a EN PROGRESO. Si el estado de la actividad es EN PROGRESO el sistema debe cambiarlo a TERMINADA	+
	Eliminar actividad	Cuando: Funcionalidad: Eliminar actividad Restricciones: - El usuario debe estar logueado como DIRECTOR, PROFESOR o ESTUDIANTE. - La actividad no debe tener comentarios asociados.	- Ingresar a la funcionalidad Proyectos asignados. - Seleccionar el proyecto - Seleccionar la funcionalidad Gestionar Actividades - Seleccionar la funcionalidad eliminar actividad en una de las actividades.	Resultado esperado: El sistema debe mostrar el mensaje Actividad eliminada.	+

5.1		El usuario debe tener proyectos asociados			
	Crear comentario	Cuando: Funcionalidad: crear comentario Restricciones: - El usuario debe estar logueado como DIRECTOR, PROFESOR o ESTUDIANTE. - El usuario debe tener proyectos asociados - El proyecto debe tener actividades creadas	- Ingresar a la funcionalidad Proyectos asignados. - Seleccionar el proyecto - Seleccionar la funcionalidad Gestionar Actividades - Seleccionar la funcionalidad comentarios - Seleccionar la funcionalidad crear comentario - Digitar los campos obligatorios - Seleccionar la funcionalidad crear comentario	Resultado esperado: El sistema debe mostrar el mensaje Comentario Creado	+
	Listar comentarios	Cuando: Funcionalidad: histórico comentarios Restricciones: - El usuario debe estar logueado como DIRECTOR, PROFESOR o ESTUDIANTE. - El usuario debe tener proyectos asociados - El proyecto debe tener actividades creadas	- Ingresar a la funcionalidad Proyectos asignados. - Seleccionar el proyecto - Seleccionar la funcionalidad Gestionar Actividades - Seleccionar la funcionalidad comentarios	Resultado esperado: El sistema debe listar todos los comentarios de la actividad.	+
	Editar comentario	Cuando: Funcionalidad: Editar comentario Restricciones: El usuario debe estar logueado como	- Ingresar a la funcionalidad Proyectos asignados. - Seleccionar el proyecto - Seleccionar la funcionalidad Gestionar Actividades	Resultado esperado: El sistema debe mostrar el mensaje comentario actualizar y redirigirlo al listado de comentarios	+

5.4		DIRECTOR, PROFESOR o ESTUDIANTE. - El usuario debe tener proyectos asociados - El proyecto debe tener actividades creadas	4- Seleccionar la funcionalidad comentarios 5- Seleccionar la funcionalidad actualizar comentario. 6- Digital el campo que sea desea actualizar		
	Eliminar comentario	Cuando: Funcionalidad: Eliminar comentario Restricciones: - El usuario debe estar logueado como DIRECTOR, PROFESOR o ESTUDIANTE. - El usuario debe tener proyectos asociados - El proyecto debe tener actividades creadas	1- Ingresar a la funcionalidad Proyectos asignados. 2- Seleccionar el proyecto 3- Seleccionar la funcionalidad Gestionar Actividades 4- Seleccionar la funcionalidad comentarios 5- Seleccionar la funcionalidad eliminar comentario.	Resultado esperado: El sistema debe mostrar el mensaje comentario eliminado.	+

Tabla 7: Matriz de casos de pruebas

Fuente: Elaboración propia

ii. Pruebas de Rendimiento

Para el diseño y ejecución de las pruebas de rendimiento, se eligieron a criterio los siguientes servicios implementados:

- Login
- Crear proyecto
- Listado de proyectos asignados
- Listado de aprobar proyectos
- Listado de todos los proyectos filtrados por estado

a. Diseño de pruebas de rendimiento.

En la siguiente tabla, se puede observar el diseño para la ejecución de las pruebas de rendimiento.

