



Universidad Laica Eloy Alfaro de Manabí

Facultad:

Ciencias de la vida y Tecnologías

Carrera:

Tecnologías de la información

Tema:

Desarrollo de Aplicación Web para la gestión procesos académicos y
documentos del proceso de titulación

Autores:

Anthony Rogers Arteaga Meza

Leif Wither Velásquez Realpe

DECLARACION DE EXPRESA AUTORIA

Nosotros, **Arteaga Meza Anthony Rogers** con cedula de ciudadanía **131566996-8** y **Velásquez Realpe Leif Wither** con cedula de ciudadanía **131621505-0** en calidad de autoras del trabajo de titulación: **“Desarrollo de Aplicación Web para la gestión procesos académicos y documentos del proceso de titulación”**

Declaramos que este trabajo es original y de nuestra autoría, hemos citado correctamente las fuentes utilizadas y respetado las leyes vigentes sobre derechos de autor. Autorizamos a la Universidad Laica “Eloy Alfaro” de Manabí a usar este trabajo con fines académicos e investigativos, así como publicarlo en su repositorio digital

Manta, 10 de Febrero de 2025



Arteaga Meza Anthony Rogers
C.I: 1315669968
e-mail Instit:
e1315669968@live.uileam.edu.ec



Velásquez Realpe Leif Wither
C.I: 1316215050
e-mail Instit:
e1316215050@live.uileam.edu.ec

| | | |
|---|--|-------------------------------------|
|  Uleam <small>UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ</small> | NOMBRE DEL DOCUMENTO: CERTIFICADO DE TUTOR(A). | CÓDIGO: PAT-04-F-004 |
| | PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR | REVISIÓN: 1 Página 1 de 1 |

CERTIFICACIÓN

En calidad de docente tutor(a) de la Facultad de Ciencias de la Vida y Tecnologías de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante Leif Wither Velásquez Realpe, legalmente matriculado/a en la carrera de Tecnologías de la Información, período académico 2024-2025(2), cumpliendo el total de 384 horas, cuyo tema del proyecto es "DESARROLLO DE APLICACIÓN WEB PARA LA GESTIÓN PROCESOS ACADÉMICOS Y DOCUMENTOS DEL PROCESO DE TITULACIÓN".

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

Manta, 17 de enero de 2025.

Lo certifico,


Ing. John Cevallos Macías
Docente Tutor(a)
Área: Software

| | | |
|---|---|------------------------------|
|  UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: CERTIFICADO DE TUTOR(A). | CÓDIGO: PAT-04-F-004 |
| | PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR | REVISIÓN: 1 Página 1 de 1 |

CERTIFICACIÓN

En calidad de docente tutor(a) de la Facultad de Ciencias de la Vida y Tecnologías de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante Anthony Rogers Arteaga Meza, legalmente matriculado/a en la carrera de Tecnologías de la Información, período académico 2024-2025(2), cumpliendo el total de 384 horas, cuyo tema del proyecto es "DESARROLLO DE APLICACIÓN WEB PARA LA GESTIÓN PROCESOS ACADÉMICOS Y DOCUMENTOS DEL PROCESO DE TITULACIÓN".

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

Manta, 17 de enero de 2025.

Lo certifico,



Ing. John Cevallos Macías
Docente Tutor(a)
Área: Software

UNIVERSIDA LAICA “ELOY ALFARO” DE MANABI

Creada el 13 de noviembre de 1985 mediante Decreto Ley No. 10, publicado en el registro Oficial No. 313

Trabajo De Titulación Modalidad: Proyecto Integrador, Previo A La Obtención Del Título
De: Ingeniero En Tecnologías De La Información

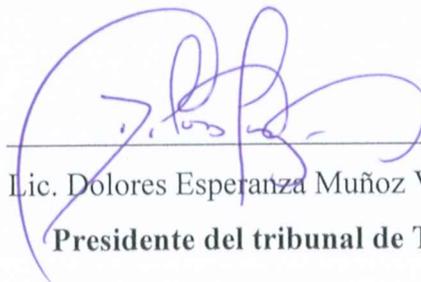
Título del Proyecto:

“Desarrollo de Aplicación Web para la gestión procesos académicos y documentos del proceso de titulación”

El tribunal examinador declara APROBADO el trabajo presentado para optar al grado de: INGENIERO EN TECNOLOGIAS DE LA INFORMACION, a los señores:

Arteaga Meza Anthony Rogers y Velásquez Realpe Leif Wither

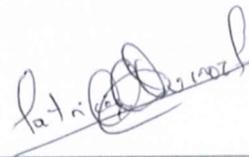
Señores Docentes:



Lic. Dolores Esperanza Muñoz Verduga, PhD
Presidente del tribunal de Titulación



Ing. Maria Esther Garcia Montes, Mg
Miembro del tribunal



Ing. Patricia Quiroz Palma, PhD
Miembro del tribunal



Agradecimientos

Agradezco a:

El cuerpo de bomberos de Manta donde pude hacer mis practicas preprofesionales y me aportaron una gran cantidad de conocimientos sobre información sobre el desarrollo y uso profesional de las tecnologías web.

Al Ing. Jhon Cevallos por la tutoría y ayudas en cuanto a el proceso de la tesis y los aportes sobre las tecnologías web de desarrollo backend donde pude conocer el framework NestJS y que sea muy importante para mi carrera.

Al Ing. Robert Moreira por los conocimientos y el adentrarnos a las bases de datos relacionales más importantes, su uso y sus conocimientos para las consistencias de datos también muy importantes en mi formación académica.

A mi familia por el apoyo incondicional en todos los semestres y etapas de mi carrera universitaria.

Arteaga Anthony



Agradecimientos

Agradezco a:

A mi familia y amigos por el apoyo que me han dado durante toda esta trayectoria que ha sido fundamental en mi crecimiento personal.

A mis profesores, por guiarme por el camino correcto y compartir su sabiduría. Sus enseñanzas han sido esenciales en mi formación académica.

A Cofarve, la empresa donde realicé mis prácticas profesionales. Su colaboración y el entorno profesional que me proporcionaron fueron cruciales para el desarrollo de mis habilidades y conocimientos prácticos.

A los YouTubers que han sido mis mentores a través de su contenido educativo. Su enseñanza me ha brindado valiosa información e inspiración a lo largo de la realización de esta tesis. En particular, agradezco a MiduDev, TodoCode, Pildorainformatica, Soy Dalto, Fatz, Nate Gentile, Alva Majo, Guinxu, JasperDev, Hdleon.net, Programador x, Absolute, Adderly Céspedes, Plano de Juego, Cinematix, Leedeo Studio, Juan Villalvazo y Hola Mundo.

Velásquez Leif



Dedicatoria

Dedico este trabajo de titulación a:

Mis Padres Mercedes Meza y Einer Arteaga los cuales siempre confiaron en mí, me enseñaron todo lo que se cómo persona y siempre me apoyaron de manera directa todos los días y me mantuvieron a flote emocionalmente.

A mi hermano Cristhian Arteaga el cual quiero mucho y es parte muy importante de mí y mi familia.

A mis amigos Acosta Lleyton, Velásquez Leif, García Jesús, Arteaga Carlos, Cevallos Angel, Mendoza Karol, a mi mejor amigo de colegio Alonzo Eduardo los cuales han sido un gran apoyo emocional en toda mi carrera en todos mis momentos y siempre están ahí para apoyarme.

Especialmente a mi amiga Tumbaco Karen la cual me apoyo desde siempre de las pocas personas que después de todo sigue conmigo y que se emociona mas que yo mi éxito y metas

A mi mascota fallecida kyubi la cual me enseñó a seguir adelante, el amor incondicional y me da fuerzas en su memoria para seguir adelante.

Arteaga Anthony



Dedicatoria

Le dedico este proyecto a mi familia, gracias al apoyo, amor, paciencia y sacrificios que han hecho, permitiéndome llegar hasta aquí. Además, también se lo dedico a mis amigos Eddu Saldarriaga, Jonatan Bravo, Alexis Chele, Lleyton Acosta, Juan Diego, Mendoza Anthony Arteaga y en especial, a mis mejores amigos Freddy Zambrano, Edson Cevallo y Fabricio Ponce, quienes confiaron en mí y me ayudaron en las situaciones difíciles.

A todos ellos, estoy profundamente agradecido por formar parte de mi vida y ser una fuente constante de apoyo e inspiración.

Velásquez Leif



Resumen

El presente trabajo de tesis presenta el desarrollo de una plataforma destinada a la web para la gestión de los procesos y documentos previos a las sustentaciones de tesis, con la finalidad de poder mejorar y controlar los procesos que actualmente se llevan antes de las sustentaciones, siendo estos los documentos habilitantes para las mismas, las asignaciones y la revisión de los documentos de tesis por los jurados de la sustentación. La plataforma se diseñó para que antes de la sustentación se pueda ver si los estudiantes cuentan con el cumplimiento completo de los documentos habilitantes y si estos han subido el respectivo trabajo de titulación el docente tutor y los jurados puedan ver dicho trabajo antes de la sustentación.

Por medio de un enfoque cualitativo y una entrevista pudimos obtener los datos necesarios requeridos sobre el proceso al cual nos enfocamos, en base a las respuestas pudimos identificar un flujo de proceso sobre el cual trabajar y además pudimos obtener todos los requerimientos sobre los cuales empezar a trabajar y programar las distintas funcionalidades. El desarrollo se realizó centrándose en el uso de microservicios usando como medio de comunicación la tecnología NATs por medio de un contenedor de Docker para así poder generar un programa ligero, pero altamente escalable vertical y horizontalmente.

Finalmente se expresaron las recomendaciones enfocadas en el uso de los microservicio y puntos a tomar en cuenta para poder continuar un desarrollo o investigación de estos.



Abstract

This thesis presents the development of a web-based platform for managing the processes and documents required before thesis defenses. The goal is to improve and control the procedures currently carried out before the defenses, including the necessary documents, assignments, and the review of thesis documents by the defense committee. The platform is designed to verify whether students have completed all the required documents and uploaded their thesis work so that the supervising professor and committee members can review it before the defense.

Through a qualitative approach and an interview, we gathered the necessary data regarding the process we focused on. Based on the responses, we identified a process flow to work with and defined all the requirements needed to develop and implement the various functionalities. The development was carried out using a microservices architecture, employing NATs technology as the communication medium through a Docker container. This approach enabled the creation of a lightweight yet highly scalable program, both vertically and horizontally.

Finally, recommendations were provided regarding the use of microservices and key aspects to consider for further development or research in this area.



| | | |
|---|---|----------------------|
|  | NOMBRE DEL DOCUMENTO: | CÓDIGO: PAT-04-F-004 |
| | CERTIFICADO DE TUTOR(A). | |
| | PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR | REVISIÓN: 1 |
| | | Página 1 de 1 |

CERTIFICACIÓN

En calidad de docente tutor(a) de la Facultad de Ciencias de la Vida y Tecnologías de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante Anthony Rogers Arteaga Meza, legalmente matriculado/a en la carrera de Tecnologías de la Información, período académico 2024-2025(2), cumpliendo el total de 384 horas, cuyo tema del proyecto es "DESARROLLO DE APLICACIÓN WEB PARA LA GESTIÓN PROCESOS ACADÉMICOS Y DOCUMENTOS DEL PROCESO DE TITULACIÓN".

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

Manta, 17 de enero de 2025.

Lo certifico,

Ing. John Cevallos Macías
Docente Tutor(a)
Área: Software



| | | |
|---|---|------------------------------|
|  | NOMBRE DEL DOCUMENTO: CERTIFICADO DE TUTOR(A). | CÓDIGO: PAT-04-F-004 |
| | PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR | REVISIÓN: 1 Página 1 de 1 |

CERTIFICACIÓN

En calidad de docente tutor(a) de la Facultad de Ciencias de la Vida y Tecnologías de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante Leif Wither Velásquez Realpe, legalmente matriculado/a en la carrera de Tecnologías de la Información, período académico 2024-2025(2), cumpliendo el total de 384 horas, cuyo tema del proyecto es "DESARROLLO DE APLICACIÓN WEB PARA LA GESTIÓN PROCESOS ACADÉMICOS Y DOCUMENTOS DEL PROCESO DE TITULACIÓN".

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

Manta, 17 de enero de 2025.

Lo certifico,

Ing. John Cevallos Macías
Docente Tutor(a)
Área: Software



Índice

| | |
|---|----|
| CAPITULO I..... | 21 |
| 1. Marco Introdutorio..... | 22 |
| 1.1 Introducción..... | 22 |
| 1.1.1 Ubicación y contextualización de la problemática..... | 23 |
| 1.2 Planteamiento del Problema..... | 23 |
| 1.3 Formulación Del Problema..... | 24 |
| 1.3.1 Problema General..... | 24 |
| 1.3.2 Problemas Específicos..... | 24 |
| 1.3.3 Pregunta De Investigación..... | 25 |
| 1.4 Diagrama Causa y Efecto..... | 25 |
| 1.5 Objetivos..... | 26 |
| 1.5.1 Objetivo General..... | 26 |
| 1.5.2 Objetivos Específicos..... | 26 |
| 1.6 Justificación..... | 27 |
| 1.7 Alcance y Limitaciones Del Estudio..... | 27 |
| 1.8 Impactos Esperados..... | 28 |
| 1.8.1 Impacto Tecnológico..... | 28 |
| 1.8.2 Impacto Social..... | 29 |
| CAPITULO II..... | 30 |



| | |
|--|----|
| 2. Marco Teórico..... | 31 |
| 2.1 Antecedentes Históricos..... | 31 |
| 2.1.1 Antecedentes de la Educación Superior..... | 31 |
| 2.1.2 Antecedentes de la Digitalización de Documentos..... | 31 |
| 2.2 Antecedentes de Investigación Relacionada..... | 31 |
| 2.3 Definiciones conceptuales..... | 34 |
| 2.3.1 Metodología de desarrollo..... | 34 |
| 2.3.2 Herramientas Tecnológicas..... | 36 |
| 2.3.3 Base de Datos..... | 37 |
| 2.3.4 Lenguaje y Framework..... | 37 |
| 2.3.5 Comunicación para microservicios..... | 39 |
| 2.4 Análisis de las Herramientas..... | 42 |
| 2.4.1 Herramientas Front End Web..... | 42 |
| 2.4.2 Conclusiones sobre Herramientas Front End Web..... | 45 |
| 2.4.3 Herramientas Back End..... | 45 |
| 2.4.4 Conclusión Herramientas Back End..... | 47 |
| 2.4.5 Herramientas de Base de Datos..... | 47 |
| 2.4.6 Conclusión sobre Herramientas de Base de Datos..... | 50 |
| 2.5 Conclusión de las investigaciones..... | 51 |
| CAPITULO III..... | 52 |
| 3 Marco Investigativo..... | 53 |
| 3.1 Tipo de Investigación..... | 53 |



| | |
|--|----|
| 3.2 Métodos de Investigación | 54 |
| 3.2.1 Método Teóricos | 54 |
| 3.3 Estrategia Operacional para la Recolección de Datos | 55 |
| 3.3.1 Población y Muestra | 55 |
| 3.3.2 Análisis de las herramientas de recolección de datos a utilizar | 56 |
| 3.4 Plan de Recolección de Datos..... | 56 |
| 3.5 Análisis de los datos..... | 57 |
| 3.5.1 Preparación de los Datos | 57 |
| 3.5.2. Análisis de Correlación..... | 57 |
| 3.5.3 Análisis Cualitativo..... | 57 |
| 3.5.4. Interpretación de Resultados..... | 57 |
| 3.6 Análisis de los resultados..... | 58 |
| 3.6.1 Entrevista a director de proceso de titulación..... | 58 |
| 3.6.2 Entrevista a secretaria encargada de los requisitos de sustentación .. | 63 |
| 3.7 Conclusión del Marco Investigativo | 65 |
| CAPITULO IV | 67 |
| 4. Marco Propositivo | 68 |
| 4.1 Descripción de la Propuesta..... | 68 |
| 4.1.1 Diseño de la arquitectura | 68 |
| 4.1.2 Backend | 69 |
| 4.1.3 Microservicio | 70 |
| 4.2 Determinación de Recursos | 70 |



| | |
|---|-----|
| 4.2.1 Humanos | 71 |
| 4.2.2 Tecnológicos | 72 |
| 4.2.3 Económicos..... | 73 |
| 4.3 Etapas de acción para el desarrollo de la propuesta..... | 74 |
| 4.3.1 Diseño del Modelado de la Base de Datos | 74 |
| 4.3.2 Diseño y Desarrollo el Backend | 77 |
| 4.3.3 Diseño y Desarrollo en el FrontEnd | 77 |
| 4.4 Etapas de acción para el desarrollo de la aplicación..... | 78 |
| 4.4.1 Fase I: Definición y Planificación | 79 |
| 4.4.2 Fase II: Diseño y Desarrollo | 109 |
| 4.4.3 Fase III: Configuración y Despliegue..... | 123 |
| 4.5 Conclusión del Capítulo | 133 |
| Capítulo V..... | 134 |
| 5. Evaluación de Resultados..... | 135 |
| 5.1 Introducción | 135 |
| 5.2 Presentación de resultados..... | 135 |
| 5.2.1 Pruebas desde el usuario estudiante..... | 135 |
| 5.2.2 Pruebas de usuario secretaria..... | 136 |
| 5.2.3 Prueba desde el usuario profesor | 139 |
| 5.2.4 Prueba desde el usuario administrador | 140 |
| 5.3 Conclusión | 143 |
| Capítulo VI..... | 144 |



| | |
|---|-----|
| 6. Conclusiones y Recomendaciones..... | 145 |
| 6.1 Conclusiones..... | 145 |
| 6.2 Recomendaciones | 146 |
| 7. Bibliografía..... | 147 |
| 8. Anexo..... | 153 |
| Entrevista a director de proceso de titulación..... | 153 |
| Entrevista a secretaria encargada de los requisitos de sustentación | 155 |
| Imágenes del código del desarrollo de la plataforma | 156 |



Índice Figuras

| | | |
|------------|--|----|
| Figura 1. | Diagrama causa y efecto..... | 25 |
| Figura 2. | Ejemplo del uso de Kafka en un sistema simple, por Bill Bejeck..... | 42 |
| Figura 3. | Arquitectura de la aplicación..... | 69 |
| Figura 4. | Arquitectura del BackEnd | 69 |
| Figura 5. | Arquitectura y lógica de un microservicio | 70 |
| Figura 6. | Tabla de base de dato..... | 76 |
| Figura 7. | Planificación de los módulos de desarrollo en Airtable. | 80 |
| Figura 8. | Planificación de las tareas de desarrollo en Airtable. | 80 |
| Figura 9. | Desarrollo de las actividades de la planificación en Airtable. | 81 |
| Figura 10. | Primeras actividades de la fase 1..... | 81 |
| Figura 11. | Primer Boceto del prototipo | 84 |
| Figura 12. | Segundo boceto diseño del prototipo | 84 |
| Figura 13. | Tercer boceto diseño del prototipo | 85 |
| Figura 14. | Cuarto boceto diseño del prototipo..... | 85 |
| Figura 15. | Vista de modelado del prototipo..... | 86 |
| Figura 16. | Modelado de Funcionamiento desde la vista secretaria | 86 |
| Figura 17. | Modelado de Funcionamiento desde la vista estudiante | 87 |
| Figura 18. | Modelado de Funcionamiento desde vista profesor como jurador y tutor. | 87 |



| | |
|---|-----|
| Figura 19. Interacción Administrador – Sistema | 97 |
| Figura 20. Gestión de Usuario | 98 |
| Figura 21. Gestión de Roles..... | 98 |
| Figura 22. Interacción secretaria – Sistema | 99 |
| Figura 23. Gestión de Carrera..... | 100 |
| Figura 24. Gestión de Requisito..... | 100 |
| Figura 25. Gestión de Estudiantes | 101 |
| Figura 26. Gestión de Profesores | 101 |
| Figura 27. Gestión de Tesis..... | 102 |
| Figura 28. Interacción profesor - Sistema..... | 102 |
| Figura 29. Gestionar Revisión de sustentación..... | 103 |
| Figura 30. Gestionar Revisión de sustentación..... | 103 |
| Figura 31. Interacción estudiante - Sistema..... | 104 |
| Figura 32. Gestionar Perfil..... | 104 |
| Figura 33. Vista Estudiante - Tesis..... | 105 |
| Figura 34. Tablas para el microservicio de autenticación | 106 |
| Figura 35. Tablas para el microservicio de Usuarios | 107 |
| Figura 36. Tablas para el microservicio de Control de Tesis | 108 |
| Figura 37. Tablas para el microservicio de sustentación | 108 |
| Figura 38. Actividades del primer módulo de fase II | 109 |
| Figura 39. Actividades del segundo módulo de fase II..... | 109 |
| Figura 40. Contenedor de NATs levantado y funcionando | 110 |



| | |
|--|-----|
| Figura 41. Imagen oficial de NATs usada | 110 |
| Figura 42. Explicación lógica de comunicación al hacer una petición | 111 |
| Figura 43. Código de controlador MainGateway | 112 |
| Figura 44. Código del controlador del microservicio | 112 |
| Figura 45. Código del servicio del microservicio | 113 |
| Figura 46. Código del método Get básico creado | 115 |
| Figura 47. Código del método GetAll básico creado | 115 |
| Figura 48. Código del método Create básico creado | 116 |
| Figura 49. Código del método Update básico creado | 116 |
| Figura 50. Código del método Delete básico creado | 117 |
| Figura 51. Código del método Restore básico creado | 117 |
| Figura 52. Definición de SEO y Font del Frontend | 118 |
| Figura 53. Arquitectura del Frontend..... | 119 |
| Figura 54. Configuración de Proxy y Cabecera del http | 120 |
| Figura 55. Interfaz Login | 121 |
| Figura 56. Configuración de middleware y otras funcionalidades de verificación. .. | 121 |
| Figura 57. Perfil del Usuario..... | 122 |
| Figura 58. Interfaz gestión de Tesis | 122 |
| Figura 59. Actividades del primer módulo de fase III | 123 |
| Figura 60. Actividades del primer módulo de fase III | 123 |
| Figura 61. Actualización de VPS digital Ocean | 124 |
| Figura 62. Comprobación de nodejs | 124 |



| | |
|--|-----|
| Figura 63. Instalación de Docker en VPS | 124 |
| Figura 64. Usuario creado en VPS de base de datos..... | 125 |
| Figura 65. Base de datos preparadas en el VPS | 125 |
| Figura 66. Configuración de Traslado del Dominio de Go daddy..... | 126 |
| Figura 67. Configuración de Cloudflare para obtener el dominio trasladado..... | 126 |
| Figura 68. Conexión a bases de datos | 127 |
| Figura 69. Servicio NATs en el contenedor levantado..... | 128 |
| Figura 70. Microservicio iniciado y corriendo en VPS | 128 |
| Figura 71. Repositorio de Github del Frontend | 129 |
| Figura 72. Plataforma de Vercel crear un nuevo proyecto | 129 |
| Figura 73. Seleccionando proyecto a desplegar..... | 130 |
| Figura 74. Configuración del despligue del frontend | 130 |
| Figura 75. Despligue de frontend Exitoso..... | 131 |
| Figura 76. Configuración de CNAME del dominio para Vercel..... | 131 |
| Figura 77. Configuración del dominio del sitio web en Vercel | 132 |
| Figura 78. Configuración de Dominio Exitoso..... | 132 |
| Figura 79. Manual de usuario | 153 |
| Figura 80. Desarrollo de Login del FrontEnd..... | 156 |
| Figura 81. Desarrollo de layout de la plataforma | 156 |



Índice Tablas

| | |
|--|----|
| TABLA 1. Tabla de Antecedentes relacionados con el tema..... | 34 |
| TABLA 2. Comparación de latencia entre RabbitMQ y NATs | 40 |
| TABLA 3. Tabla de resultados sobre pruebas de latencia RabbitMQ y NATS | 41 |
| TABLA 4. Comparación de framework Frontend..... | 45 |
| TABLA 5. Comparación de framework Backend | 47 |
| TABLA 6. Comparación de bases de datos | 50 |
| TABLA 7. Recursos Humanos | 71 |
| TABLA 8. Recuso tecnológico (Hardware) | 72 |
| TABLA 9. Recursos Tecnológicos (Software) | 72 |
| TABLA 10. Recursos Económicos..... | 73 |
| TABLA 11. Caso de uso Autenticación de Usuarios..... | 88 |
| TABLA 12. Caso de uso Registro de Usuario..... | 88 |
| TABLA 13. Caso de uso Modificación del Usuario..... | 89 |
| TABLA 14. Caso de uso Activa o Desactivar Usuario..... | 89 |
| TABLA 15. Caso de uso Registro de Carrera | 90 |
| TABLA 16. Caso de uso Modificación del Carrera | 90 |
| TABLA 17. Caso de uso Activa o Desactivar Carrera | 90 |
| TABLA 18. Caso de uso Registro de Requisito | 91 |
| TABLA 19. Caso de uso Modificación del Requisito | 91 |



| | |
|--|-----|
| TABLA 20. Caso de uso Activa o Desactivar Requisito | 92 |
| TABLA 21. Caso de uso Activar o Desactivar Rol | 92 |
| TABLA 22. Caso de uso Activar o Desactivar los Requisito Cumplido del estudiante. 92 | |
| TABLA 23. Caso de uso Registro de Tesis | 93 |
| TABLA 24. Caso de uso Modificación del Tesis | 93 |
| TABLA 25. Caso de uso Activa o Desactivar Tesis | 94 |
| TABLA 26. Caso de uso Modificación del Perfil..... | 94 |
| TABLA 27. Caso de uso Registro de Sustentación | 95 |
| TABLA 28. Caso de uso Modificación del Sustentación | 95 |
| TABLA 29. Caso de uso Agregar Comentario referente al documento de tesis | 95 |
| TABLA 30. Caso de uso Aprobación de Tesis | 96 |
| TABLA 31. Caso de uso Subida de Documento de Tesis | 96 |
| TABLA 32. Caso de uso Reemplazo de Documento de Tesis..... | 96 |
| TABLA 33. Pruebas realizadas en usuarios estudiante | 136 |
| TABLA 34. Pruebas realizadas en usuarios de tipo secretaria | 139 |
| TABLA 35. Pruebas realizadas en usuarios profesor | 140 |
| TABLA 36. Pruebas realizadas en usuario administrador..... | 143 |



CAPITULO I



1. Marco Introdutorio

1.1 Introducción

En la actualidad, donde los procesos y la documentación de organizaciones y empresas transitan hacia la era digital, la tecnología ha facilitado la gestión y control de estos procesos. La web, en particular, domina gran parte de esta transformación tecnológica. Aunque hay empresas u organizaciones que no todo su proceso ha logrado adaptarlo por completo, ya que este cambio requiere más trabajo para flexibilidad y personalización en la gestión planificada.

Actualmente muchas instituciones no cuentan con un software o plataforma de gestión para su proceso de documentales en el proceso de titulación ya que no cuenta con una regulación digital. Esto llega a generar problemas a todo lo involucrado en el proceso de titulación ya que al no contar con una normativa esta entidad utilizar varias plataformas y varios procesos que transcurren cada semestre llegar a tener como resultado retraso, normativas inesperada y confusión.

El objetivo de la tesis es desarrollar una plataforma web que solucione el problema analizando los procesos involucrados que requiera la entidad para regularizar una normativa y procesos que deben realizar y a partir de este diseñar y construir este prototipo.

La plataforma o prototipo construido mediante el levantamiento de requisitos ofrecerá una regulación de las normativas y procesos documentales y facilitando la gestión del proceso de sustentación del estudiante eficiente y eficaz.



1.1.1 Ubicación y contextualización de la problemática

Los procesos de titulación Universitarias un proceso muy importante académicamente hablando tanto para la institución, estudiantes y docentes implicados en esta, por eso mismo este suele ser el proceso más extenso y dividido por partes en algunas instituciones, este además es el que más actores puede tener, también cuenta con la característica de ser el que más documentación de suma importancia genera en una carrera. Según esto, el proceso de titulación según cómo se administre puede generar mayor complejidad al llevarse.

El orden y la centralización de los procesos son fundamentales en cualquier actividad, académica o educativa. Mantener un proceso ordenado y supervisado incrementa significativamente la eficiencia de cualquier organización. Cuando existen muchos actores involucrados, poder gestionar su participación de manera efectiva mejora todo el proceso y facilita el análisis de su funcionamiento y desempeño.

La falta de existencia de un aplicativo que pueda gestionar el proceso de sustentación en titulación permite la implementación de una plataforma web y que está presente cambios positivos y mejoras en la calidad del proceso además de mejorar los tiempos dedicados a los actores implicados. Esto también simplificaría la necesidad de buscar los recursos y el lugar donde almacenarlo.

1.2 Planteamiento del Problema

Para el sistema de educación superior actual, uno de los procesos más importantes si no es que más peso tiene sin duda es el proceso de titulación de los estudiantes universitarios, este en algunos casos es llevado de manera tradicional. Por eso la gestión de los procesos académicos en este ámbito se gestiona de manera diferente y de maneras en las que no son eficientes comparadas con las tecnologías actuales.



Se ha identificado que la parte de la sustentación no se lleva con un software que pueda gestionar de manera eficiente la documentación requerida y generada al avanzar mediante los procesos estipulados por la Universidad para finalizar satisfactoriamente el proceso. Los participantes, Jurado evaluador y docente designado a la tesis tienen una manera sencilla centralizada de mantener el seguimiento en los procesos finales de la tesis.

1.3 Formulación Del Problema

El problema principal de este proyecto es que no hay una plataforma para organizar todos los procesos previos referente a las sustentaciones de tesis. Esto causa muchas dificultades que suele causar ciertos inconvenientes a la hora de proseguir la fase.

1.3.1 Problema General

No tener un sistema para organizar los procesos previos a la sustentación de tesis *hace que sea más complicado para los estudiantes llegar al final de su etapa de estudios.* Frecuentemente, surgen imprevistos que retrasan el progreso innecesariamente este proceso.

1.3.2 Problemas Específicos

- Los estudiantes no entregan los documentos a tiempo: Muchas veces, los estudiantes no cumplen con los requisitos porque dejan todo para el último momento. Esto hace que les falten papeles importantes o que no puedan cumplir con los plazos establecidos.
- No hay un buen seguimiento para los estudiantes: Como no hay una plataforma, los profesores y el jurado no pueden hacer un buen



seguimiento del proceso. Esto hace que todo sea más complicado y menos organizado.

1.3.3 Pregunta De Investigación

¿El desarrollo de una aplicación web puede facilitar la gestión del proceso previo a la sustentación de tesis, promoviendo una organización y eficiencia óptimas?

1.4 Diagrama Causa y Efecto

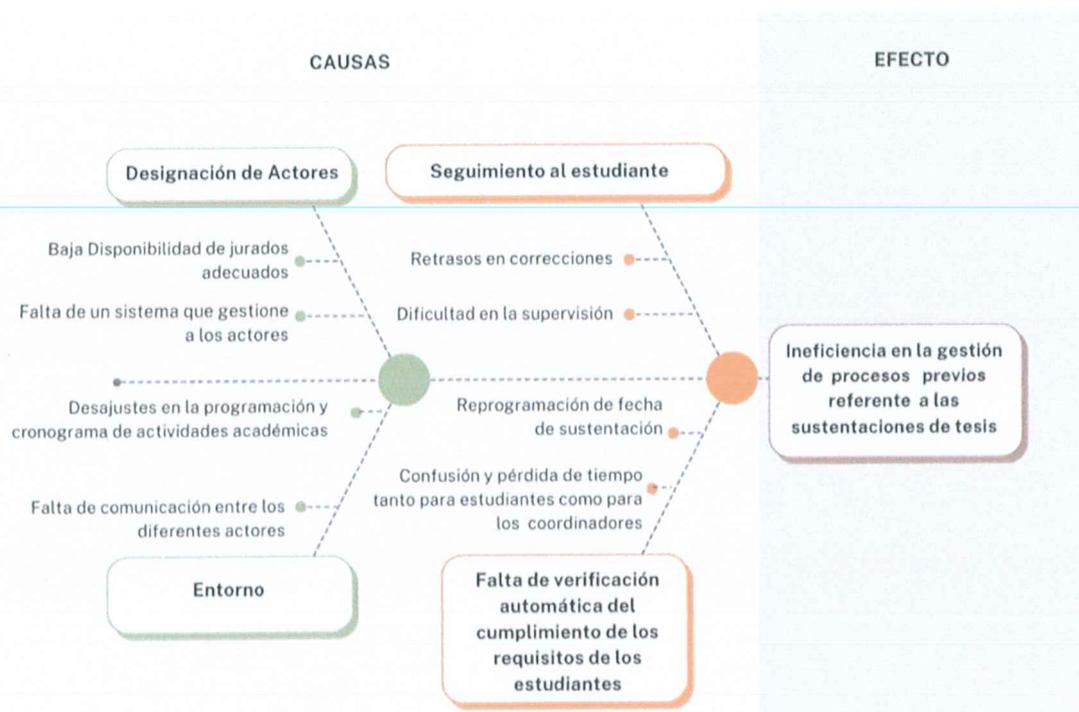


Figura 1. Diagrama causa y efecto

El diagrama de Ishikawa, también reconocido como causa-efecto, se grafica como una espina de pescado, donde la cabeza del pez es el efecto, las espinas son los problemas y las ramificaciones son las causas. Es una herramienta utilizada para identificar la



problemática que genera cierta causa, la cual, a su vez, tiene un efecto. En nuestro caso, se hizo uso de este diagrama debido al problema principal relacionado con la falta de algún aplicativo para la administración del proceso de sustentación de tesis.

Para entender mejor la posible causa de este problema, hemos organizado el diagrama con varios puntos principales, como:

- Ausencia de un sistema automatizado de asignación de fechas.
- Designación de actores.
- Falta de verificación automática del cumplimiento de los requisitos previos.
- Falta de un repositorio digital para el seguimiento y la revisión de los documentos en proceso de desarrollo de la tesis.

Entonces, como conclusión, el diagrama de causa-efecto nos ayuda a identificar las causas, los problemas y los efectos de una manera estructurada para tener un mejor entendimiento a la hora de abordar las causas potenciales, buscar una mejor solución y promover un mejor entendimiento de este.

1.5 Objetivos

1.5.1 Objetivo General

- Desarrollar una Aplicación Web para la gestión de procesos académicos y documentos habilitantes previos al proceso de sustentación.

1.5.2 Objetivos Específicos

- Definir la arquitectura de la API de microservicios considerando los requerimientos necesarios en el proceso de sustentación de tesis.



- Desarrollar las funcionalidades principales de la aplicación de los distintos usuarios, garantizando la cobertura de los casos de uso aplicando microservicios.
- Realizar un despliegue de prueba que integre todos los microservicios para validar su correcto funcionamiento y comunicación en un entorno controlado.

1.6 Justificación

La importancia de la creación de un software para la gestión del proceso previo a la sustentación de titulación se basa en la optimización y solución de los problemas de gestión de los actores del proceso. De centralizar en un solo punto, en este caso la página web para tener un control de manera eficiente y sencilla de donde se está llevando todos los procesos relacionados, la gestión de documentos de igual manera se ve beneficiado por la centralización del proceso ya que la generación y guardado de estos se soluciona gracias al fácil acceso a estos por medio de la misma plataforma, la creación de un software para los procesos académicos relacionados con la sustentación no solo beneficia de manera grande a los estudiantes, sino que es beneficioso para los actores implicados en este ya que se verán beneficiados gracias a ser una manera sencilla de controlar el proceso y las etapas de este.

1.7 Alcance y Limitaciones Del Estudio

El alcance que puede tener la implementación de la plataforma web puede verse limitado por la infraestructura tecnología que tiene disponible la institución u organización lo cual afectará el rendimiento del esté. La adaptación de la plataforma web por parte de los usuarios puede verse comprometida por requerir tiempo, dificultad en la adaptación y la aceptación este nuevo sistema. Por lo cual deberá contar con una capacitación donde todos los usuarios involucrados puedan adaptarse de acuerdo con el



diseño y funcionalidades que esta contará. Además, se verá limitada por normativas y políticas de las organizaciones la cual deben cumplirse.

1.8 Impactos Esperados

La página web para la gestión del procesos de sustentación para la titulación se espera que se mejore la facilidad en las distintas partes de los procesos y el seguimiento de estos al centralizarlos, que se pueda integrar de manera más fácil y eficiente a los actores responsable en cada parte del proceso de titulación y presentación de tesis, que se mejore así la comunicación entre los actores y se tenga una muy buena retroalimentación sobre el proceso y que además se mejore la calidad de vida de estos.

Con el uso de Microservicios en esta se espera una mejor consistencia y rendimiento de la plataforma ya que darle responsabilidades de funciones a un servicio único hace que este sea mucho más rápido, además en caso de comprometerse una base de datos al tenerlas separadas hace que la seguridad de las demás no se vean comprometidas porque esos datos no están en esa base de datos.

1.8.1 Impacto Tecnológico

Se espera que el uso de una plataforma web facilite el uso y la accesibilidad del programa, que sea sencilla para los estudiantes y docentes actores de estos procesos, que se puedan gestionar los documentos de manera sencilla para que sea fácil llevar a cabo los requerimientos dados. Que el proyecto gracias a la modularización que dan los frameworks sea escalable y expandible a más mejoras y cambios. También que esta permita que parte de la interacción que se lleva a cabo con tutores de proyecto o jurado pueda ser más sencilla y de manera remota brindando la posibilidad de mantener una comunicación eficiente con ellos.



1.8.2 Impacto Social

Se espera que la plataforma web mejore la experiencia del estudiante gracias a la centralización de los procesos haciendo que sepa siempre donde está siendo gestionado su proceso de titulación, que los docentes puedan coordinar de manera más sencilla sus procesos sin mucho esfuerzo para aliviar el trabajo de este, que al nivel como comunidad académica los encargados del proceso como los actores y estudiantes tengan una comodidad más grande y una respuesta más satisfactoria a cada paso que se da en este mismo.



CAPITULO II



2. Marco Teórico

2.1 Antecedentes Históricos

2.1.1 Antecedentes de la Educación Superior

En la antigua Grecia clásica se sentaron las bases de la educación superior, la academia de Platón el primer pilar principal de este, la cual fue fundada hace más de 2400 años otro pilar fundamental e inicial en la educación superior es el Liceo de Aristóteles en estas principalmente se trabajaba en base a la filosofía.

En el siglo XII se produce el auge de las universidades en Europa durante la edad media en este mismo siglo la invención de la imprenta favoreció mucho la difusión del conocimiento. En Hispanoamérica se sabe que en el colonialismo se fundan Universidades con Influencias religiosas una en Santo Domingo y otra en San Marcos de Perú, Después de la independencia se fundan las Universidades públicas y privadas en base al modelo europeo.

2.1.2 Antecedentes de la Digitalización de Documentos

Desde la creación de los primeros sistemas informáticos se ve como los archivos digitales se necesitan en estos, desde archivos de datos planos hasta archivos de configuración, pero los archivos de actividades o procesos físicos a digitales llegan desde la segunda mitad del siglo XX gracias al avance de las computadoras para uso personal y profesional en la sociedad, sentando las bases de la necesidad de la digitalización de los documentos en los trabajos.

2.2 Antecedentes de Investigación Relacionada

En base al tema de la aplicación para la gestión de los procesos de titulación se tomaron como antecedentes referenciales ciertas tesis que pudieron encontrar en la web



de Google académico los cuales toman como relación hecha de haber desarrollado una aplicación con objetivo el proceso de titulación en las diferentes universidades en las que se desarrollaron estas, en base a esto se pudo analizar:

Según (ELIZABETH, 2022) la creación de un software para la gestión de los procesos de titulación tiene un valor grande cuando se habla de la fiabilidad que se le da al proceso relacionado con la verificación de las tesis, esto también acelerara la asignación y control de estos actores a la tesis, también propone una fácil revisión por parte de administración a estos gracias al software y al registro de esas asignaciones en este.

Para (RIVILLA, 2017) el sistema web desarrollado e implementado presenta una solución para adaptar la metodología de titulación nueva que se aplicó en la institución politécnica salesiana, junto con eso el proceso anterior se llevaba de manera manual el cual generaba inconvenientes en cuanto a la coordinación de los docentes y tutores con los estudiantes asignados a este proceso, en esta tesis se proponen y brindan herramientas para la gestión del proceso en el que se englobe a los actores de manera sencilla.

En cuanto a lo propuesto por (MERCHAN, 2020) se hace énfasis en que la aplicación de un software como el que ellos proponen ayuda a la hora de evitar las tesis duplicadas o plagios en esta ya que de manera sencilla se permite la búsqueda de los temas anteriores con los que se sustentaran en el momento, también plantean el objetivo de que el área administrativa lleve un mejor control de los estados y fases de la titulación.

Para (Joseph Saldaña Torres, 2015) los problemas encontrados en el sistema que se manejaba era que toda la documentación del proceso de sustentación de tesis y anteproyecto de tesis eran llevados por Word y por impresiones lo que dificultaban la gestión y organización de todos procesos de sustentación en cada periodo académico.



Al acabar de analizar las tesis investigadas nos podemos dar cuenta que los factores comunes que presentan problemas en estos procesos son la documentación y las asignaciones de los actores en las sustentaciones.

| Tema | Autor/es | Relevancia |
|---|---|--|
| SOFTWARE INFORMÁTICO PARA EL CONTROL Y REGISTROS DE TEMAS DE TITULACIÓN EN LA CARRERA TECNOLOGÍAS DE LA INFORMACIÓN. | VERA BARZOLA CYNTHIA ELIZABETH | El crear un software aumenta aumento la facilidad que tenían los estudiantes y docentes de poder revisar los documentos subidos para poder iniciar el proceso de sustentación |
| IMPLEMENTACION DE SISTEMA WEB PARA LA GESTION Y CONTROL DE LOS PROCESOS DE LA UNIDAD DE TITULACION DE LA CARRERA DE INGENIERIA EN SISTEMAS DE UNIVERSIDAD SALESIANA. | JORGE ENRIQUE MENDOZA RIVILLA | El software cambio todo el proceso y lo agilizo ya que antes todo se hacia de manera manual y con todos los documentos de manera física teniendo dificultad para le lectura de estos y su organizacion |
| APLICACIÓN INFORMATICA PARA PROCESOS DE TITULACION DE LA | JONATHAN JAIR SALAZAR MERCHAN | Junto con la gestión de las tesis se propuso el validar el plagio que pueda haber referente a otras tesis de la institución. |



UNIVERSIDAD ESTATAL

DEL SUR DE MANABÍ.

| | | |
|--|---|---|
| SISTEMA WEB PARA LA GESTIÓN Y ADMINISTRACIÓN DE ANTEPROYECTO Y TESIS DE GRADO | Joseph Saldaña Torres, Rafael Zúñiga Burgos | En esta tesis lo mas importante fue el proponer la organización de los documentos ya que era un problema que todos fueran Word pesados. |
|--|---|---|

TABLA 1. Tabla de Antecedentes relacionados con el tema

2.3 Definiciones conceptuales

2.3.1 Metodología de desarrollo

En cuanto a la metodología de desarrollo usada para la creación de la plataforma web se opta por la metodología por la metodología personalizada denominada Agile Inception.

Según (achdeva, 2016) dice que: *"Agile Inception es una metodología ágil adaptada, diseñada específicamente para proyectos académicos o de pequeño alcance que requieren flexibilidad, simplicidad y un enfoque en la entrega constante de resultados. Que elimina formalidades innecesarias de metodologías estándar como Scrum, centrándose en una planificación inicial ligera, ejecución directa y reuniones rápidas de retroalimentación, ajustándose así a las necesidades del equipo y del proyecto " en 2001 por Jonathan Rasmusson".*

Entonces, la metodología Agile Inception es una metodología ágil adaptada, diseñada especialmente para proyectos académicos o de alcance limitado que requieran flexibilidad y un enfoque sencillo en la entrega constante de resultados. Este enfoque



elimina formalidades innecesarias que pueden encontrarse en metodologías como Scrum, centrándose únicamente en la ejecución de la planificación inicial ligera a través de la ejecución directa de la actividad y reuniones rápidas de muestra de resultado.

Sobre los componentes de la estructura que conlleva Agile Inception son:

- Equipo
- Facilitador
- Revisiones de retroalimentación

En el flujo de trabajo de la metodología AFD se basa en los siguientes pasos:

- Planificación inicial
- Ejecución ágil
- Revisiones rápidas

Una de las grandes ventajas de esta metodología es su capacidad de adaptar el modo de trabajo del equipo, permitiendo que la planificación, la ejecución del proyecto y las revisiones periódicas se desarrollen de manera coordinada. Esto es fundamental, ya que nunca se pierde de vista los objetivos finales. Además, el equipo puede añadir ciertas reglas o complementar la metodología con otras, según las necesidades del proyecto.

En cada ciclo, se realiza una retroalimentación para identificar posibles mejoras, lo que garantiza que el equipo avance según los objetivos establecidos. Este enfoque no solo contribuye a la productividad, sino que también permite una mayor claridad en la ejecución del proyecto.



2.3.2 Herramientas Tecnológicas

- Control de versiones: Para el control de versión de opta por usar las herramientas Git y GitHub, Git es un software de control de versión muy popular por su facilidad de usar, su portabilidad y sus múltiples herramientas anexadas a él, este software desarrollado por Linus Torvald permite el alojamiento de los archivos de un proyecto en cualquier lenguaje de programación para así poder controlar las versiones del repositorio en base a las actualizaciones, además de poder gestionar estos por ramas que pueden ser integradas a la rama principal con el fin de que cada equipo pueda trabajar individualmente, probar sus funcionalidad y luego unir las al proyecto principal de manera sencilla.
- IDE: Como IDE de desarrollo se usará el Visual Studio Code, la selección de esta se basa en la gran cantidad de lenguajes los cuales puede soportar y además adaptar a este ide tales como: Python, JS, TS, C#, C++, etc...., También la gran cantidad de complementos interesantes para la facilidad de la codificación de proyectos.
- Control de tareas: Para el control de tareas en el proceso de desarrollo se selección la herramienta Airtable la cual permite gestiona un proyecto en el cual se puede añadir al equipo como parte de esta y asignar tareas con un límite temporal de este para verificar el cumplimiento de estas. Con Airtable se puede gestionar y asignar roles a las tareas pertinentes en el desarrollo y así llevar un registro temporal de estos.
- Herramienta de Responsive Design: ResponsivelyApp: Herramientas que permite visualizar el comportamiento responsive que tiene la página web a través mediante la simulación de múltiple dispositivo.



2.3.3 Base de Datos

Los Gestores de base de datos son programas que permiten la característica de fundamental importancia para cualquier tipo de programa llamada persistencia de datos. Las bases de datos se encargan de mantener los registros e información que se requiere o genera una aplicación. Estos junto a los Gestores de base de datos permiten el poder usar operación SQL para crear la estructura relacional o no relacional de datos de un proyecto

Para este proyecto se usará la base de datos llamada PostgreSQL ya que es muy reconocida por su estabilidad, escalabilidad que puede llegar a tener y las funciones ya sean complejas y sencillas que tienen integrados.

2.3.4 Lenguaje y Framework

- Lenguajes: Los lenguajes de programación son los fundamentales a escoger y también depende del framework y la plataforma a la que se esté apuntando crear el programa, en este caso como plataforma web y como proyecto vamos a usar estos lenguajes:
 1. HTML: Lenguaje de Etiquetas usado para la creación de páginas web en lo que a interfaz de usuario se refiere.
 2. CSS: Lenguaje usado conjunto con HTML para dar estilos y dinamismo a las etiquetas declaradas en los archivos HTML
 3. JavaScript: Lenguaje lógico que permite programar funciones dinámicas y lógicas, operacionales y de variables para crear funciones complejas en una página Web



4. TypeScript: Basado en JavaScript y en unión con C# para el uso de clases y objetos dentro de este, de código abierto, con más funcionalidades que JavaScript, pero con el mismo objetivo.
 5. SQL: Lenguaje de consulta comúnmente usado en la mayoría de las bases de datos, permite la creación de estructuras de datos consulta y funciones complejas de una base de datos.
 6. JSX: Es una extensión de JavaScript que nos permite usar JavaScript, HTML y CSS en el mismo archivo. Fue creada por Facebook y se utiliza en los archivos de React.
 7. TSX: Es una extensión de TypeScript que nos permite usar TypeScript, HTML y CSS en el mismo archivo. También fue creada por Facebook y se utiliza en los archivos de React para habilitar el soporte de tipado estático y mayor seguridad en el código.
- Frameworks
 - FrontEnd: Para el FrontEnd se usará el framework NextJS 14 y como lenguaje JavaScript. Es un framework que se encargar la parte visual y lógica con la que interactuará los usuarios.
 - Backend: Para el Backend se usará NestJs, framework que usa TypeScript como lenguaje. NestJs es un framework para la creación de backends que funciona de manera modular lo que permite la escalabilidad de proyecto de manera sencilla y además la segmentación por funciones dentro de este. Este fue escogido ya que le proporciona al programador la solución a muchos procesos pesados de codificar y los brinda de manera sencilla



para usar, además de tener mucha documentación y una comunidad amplia.

- Tailwind: Es un framework de CSS que ayuda a mejorar la velocidad y flexibilidad de escritura y su funcionamiento en si se basa escanear todo lo HTML y componentes de JavaScript para así generar lo estilos correspondiente a la clase dada a la etiqueta.