Nombre del servicio	Número de usuarios concurrentes prueba #1	Número de usuarios concurrentes prueba #2	Número de usuarios concurrentes prueba #3	Número de usuarios concurrentes prueba #4	Intervalo de tiempo entre cada ejecución en segundos	Número de ejecuciones por usuario
Login	10	50	150	300	1	1
Crear proyecto	10	50	150	300	1	1
Listado de proyectos asignados	10	50	150	300	1	1
Listado de aprobar proyectos	10	50	150	300	1	1
Listado de todos los proyectos filtrados por estado	10	50	150	300	1	1

Tabla 8: Tabla diseño de pruebas de rendimiento

Fuente: Elaboración propia

b. Ejecución pruebas de rendimiento

Resultado Prueba de rendimiento # 1:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de By...
HTTP Requ...	50	466	43	2009	490.46	0.00%	13.4/sec	78.34	7.88	6005.0
Total	50	466	43	2009	490.46	0.00%	13.4/sec	78.34	7.88	6005.0

Tabla 9: Tabla resultados prueba de rendimiento 1

Fuente: Jmeter

Resultado Prueba de rendimiento # 2:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de By...
HTTP Requ...	300	1145	40	5536	1194.09	0.00%	6.3/sec	36.76	3.70	6005.0
Total	300	1145	40	5536	1194.09	0.00%	6.3/sec	36.76	3.70	6005.0

Tabla 10: Tabla resultados prueba de rendimiento 2

Fuente: Jmeter

Resultado Prueba de rendimiento # 3:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de By...
HTTP Requ...	1050	2319	22	9903	1968.27	0.00%	11.4/sec	67.02	6.74	6005.0
Total	1050	2319	22	9903	1968.27	0.00%	11.4/sec	67.02	6.74	6005.0

Tabla 11: Tabla resultados prueba de rendimiento 3

Fuente: Jmeter

Resultado Prueba de rendimiento # 4:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de By...
HTTP Requ...	2550	4078	14	18719	3579.18	0.00%	18.5/sec	108.62	10.93	6005.0
Total	2550	4078	14	18719	3579.18	0.00%	18.5/sec	108.62	10.93	6005.0

Tabla 12: Tabla resultados prueba de rendimiento 4

Fuente: Jmeter

Para la ejecución de las pruebas se realizó el despliegue de la aplicación en docker en el ambiente local, siendo clave los recursos de la maquina en donde se realizó la ejecución de las pruebas.

Como parte de las conclusiones, se puede observar en la columna % Error en todas las pruebas fue de 0%, demostrando como se mencionó anteriormente, que los recursos del sistema, influyen para que la aplicación pueda funcionar en casos donde la carga sea alta. El objetivo de las pruebas de rendimiento era generar el error al aumentar el número de peticiones, seguramente si se realizan las prueba en un ambiente que no sea controlado como en local, podría afectar el rendimiento y posiblemente generar errores.

3.9.3 Resumen Pruebas Funcionales

Iteración	Nro. de casos de prueba	Ejecutados		OK		NOK		PDTE		Anuladas		Correctitud	Densidad NC	Confiabilidad	Cobertura
i	20	20	100%	17	85%	3	15%	0	0%	0	0%	85%	15%	85%	100%
ii	5	4	80%	4	80%	0	0%	1	20%	0	0%	100%	0%	100%	80%
Total	25	24	96%	21	84%	0	0%	1	4%	0	0%	88%	0%	100%	96%

Tabla 13: Tabla resultados pruebas funcionales

Fuente: Elaboración propia

Del ciclo 1 se ejecutaron 20 casos, dando como resultado en el ciclo 1 correctitud 85%, confiabilidad del 85%, una cobertura del 100% y tres hallazgos reportados con una densidad de NCs del 15% .

En el ciclo 2, se ejecutaron cinco casos de prueba, los cuales se solucionaron dos NCs del ciclo 1. Un caso de prueba quedó pendiente debido a la estimación y alcance de las pruebas.