2.3.5 Comunicación para microservicios

Para la comunicación de los microservicios se usará la tecnología llamada NATs la cual permitirá la interconexión y comunicación de los microservicios construidos para la aplicación, usando así una comunicación de mensajes que validad hacia donde se dirige esta solicitud y que servicio debe encargarse de esta.

2.3.5.1 NATs

Como método de comunicación para los microservicios se escogió NATs para suplir la necesidad de un método de comunicación entre el Gateway principal y cada microservicio gracias a las ventajas que esta posee.

Para esta selección nos basamos en lo que expone (Agung Nur Aprianto, 2024) sobre NATs en su estudio comparativo donde se muestran datos para respaldar la eficiencia de este método de comunicación.

Aquí se presenta una tabla en la que compara la estabilidad en cuanto la latencia que tiene NATs siendo comparado con RabbitMQ en el que podemos apreciar como este brinda una latencia consistente, aunque la carga y numero de datos sea mayor:



| Valor de datos | Latencia por el número de mensajes | |
|----------------|------------------------------------|----------|
| | RabbitMQ | NATs |
| 1 | 2.509000 | 1.820000 |
| 10 | 1577.940186 | 2.015900 |
| 100 | 1599.461060 | 1.824140 |
| 1000 | 1168.534546 | 1.563516 |
| 10000 | 1087.672241 | 1.352136 |

TABLA 2. Comparación de latencia entre RabbitMQ y NATs

Gracias a esta comparación podemos ver la gran estabilidad que tiene NATs en cuanto a su latencia en cuanto a un número mayor de datos siempre manteniéndose estable y con una latencia parecida, aunque el valor de los datos aumente.

2.3.5.2 RabbitMQ

Para introducir a Rabbit tomaremos parte de lo que dijo (Richardson, 2014), cuando nace RabbitMQ en el año 2006 se convierte en la tecnología de comunicación por mensajes que se usa en TI, esta no solo era una herramienta de mensajería si no que era una herramienta muy potente de intercomunicación, RabbitMQ utiliza el protocolo AMQP inicialmente usada en las empresas más atrevidas a la nueva tecnología, pero solo como un método de mensajería, cuando llego el Cloud Computing y las necesidades de una arquitectura descentralizada y comunicación entre estas RabbitMQ fue la opción que satisfacía las necesidades de ese tiempo.

Pero en comparación con las tecnologías actuales y una necesidad de transferencias de datos más altas en los softwares de estos momentos RabbitMQ es inferior en esa área según nos muestra (Agung Nur Aprianto, 2024) en su tabla de conclusión sobre la comparación de NATS y RabbitMQ



| Valor de datos | Latencia por número de Mensajes | | Latencia por tamaño de mensajes | |
|----------------|---------------------------------|------|---------------------------------|------|
| | RabbitMQ | NATS | RabbitMQ | NATS |
| 1 | X | ✓ | X | ✓ |
| 10 | X | ✓ | X | ✓ |
| 100 | X | ✓ | X | ✓ |
| 1000 | X | ✓ | X | ✓ |
| 10000 | X | ✓ | X | ✓ |

TABLA 3. Tabla de resultados sobre pruebas de latencia RabbitMQ y

NATS

2.3.5.3 Kafka

La herramienta de Kafka es una herramienta usada en la creación y comunicación de microservicios ya que esta es una plataforma que permite la transmisión de datos y la lectura de flujos de datos que pueden ser distribuidos entre distintos usuarios.

Para describir mejor la tecnológica nos apoyamos de lo que describe (Bejeck) refiriéndose a Kafka como un transmisor se eventos el cual consume o transporta la información obtenida de los suscriptores a estos trasmisores, este se propone como una manera de brindarle a los usuarios funciones dependiente de sus acciones, o como el los llama eventos, el hecho de querer ver una película, revisar los zapatos en una tienda digital, son tantos datos que se obtienen y tantos otros que se tienen que mandar al cliente que se necesita una manera eficaz para poder tomar un accionar ante esos eventos.

El también describe los eventos de manera técnica, un evento en Kafka consta de 3 partes. La llave (key) el identificador del evento el cual puede ser usado como ejemplo

en una tienda online el id de un cliente usado como identificador y desencadénate de eventos basado en lo que estuviera haciendo. El valor (value) es el evento en sí mismo la información producida o necesaria para el evento y la información de fecha y hora (timestamp) que obtiene cuando el evento ocurre, gracias esto define como Kafka en una estructura de microservicios en empresas como una manera de redirigir peticiones y funciones de usuarios a distintos servicios de manera sencilla y rápida cuando uno de estos realice un 'evento'.

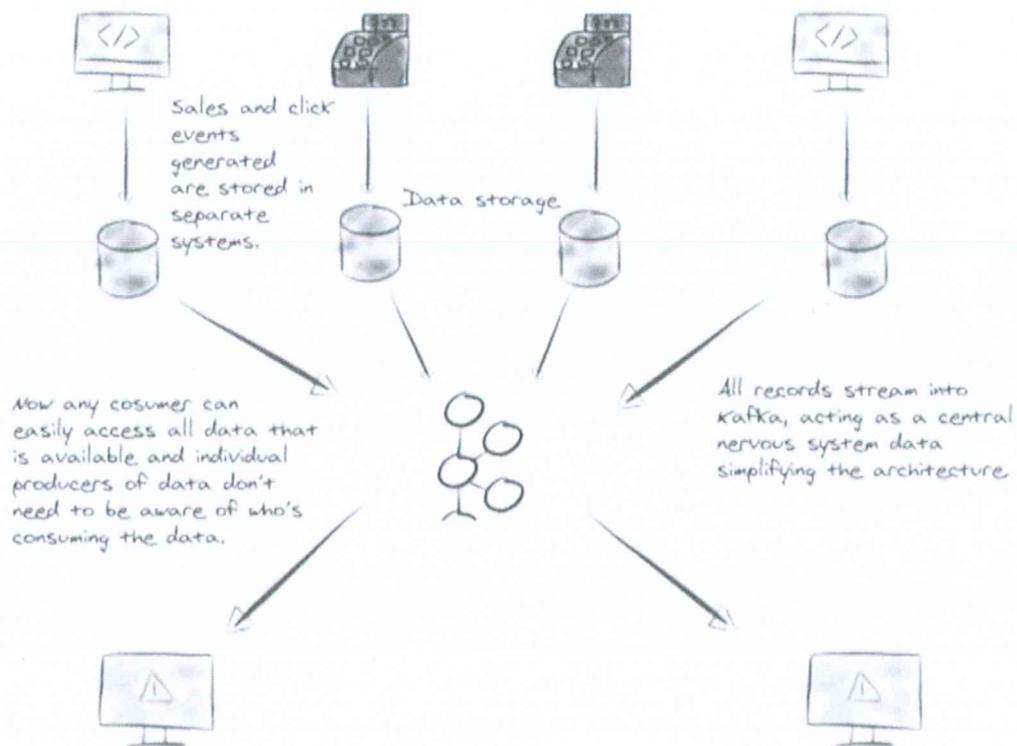


Figure 1.3 Using the Kafka event streaming platform with a simplified architecture

Figura 2. Ejemplo del uso de Kafka en un sistema simple, por Bill Bejeck

2.4 Análisis de las Herramientas

2.4.1 Herramientas Front End Web

(Doglio, F, 2024) comento lo siguiente: El desarrollo del FrontEnd existen variedades de herramientas realizada por la comunidad u organización con diferentes

propósitos. Por ejemplo, podemos encontrar como Figma que una herramienta destinada al diseño de interfaces, mientras que otra como Responsive te ayudan a mostrar el resultado de cómo se verían la página web simulado en varios dispositivos.

Entonces el momento de programar, tenemos varios framework que cada uno fue pensando para algo en específico. Por ejemplo, Angular, desarrollado por Google, ha recuperado popularidad con su versión 19 y muchas empresas lo están usando. React, de Facebook, es súper flexible y sigue siendo de los favoritos. Por otro lado, Vue.js nació porque su creador quería un enfoque más sencillo que React. Y no olvidemos Laravel para proyectos full-stack y Next.js, que lleva React al siguiente nivel con características avanzadas.

Entonces basado en eso conocimiento para saber que framework utilizar se hizo una tabla comparativa sobre la ventaja, desventaja y lenguaje que usan, continuación la tabla.

| Herramienta | Característica | Ventaja | Desventaja | Lenguaje |
|----------------|------------------|--------------------|---------------|------------|
| Angular | -Uso de DOM | -Alta calidad | -Curva de | TypeScript |
| | -Data Binding | -Desarrollo | aprendizaje | |
| | -Enlace | multiplataforma | -Mayor | |
| | bidireccional de | -Velocidad | complejidad | |
| | datos | -Web ligeras | -Sobrecarga | |
| | -Compatibilidad | | inicial | |
| | móvil | | | |
| NextJS | -Lenguaje JSX | -Fácil de Aprender | -Necesidad de | JavaScript |
| | -Virtual DOM | -Componentes | conocimiento | |
| | -Ciclos de Vida | React | | |



| | | | | | |
|----------------|-------------------------------------|----------------------------------|-----------------------------------|----------------------|---------------------|
| | -Gestión de estados | -DOM virtual | -Interfaces interactivas | -Peso de bibliotecas | solido en HTML y JS |
| | | -Documentación accesible | | -Solo interfaz | |
| Vue | -Modularidad | -Tamaño reducido | -Barrera del idioma | JavaScript | |
| | -Reactividad | -Componentes individuales | | | |
| | -Componente - Web | -Fácil de aprender | | | |
| | -Virtual DOM | -Sencilles de pruebas unitarias | | | |
| | -Mixins | -Comunidad | | | |
| | -SPA | -Documentación | | | |
| Laravel | -Arquitectura | -Fácil de usar | -Depende de | Php | |
| | -MVC | -Su documentación | Symfony | | |
| | -Los modelos trabajan con los datos | -Paquetes | | | |
| | -Estructura solida | -Sistemas de Rutas | | | |
| | -Artisan | | | | |
| | Modularidad | | | | |
| Django | -Api Robusta | -Seguro | -No compatible con ciertos plugin | Python | |
| | -Vistas genéricas | -Compatible con otros frameworks | | | |
| | -MVC | | -Pesado | | |



| | | |
|-------------------------|---------------------------------|-----------------------|
| -Soporte a múltiples DB | -Herramientas de administración | -Curva de aprendizaje |
| | -ORM | |

TABLA 4. Comparación de framework Frontend

2.4.2 Conclusiones sobre Herramientas Front End Web

De todos los frameworks analizados, se ha optado por utilizar Next.js debido a la facilidad que ofrece al desplegar en los servicios de Vercel, su documentación detallada y la posibilidad de implementar funcionalidades adicionales que otros frameworks no permiten. Además, Next.js permite utilizar de manera organizada los "server components" y el "server-side rendering" de forma eficiente.

2.4.3 Herramientas Back End

Los frameworks de Backend son muchos y para distintos tipos de lenguaje, estos pueden llegar a cumplir las funciones de una API REST muy fácilmente pero tienen características que los diferencian y pueden ayudar a sobresalir como Express la base de la mayoría de frameworks basados en JavaScript que contiene todas las herramientas necesarias para levantar un proyecto y muy flexible a la hora de cambiar este y hacer lo que necesites, o como Django siendo muy robusta en los procesos difíciles de hacer, pero en el contexto de los microservicios que se aplicara se compararon solo Django, Express, Nest y Ruby.

| Herramienta | Característica | Ventajas | Desventaja | Lenguaje |
|---------------|-----------------------------------|----------|-----------------------------------|----------|
| Django | -Api Robusta -Vistas genéricas | -Seguro | -No compatible con ciertos plugin | Python |



| | | | | |
|----------------|-----------------------------------|----------------------------------|--|------------|
| | -MVC | -Compatible con otros frameworks | -Pesado | |
| | -Suporte a múltiples DB | -Herramientas de administración | -Curva de aprendizaje | |
| | | -ORM | | |
| Express | -Middleware para petición HTTP | -Sencillos | -Mucha responsabilidad del desarrollador | JavaScript |
| | -Compatible con muchas BD | -Facilidad | -Menos estructura | |
| | -Fácil estructuración de app | -Comunidad | -Menos convenciones | |
| | -Flexible | -Rendimiento | -Baja robustez | |
| Nest | -Centrado en arquitectura | -Hace el trabajo pesado por ti | -Dependencias circulares | TypeScript |
| | -Orientado a objeto | -Coherencia | | |
| | -Inspirado en Angular | -Ejemplos Documentados | | |
| | -Capacidad de crear proyectos web | -Facilidad | | |
| | -Promueve el uso de TS | -Comunidad | | |
| Ruby | -Dinámico | -Comunidad | -Lenguaje interpretado | Ruby |
| | -Flexible | -Múltiple librería | -Alto uso de memoria | |
| | -Legible | -Actualizaciones constantes | | |
| | -OpenSource | | | |



| | | |
|------------|----------------|-------------|
| -Expresivo | -Facilidad | -No soporta |
| | -Permite crear | paralelismo |
| | DSLs | -Decrece en |
| | | popularidad |

TABLA 5. Comparación de framework Backend

2.4.4 Conclusión Herramientas Back End

En conclusión de la comparación sobre todas las opciones de Frameworks BackEnd para el desarrollo de este proyecto de titulación se seleccionara a NestJs como la mejor opción ya que es un framework basado en módulos de lenguaje TypeScript creado en base al Framework de Express, Esta es la mejor elección en cuanto a la arquitectura de *microservicios que se va a desarrollar, ya que esta te ayuda a simplificar las tareas más complejas de crear manualmente y configurar para su funcionamiento, además tiene librerías exclusivas para el uso y configuración de microservicios y como extra presenta pocas desventajas en comparación a las demás herramientas analizadas previamente.*

2.4.5 Herramientas de Base de Datos

Dentro de las bases de datos podemos encontrar muchas opciones para estas pero si vamos a las más populares la lista se reduce dentro de todo, la mayoría con opciones muy viables y muy buenas, MongoDB siendo una base no relacional muy liviana y sencilla de conectar o su versión en la nube llamada Mongo Atlas es muy buena en cuanto a portabilidad u otras como Oracle que es el extremo opuesto a este que aunque es muy buena requiere de pagos, conocimientos en esa base de datos y es muy exigente en cuanto al uso ya que tiene un gran consumo de recursos en su ejecución.



| Herramienta | Característica | Ventajas | Desventaja |
|-------------|------------------------------------|---------------------------------------|--|
| MongoDB | -Código abierto | -Gran rendimiento | -Problemas de fiabilidad |
| | -Escalado Automático | -Soporte a esquemas dinámicos | -Problemas con diseños sin esquemas |
| | -Gran disponibilidad | -Lenguaje de consulta enriquecido | |
| | -Similitud a objetos JSON | -Alta disponibilidad por replicación | |
| | | -Escalabilidad Horizontal | |
| MySQL | -Conectividad TCP/IP, Sockets Unix | -Hasta 64 índices por tablas | -No incluyen características de objetos como la herencia |
| | -Alta estabilidad | -Sistema de privilegios y contraseñas | |
| | -Múltiples métodos de almacenado | -Verificación basada en host | |
| | -Optimización multiprocesador | -Soporte de caracteres | |
| PostgreSQL | -Gran escalabilidad | -Mecanismos de Abstracción | -Complejidad |
| | -Rollback | -Fácil reutilización | -Rendimiento en consultas complejas |
| | -Integridad referencial | -Modelado natural | -Curva de aprendizaje |
| | -Herencia de tablas | -Alta seguridad | |
| | -Acceso concurrente | -Escalabilidad | |
| | -Comunidad | | |
| | -Herramientas de administración | | |



-Uso por medio de
contenedores

Oracle

-Modelo relacional

-Interfaz gráfica de
administración

intuitiva

-Control de acceso
avanzado

-Seguridad de datos
integral

-Lenguaje de
programación de

bases de datos

completo (PL/SQL)

-Disponibilidad
continua

-Gestión de usuarios
simplificada

-Permite a los usuarios crear
segmentaciones de datos

-Aumenta la seguridad de
los datos y facilita un

proceso de transacciones sin
inconvenientes

-Optimiza de manera
efectiva las consultas

complejas

-Ofrece un soporte dinámico
que simplifica la integración

sin problemas con diversas
plataformas

-Proporciona sugerencias

inteligentes que asisten a los
desarrolladores en su trabajo

-Su estructura de tarifas no
resulta adecuada para

empresas emergentes y
pequeñas

-La apariencia del diseño
parece menos favorable en

comparación

-La apariencia del diseño

parece menos favorable en

comparación con otras

-Requiere una cantidad

significativa de recursos y

experimenta retrasos

SQL Server

-Soporte de
transacciones

-Escalabilidad,
estabilidad y

seguridad

-Control semántico de los
datos

-Procesamiento de consultas
-Control de recuperación

-Lenguaje estandarizado.

-Sin habilidades de
codificación

-Costo

-Complejidad

-Limitaciones de
rendimiento



-
- Soporta procedimientos almacenados
 - Múltiples vistas de datos.
 - Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML
 - Además permite administrar información de otros servidores de datos
-

TABLA 6. Comparación de bases de datos

2.4.6 Conclusión sobre Herramientas de Base de Datos

Para concluir con la elección de la mejor opción entre todas las bases de datos se escogió PostgreSQL para el proyecto, no por una característica sumamente significativa ya que todas las opciones son muy buenas, si no más por un tema de la comodidad de uso, el primer hecho es que esta puede ser usada de manera gratuita, Su lenguaje mantiene mucho la base del lenguaje SQL, pero brindando también funciones propias de PostgreSQL y también permitiendo la alteración sencilla de las tablas o datos una vez ya creados siendo que en una arquitectura de microservicios con bases de datos separados son de gran ayuda para cambios necesario para añadir parámetros en tablas para el control



de entrada o edición de datos en una arquitectura con distintas bases de datos que conectan un solo programa.

2.5 Conclusión de las investigaciones

En conclusión gracias a las investigaciones realizadas se pudo obtener mucha información relevante en primer instancia sobre el tema principal a tratar, como el hecho de que en distintas entidades de educación superior ya se ha logrado proponer un software que solucione estos problemas, Gracias a estas también se pudo tener un referente sobre los problemas comunes que se pueden generar al momento de desarrollar el tema y el cómo se mejoró el área a tratar cuando se implementaron estos proyectos.

También la investigación en el apartado de herramientas tecnológicas aportó una visión más amplia sobre todas las opciones posibles para la implementación, pero a su vez un punto de comparación entre ellas basado en la visión principal de la arquitectura del proyecto para así de esta manera poder identificar los puntos fuertes y débiles a los que puedan presentar un impedimento en el desarrollo futuro de la propuesta ya que en algunos sus enfoques no están tan desarrollados para el ámbito que se tiene pensado, esto permitió hacer una selección más sencilla en torno a todas las opciones presentadas.



CAPITULO III



3 Marco Investigativo

En el presente capítulo se describirán los métodos y tipos de investigación que se llevaron a cabo. Además, se abordarán las técnicas de recopilación de datos utilizadas y los datos a analizar, los cuales nos permitirán la obtención de los resultados buscados, útiles para presentar una solución.

3.1 Tipo de Investigación

En la investigación propuesta se detallan los diferentes enfoques exploratorio y descriptivo. El enfoque exploratorio busca el descubrimiento de ideas que nos permitan formular hipótesis y obtener diferentes puntos de vista sobre el tema, lo cual facilita o ayuda a encontrar un enfoque para la solución propuesta y a considerar varias medidas o alternativas para lograr las características deseadas en la investigación. Además, el enfoque exploratorio servirá como base para la formulación de preguntas de investigación, específicamente para la encuesta, donde se identificarán las diferentes variables relevantes para el análisis posterior, se explorarán dinámicas subyacentes, y se identificarán los diferentes patrones que pueden emerger en una problemática. La revisión de literatura y el análisis preliminar de los datos nos permitirán definir conceptos clave.

El enfoque descriptivo se caracteriza por su diseño estructurado, que nos permite definir los diferentes puntos o atributos que pueden ser relevantes al tema, así como sus propiedades. Esto nos facilita las oportunidades de procesar y recopilar los datos obtenidos para la caracterización detallada de los diferentes puntos de vista expresados por los usuarios. Mediante técnicas cualitativas y cuantitativas, se clasificarán y analizarán las diferentes perspectivas o visiones que hemos considerado relevantes al *tema*.



3.2 Métodos de Investigación

Se empleará una metodología cualitativos para obtener una visión completa o panorama general del proyecto. Los métodos cualitativos se dedicarán a contextualizar y entender las opiniones de manera más profunda, capturando las diferentes experiencias de los participantes o es decir los involucrado.

El método cualitativo se utilizará para la contextualización y comprensión de las diferentes opiniones, comentarios y perspectivas de los involucrados. Este enfoque nos permitirá explorar la diferente profundidades, experiencias y percepciones de los participantes, proporcionando un entendimiento detallado y la interpretación de los datos a través de encuestas y otras fuentes cualitativas.

3.2.1 Método Teóricos

3.2.1.1 Método bibliográfico

El método bibliográfico es utilizado para la recolección de datos a través de las diferentes perspectivas que cada autor presenta en lecturas académicas, repositorios en línea, artículos web, entre otros. Esto nos facilita la extracción y, a su vez, la interpretación de los diferentes conceptos obtenidos que se encuentran presentes en el objeto de estudio.

3.2.1.2 Método Deductivo

El método deductivo fue utilizado durante la investigación con la finalidad de obtener conclusiones a partir del conocimiento previamente recopilado por medio de varias fuentes de información, fundamentando así los conceptos determinados que se han planteado en el desarrollo del proyecto. La esencia de este enfoque radica en la capacidad de extraer información lógica y partir de premisas previamente establecidas, lo cual nos proporcionará un marco conceptual sólido que orienta y respalda nuestro proyecto.



3.2.1.3 Método Inductivo

El método inductivo es crucial, ya que juega una parte importante en este proceso debido a que nos brinda la capacidad de generar ideas y formular hipótesis sobre cómo solucionar o ajustar las diferentes versiones de la solución del proyecto. Asimismo, al enfrentar la complejidad de resolución de problemas, esto lo convierte en una herramienta valiosa, ya que nos ayuda a identificar tendencias, patrones y relaciones entre ellas.

3.2.1.4 Método Técnico

El método técnico de la investigación proporciona diferentes perspectivas y opiniones de los involucrados, así como comentarios sobre posibles mejoras. Por ello, optamos por la aplicación de una encuesta como herramienta fundamental para la recopilación de datos. La elección de la encuesta como técnica está sustentada en la necesidad de capturar conocimientos prácticos, evaluar comentarios y opiniones sobre mejoras, y obtener experiencias directas de los participantes.

3.3 Estrategia Operacional para la Recolección de Datos

3.3.1 Población y Muestra

En lo que es la población se tiene que tomar en cuenta que estos serán los docentes, administrativos, estudiantes, directores de proceso y secretarios administrativos. Docentes los cuales tendrán el papel de Docentes tutores y también de jurado seleccionado para la recopilación de datos, el personal administrativo siendo este mediador en algunos procesos de titulación como los son los cambios de temas de



titulación académica, y los estudiantes como actores principales de la tesis y de los procesos a completar para la correcta finalización del proceso.

Pero de entre todos estos se seleccionó como muestra para la investigación al director del proceso y a la secretaria de este mismo para desarrollar una entrevista profesional porque al evaluar el tipo de software son los que más van a interactuar y administrar los datos, requerimientos y usuarios del mismo.

3.3.2 Análisis de las herramientas de recolección de datos a utilizar

En las herramientas que llevaremos a cabo para la recolección y análisis de datos necesarios para el software de gestión tendremos los siguientes:

3.3.2.1.2 Web

Como otro recurso informativo, la web deberá encontrar referencia bibliográfica sobre procesos similares en los que se hayan propuesto una solución parecida a la desarrollada aquí, así también obtener ideas que se pueden usar y también con estos datos esclarecer dudas que se pueden generar en cuanto a la incertidumbre de cómo se receptorá el sistema, ya que estos ya se finalizaron y pueden brindar esa información anticipada.

3.4 Plan de Recolección de Datos

Para obtener una comprensión profunda y precisa de las necesidades y expectativas de los actores clave del sistema, se implementará un plan de recolección de datos basado en una entrevista profesional detalladas. Este plan se desarrollará con varias preguntas focalizadas la máxima eficacia y relevancia de los datos recopilados. Con la entrevista desarrollada se procederá a seleccionar al participante de esta seguido de que se tome la encuesta y se implemente el análisis de las respuestas.



3.5 Análisis de los datos

El análisis de los datos es crucial para obtener conclusiones claras y útiles en esta investigación. A continuación, se describe cómo se llevará a cabo:

3.5.1 Preparación de los Datos

Primero, se revisarán y organizarán los datos recolectados. Esto implica limpiar las respuestas de algo repetitivas que se dieron en el momento de la entrevista y las redundancias de explicación que ya se presentaban en otras preguntas.

3.5.2. Análisis de Correlación

Se analizará las preguntas en base al tema de relación que este tiene con el tema implicado y hacia que área de esta se enfoca para poder catalogar las respuestas a cosas como una parte del proceso, documentos específicos o actores implicados para así obtener los datos necesarios para los requerimientos del programa.

3.5.3 Análisis Cualitativo

Las respuestas abiertas de las entrevistas se examinarán para identificar temas comunes y entender mejor las opiniones y experiencias de los participantes. Este análisis nos dará un contexto más profundo y ayudará a interpretar las necesidades y requerimientos que tienen los actores implicados en el proceso.

3.5.4. Interpretación de Resultados

Finalmente, se interpretarán los resultados obtenidos para evaluar si cumplen con los objetivos de la investigación. Esto permitirá sacar conclusiones y sugerir mejoras para el sistema de gestión de proyectos de titulación.



3.6 Análisis de los resultados

Basados en la entrevista profesionales hecha al Ing. John Cevallos siendo director del proceso de titulación actual de la facultad de Tecnologías de la información y a la secretaria Maria Elena Garcia encargada de los requisitos para la sustentación de los estudiantes, se expondrá las respuestas dadas a las preguntas hechas basadas en el área de aplicación del software desarrollado en esta tesis.

3.6.1 Entrevista a director de proceso de titulación

¿Cómo se lleva a cabo el proceso de presentación de tesis en su totalidad, desde la inscripción del estudiante hasta la entrega final de la calificación? ¿Cuáles son las etapas clave y qué acciones realiza cada actor en cada una de ellas?

Se necesita constatar los documentos listos para la sustentación, luego el docente tiene que emitir un certificado de que el trabajo de titulación está listo para sustentar, luego el docente tutor debe pasar el documento al anti plagio, luego el certificado, el documento de titulación terminado y el anti plagio se envía a la directora de titulación, se emite una lista de los estudiantes listos para la titulación, luego de eso se asigna el jurado que son dos jurados y la decana, luego se hace un proceso de revisión y calificación.

Pregunta Complementaria: ¿Cuáles son los principales desafíos o inconvenientes que enfrentan los estudiantes, los jurados y los coordinadores durante este proceso? ¿Se han implementado cambios recientes para solucionar estos problemas?

Los estudiantes muchas veces esperan que a lo últimos tienen inconveniente detipo administrativo, ya cuando están listo para sustentar no pueden hacerlo por eso problema, a veces es un poco difícil por ejemplo necesitas tener todas las asignaturas



aprobadas, la tienen aprobadas por ejemplo lo docente tiene que hacer un acta que le envía a secretaria y secretaria deber ser ingresada al sistema ese proceso dilatar. Otro problema que existe que lo tiempo es muy corto porque deber tener todo terminado pero todo porque el sistema se cierra como 3 semana máximo tienen que dar sustentaciones y hay tiene que ver revisiones , luego esa semana de revisiones puede haber una post sustentaciones pero hay estudiante que no aprueban entonces si analizamos el proceso debería tener 1 mes pero muchas veces solo son dos semana por eso mismo se realizan la pre sustentaciones para que ya vayan seguros a la sustentaciones y en la sustentación ya le dan por adelantado algunas parte para que el proceso no sea lento, tal vez la falta de automatización la mayoría se manejar con Excel, entonces muchas información de repetidas y se tratar de compartir pero cada docente manejar de diferente manera sus sistema.

¿Cuáles son todos los actores que intervienen en el proceso de presentación de tesis, y qué roles desempeñan cada uno de ellos (estudiantes, jurados, coordinadores, administrativos, etc.)?

Estudiantes proyecto y documentación, el tutor pendiente del estudiante y en la sustentación estar ahí, responsable de titulación estructurar los jurados, validación de los certificados. El estudiante avanzar parte de su proyecto e ir presentando lo avances con su respectivo tutor. Cuanto el tutor deber está pendiente del estudiante, revisar la parte de presentación, producto, coordina con el estudiante de pre-sustentación.

Secretaria coordina toda la parte de documento, certificado de acta, pendiente del docente que estén trabajando, certificado de tutor por el anti-plagio. Los jurados una revisión exhaustiva y rápida. Nota de calificación de documento y la tesis, para asegurar que documento está bien pero también el producto.



¿Qué tipo de documentos deben presentar los estudiantes para la sustentación de su tesis, y qué tipo de documentación es generada por los jurados después de la evaluación (como actas, comentarios, calificaciones)?

Los requisitos, después de la sustentación se genera un acta el mismo día y la secretaria ingresa en el sistema todo. Luego de la sustentación viene el acta y el mismo que termina la sustentación, la secretaria ingresar la nota y automáticamente de generar el acta y regresar a secretaria.

El repositorio digital se solicitar al estudiante que documento este firmado y escaneado de buena calidad y que se ubique en ese repositorio que luego será subido a la *institución*.

¿Cómo se pueden dividir los subprocesos de la presentación de tesis en etapas más específicas? Ejemplos: inscripción, revisión preliminar de la tesis, asignación de jurado, programación de la sustentación, evaluación y entrega de resultados.

Revisión de documentación, Entrega de documentación por docente, Asignación de jurado, Revisión que puede incluir una pre-sustentación, Asignación de fecha y hora de sustentación, sustentación y Generación del acta (documento final en repositorio).

¿Existen tiempos establecidos o recomendaciones para cada fase del proceso de presentación de tesis? ¿Cuáles son los tiempos estándar para la carga de la tesis, la asignación de jurados, la revisión, y la entrega de calificaciones?

Revisión se da desde el inicio del semestre, asignación inmediatamente se cumple requisitos, revisión por docentes una semana, se asigna la fecha después de petición de cambios, sustentación el mismo día, la generación del acta el mismo día, una semana para subir el documento.



¿En caso de retrasos o problemas inesperados (como la ausencia de un jurado o dificultades técnicas), ¿existen protocolos para extender los tiempos establecidos o para ajustar las fechas de entrega?

Si un jurado falta se reasigna rápidamente, si es del estudiante se reasigna, si es más complejo se reasigna el siguiente semestre. Se permite sustentación virtual en casos específicos. Si es por jurado se hacer el cambio inmediato, si hay un problema de estudiante si es problema es leve se dar una fecha extra, pero si es grave se le asignar el siguiente semestre. Si el caso es por enfermedades por no pueda sustentar grave o leve se puede hacer sustentación virtual. También se puede dividir la sustentación un estudiante puede dar y le otro es para próximo semestre.

¿Cómo se lleva a cabo el proceso de selección de los jurados para la sustentación de tesis? ¿Qué criterios se toman en cuenta (experiencia, especialización en el tema, disponibilidad)?

El jurador asignador depender del área de conocimiento o la lógica de negocio que llevar la tesis, también se tomar la experiencia de docente por cierto área o dominio a un tema.

**¿Cuántos jurados participan generalmente en la evaluación de una tesis?
¿Este número varía según el tipo de tesis o el área de conocimiento?**

Dos siempre son los que califican y revisan, allí la decana puede ser ella mismo que asignar alguien más. Por esta ocupada por la institución. También si hay algún problema por los dos docentes no se poner de acuerdo ella interviene con pregunta o tratar de solucionar el conflicto entre ellos.



¿Cómo se organiza la comunicación entre los jurados y los estudiantes durante el proceso de revisión de la tesis? ¿Los jurados pueden interactuar con los estudiantes para hacer preguntas o sugerencias antes de la sustentación?

Normalmente cuando hay problema se comunican con el docente, con la pre-sustentación tiene opiniones, que es validar. Normalmente en la pre-sustentación tiene comunicación directa.

¿Existe algún miembro especial dentro del jurado, como un presidente o moderador, que desempeñe un rol adicional o tenga más responsabilidades que los otros jurados? ¿Cómo influye este rol en el proceso?

Quien dirige el proceso de cada etapa de la sustentación, toda la información base, también hace pregunta para que el jurado pueda tomar decisiones. También el momento que se analiza la calificación tiene voz voto, solo solucionar si es que hay un problema que no esté de acuerdo.

¿El miembro especial tiene algún poder adicional en caso de desacuerdos entre los otros jurados o en la toma de decisiones finales?

Es quien dirige el proceso de sustentación, dirige las etapas es el que emite la información básica de la sustentación, tiene voz, pero no voto si no hay conflicto.

3.6.1.1 Conclusiones de la entrevista

Los datos importantes para resaltar y de alta prioridad para el software serían, los actores principales en el proceso antes de la sustentación como los son los docentes tutores de tesis y los jurados que se encargan de ver y revisar las tesis de los estudiantes, otro tema a tomar en cuenta son los documentos habilitantes y previos para la sustentación que son los documentos habilitantes donde se destaca el problema puntualidad al entregar.



3.6.2 Entrevista a secretaria encargada de los requisitos de sustentación

¿Cuánto tiempo toma subir la información de datos generales y cómo se hace este proceso?

Subir la información depende de cada estudiante. Normalmente, todo se entrega digitalizado en formato PDF o como documento físico. Estos datos son importantes porque sirven para elaborar el acta de grado y también incluyen información sobre dónde se hizo la práctica, el tema de vinculación, y los proyectos que se desarrollaron.

¿Cuánto tiempo tienen los estudiantes para entregar esta información?

Idealmente, los estudiantes deberían entregarla una vez finalizada la práctica, porque se necesita para legalizar cada periodo. Pero a veces son irresponsables y no entregan a tiempo, así que se hace un seguimiento mensual para asegurarse de que cumplan.

¿Cómo se organiza la recepción de documentos?

Se hace una planificación, semestre por semestre. Hay un periodo establecido, como una semana, para que los estudiantes entreguen todo. Luego, la Secretaría General *recopila los documentos y verifica que estén completos porque el proceso de defensa de tesis es a nivel institucional y todas las unidades académicas trabajan en lo mismo al mismo tiempo.*

¿Qué pasa si un estudiante cumple con todo?

Cuando un estudiante cumple, puede avanzar en el proceso. Pero deben esperar a que los docentes suban las últimas notas antes de continuar. A veces, eso retrasa un poco porque el récord académico demora más tiempo en generarse. Luego, el analista general revisa todo y confirma que esté completo.



¿Qué documentos son necesarios para el registro del título?

- Copia del título de bachiller.
- Otra certificación relacionada al bachillerato (creo que es un acta o constancia).
- Dos copias de la cédula.
- Récord académico.
- Certificado de no adeudar a la institución.
- Certificado de devolución de libros a la biblioteca.
- Si vienes de otra universidad, también necesitas todos los documentos de ingreso, como los de fiscalía y otros trámites habilitantes.

El récord académico es lo que más demora.

¿Qué pasa si un estudiante no cumple con los requisitos?

Si no cumplen, no pueden avanzar en nada. Es su responsabilidad. Por ejemplo, si *no sustentan en este semestre, tendrán que hacerlo el próximo. Pero si dejan pasar más tiempo, podrían tener que repetir toda la fase de resultados, lo que también tendría un costo extra.*

¿Qué pasa antes y después de la sustentación?

Antes, se verifica que toda la documentación esté completa. Después, solo queda el acta de grado, que confirma que el estudiante ya cumplió con todos los pasos.

¿Qué problemas o imprevistos suelen enfrentar?



Lo más común es que algunos estudiantes no entreguen la documentación a tiempo. Eso hace que el proceso sea más lento y que algunos tengan que esperar más tiempo. Pero la Secretaría se encarga de hacer un seguimiento y avisar cuando algo falta.

¿Cómo es el proceso después de recibir la documentación?

Primero, la Secretaría revisa que todo esté completo y lo notifica a la Secretaría General. Ellos organizan el proceso de sustentación y avisan a los estudiantes que ya están listos. Es como una cadena donde todos tienen que hacer su parte para que el estudiante pueda avanzar.

3.6.2.1 Conclusiones de la entrevista

Después de realizar la entrevista se pueden destacar dentro de toda la información el hecho de que parte importante previa a la sustentación son los documentos habilitantes a esta que gestiona la secretaria para que luego sean enviados a los superiores de la institución para poder validar la sustentación y la entrega de título después, pero estos son importantes para el proceso previo de la sustentación junto con su misma asignación.

3.7 Conclusión del Marco Investigativo

Para concluir el marco investigativo se obtuvo esta información relevante, Gracias a la entrevista al director actual del proceso de titulación en la Facultad de Tecnologías de la Información se obtuvo la información necesaria sobre el proceso en términos generales, pero también en base al enfoque que se tiene como personal administrativo de primer nivel en el proceso, información de provecho para la creación y requisitos de funcionalidades para los usuarios administrativos que usaran la plataforma, añadido a eso esta entrevista además se nos proporcionó los datos necesario para el enfoque de docente tutor y jurado de sustentación que son también actores importantes e implicados



directamente en el proceso de titulación, con la entrevista planteada datos como tiempos y flujos de trabajo de dicho procesos fueron obtenidos que son parte esencial para definir los casos de uso para cada usuario.

En la entrevista hacia la secretaria encargada de los documentos habilitantes del proceso de titulación se pudo identificar que la parte del estudiantado tiene un papel muy importante en cuando al hecho de mantener su matriculación en regla pero que no es un tema que deba llevar la parte administrativa para la sustentación gracias a esto se pudo filtrar los documentos realmente necesarios para la habilitación de este proceso, como lo son las certificaciones de prácticas, la de los idiomas necesarios en la carrera, y los *certificados para la vinculación con la sociedad, información importante para las funcionalidades que tendrán los usuarios administrativos y requerimientos que pedirá el sistema a la hora de su desarrollo.*

En general gracias al trabajo investigativo que se realizó en este capítulo se pudo extraer información de gran valor directamente de los actores principales y que están directamente relacionados con el proceso en cuestión que es de gran valor en base a su experiencia y posición en este, ya que con este se puede sacar un gran valor al poder alinear el foco del desarrollo que se tendrá con la información obtenida por parte de estos.



CAPITULO IV



4. Marco Propositivo

4.1 Descripción de la Propuesta

El objetivo de la propuesta es desarrollar una plataforma web que permita la asignación de manera correcta de las fechas para la pre-sustentación y sustentación de las tesis de los estudiantes registrados en un proceso de tesis facilitando así la gestión de estas, y brindando una plataforma donde consultar estas.

Además de esto la plataforma tiene el objetivo de facilitar el proceso anterior a su *presentación como lo es la revisión de las tesis por parte de los jurados y los comentarios* o cambios que requieran hacer en el documento o propuesta para que este proceso sea más inmediato y con una comunicación más directa dentro de este programa.

Se proporcionará una forma de guardar los documentos necesarios, generados y después del proceso en la nube, para consultar y acceder a estos de manera sencilla y centralizada para suplir la necesidad de estos en el progreso del proceso.

4.1.1 Diseño de la arquitectura

La arquitectura de esta aplicación es la arquitectura clásica de tres niveles la cual se define como: Primer nivel FrontEnd o el nivel de interacción con el usuario. El segundo nivel el de Backend el cual es el intermediario del primer y tercer nivel de la aplicación siendo este el cual interactúe con los datos y le envíe los que requiera el primer nivel. El tercer nivel o el nivel de base de datos este es el que se encargará de guardar los datos de

la aplicación y donde el segundo nivel guardara consultara o eliminara datos según sea necesario.

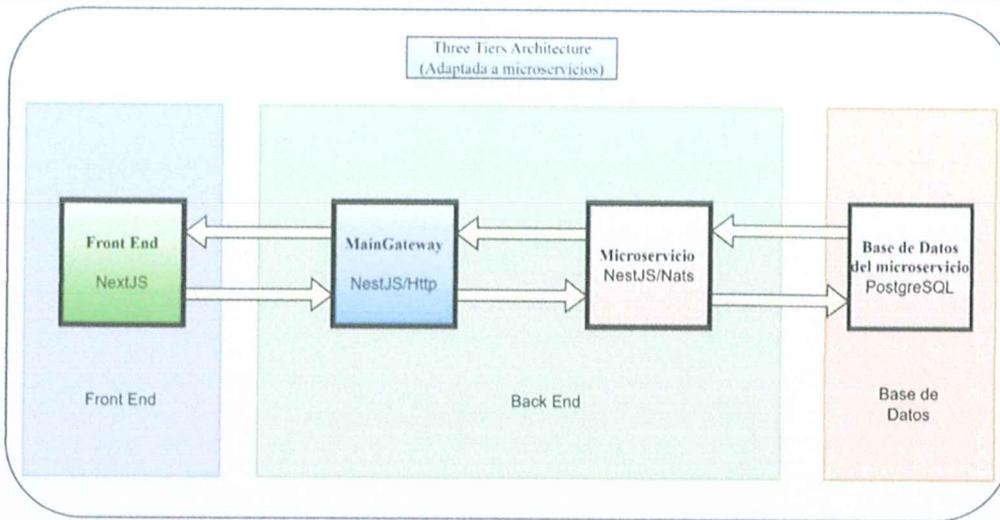


Figura 3. Arquitectura de la aplicación

4.1.2 Backend

En el Backend se optó por la distribución de tipo microservicios con un broadcast para la centralización de la comunicación entre todos los servicios requeridos dentro del sistema. Arquitectura del BackEnd

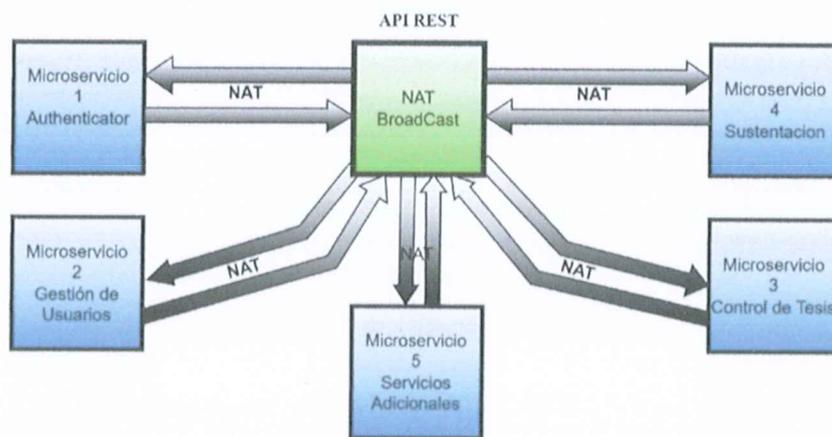


Figura 4. Arquitectura del BackEnd

4.1.3 Microservicio

Para la arquitectura del microservicio se va a utilizar la arquitectura típica de los módulos de Nest y sus componentes para armar cada uno de estos, pero haciendo uso de las funciones de la biblioteca microservices para integrarla a la red de comunicación y que además esta pueda oír los mensajes enviados por el Main Gateway para que el microservicio se encargue de la lógica desarrollada en el EndPoint al que se está llamando.

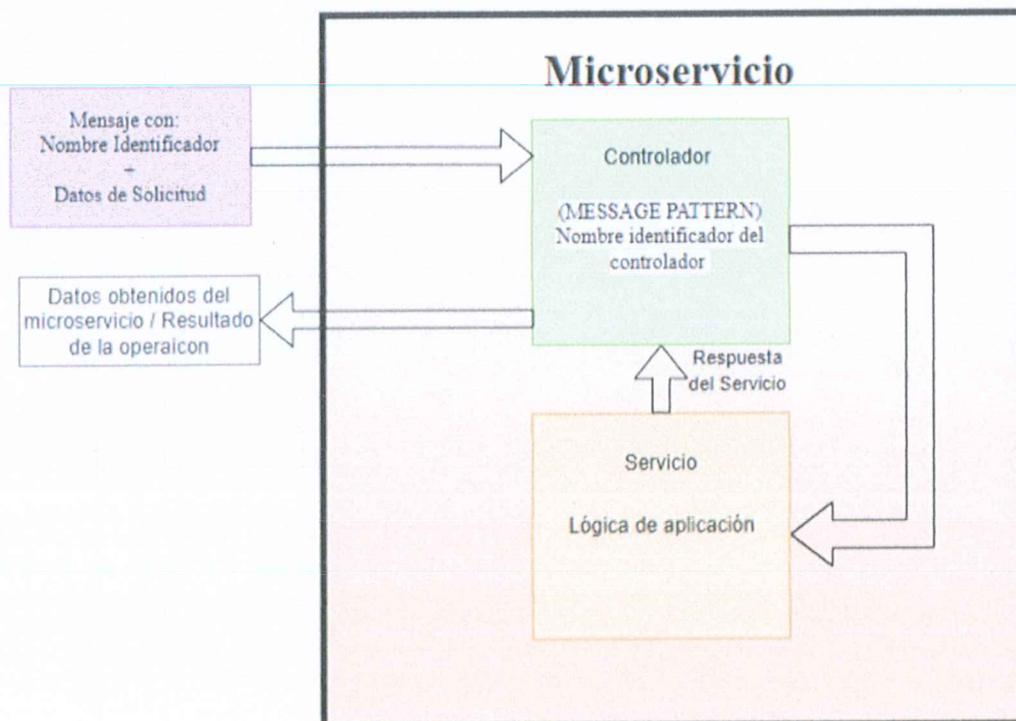


Figura 5. Arquitectura y lógica de un microservicio

4.2 Determinación de Recursos

Para lograr el desarrollo y el despliegue sea exitoso de la plataforma basada en microservicios, es necesario identificar y la asignación adecuada de los recursos



necesarios. A continuación, se describirán los recursos económicos, humanos y tecnológicos que se requiere para establecer el proyecto.

4.2.1 Humanos

En este punto se identifican los roles y funcionalidades de cada participante que forma parte del proyecto.

| Recursos Humanos | Rol | Responsabilidades |
|------------------------------|-----------------|--------------------------|
| Arteaga Meza | Desarrollador y | Desarrollador de |
| Anthony Rogers | Documentador | microservicios y llevar |
| | | la documentación de |
| | | los procesos de |
| | | desarrollo. |
| Velásquez Realpe Leif | Desarrollador y | Desarrollador y |
| Wither | Tester | Diseñador de la |
| | | plataforma y prueba de |
| | | funcionalidad y |
| | | usabilidad. |

TABLA 7. Recursos Humanos



4.2.2 Tecnológicos

Los recursos tecnológicos se mostrarán en la siguiente tabla donde se visualizará la tecnología utilizada en proyecto con su respectiva descripción.

| Hardware | Descripción |
|----------------|---|
| 2 computadoras | Equipos computaciones para realizar la ejecución de desarrollo de la web. |

TABLA 8. *Recuso tecnológico (Hardware)*

| Software | Descripción |
|--|--------------------|
| IDE | Visual Studio Code |
| Sistema de Control de Versiones | Git |
| Herramienta de simulación de dispositivo | Resposinve |
| Framework | Next js y Nest js |
| SGBD | Postgresql |
| Servidor Web | Ubuntu |
| Plataforma de Despliegue | Vercel y Cpanel |

TABLA 9. *Recursos Tecnológicos (Software)*



4.2.3 Económicos

En lo siguiente se describen los costo económico que se requieren a la hora de desarrollar un proyecto, que son recurso necesario. Tomando en cuenta que los desarrolladores tuvieron un aproximado de 480 horas trabajada.

| Cantidad | Recurso | Característica | PVU | Sub Total |
|--------------|--------------------------|---|---------------|---------------|
| 2 | Dominio | Go Daddy | \$1 | \$ 2 |
| 1 | VPS | Digital Ocean | \$5 / mes | \$ 10 |
| 1 | VPS | Atukhosting | \$12 / mes | \$ 24 |
| 1 | Desarrollador Frotend | Pago a desarrollador - Horas trabajadas: 480 | \$5.50 /horas | \$ 2640 |
| 1 | Desarrollador Backend | Pago a desarrollador - Horas trabajadas: 480 | \$5.50 /horas | \$ 2640 |
| Total | | | | \$5316 |

TABLA 10. Recursos Económicos

Cabe aclarar que dominio de go daddy que se utilizo fue “Online” es con un precio significativamente bajo solo por un año, pero su desventaja es que a extender este dominio por mas año te sale más costoso que otros y además de encontrarlo siempre en descuento sale beneficioso para hacer varias pruebas si uno está comenzando en este mundo.



4.3 Etapas de acción para el desarrollo de la propuesta

4.3.1 Diseño del Modelado de la Base de Datos

Tabla Usuario: Esta tabla almacena los usuarios de la aplicación y sus datos que sirven para heredar a los distintos usuarios.

Tabla Carrera: Almacena las carreras existentes en el centro educativo superior

Tabla Rol: Almacena los roles que tendrán los distintos tipos de usuarios.

Tabla Jurado: Almacena los datos de un usuario tipo jurado, los cuales son parte de las sustentaciones.

Tabla Estudiante: Almacena los datos de un usuario de tipo estudiante, los cuales son participes directos de la sustentación de sus tesis.

Tabla Docente_Tutor: Almacena los datos de un usuario de tipo Docente Tutor implicados como docentes encargados de la tesis asignada.

Tablas Tesis: Almacena los datos necesarios de la tesis requeridos en la sustentación y procesos con jurados.