Para mayor detalle de las tres no conformidades, correspondieron a dos de la capa de presentación y una funcional, como se muestra en la siguiente tabla:

No.	CP-No.	Descripción	Estado	Severidad	Origen
1	1.1	Al seleccionar un proyecto para su aprobación o rechazo, la lista de checkbox cuando está habilitada no se muestra deshabilitada si el estado de solicitud cambia RECHAZADA,	Pendiente	Presentación	Aprobación de proyectos

2	1.2	En las tablas de listado de proyectos, no se adapta de acuerdo con el tamaño del texto, generando un desborde del texto.	Resuelta	Presentación	Aprobación de proyectos, ver información de proyecto Proyectos asignados
3	1.3	Al consultar proyectos asignados, se visualizaban proyectos que no pertenecían al usuario	Resuelta	Funcional	Consultar proyectos asignados

Tabla 14: Tabla de no conformidades

Fuente: Elaboración propia

3.9.4 Pruebas Piloto del Proyecto

3.9.4.1.1 Escenario de la prueba piloto

Inicialmente la prueba piloto se planteó para ejecutarse durante un mes con el grupo de investigación LIDIS, pero debido a los obstáculos de tiempo y correcciones en la implementación, se realizó una prueba piloto con el director del grupo LIDIS. El objetivo de la prueba es validar las funcionalidades y recibir la retroalimentación del grupo que hará uso de la aplicación. Para ejecutar la prueba, se crearon dos usuarios, uno con el perfil de ESTUDIANTE y otro con el de DIRECTOR, donde se validó por cada rol las funcionalidades que se muestran en la siguiente tabla:

Funcionalidades	Rol
Crear proyecto	Estudiante, Director
Crear actividades	Estudiante, Director
Crear comentario	Estudiante, Director
Editar comentario	Estudiante, Director
Crear usuario	Director
Editar usuario	Director
Asociar usuario a proyecto	Director
Aprobar proyecto	Director
Rechazar proyecto	Director
Listar todos los proyectos	Estudiante, Director

Tabla 15: Tabla funcionalidades prueba piloto

Fuente: Elaboración propia

3.9.4.1.2 Resultados obtenidos de la prueba piloto

Como resultado de la prueba piloto, se obtuvieron comentarios para trabajos futuros y correcciones de posibles NCs, en la siguiente tabla se relacionan los comentarios obtenidos:

Funcionalidad	Comentarios sobre la funcionalidad
Crear proyecto	En la parte de la selección de listas de entregables para la funcionalidad de

	crear proyecto, se realizó la aclaración de que la lista era dinámica, es decir, podría modificarse en base de datos, sin necesidad de realizar implementación.
Texto de Proyectos aprobados en la interfaz para rol de estudiante	En el menú para el rol de estudiante o profesor, la funcionalidad de ver el listado de proyectos aprobados, se visualiza como Proyectos Aprobados, pero con el análisis de la prueba piloto se cambió el nombre a Mis Proyectos para que el texto sea diciente y no genere confusiones para el usuario
Rol de Profesor en la aplicación	El rol de profesor, cuenta con los mismos accesos y permisos que el rol de estudiante, en esta prueba piloto se realizó el análisis del objetivo para este rol, cuya función es servir de acompañamiento al estudiante en sus proyectos.

Tabla 16: Tabla resultados prueba piloto

Fuente: Elaboración propia

3.10 Acceso a la aplicación

El despliegue de la aplicación, se hizo en AWS, esto para pruebas y entregas del proyecto, se espera que el despliegue final se haga en servidores de la universidad dejando indicado como hacer este proceso.

Puede acceder a la aplicación a través del siguiente enlace:

<http://project-manager-frontend-usb.s3-website-us-east-1.amazonaws.com/projectManagement#/login>

Nombre de usuario: usuario.director

Contraseña: Usb2022*

Rol: DIRECTOR

Nombre de usuario: usuario.estudiante

Contraseña: Usb2022*

Rol: ESTUDIANTE

3.11 Estándares de ingeniería empleados durante el proyecto

- Aplicación del estándar internacional para especificación de requisitos de software IEEE 29148 2011.
- Implementación de patrones de diseño en la fase de desarrollo.
- Se puso en práctica algunos de los principios SOLID.
- Aplicación de integración continua para los despliegues.
- Para el modelo de diagramas, se diseñaron con lenguaje UML.