Tabla Sustentación: Almacena datos necesarios para la sustentación de las tesis como la tesis estudiantes y jurados implicados en esta.

Tabla Jurado_Sustentación: Contiene los datos enlazados de más de un jurado con las sustentaciones asignadas.

Tabla Requisito: Almacena la información sobre los documentos o procesos requeridos para llevar a cabo la sustentación.



Tabla Requisitos Cumplidos: Almacena la relación entre los estudiantes y los requisitos para comprobar el cumplimiento de estos.

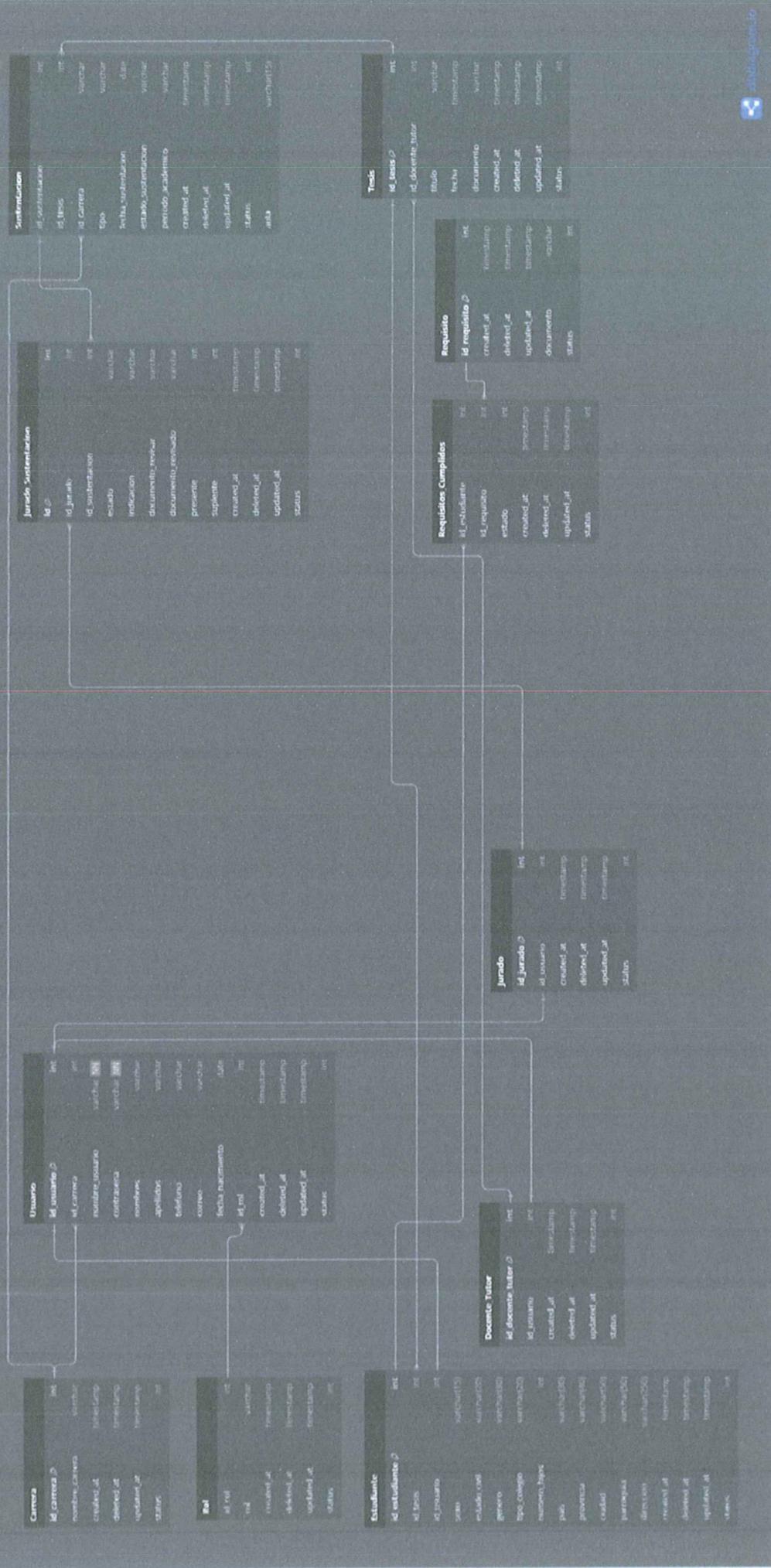


Figura 6. Tabla de base de dato



4.3.2 Diseño y Desarrollo el Backend

Para el desarrollo del Backend por microservicios se inició por el levantamiento de la red de comunicación:

Docker/NATs: Con Docker se pueden levantar servicios en de manera remota que se pueden usar de manera local, uno de estos es NATs, este presta el método de comunicación son el llamado al puerto respectivo en este caso el puerto 4222 para la comunicación de este, con levantar el contenedor del servicio y habilitar el puerto la comunicación está habilitada para los servicios que lo requieran.

Diagram.io/PostgreSQL: Su utilizo diagram.io para la esquematización de la base de datos para la aplicación ya que esta cuenta con una manera sencilla de armarla basándose en código como si fuera un ide de programación sencillo intuitivo y muy bueno, después de su uso se pasó este esquema a SQL y se crearon la base de datos para los servicios con PostgreSQL por su sencillas y potencia.

Microservicios/NestJs: Con el contenedor de Docker levantado ahora se necesita hacer uso de este servicio. Con el framework de NestJs y su biblioteca de Microservicios se levantó un proyecto principal llamado MainGateway que es un API de tipo REST que gestiona las entradas de peticiones HTTP para redirigirlas a mensajes que conectan los servicios en la red NATs del contenedor, después se crearon los microservicios que comunicaran con el MainGateway y sus respectivas bases de datos, creando un proceso explicado así:

4.3.3 Diseño y Desarrollo en el FrontEnd

Figma: Se utilizó Figma con el propósito de diseñar las diferentes interfaces de la plataforma web. Esto permitió crear interfaces detalladas y visualizar el diseño del



producto final, asegurando que cumpliera con los diferentes objetivos definidos para el proyecto.

Next.js 15: Se usó Next.js 15 para el desarrollo de la plataforma web en la parte de cliente, es decir, la parte visual. Este framework basado en React, agilizó y mejoró la eficiencia del proceso de desarrollo gracias a las funcionalidades que ofrece. Además, se integraron librerías adicionales para complementar funcionalidades específicas.

Git/Github: Se utilizó Git para el control de versiones del proyecto, complementado con la plataforma GitHub para el alojamiento del código. Estas herramientas permitieron una gestión sencilla y eficiente del desarrollo.

Vercel: El proyecto fue desplegado en la plataforma Vercel, que facilita la gestión y publicación de este. Al vincular la cuenta de GitHub, todos los proyectos alojados allí pueden publicarse con pocos pasos. Además, cuenta con una integración de despliegue automatizado que actualiza el proyecto a medida que se suben cambios a GitHub.

4.4 Etapas de acción para el desarrollo de la aplicación

Fase I: Definición y Planificación (1 de abril - 15 de mayo)

- Definición de los requisitos funcionales y no funcionales del sistema
- Modelado y Diseño de la arquitectura del sistema
- Casos de uso
- Diagramas UML de los casos de uso
- Modelo lógico entidad relación de la base de datos del software

Fase II: Diseño y Desarrollo (16 de mayo - 15 de agosto)



- Definición de las historias de usuario
- Definir el método de comunicación y estructura lógica entre el Main Gateway y los microservicios
- Planificación de los métodos REST base para cada entidad de la API
- Desarrollo de Especifico de cada método REST base planificado.
- Desarrollo de interfaces de usuario en para el sistema.

Fase III: Configuración y Despliegue (1 de septiembre - 27 de noviembre)

- Configuración de los VPS para alojamiento de los microservicios y base de datos (Digital Ocean y Cpanel)
- Configuración del dominio con CloudFlare para el FrontEnd
- Despliegue de la aplicación FrontEnd y Backend

4.4.1 Fase I: Definición y Planificación

En la planificación del desarrollo de la plataforma se hizo uso de la Herramienta Airtable donde se fue dividiendo por módulos y su actividad respectiva para cumplir el plan de desarrollo que van por varias fases con su fecha de inicio y fecha de finalización. A continuación, se mostrará una imagen de los módulos en la herramienta utilizada.

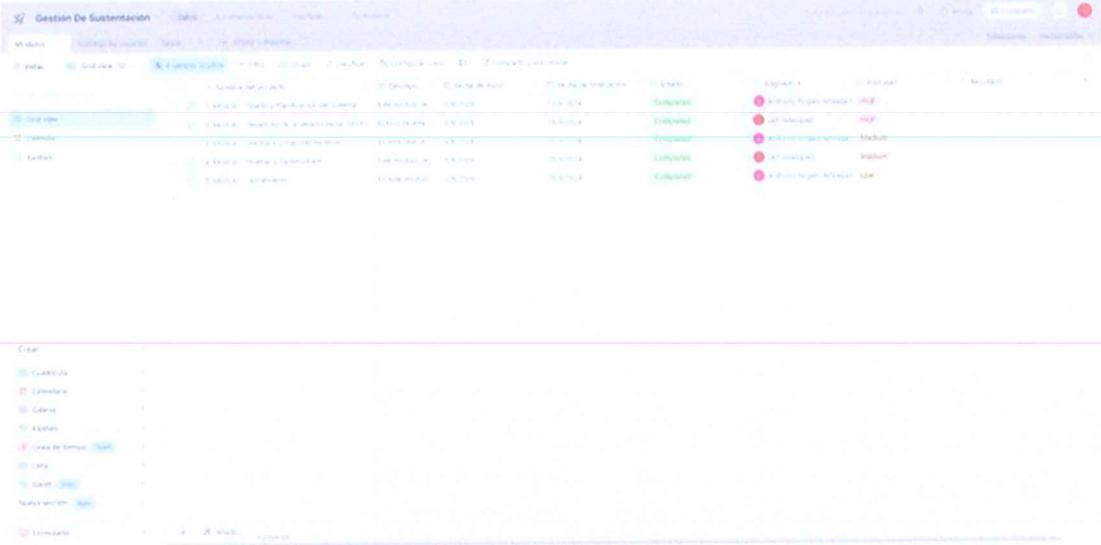


Figura 7. Planificación de los módulos de desarrollo en Airtable.

Entonces cada módulo tiene sus respectivas actividades por realizar que son asignada a la persona capacitada a esa actividad y lo cual también cuenta con un nivel de prioridad cada tarea. A continuación, una imagen de las actividades de los módulos.



Figura 8. Planificación de las tareas de desarrollo en Airtable.

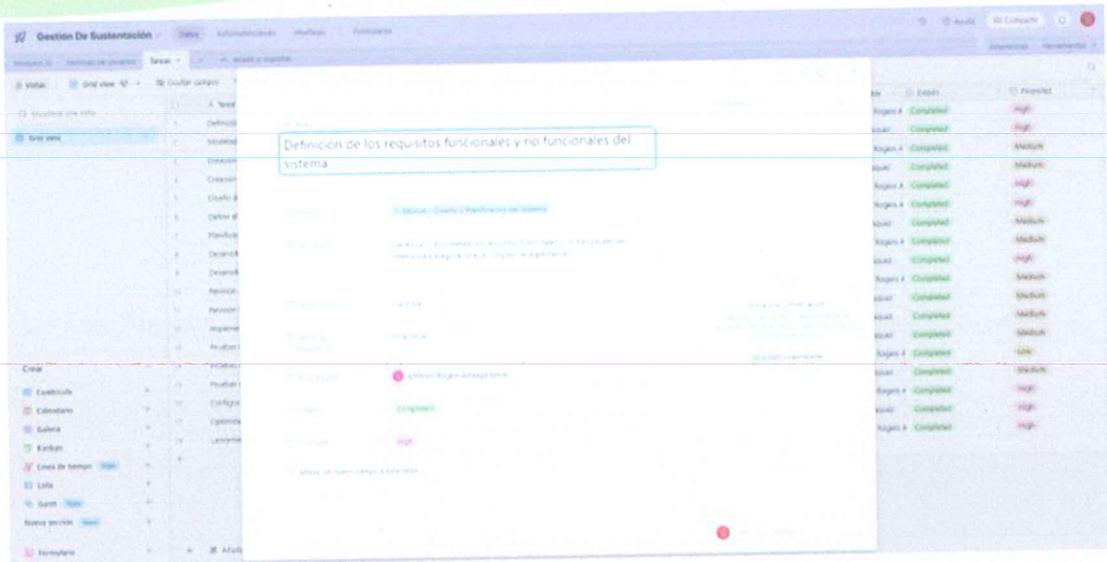


Figura 9. Desarrollo de las actividades de la planificación en Airtable.

Como primera fase comenzaremos con la siguiente actividad que son levantamiento de requisitos y funcionalidades, además la parte de diseño y usabilidad.

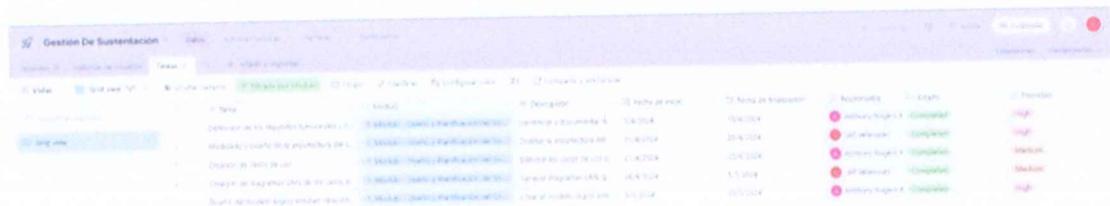


Figura 10. Primeras actividades de la fase 1.

4.4.1.1 Requisitos del sistema

Funcionales

Se tiene que establecen los requisitos de funcionamiento que se van a desarrollar en la aplicación para poder tener en cuenta los puntos importantes a la hora de ir



desarrollando cada una de las funcionalidades y que estas cumplan lo planificado de esta manera:

- El sistema permitirá el registro y autenticación de usuarios, incluyendo estudiantes, docentes y jurados.
- Se asignarán roles y permisos a los usuarios según su función en el proceso de titulación. Los usuarios podrán subir y almacenar documentos necesarios para el proceso de titulación.
- Los documentos podrán ser verificados y aprobados por los docentes y jurados.
- Se enviarán notificaciones automáticas a los usuarios sobre cambios la fecha asignada para la sustentación de la tesis.
- Validación y almacenamiento de los datos ingresados a la plataforma.
- Implementar un middleware para la protección de las rutas por roles.
- Establecer comunicación con los diferentes protocolos de comunicación entres servicios.
- Manejo de errores para detectar la causar del problema.
- Creación de interfaces intuitivas y responsiva para una experiencia fluida.
- Detección automática de la expiración del token y redireccionamiento al inicio de sesión.
- El sistema debe permitir la conexión o enrutamiento de solicitudes entre microservicios a través de un api Gateway que lo distribuya.



No Funcionales

En la parte de los requisitos no funcionales especificamos la manera lógica y tecnológica de cómo va a ser desarrollada la aplicación y ciertas funciones, es importante definir las ya de esta manera cada parte del equipo puede tomar en cuenta las limitaciones y características de las tecnologías o las lógicas implementadas y así tomar medidas a favor o en contra de estas de manera dinámica.

- El MainGateway recibe una entrada de un método http convencional de un servicio REST corriente.
- Se identifica la solicitud y al entrar por su controlador se inicia la función que envía por medio del NATs el mensaje que identifica a que método de que microservicio se quiere llamar.
- Al entrar el mensaje en la función correspondiente el servicio ejecuta la lógica que contiene este para devolverle al MainGateway la respuesta de esta.
- El MainGateway obtiene la respuesta del Microservicio y arma una respuesta estructurada para el FrontEnd dependiendo de cómo se resuelva la petición.
- Implementación de autenticación por medio de jwt

4.4.1.2 Modelado y Diseño los prototipos del sistema

Sobre el diseño de sistema se tuvo en cuenta los requisitos, roles y funcionalidades que necesitaba el sistema en cual se fue diseñando en una hoja de papel bocetos de como debería funcionar y navegar el usuario.

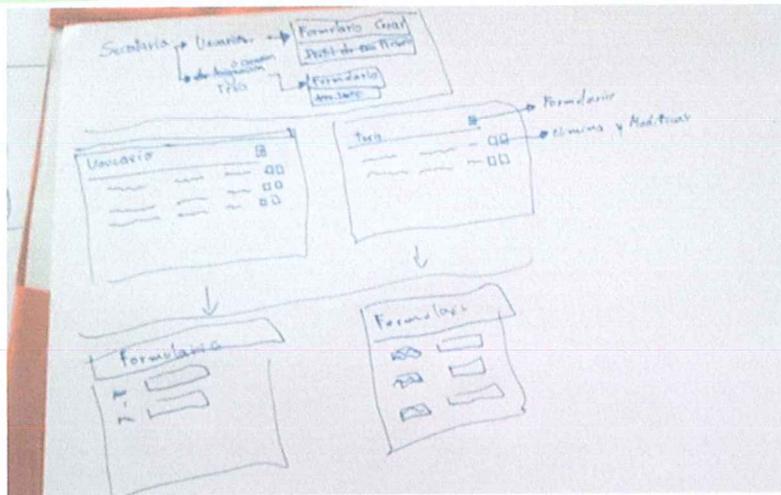


Figura 11. Primer Boceto del prototipo

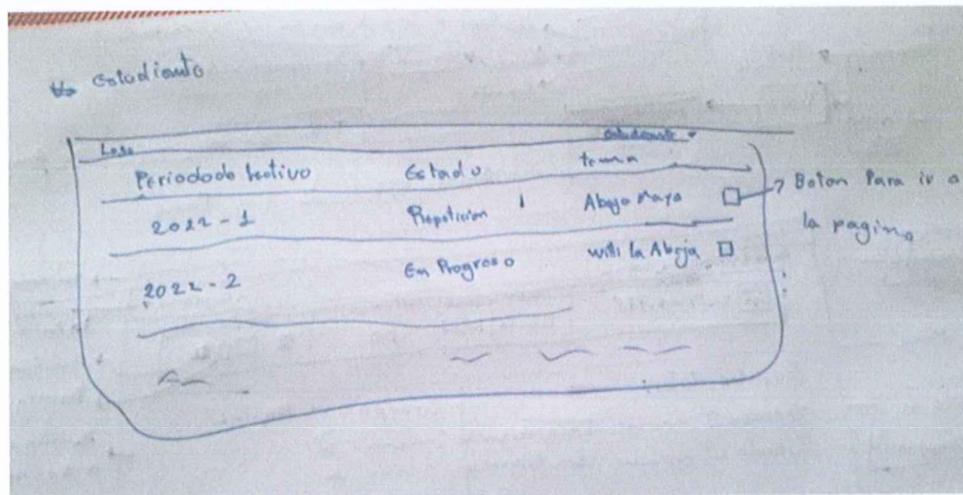


Figura 12. Segundo boceto diseño del prototipo

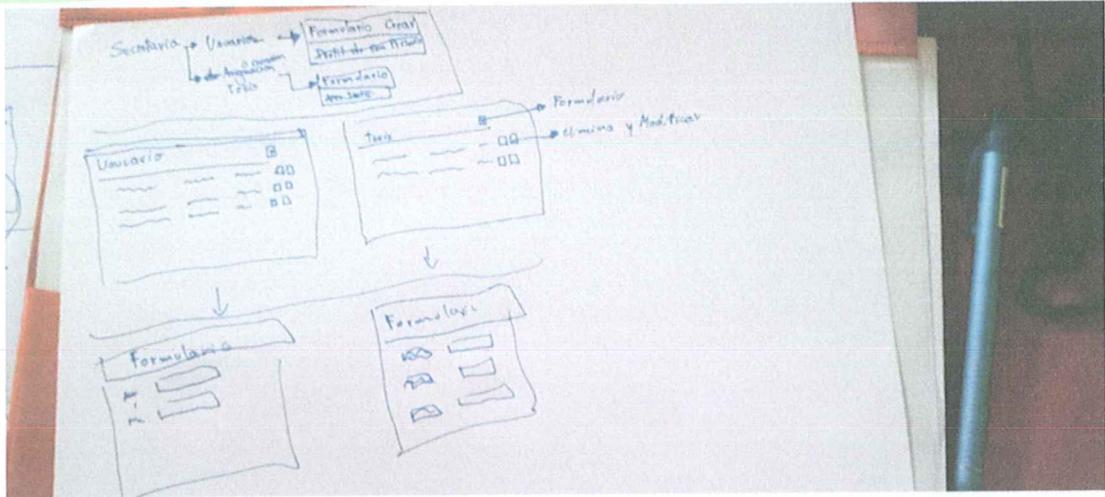


Figura 13. Tercer boceto diseño del prototipo

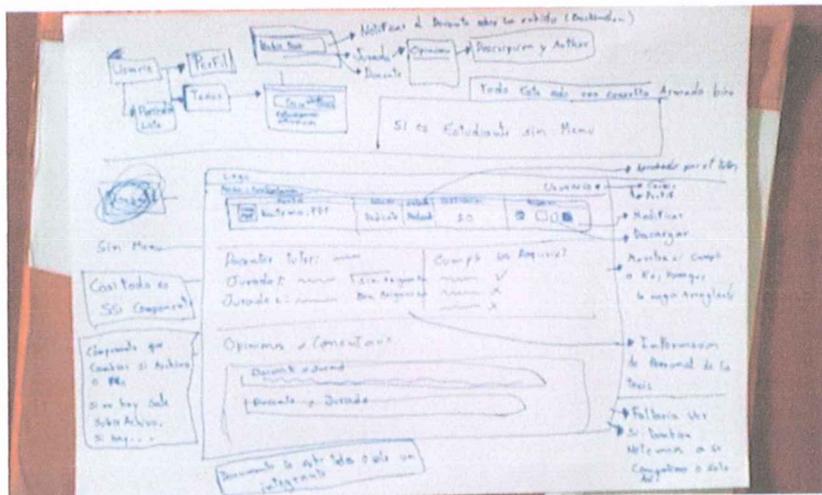


Figura 14. Cuarto boceto diseño del prototipo



Luego de eso se fue armando un prototipo visual con la herramienta de Figma tomando de referencia los bocetos anteriores realizada en hojas, en cual se fue dividiendo mejor la funcionalidades y control de sistema.