4. ANÁLISIS DE RESULTADOS

- 4.1 Análisis de impactos
- 4.2 Utilización de herramientas
- 4.3 Análisis de diseño de componentes
- 4.4 Cumplimiento de objetivos
- 4.5 Conclusiones
- 4.6 Experiencia de diseño en ingeniería de software
- 4.7 Trabajos futuros

4. ANÁLISIS DE RESULTADOS

4.1 Análisis de Impactos

La propuesta de ingeniería discutida en el presente documento tiene diferentes impactos en el escenario en el cual se contextualiza el problema. Estos impactos pueden ser analizados y discutidos desde las siguientes dimensiones o perspectivas: ambiental, social, económica, técnica, e individual [1].

- Social: Aporta a la comunidad científica de la universidad San Buenaventura Cali generando interacciones en una plataforma unificada para profesores y estudiantes. Sumado a esto, los nuevos estudiantes tendrán la posibilidad de extraer conocimiento de proyectos que ya se hayan trabajado en dicha comunidad.
- Económico: Representa un ahorro en la contratación de gerente de proyecto que posiblemente no esté enfocado en el área científica, adicionalmente permitiendo la mejora del aplicativo desde un semillero de investigación.
- Técnica: El proyecto cuenta con almacenamiento de datos permitiendo la durabilidad de la información. A partir de lo anterior para trabajos futuros relacionados con esta aplicación se podrá usar la información almacenada para la generación de reportes de indicadores.
- Individual: Como individuo proporciona una ruta ordenada para proyectos científicos que a largo plazo contribuye a la paz mental y emocional del individuo. Facilitar y centralizar la información de los proyectos. Adicionalmente, como individuo adquiere conocimiento, tomando como base, otros proyectos que ya se hayan creado en la aplicación.

4.2 Utilización de herramientas

- Visual paradigm community edition 16.3
- Justmind 9.7.2
- Visual Studio Code
- Postman
- Smartgit
- Eclipse IDE for enterprise java and web developers
- PgAdmin 6.1
- Jmeter
- Heroku
- Docker para el despliegue de la aplicación en local
- Justmind para los prototipos de pantalla

4.3 Análisis de diseño de componentes

En el proceso de ingeniería de software se utilizaron las vistas de alto nivel, vista de procesos, vista lógica y vista de desarrollo.

- Vista de alto nivel: Modelo de diagrama C4 para entender como los usuarios interactúan con el sistema.
- Vista de procesos: Representado en un diagrama actividades para entender el flujo en la creación de un proyecto y el flujo de iniciar sesión en la aplicación.
- Vista lógica: Usando los diagramas de clases se logró la descomposición de los módulos del sistema con sus respectivos atributos.
- Vista desarrollo: Es representada mediante el diagrama de componentes para descomponer el contenedor de backend entendiendo los módulos de la aplicación.

4.4 Cumplimiento de Objetivos

- Especificar los requisitos que definan el alcance de la aplicación del banco de proyectos: En la elaboración del SRS se especificaron todos los requisitos que están dentro del alcance de la aplicación.
- Diseñar una solución acorde con la especificación de requisitos: Para el diseño de la solución, se hicieron los prototipos de pantalla, adicionalmente, las entrevistas con un profesor del programa ingeniería de sistemas de la universidad San Buenaventura Cali, lograron contribuir para la creación y modificación de prototipos de pantalla, así como para el documento SRS.
- Implementar la solución basada en el diseño: Para el cumplimiento de este objetivo, se tuvieron en cuenta los prototipos de pantalla realizados en la fase del diseño, donde en las iteraciones 4 y 5 véase 3.1.1 se obtuvo el primer prototipo funcional de la aplicación, que dio como resultado, una mejora respecto a los comentarios generados en estas iteraciones.
- Validar la herramienta con el grupo de investigación LIDIS: Se realizó la prueba piloto con el director del grupo de investigación LIDIS.

4.5 Conclusiones

- El implementar una aplicación que hiciera seguimiento y trazabilidad a los proyectos del grupo de investigación LIDIS, permitirá a los profesores y estudiantes interactuar de forma colaborativa, aportando nuevas ideas para futuros proyectos y tendrán la posibilidad de tener un repositorio de información organizado.
- Surgieron ideas para el mejoramiento de la aplicación, generando expectativas sobre el acoplamiento con otros sistemas.
- La implementación cuenta con tecnologías y herramientas que actualmente se usan en el mercado de desarrollo de software, puede generar conocimiento y directrices de aprendizaje.
- Para futuros despliegues se recomienda usar servidores en la nube como AWS, ya que se evidenció que los servidores de la universidad San Buenaventura Cali quedan obsoletos para incorporar nuevas tecnologías.

4.6 Experiencia de diseño en ingeniería de software

Lo presentado en las secciones de Análisis de Impactos (ver sección [4.1](#)), utilización de herramientas (ver sección [4.2](#)) y Análisis de diseño de componentes (ver sección [4.3](#)) se argumenta como una evidencia de que, a lo largo del proceso de formación de la carrera de ingeniería de sistemas, que culmina con esta experiencia de proyecto de grado, se lograron alcanzar los siguientes resultados de aprendizaje:

- Capacidad para identificar, formular y resolver problemas complejos de ingeniería mediante la aplicación de principios de ingeniería, ciencia y matemáticas.
- Capacidad para aplicar diseño de ingeniería en la producción de soluciones que satisfagan necesidades específicas, considerando tanto la salud pública, la seguridad

y el bienestar, como factores de carácter global, cultural, social, ambiental y económico.

- Capacidad para reconocer responsabilidades éticas y profesionales en el ejercicio de la ingeniería y tomar decisiones de manera informada sobre soluciones de ingeniería considerando contextos de carácter global, económico, ambiental y social.
- Capacidad para planear y realizar experimentación adecuada, analizar e interpretar datos, y aplicar juicio de ingeniería en la obtención de conclusiones.

4.7 Trabajos Futuros

Como trabajos futuros se proponen:

- Implementar la generación de reportes tomando como base la información ya registrada de cada proyecto.
- Implementar la funcionalidad para adjuntar enlaces para cada actividad.
- Mejorar la interfaz para que la tabla se comporte de manera responsiva.
- Mejorar los campos de texto en la creación de proyecto para que sean más grandes.
- Implementar un panel principal donde se puedan ver los proyectos y sus estados.
- Desarrollar un api restfull adicional para la consulta de datos y que la aplicación solo reciba esa información.
- Realizar pruebas automatizadas para el frontend.
- Optimizar código de la implementación.