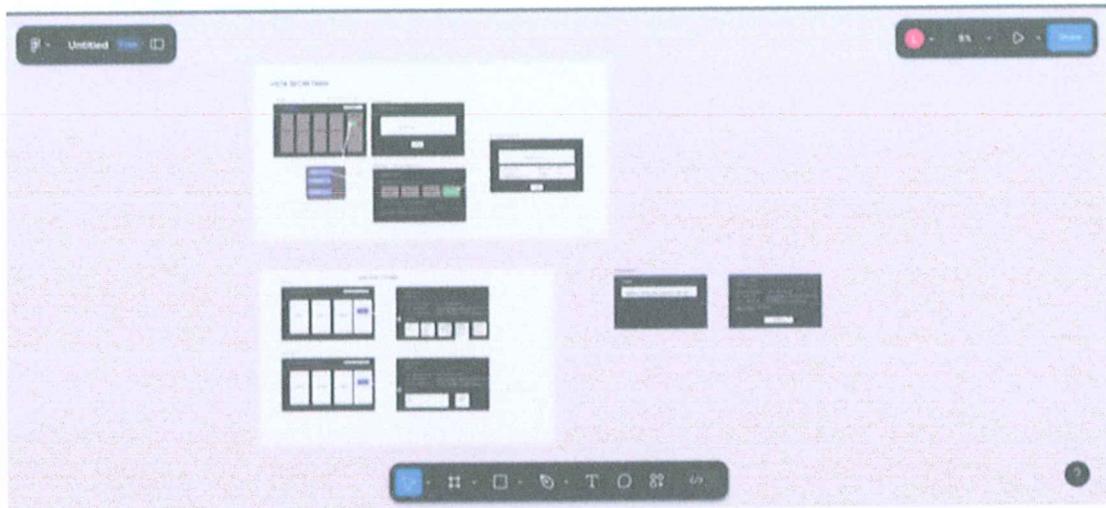


Figura 15. Vista de modelado del prototipo

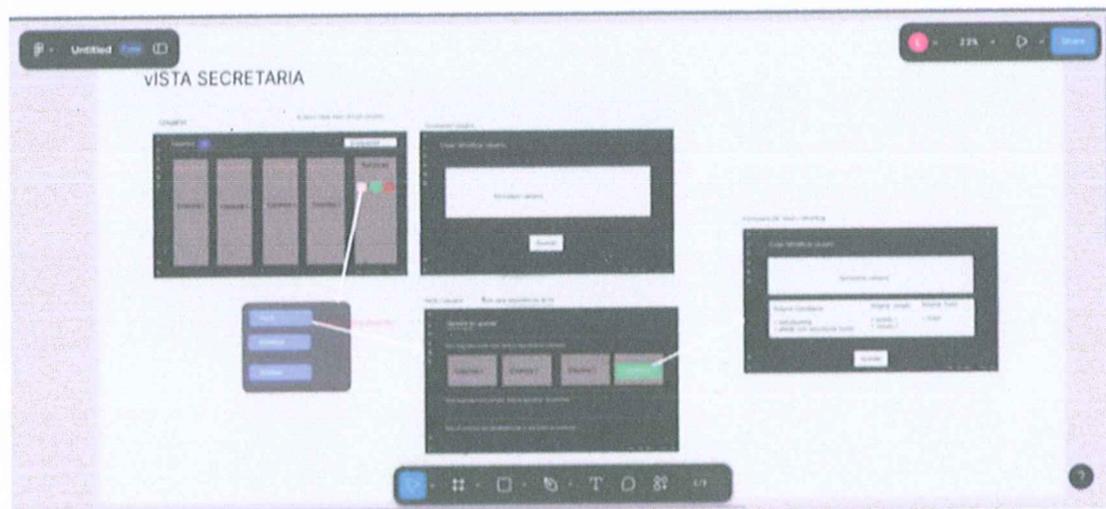


Figura 16. Modelado de Funcionamiento desde la vista secretaria

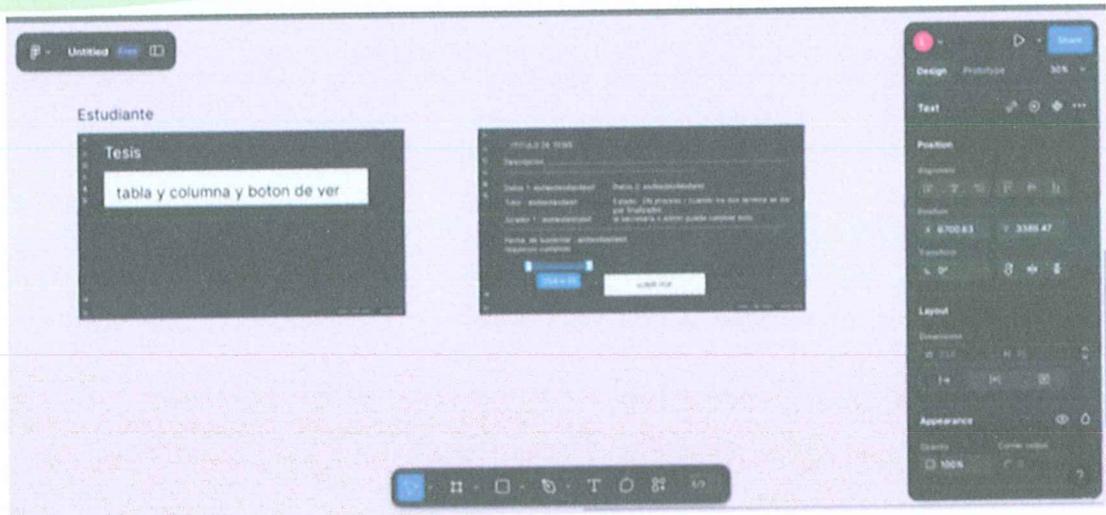


Figura 17. Modelado de Funcionamiento desde la vista estudiante

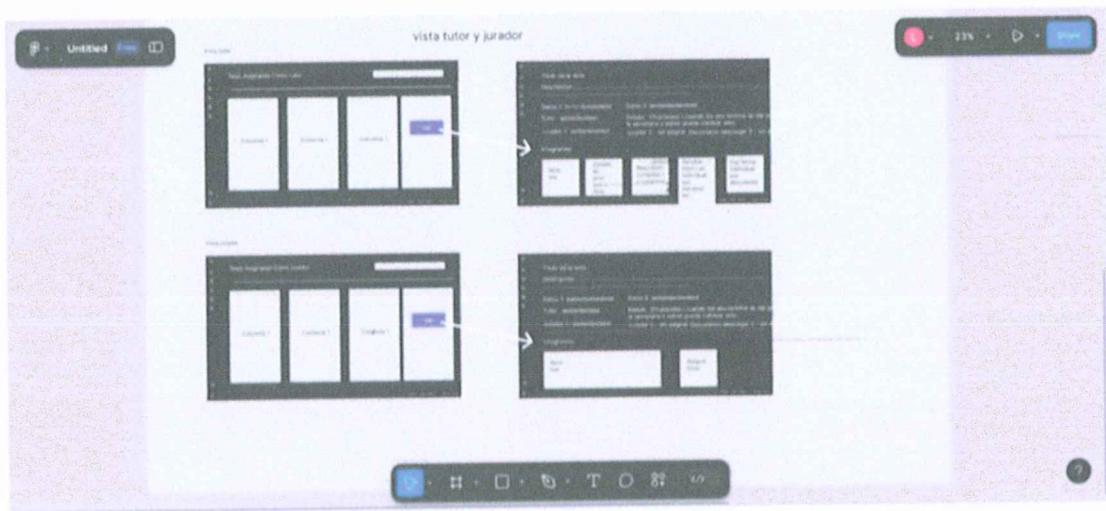


Figura 18. Modelado de Funcionamiento desde vista profesor como jurador y tutor.

En conclusión, realizar un boceto puede ayudar a definir varias funcionalidades, la navegación por tipo de usuario, entre otras cosas. Este proceso es de gran importancia, ya que, a partir de los requisitos y funcionalidades levantados al comienzo, se pudo



bocetar una idea general del sistema, la cual se fue puliendo a medida que se avanzaba con el proyecto.

4.4.1.3 Casos de uso

En los siguientes párrafos se describirá cada caso de uso que son básicamente una herramienta que permite modelar la funcionalidad a partir de una descripción, requisito, paso de flujo de la acción y una postcondición que indica el éxito de la operación.

4.4.1.3.1. Autenticación de Usuarios

| | |
|---|---|
| Identificador: CU-01 | Descripción: Permite a los usuarios ingresar a la plataforma utilizando sus credenciales. |
| Actor: Usuario | |
| Pre-Condición: Tener una cuenta habilitada | |
| Flujo Principal: | |
| <ol style="list-style-type: none"> 1. El usuario accede a la página de Login. 2. El usuario ingresar los datos. 3. El sistema valida lo datos ingresados. 4. Si las credenciales son correctas, el sistema genera una cookie o token de sesión. 5. El usuario es redirigido a la plataforma. | |
| Postcondición: El usuario tiene acceso a las funcionalidades asignadas a su rol. | |

TABLA 11. Caso de uso Autenticación de Usuarios

4.4.1.3.2. Registro de Usuario

| | |
|--|--|
| Identificador: CU-02 | Descripción: Creación de usuarios a partir de su rol correspondiente |
| Actor: administrador | |
| Pre-Condición Tener acceso al sistema con un rol de administrador | |
| Flujo Principal: | |
| <ol style="list-style-type: none"> 1. El usuario accede a la página de gestión de usuarios. 2. Pulsar el botón de crear. 3. Llenar los datos del formulario correspondiente con la información de la persona quien lo va a usar. 4. Se enviar al sistema. 5. El sistema valida los datos ingresados. 6. El sistema confirma el registro exitoso con un mensaje | |
| Postcondición: El nuevo usuario está habilitado en el sistema con el rol asignado. | |

TABLA 12. Caso de uso Registro de Usuario



4.4.1.3.3. Modificación del Usuario

| | |
|--|---|
| Identificador: CU-03 | Descripción: Modificación de los datos del usuario existente. |
| Actor: Usuario | |
| Pre-Condición: Usuario creado | |
| Flujo Principal: 1. Seleccionar el usuario a modificar. 2. Ingresar lo datos nuevos. 3. Grabar modificación. 4. Sistema Valida los datos modificados. 5. El sistema confirma la modificación con un mensaje de éxito. | |
| Postcondición: Los datos del usuario se actualizan en el sistema. | |

TABLA 13. Caso de uso Modificación del Usuario

4.4.1.3.4. Activa o Desactivar Usuario

| | |
|--|---|
| Identificador: CU-04 | Descripción: Activación o desactivación de la cuenta de usuario |
| Actor: Administrador | |
| Pre-Condición: El usuario debe estar registrado | |
| Flujo Principal: Seleccionar al usuario a activar o desactivar Pulsa el botón correspondiente (“Activar” o “Desactivar”). Sistema retorna mensaje de existo | |
| Postcondición: El estado de la cuenta se actualiza en el sistema. | |

TABLA 14. Caso de uso Activa o Desactivar Usuario



4.4.1.3.5. Registro de Carrera

| | |
|---|---|
| Identificador: CU-05 | Descripción: Permite crear una nueva carrera en el sistema. |
| Actor: Administrador | |
| Pre-Condición: Ninguna | |
| Flujo Principal: <ol style="list-style-type: none">1. Ingresar a la página de Carrera2. Pulsar el botón de crear3. Llenar los datos del formulario correspondiente con la información.4. Se enviar al sistema5. Sistema valida los datos6. Sistema enviar la respuesta de existo. | |
| Postcondición: La nueva carrera está disponible en el sistema. | |

TABLA 15. Caso de uso Registro de Carrera

4.4.1.3.6. Modificación del Carrera

| | |
|---|--|
| Identificador: CU-06 | Descripción: Modificación de los datos de la carrera |
| Actor: Administrador | |
| Pre-Condición: Carrera creada. | |
| Flujo Principal: <ol style="list-style-type: none">1. Seleccionar la carrera a modificar.2. Ingresar lo datos nuevos.3. Grabar modificación4. Sistema Valida los datos enviados.5. Retorna un mensaje de existo. | |
| Postcondición: Los datos de la carrera se actualizan correctamente en el sistema. | |

TABLA 16. Caso de uso Modificación del Carrera

4.4.1.3.7. Activa o Desactivar Carrera

| | |
|---|---|
| Identificador: CU-07 | Descripción: Activación o desactivación de la carrera |
| Actor: Administrador | |
| Pre-Condición: Carrera creada | |
| Flujo Principal: <ol style="list-style-type: none">1. Seleccionar a la carrera activar o desactivar.2. Pulsa el botón correspondiente (“Activar” o “Desactivar”).3. Sistema retorna mensaje de existo. | |
| Postcondición: El estado de la carrera se actualiza correctamente. | |

TABLA 17. Caso de uso Activa o Desactivar Carrera



4.4.1.3.8. Registro de Requisito

| | |
|---|--|
| Identificador: CU-08 | Descripción: Creación de los requisitos para lo estudiante |
| Actor: Administrador | |
| Pre-Condición: Ninguno | |
| Flujo Principal: <ol style="list-style-type: none">1. Ingresar a la página de Requisito.2. Pulsar el botón de crear.3. Llenar los datos del formulario correspondiente con la información.4. Se enviar al sistema.5. Sistema valida los datos.6. Sistema enviar la respuesta de existo. | |
| Postcondición: El requisito queda registrado y disponible en el sistema. | |

TABLA 18. Caso de uso Registro de Requisito

4.4.1.3.9. Modificación del Requisito

| | |
|--|--|
| Identificador: CU-09 | Descripción: Modificación de los datos del requisito |
| Actor: Administrador | |
| Pre-Condición: Requisito creado. | |
| Flujo Principal: <ol style="list-style-type: none">1. Seleccionar el requisito a modificar.2. Ingresar lo datos nuevos.3. Grabar modificación.4. Sistema Valida los datos enviados.5. Retorna un mensaje de existo. | |
| Postcondición: Los datos del requisito se actualizan correctamente. | |

TABLA 19. Caso de uso Modificación del Requisito



4.4.1.3.10. Activa o Desactivar Requisito

| | |
|--|---|
| Identificador: CU-010 | Descripción: Activación o desactivación del requisito |
| Actor: Administrador | |
| Pre-Condición: Requisito creado. | |
| Flujo Principal: 1. Seleccionar al requisito activar o desactivar. 2. Pulsa el botón correspondiente (“Activar” o “Desactivar”). 3. Sistema retorna mensaje de existo | |
| Postcondición: El estado del requisito se actualiza correctamente | |

TABLA 20. Caso de uso Activa o Desactivar Requisito

4.4.1.3.11. Activar o Desactivar Rol

| | |
|---|---|
| Identificador: CU-11 | Descripción: Activación o desactivación de rol. |
| Actor: Administrador | |
| Pre-Condición: Rol creado. | |
| Flujo Principal: 1. Seleccionar al rol activar o desactivar. 2. Clic al botón correspondiente. 3. Sistema retorna mensaje de existo. | |
| Postcondición: El estado del rol se actualiza correctamente. | |

TABLA 21. Caso de uso Activar o Desactivar Rol

4.4.1.3.12. Activar o Desactivar los Requisito Cumplido del estudiante

| | |
|--|--|
| Identificador: CU-12 | Descripción: Activación o Desactivación de lo requisito cumplido por parte del estudiante. |
| Actor: Administrador | |
| Pre-Condición: Estudiante creado | |
| Flujo Principal: 1. Ir a la página de Estudiantes. 2. Seleccionar el botón de redirección de página del estudiante referente. 3. Seleccionar el requisito Activa o desactivar. 4. Sistema Valida los datos enviados. 5. Retorna un mensaje de existo. | |
| Postcondición: El estado del requisito del estudiante queda actualizado en el sistema. | |

TABLA 22. Caso de uso Activar o Desactivar los Requisito Cumplido del estudiante



4.4.1.3.13. Registro de Tesis

| | |
|--|--|
| Identificador: CU-13 | Descripción: Creación de la tesis con la asignación del tutor. |
| Actor: Administrador y secretario | |
| Pre-Condición: Tener Estudiantes y profesor con el sub-rol de Tutor Habilitado | |
| Flujo Principal: <ol style="list-style-type: none">1. Ingresar a la página de Tesis.2. Pulsar el botón de crear.3. Llenar los datos del formulario correspondiente con la información y seleccionar el tutor de la tesis.4. Se enviar al sistema.5. Sistema valida los datos.6. Sistema enviar la respuesta de existo. | |
| Postcondición: La tesis queda registrado y disponible en el sistema. | |

TABLA 23. Caso de uso Registro de Tesis

4.4.1.3.14. Modificación del Tesis

| | |
|---|--|
| Identificador: CU-14 | Descripción: Modificación de los datos de la tesis |
| Actor: Administrador y secretario | |
| Pre-Condición: Tesis creada | |
| Flujo Principal: <ol style="list-style-type: none">1. Seleccionar la tesis a modificar2. Ingresar lo datos nuevos3. Grabar modificación4. Sistema Valida los datos enviados5. Retorna un mensaje de existo | |
| Postcondición: Los datos de la tesis se actualizan correctamente | |

TABLA 24. Caso de uso Modificación del Tesis



4.4.1.3.15. Activa o Desactivar Tesis

| | |
|--|---|
| Identificador: CU-15 | Descripción: Activación o desactivación del requisito |
| Actor: Administrador y secretaria | |
| Pre-Condición: Requisito creado | |
| Flujo Principal: <ol style="list-style-type: none">1. Seleccionar al requisito activar o desactivar2. Clic al botón3. Sistema retorna mensaje de existo | |
| Postcondición: El estado de la tesis se actualiza correctamente. | |

TABLA 25. Caso de uso Activa o Desactivar Tesis

4.4.1.3.16. Modificación del Perfil

| | |
|--|---|
| Identificador: CU-16 | Descripción: Modificación de los datos del estudiante |
| Actor: Estudiante | |
| Pre-Condición: Solo a Estudiantes. | |
| Flujo Principal: <ol style="list-style-type: none">1. Se dirige a la página perfil.2. Pulsar el botón modificar.3. Introduce lo nuevos datos a actualizar.4. Enviar lo datos al sistema. | |
| Postcondición: Los datos del estudiante se actualizan correctamente. | |

TABLA 26. Caso de uso Modificación del Perfil

4.4.1.3.17. Registro de Sustentación

| | |
|--|--|
| Identificador: CU-17 | Descripción: Creación de sustentación del estudiante con la asignación de los jurados. |
| Actor: Administrador y secretaria | |
| Pre-Condición: Jurados Habilitados y Tesis del estudiante. | |
| Flujo Principal: <ol style="list-style-type: none">1. Ingresar a la página de Tesis.2. Seleccionar la tesis del quien corresponda.3. Pulsar el botón crear sustentación4. Llenar el formulario con los datos respectivos y se seleccionar a lo jurados.5. Sistema valida los datos.6. Sistema enviar la respuesta de existo. | |
| Postcondición: La sustentación queda registrado y disponible en el sistema | |



TABLA 27. *Caso de uso Registro de Sustentación*

4.4.1.3.18. Modificación del Sustentación

| | |
|--|--|
| Identificador: CU-18 | Descripción: Modificación de los datos de la sustentación. |
| Actor: Administrador y secretaria | |
| Pre-Condición: Sustentación creada. | |
| Flujo Principal: | |
| <ol style="list-style-type: none"> 1. Seleccionar la sustentación a modificar. 2. Ingresar lo datos nuevos. 3. Grabar modificación 4. Sistema Valida los datos enviados. 5. Retorna un mensaje de existo. | |
| Postcondición: | |

TABLA 28. *Caso de uso Modificación del Sustentación*

4.4.1.3.19. Agregar Comentario referente al documento de tesis

| | |
|--|--|
| Identificador: CU-19 | Descripción: Los jurados añadirán comentarios de críticas sobre el documento de tesis. |
| Actor: Jurado | |
| Pre-Condición: Documento de Tesis. | |
| Flujo Principal: | |
| <ol style="list-style-type: none"> 1. Pulsar el botón agregar comentario. 2. Ingresar el comentario. 3. Pulsar enviar. 4. Sistema Valida los datos enviados. 5. Retorna un mensaje de existo. | |
| Postcondición: Se agregado exitosamente el sistema el comentario de la tesis. | |

TABLA 29. *Caso de uso Agregar Comentario referente al documento de tesis*

4.4.1.3.20. Aprobación de Tesis

| | |
|--|--|
| Identificador: CU-20 | Descripción: Los jurados aprobarán o desaprobarán la tesis |
| Actor: Jurados | |
| Pre-Condición: Los jurados aprobarán o desaprobarán la tesis. | |
| Flujo Principal: | |
| <ol style="list-style-type: none"> 1. Pulsar el botón calificar. 2. Seleccione el dato correspondiente. 3. Enviar datos. 4. Sistema Valida los datos enviados. 5. Retorna un mensaje de existo. | |
| Postcondición: Se asignado correctamente en el sistema la calificación del estudiante. | |



TABLA 30. Caso de uso Aprobación de Tesis

4.4.1.3.21. Subida de Documento de Tesis

| | |
|---|--|
| Identificador: CU-21 | Descripción: El estudiante subirá el documento de tesis. |
| Actor: Estudiante | |
| Pre-Condición: Ninguna | |
| Flujo Principal: <ol style="list-style-type: none">1. Seleccionar el botón de subir documento.2. Seleccionar el documento.3. Enviar el documento.4. Sistema Valida los datos enviados.5. Retorna un mensaje de existo. | |
| Postcondición: Se ha subido correctamente en sistema el documento de tesis. | |

TABLA 31. Caso de uso Subida de Documento de Tesis

4.4.1.3.22. Reemplazo de Documento de Tesis

| | |
|--|---|
| Identificador: CU-22 | Descripción: El estudiante reemplazara el documento de tesis. |
| Actor: Estudiante | |
| Pre-Condición: Documento de tesis. | |
| Flujo Principal: <ol style="list-style-type: none">1. Seleccionar el botón de reemplazar documento.2. Seleccionar el documento.3. Enviar el documento.4. Sistema Valida los datos enviados.5. Retorna un mensaje de existo. | |
| Postcondición: Se ha actualizado correctamente en sistema el documento de tesis. | |

TABLA 32. Caso de uso Reemplazo de Documento de Tesis

4.4.1.4 Diagramas UML de los casos de uso

En los siguientes se mostrará varios diagramas de uso para que son en sin son una herramienta de análisis de sistema que permite visualizar a través de un diseño lo proceso que conllevar un usuario.



Administrador y secretaria:

Interacción Administrador - Sistema

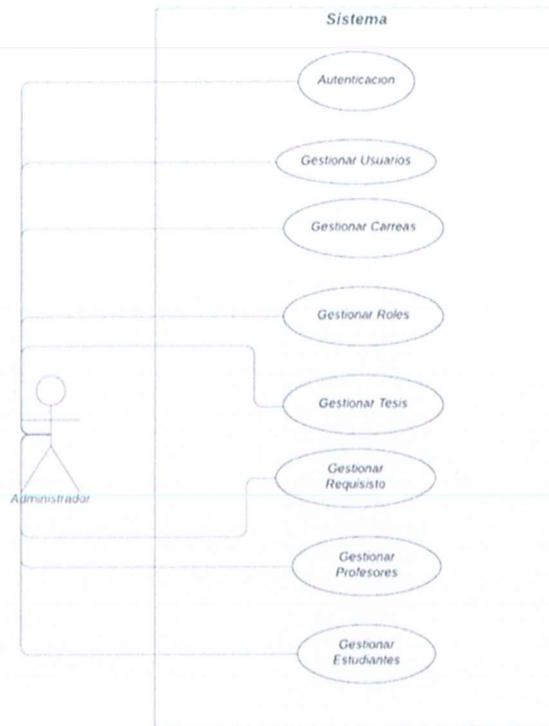


Figura 19. Interacción Administrador – Sistema



Gestionar Usuario:

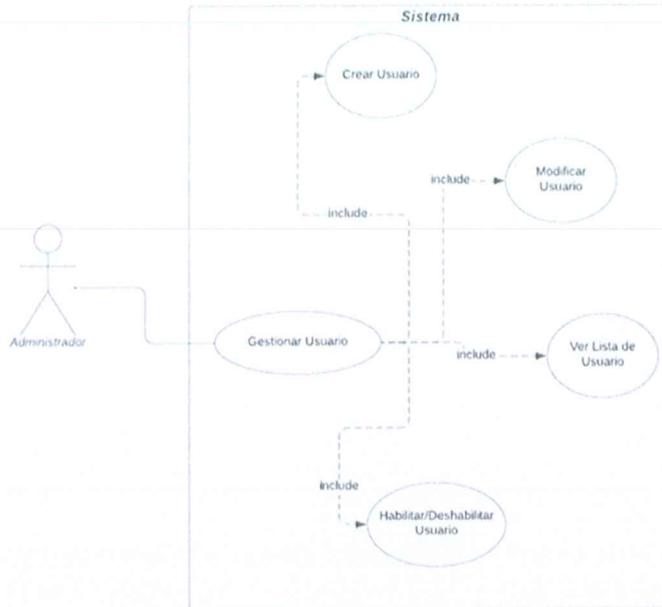


Figura 20. Gestión de Usuario

Gestionar Role

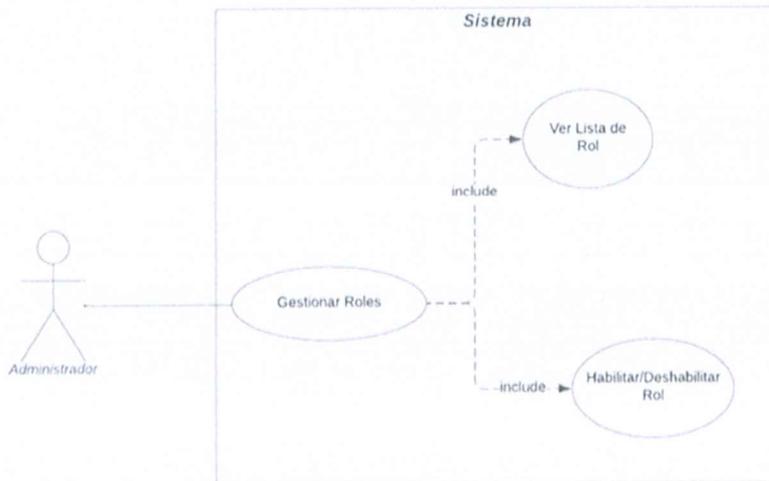


Figura 21. Gestión de Roles



Interacción secretaria - Sistema

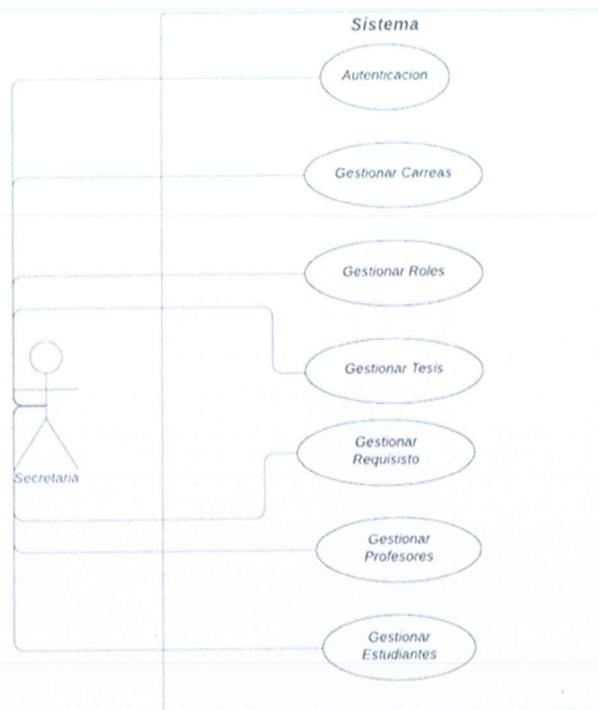


Figura 22. Interacción secretaria – Sistema

Gestionar Carrera:

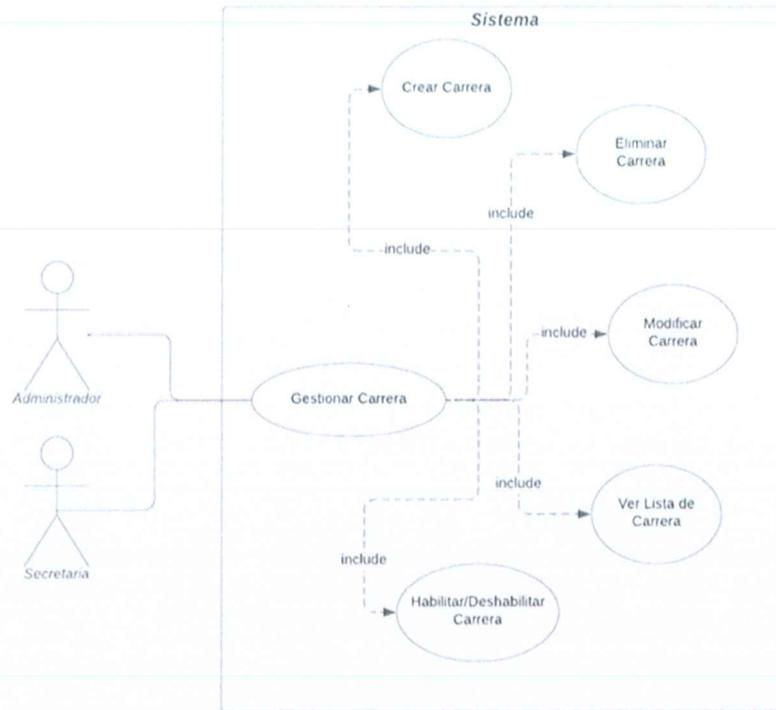


Figura 23. Gestión de Carrera

Gestionar Requisito:

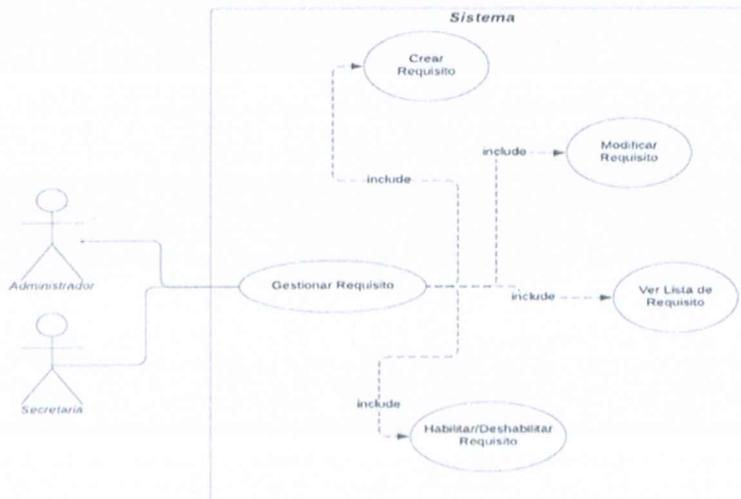


Figura 24. Gestión de Requisito



Gestionar Estudiante:

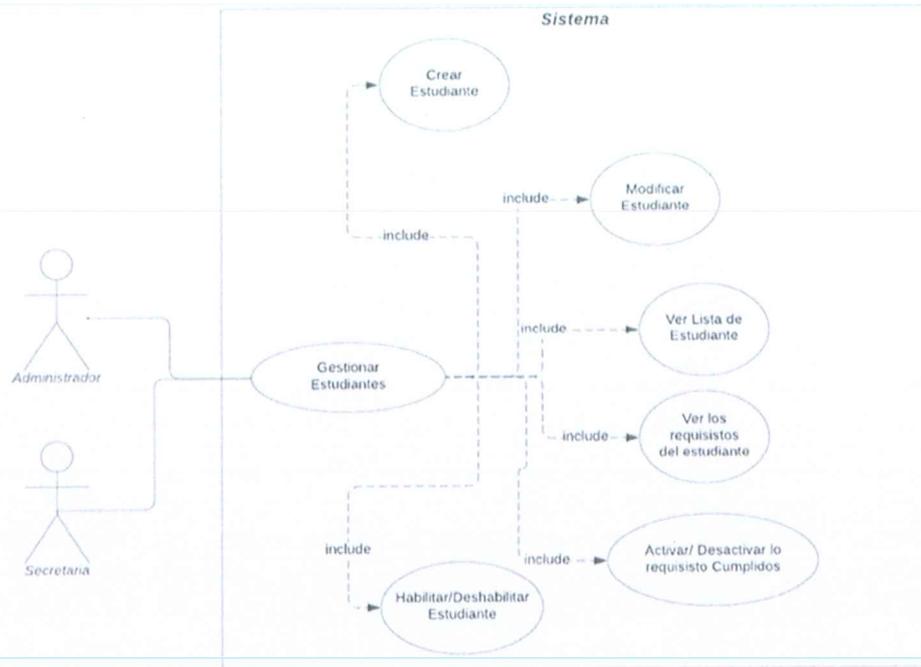


Figura 25. Gestión de Estudiantes

Gestionar Profesores:

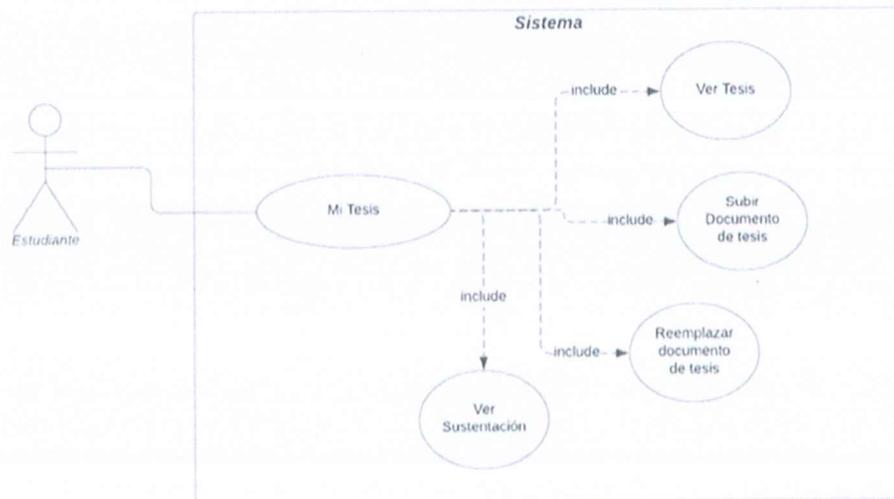


Figura 26. Gestión de Profesores



Gestionar Asignación de tesis:

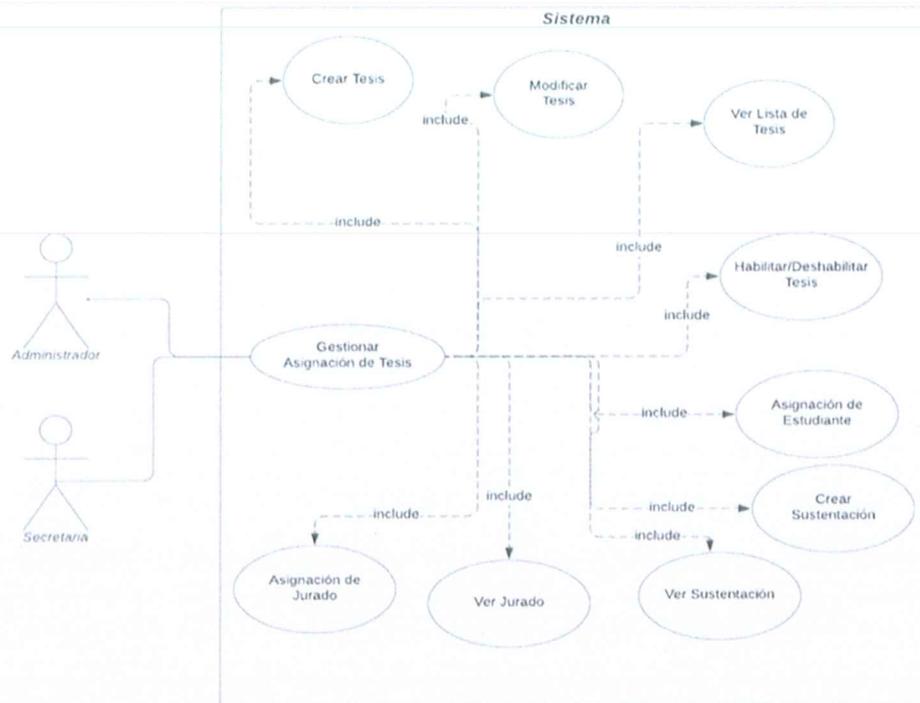


Figura 27. Gestión de Tesis

Interacción profesor - Sistema

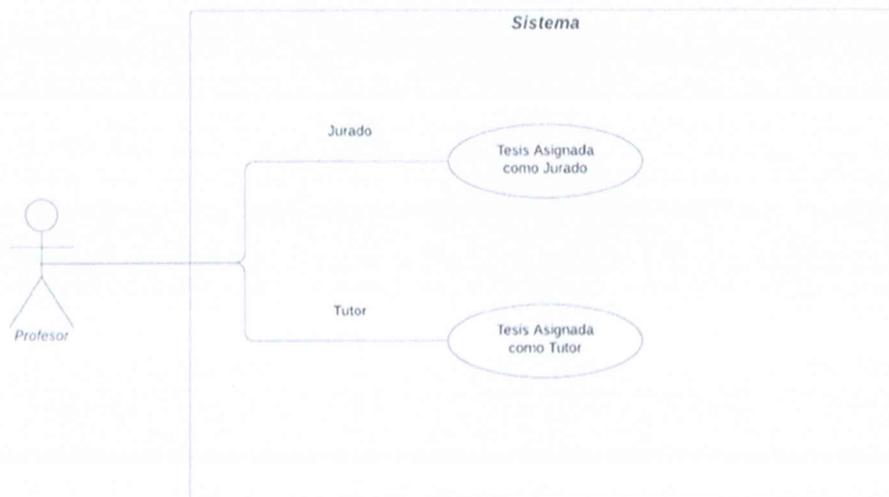


Figura 28. Interacción profesor - Sistema



Gestionar Revisión de sustentación:

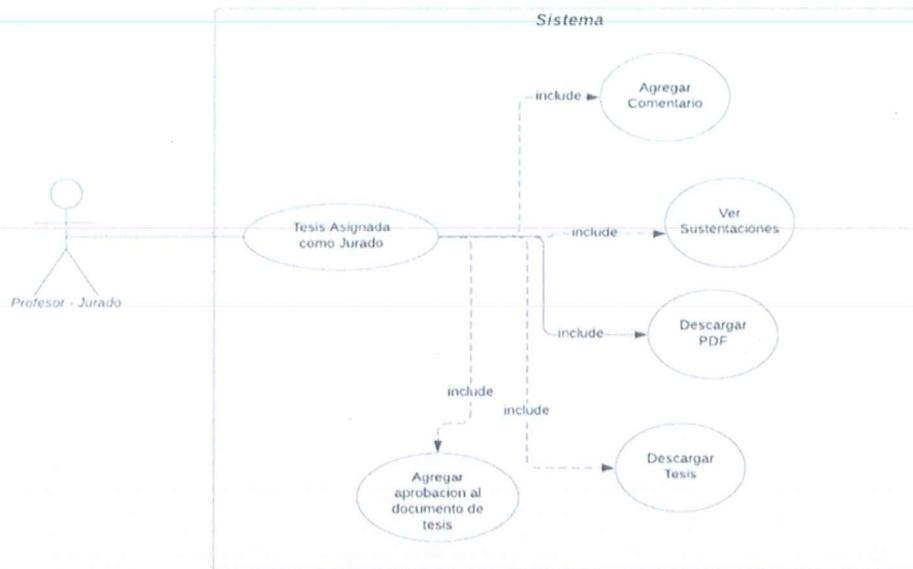


Figura 29. Gestionar Revisión de sustentación

Gestionar Revisión de sustentación:

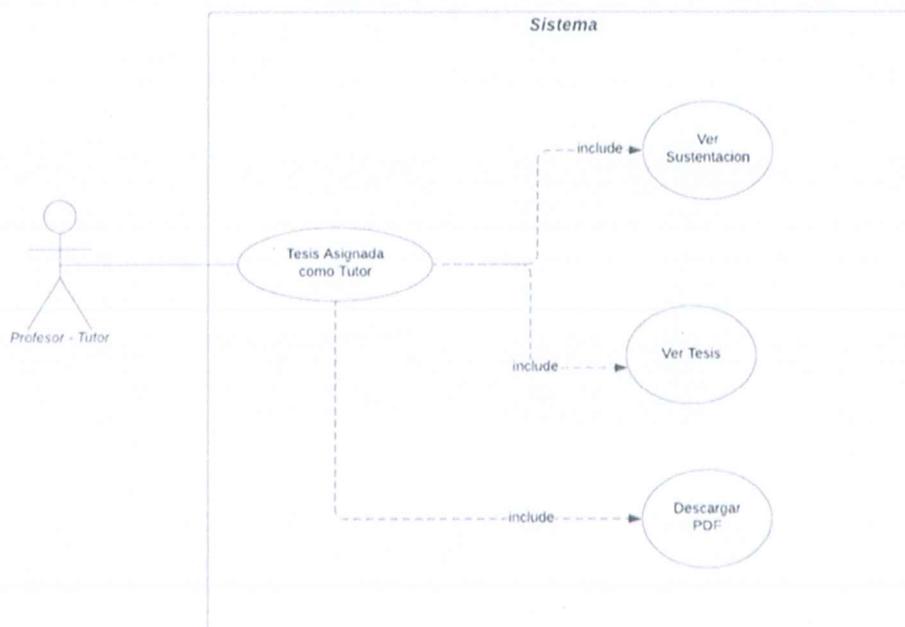


Figura 30. Gestionar Revisión de sustentación



Interacción estudiante - Sistema

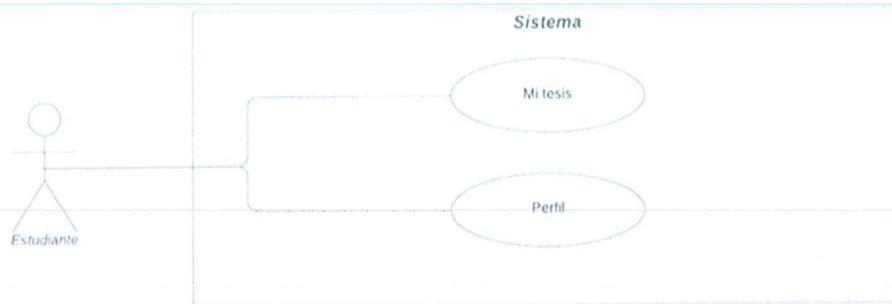


Figura 31. Interacción estudiante - Sistema

Gestionar Perfil:

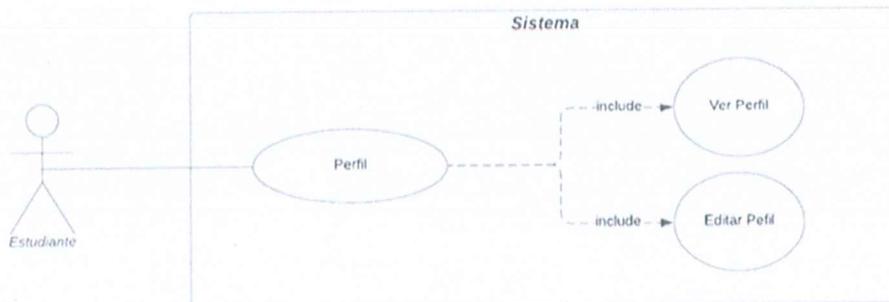


Figura 32. Gestionar Perfil

Vista Estudiante - Tesis

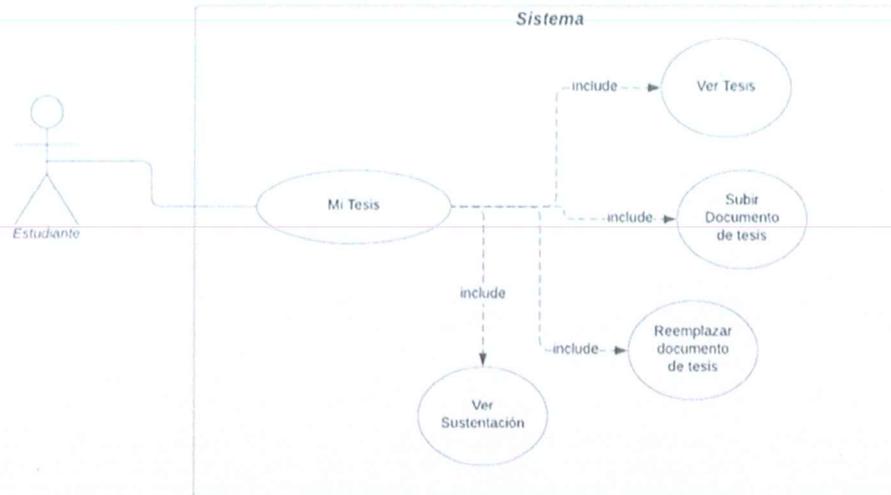


Figura 33. Vista Estudiante - Tesis

4.4.1.5 Modelo lógico entidad relación de la base de datos del software

Aunque ya se mostró de manera general la base de datos esta tiene que ser desestructurada para cumplir con los requerimientos de cada microservicio, es decir debemos separar las tablas en función a los microservicios que se van a tener en el software, primero se realizó una base de datos relacionada completamente para poder tener en cuenta las relaciones que puedan tener al momento de separarlas para así saber en cuales se deben hacer las validaciones de datos respectivas.

Base de datos para la Autenticación

En esta base de datos crearemos lo que son las tablas de Usuario, Rol y Carrera en conjunto para crear el microservicio de autenticación, esto será así para en un caso



práctico a la hora de iniciar sesión poder obtener su rol y la carrera asociada al usuario y puedan funcionar sin tener que llamar a otro microservicio

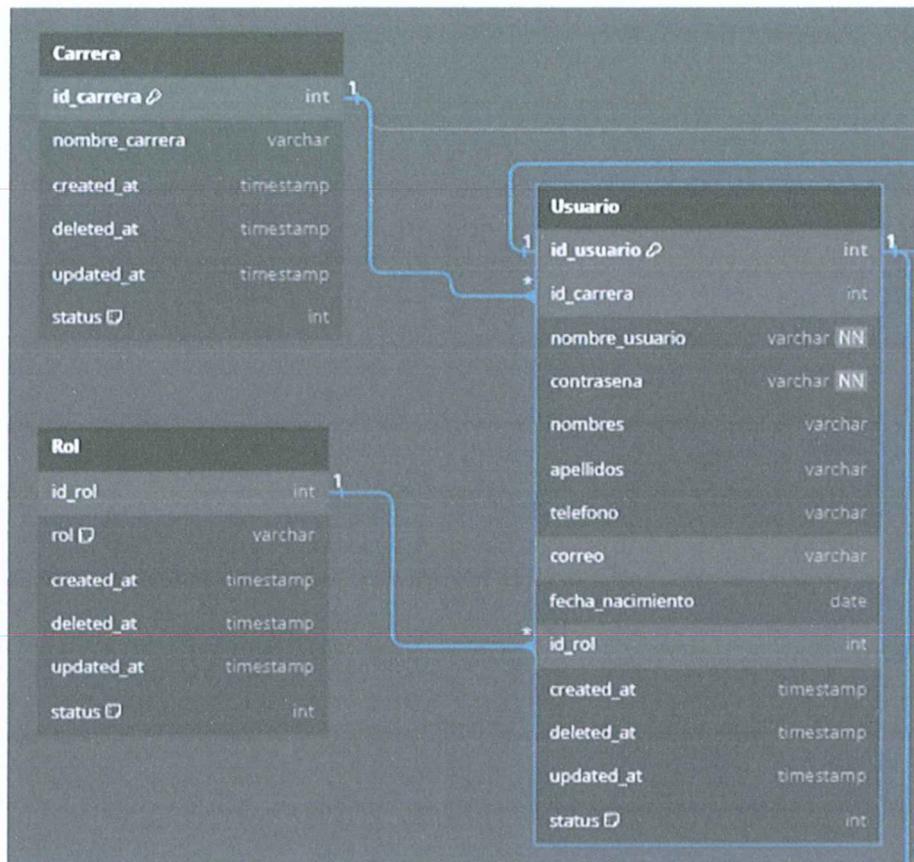


Figura 34. Tablas para el microservicio de autenticación



Base de datos para Usuarios

Ahora vamos a la parte de la gestión de los usuarios dentro de la plataforma, una vez creado los usuarios tenemos que definir el hecho de que sean estudiantes docentes y jurados, que son los usuarios importantes en la plataforma entonces la gestión de estos irá en una base de datos única para el microservicio que se encargara de estos

| Estudiante | |
|---------------|--------------|
| id_estudiante | int |
| id_tesis | int |
| id_usuario | int |
| sexo | varchar(15) |
| estado_civil | varchar(20) |
| genero | varchar(30) |
| tipo_colegio | varchar(20) |
| numero_hijos | int |
| pais | varchar(30) |
| provincia | varchar(40) |
| ciudad | varchar(50) |
| parroquia | varchar(50) |
| direccion | varchar(250) |
| created_at | timestamp |
| deleted_at | timestamp |
| updated_at | timestamp |
| status | int |

| Docente_Tutor | |
|------------------|-----------|
| id_docente_tutor | int |
| id_usuario | int |
| created_at | timestamp |
| deleted_at | timestamp |
| updated_at | timestamp |
| status | int |

| Jurado | |
|------------|-----------|
| id_jurado | int |
| id_usuario | int |
| created_at | timestamp |
| deleted_at | timestamp |
| updated_at | timestamp |
| status | int |

Figura 35. Tablas para el microservicio de Usuarios

Base de datos para el control de tesis

En las tablas con relación a la tesis y al control de los documentos necesarios para que esta pueda ser habilitada en su defensa se las separa para que estas armen un microservicio basado en el control de la tesis y los documentos requeridos.

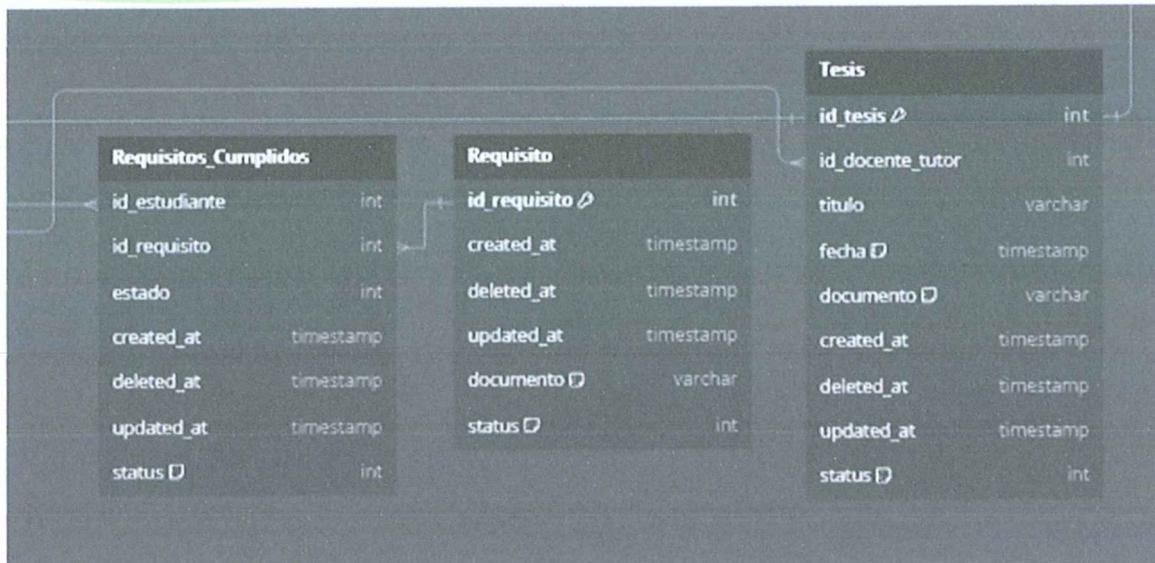


Figura 36. Tablas para el microservicio de Control de Tesis

Base de datos para la asignación de sustentación

Como ultima separación tenemos la de los servicios para la asignación de sustentación donde recae la tabla de Jurado sustentación y la propia sustentación como tal, para poder proveer los datos relacionados a esta en sus funciones.