5. REFERENCIAS

5. REFERENCIAS

- [1] Universidad del Valle, "Objetivos grupo de investigación," 2022. <http://gyepro.univalle.edu.co/mision.html> (accessed Mar. 26, 2022).
- [2] uniandes, "Quienes somos," 2022. <https://ingeco.uniandes.edu.co/index.php/quienes-somos/objetivos1> (accessed Mar. 26, 2022).
- [3] Universidad de San Buenaventura Cali, "Semilleros de investigación." <https://www.usbcali.edu.co/node/5841> (accessed Mar. 26, 2022).
- [4] M. A. Rosario Villarreal, Y. Ocaña Fernández, C. H. Capillo Chávez, A. B. Lavado Rodríguez, M. el Homrani, and S. M. Arias Romero, "Factores que inciden en la gestión de proyectos de investigación científica," *Apuntes Universitarios*, vol. 9, no. 1, May 2019, doi: 10.17162/au.v1i1.349.
- [5] A. Rodríguez Fuentes and M. J. Caurcel Cara, "Aproximación cualitativa del escudriño en Psicología educativa," *Propósitos y Representaciones*, vol. 7, no. 1, p. 01, Mar. 2019, doi: 10.20511/pyr2019.v7n1.301.
- [6] J. Fernando, M. Molina Tutor, A. Eilen, and V. Jaimes, "BANCO DE PROYECTOS DE TRABAJO DE GRADO PARA LA FACULTAD DE INGENIERIA Autores NELSON MARTÍNEZ ÁLVAREZ," 2019.
- [7] A. E. OVIEDO B and J. J. SIERRA A, "Aplicación web de inscripciones y seguimientos de proyecto de grado, proyectos integradores y prácticas empresariales, para el portal web del programa de ingeniería de sistemas," 2007.
- [8] D. P. HERRERA OCAMPO and C. A. OLARTE CAMPO, "DISEÑO DE UNA HERRAMIENTA MODULAR QUE FACILITE LA IMPLEMENTACIÓN DEL BANCO DE PROYECTOS DE LA FUNDACION RED DE APOYO SOCIAL DE ANTIOQUIA – RASA," 2015.
- [9] Á. M. Sánchez Gómez, "Guía para la gestión de proyectos de investigación de la Universidad EAFIT," 2015.
- [10] S. Núñez González *et al.*, "PROPOSAL OF ACTIONS FOR THE DESIGN AND IMPLEMENTATION OF THE PROJECT MANAGEMENT PROCEDURE AT THE UNIVERSITY OF PINAR DEL RÍO, CUBA," 2021.
- [11] J. M. Hernández, D. A. Franco, and R. J. Martelo, "Sistema de información de la producción científica de una institución universitaria," 2020, doi: 10.48082/espacios-a20v41n48p18.
- [12] F. De, A. Javier, and S. Alvarado, "DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE LOS PROYECTOS DE INVESTIGACIÓN QUE REALIZA EL CENI DE LA UNIVERSIDAD TÉCNICA DE AMBATO," 2010.
- [13] O. D. Medina Martínez, M. Pérez de Armas, F. Toledo Álvarez, A. Carrillo Ramos, R. J. Rosales González, and J. A. Trujillo Arteaga, "Sistema de información para la gestión de la producción científica de la Universidad de Ciencias Médicas de Cienfuegos," *MediSur*, vol. 17, pp. 764–770, 2019.
- [14] J. MEDRANO COLORADO, "SISTEMA WEB PARA LA GESTIÓN DEL CONCURSO DE PROYECTOS DE INVESTIGACIÓN ESTUDIANTILES, PROMOCIÓN Y DIFUSIÓN DE LAS ACTIVIDADES DESARROLLADAS POR EL CID PUCESE," 2014. Accessed: May 23, 2022. [Online]. Available: <https://repositorio.pucese.edu.ec/bitstream/123456789/33/1/MEDRANO%20COLORADO%20JOHN.pdf>
- [15] Atlassian, "JIRA SOFTWARE PRICING," Apr. 10, 2022. https://www.atlassian.com/es/software/jira/pricing?&aceid=&adposition=&adgroup=117628438529&campaign=11821722911&creative=566038231644&device=c&keyword=jira%20pricing&matchtype=e&network=g&placement=&ds_kids=p59412605168&ds_e=GOOGLE&ds_eid=700000001550060&ds_e1=GOOGLE&gclid=Cj0KCOjwhLKUBhDiARIsAMaTlnFWN5iJX6ZLBPSN_3yEu23dIX40wxHjrrD6mWRrnv4wOke_afq29LMaAkBrEALw_wcB&gclidsrc=aw.ds (accessed May 24, 2022).
- [16] Trello, "Trello Pricing," 2022. <https://trello.com/es/pricing> (accessed May 24, 2022).

- [17] Atlassian, "ASANA PRICING," Apr. 10, 2022. <https://asana.com/es/pricing> (accessed May 24, 2022).
- [18] Office365, "Microsoft Planner," 2022. <https://www.microsoft.com/es-co/microsoft-365/business/task-management-software> (accessed May 24, 2022).
- [19] Notion, "Notion Pricing," 2022. https://www.notion.so/pricing?utm_source=google&utm_campaign=2075789707&utm_medium=80211061161&utm_content=476117427519&utm_term=notion%20pricing&targetid=aud-1223925334303:kwd-342110974785&gclid=Cj0KCQjwhLKUBhDiARIsAMaTLnFxZk9CXwZp3uthIRLbiex2q2ZBY6kweQCS2SKTgT_ZwEelMDRVDGYaAutZEALw_wcB (accessed May 24, 2022).
- [20] Monday.com, "Monday Pricing," 2022. <https://monday.com/lang/es/pricing> (accessed May 24, 2022).



6. ANEXOS

1. Especificación de Requisitos de software - SRS_V2.0_Andres_Bastidas.docx
2. Código fuente backend rama master:

<https://github.com/andresbastidas/project-manager-backend>

3. Código fuente frontend rama master:

<https://github.com/andresbastidas/project-management-frontend>