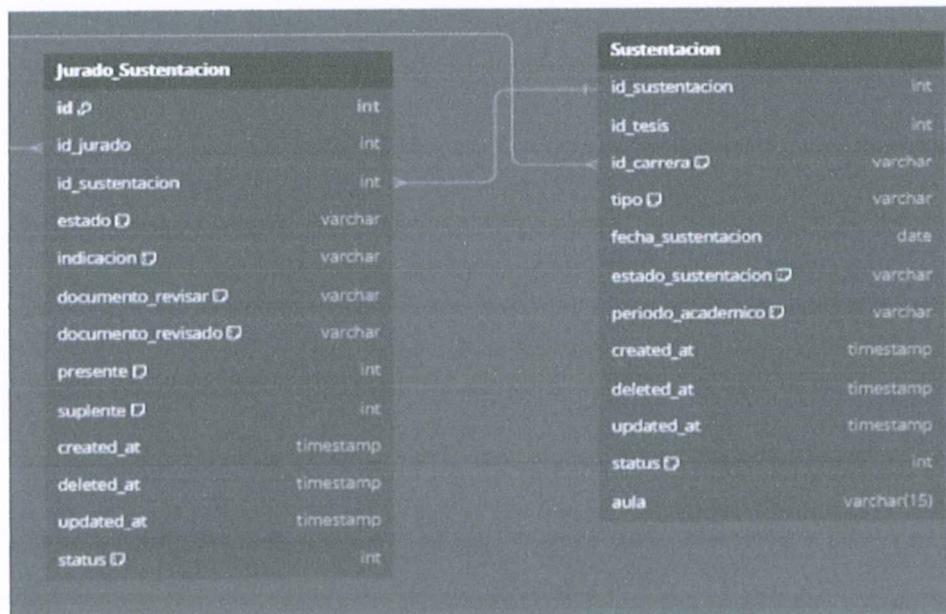


Figura 37. Tablas para el microservicio de sustentación



4.4.2 Fase II: Diseño y Desarrollo

Comenzando la segunda fase donde tenemos el desarrollo de interfaces y de los microservicios siguiendo las actividades definidas en la planificación de los módulos de esta fase.

| Actividad | Responsable | Estado | Fecha de Inicio | Fecha de Finalización |
|---|------------------|------------|-----------------|-----------------------|
| Definición de requisitos de desarrollo | Antonio Rojas A. | Completado | 15/03/2024 | 15/03/2024 |
| Planificación de desarrollo de interfaces | Antonio Rojas A. | Completado | 15/03/2024 | 15/03/2024 |
| Desarrollo de interfaces de usuario | Antonio Rojas A. | Completado | 15/03/2024 | 15/03/2024 |
| Desarrollo de interfaces de usuario | Antonio Rojas A. | Completado | 15/03/2024 | 15/03/2024 |

Figura 38. Actividades del primer módulo de fase II

| Actividad | Responsable | Estado | Fecha de Inicio | Fecha de Finalización |
|---|------------------|------------|-----------------|-----------------------|
| Definición de requisitos de desarrollo | Antonio Rojas A. | Completado | 15/03/2024 | 15/03/2024 |
| Planificación de desarrollo de interfaces | Antonio Rojas A. | Completado | 15/03/2024 | 15/03/2024 |
| Desarrollo de interfaces de usuario | Antonio Rojas A. | Completado | 15/03/2024 | 15/03/2024 |
| Desarrollo de interfaces de usuario | Antonio Rojas A. | Completado | 15/03/2024 | 15/03/2024 |

Figura 39. Actividades del segundo módulo de fase II

4.4.2.2 Métodos de comunicación y Estructura lógica

Como ya se explicó anteriormente en la investigación para parte de la comunicación se usará NATS, pero esta red de comunicación será creada a través de un contenedor de Docker

Levantar el contenedor NATs

Se utilizará la imagen oficial de NATS en Docker Hub para la creación del contenedor que nos proveerá el servicio NATS la versión a usar al momento de ser escrita esta tesis es la 2.10.24 – scratch, para que este funcione de manera correcta al hacer el contenedor debemos definir el puerto de cliente para este, que en NATs es el 4222 entonces para no complicarnos mucho a la hora de hacer el Docker run definiremos el



puerto 4222:4222 tanto para la maquina local como para la del contenedor por lo que el comando para levantar el seria este:

- `docker run -p 4222:4222 nats`

En nuestro caso le pusimos al contener el `--name` para renombrarlo como `nats-main`, pero quedaría de esta manera:

| Name | Container ID | Image | Port(s) | CPU (%) | Last started | Actions |
|-----------|--------------|------------|-------------------|---------|----------------|---------|
| nats-main | ffb2e2a22150 | nats:<none | 4222:4222 ↻ 62... | 0.09% | 57 seconds ago | ⏏ ⋮ 🗑 |

Figura 40. Contenedor de NATs levantado y funcionando

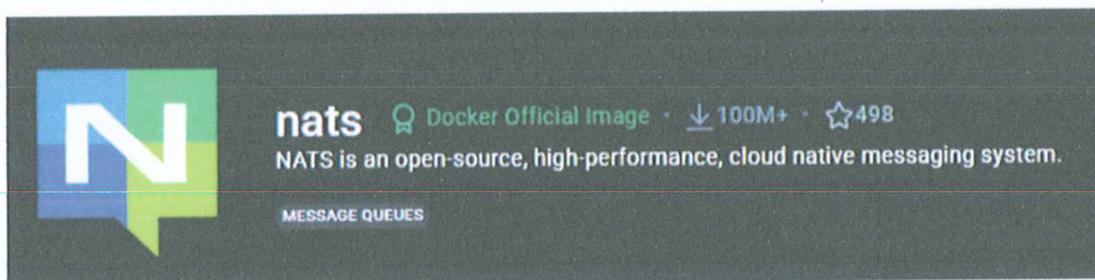


Figura 41. Imagen oficial de NATs usada

Estructura lógica

En la estructura lógica del funcionamiento nos referimos a cómo van a interactuar de manera general el main Gateway y el microservicio una vez estén en funcionamiento y se realice una petición en esta, de forma que todos los servicios que se desarrollen sigan esta metodología, la cual será explicada con una imagen a continuación:

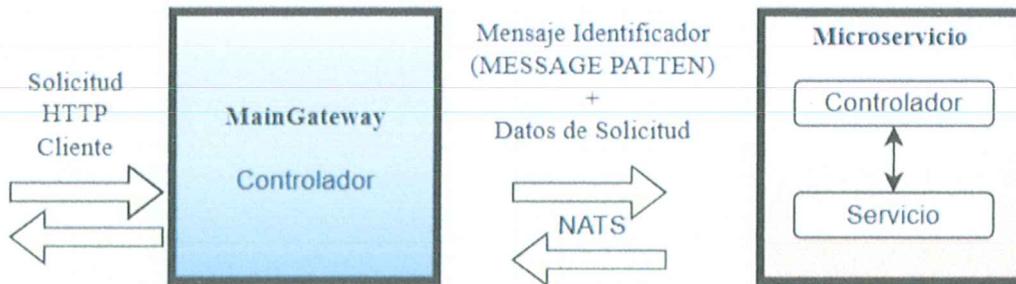


Figura 42. Explicación lógica de comunicación al hacer una petición

Como se aprecia en la imagen al hacer una solicitud siempre será una solicitud http que el MainGateway transformara a un mensaje de NATs y enviará los datos necesarios al microservicio en caso de que este lo requiera estableciendo la comunicación y obteniendo los datos de esta, esto en código sería algo así:

MainGateway

En el main Gateway tendremos para este ejemplo un controlador en que mostramos la solicitud http de tipo post para crear una tesis, en esta se puede observar cómo recibe un body y después de las validaciones pertinentes donde está marcado, enviar un mensaje a través del cliente proxy con el mensaje identificador y con los datos que requiere el microservicio, que en este caso son los datos necesarios para crear el dato de tesis:



```
@ApiTags('Tesis')
@Post()
async Create(@Body() Tesis: tesisDTO) {
  try {
    if (Tesis.id_docente_tutor) {
      const Exist = await firstValueFrom(
        this.client.send({ cmd: 'GetDocenteTutor' }, Tesis.id_docente_tutor),
      );
      if (!Exist) {
        return FailResponse('Docente Tutor no valido');
      }
    }
    const Data = await firstValueFrom(
      this.client.send({ cmd: 'CreateTesis' }, Tesis),
    );
    return SuccessResponse(Data);
  } catch (e) {
    return FailResponse(ExeptValidator(e));
  }
}
```

Figura 43. Código de controlador MainGateway

Microservicio

En la parte del microservicio tendremos el controlador que identificara el mensaje para saber a qué método estamos llamando con el decorador MessagePattern, obtener los datos necesarios enviados por el MainGateway y que este se encargue también de comunicar al servicio respectivo con la lógica para funcionar:

```
async Create(Data: tesisDTO) {
  try {
    const Tesis: Partial<tesis> = {
      ...Data,
      created_at: new Date(),
    };
    return await this.repository.save(Tesis);
  } catch (error) {
    throw new RpcException(error);
  }
}
```

Figura 44. Código del controlador del microservicio



```
@Controller('tesis')
export class TesisController {
  constructor(private readonly Services: TesisService) {}
  @MessagePattern({ cmd: 'CreateTesis' })
  async Create(Tesis: tesisDTO) {
    return await this.Services.Create(Tesis);
  }
}
```

Figura 45. Código del servicio del microservicio

Al finalizar la lógica de esto el MainGateway le da una respuesta al cliente dependiendo del desenlace de la operación.

4.4.2.3 Planificación de los métodos REST base

Para los métodos rest Básicos de un api se plantearon unas bases de estos para que se pueda trabajar con ellos y poder crear la estructura básica rest de cada entidad de los microservicios sin tomar en cuenta las necesidades específicas posteriores a la hora de ir desarrollando la aplicación, estos métodos bases son los típicos Get, GetAll, Create, Delete, Update y Restore:

- Get: Este método tendrá una entrada de parámetro que será la llave primaria o id de la tabla para realizar la búsqueda de ese único dato que devolverá.
- GetAll: Este método obtendrá 3 parámetros que serán: Pagina(page), Limite(limit) y Like. Los dos primeros serán los parámetros de paginación para la obtención de todos los datos de esa tabla, si no se mandan estos tienen valores por default que son: Pagina = 1 Limite = 10, para el parámetro Like si el método lo requiere se usara este para poder hacer una



búsqueda de datos en base a un string enviado, usado para filtrar por buscador.

- Create: El método de creación para los datos, enviara un body json que tendrá la estructura completa o parcial dependiendo de la tabla a la que se haga la solicitud y retornara el dato creado al finalizar.
- Update: Método para actualizar datos obtendrá el parámetro de la id que apuntara al dato a actualizar y un body json con los datos que pueden ser completos o parciales a actualizar de ese dato.
- Delete: Sera como el método de actualización ya que los datos eliminados solo actualizaran el status de true a false, obtendrá un parámetro que será la id del dato a soft-eliminar
- Restore: Método que será contrario al eliminar, actualizará el status de false a true y recibirá como parámetro la id del dato a restaurar.

Gracias a la planificación de los métodos básicos rest se puede empezar un desarrollo temprano y básico de estos métodos para los microservicios haciendo todo más rápido y que si luego se necesitará algún End Point específico tener una base ya creada para esto.



4.4.2.4 Desarrollo de Especifico de cada método REST base planificado.

Aquí planteamos la parte del código de cómo se estructuro cada método rest base planificado en el main Gateway:

Get: Método que obtiene un paramento que es la id y devuelve el dato como respuesta

```
@ApiTags('Carrera')
@Get('/:id')
async Get(@Param('id') id: number) {
  try {
    const data = await firstValueFrom(
      this.client.send({ cmd: 'GetCarrera' }, id),
    );
    return SuccessResponse(data);
  } catch (e) {
    return FailResponse(e);
  }
}
```

Figura 46. Código del método Get básico creado

GetAll: Método que tiene 2 parámetros principales página y límite de la paginación y el Like si lleva búsqueda

```
@ApiTags('Estudiante')
@Get()
@ApiQuery({ name: 'like', required: false })
async GetAll(
  @Query() Pagination: PaginationDto,
  @Query('Like') Like: string,
) {
  try {
    const DataEstudiante = await firstValueFrom(
      this.client.send({ Data: estudiante[]; meta: any } >> {
        cmd: 'GetAllEstudiante' },
        { Pagination, Like },
      ),
    );
    const { Data } = DataEstudiante;
    const CompleteData = await Promise.all(
      Data.map(async (estudiante) => {
        const UserData = await firstValueFrom(
          this.client.send({ cmd: 'GetUsuario' }, estudiante.id_usuario),
        );
        return { ...UserData, ...estudiante };
      }),
    );
    return PaginatedSuccessResponse({
      data: CompleteData,
      meta: DataEstudiante.meta,
    });
  }
}
```

Figura 47. Código del método GetAll básico creado



Create: Método que obtiene un body json del tipo de entidad y devuelve su respuesta

```
@ApiTags('Carrera')
@Post()
async Create(@Body() Carrera: carreraDTO) {
  try {
    const Data = await firstValueFrom(
      this.client.send({ cmd: 'CreateCarrera' }, Carrera),
    );
    return SuccessResponse(Data);
  } catch (error) {
    return FailResponse(error);
  }
}
```

Figura 48. Código del método Create básico creado

Update: Método que obtiene el parámetro de la id y el body de los datos a actualizar, devuelve la información de la operación

```
@ApiTags('Carrera')
@Put('/:id')
async Update(@Param('id') id: number, @Body() CarreraData: carreraDTO) {
  try {
    const data = await firstValueFrom(
      this.client.send({ cmd: 'UpdateCarrera' }, { id, CarreraData }),
    );
    return SuccessResponse(data);
  } catch (e) {
    return FailResponse(e);
  }
}
```

Figura 49. Código del método Update básico creado



Delete: Método que obtiene un parámetro y actualiza el status de los datos false, devuelve el resultado de la operación:

```
@ApiTags('Carrera')
@Delete('/:id')
async Delete(@Param('id') id: number) {
  try {
    const data = await firstValueFrom(
      this.client.send({ cmd: 'DeleteCarrera' }, id),
    );
    return SuccessResponse(data);
  } catch (e) {
    return FailResponse(e);
  }
}
```

Figura 50. Código del método Delete básico creado

Restore: Método que obtiene la id del dato y cambia el status de este de false a true, devuelve el resultado de la operación

```
@ApiTags('Carrera')
@Put('/:id/restore')
async Restore(@Param('id') id: number) {
  try {
    const data = await firstValueFrom(
      this.client.send({ cmd: 'RestoreCarrera' }, id),
    );
    return SuccessResponse(data);
  } catch (error) {
    return FailResponse(error);
  }
}
```

Figura 51. Código del método Restore básico creado

4.4.2.5 Desarrollo de interfaces de usuario

Para el desarrollo de las interfaces, se comenzó con la configuración de layout principal es decir el marco que va a utilizar desde principio fin donde definimos el SEO y Font.



Figura 53. Arquitectura del Frontend

Por otra parte, también es importante configurar el proxy al comenzar un proyecto, ya que siempre se preferirá contar con un solo punto para realizar cambios, en lugar de buscar archivo por archivo para modificar la URL cuando se cambian el puerto o el dominio al que se realizan las peticiones a la API. El proxy facilita este proceso y, además, ayuda a redireccionar las solicitudes, de modo que un atacante no obtenga información sobre la ubicación exacta del servidor. Asimismo, es fundamental configurar correctamente las cabeceras.



```
async rewrites() {
  return [
    {
      source: '/administracion/v1/:path*',
      destination: 'http://localhost:3000/:path*',
    },
  ];
},

async headers() {
  return [
    {
      source: '/*', // Aplica a todas las reglas
      headers: [
        {
          key: 'X-Content-Type-Options',
          value: 'nosniff',
        },
        {
          key: 'X-Frame-Options',
          value: 'DENY',
        },
        {
          key: 'Strict-Transport-Security',
          value: 'max-age=31536000; includeSubDomains',
        },
        {
          key: 'X-XSS-Protection',
          value: '1; mode=block', // Protección contra ataques de XSS
        },
      ],
    },
  ];
}
```

Figura 54. Configuración de Proxy y Cabecera del http.

Ya una vez configurado todo lo inicial, vamos a desarrollar poco a poco la plataforma y sus funcionalidades, siguiendo el orden de la estructura establecida. Lo primero que se desarrollará será la interfaz de inicio de sesión, junto con sus funcionalidades de autenticación. Luego, se procederá a desarrollar el layout o marco que se utilizará en las interfaces de la plataforma, comenzando con el menú, que es generado a partir del rol de usuario en el servidor y luego renderizado por el cliente. A partir de allí, se continuará desarrollando las demás interfaces.

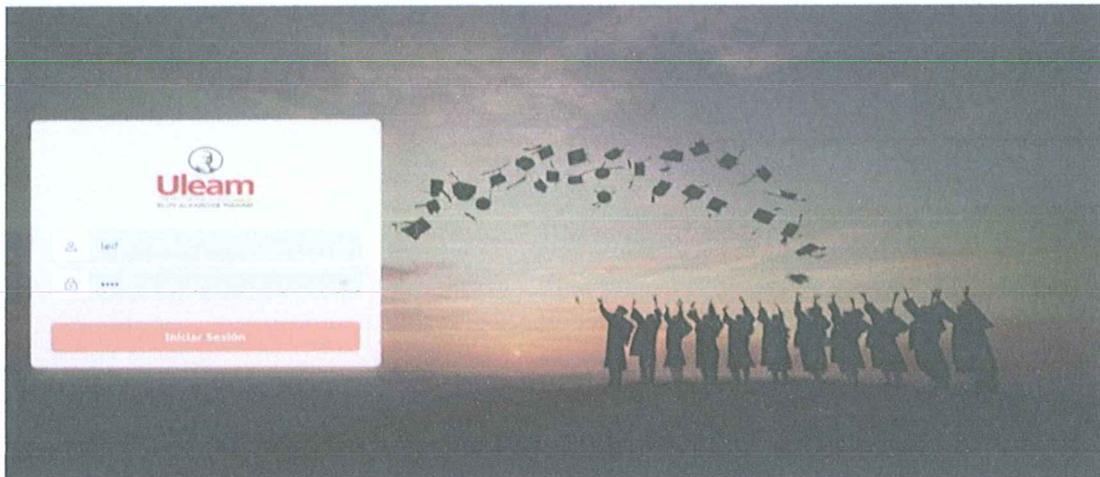


Figura 55. Interfaz Login

Por último, la configuración de middleware donde crear un archivo `middleware.tsx` en la raíz del `src` no es posible tener más de un middleware como otro framework como `django`. Entonces se configurará primero donde se va a aplicar el middleware, luego se creará las funcionalidades de obtener el token y descifrar el token para verificar que sea válido y la ruta que quiere acceder también sea validar dependiendo del rol del usuario.

```
1 import { NextRequest, NextResponse } from 'next/types';
2 import { validateToken } from '@services/auth';
3 import { validatePermission } from '@services/auth';
4 import { listNavigation } from '@services/navigation';
5
6 export type Middleware = (request: NextRequest) => NextResponse;
7
8 export async function middleware(request: NextRequest): NextResponse {
9   const dataCookie = request.cookies.get('authToken');
10  const { data_val } = await validateToken(dataCookie);
11  const url_path = request.nextUrl.pathname;
12
13  if (dataCookie && url_path === '/') {
14    return NextResponse.redirect(new URL('/', request.url));
15  }
16  if (dataCookie && url_path.startsWith('/api/')) {
17    return NextResponse.redirect(new URL('/', request.url));
18  }
19  if (dataCookie && validatePermission(request, dataCookie));
20  return NextResponse.redirect(new URL('/', request.url));
21  return NextResponse.next();
22 }
23
24 export const config = {
25   matcher: ['/api/*', '/admin/*', '/auth/*'],
26 };
27
```

Figura 56. Configuración de middleware y otras funcionalidades de verificación.



A continuación, se mostrará algunas de las interfaces desarrolladas. Como primero tenemos a la interfaz de perfil donde el usuario visualizara toda la información que se encuentre vinculada con su cuenta.

Perfil

| | | | |
|---------------------|--------------|-----------|-------------------------------|
| Nombre | Leif Wither | Apellidos | Velasquez Rialpe |
| Usuario | leif | Correo | leif@hotmail.com |
| Teléfono | 099999884848 | Carrera | Tecnologías de la información |
| Fecha de Nacimiento | | | |

Figura 57. Perfil del Usuario

Tesis

| Título de Tesis | Tutor | Fecha | Acciones |
|---|------------------------------------|--------------------------|--------------|
| Vida en la tecnología | Juan Luis Zamora Fernández | 2025-04-08T05:00:00.000Z | DESHABILITAR |
| Automatización en la agricultura | Juan Luis Zamora Fernández | 2025-07-08T05:00:00.000Z | DESHABILITAR |
| Inteligencia artificial aplicada en tesis | Ariel Alexander González Rodríguez | 2025-07-05T05:00:00.000Z | DESHABILITAR |
| Sistema de Invernadero con IA | wither Eduardo Velasquez | 2025-08-10T05:00:00.000Z | DESHABILITAR |

1

Figura 58. Interfaz gestión de Tesis



4.4.3 Fase III: Configuración y Despliegue

Ya nos encontramos en la última fase es que son las configuraciones de los servidores y configuración de la plataforma donde se va a desplegar el proyecto, lo cual, una vez configurado ya solo dejaría el despliegue y revisión de las funcionalidades según lo planificado.

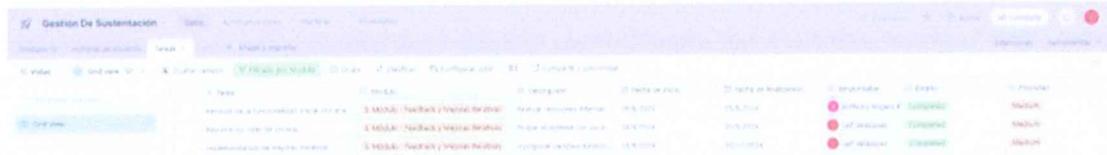


Figura 59. Actividades del primer módulo de fase III

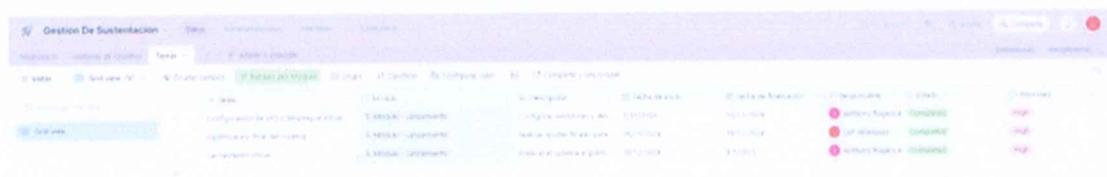


Figura 60. Actividades del primer módulo de fase III

4.4.3.1 Configuración de los VPS para alojamiento

Configuración de VPS Digital Ocean

En términos generales este VPS básico comprado de Digital Ocean cuya función será el poder desplegar los distintos microservicios dentro de él, para esto tendremos que cerciorarnos de que este cumpla con los requerimientos que tenemos:



Primero tendremos que actualizar el VPS usando apt update && apt upgrade, esto se hará ejecutando una conexión SSH al VPS con las credenciales con las que se compró

```
Last login: Thu Jan  9 21:01:07 2025 from 38.58.233.3
root@UbuntuVPSTesis:~# apt update && upgrade
Hit:1 http://mirrors.digitalocean.com/ubuntu oracular InRelease
Get:2 http://mirrors.digitalocean.com/ubuntu oracular-updates InRelease [126 kB]
Get:3 http://mirrors.digitalocean.com/ubuntu oracular-backports InRelease [126 kB]
Hit:4 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Get:5 http://security.ubuntu.com/ubuntu oracular-security InRelease [126 kB]
```

Figura 61. Actualización de VPS digital Ocean

Al contar con el VPS actualizado lo segundo es comprobar que este tenga nodejs ejecutando un node --version instalado, si no tendríamos que instalar este con un apt install nodejs

```
root@UbuntuVPSTesis:~# node --version
v20.16.0
```

Figura 62. Comprobación de nodejs

Y Finalmente se necesita instalar Docker en el VPS ya que es como levantaremos la red de comunicación NATS donde se comunicarán los distintos microservicios con apt install Docker.io

```
root@UbuntuVPSTesis:~# apt install docker.io
docker.io is already the newest version (26.1.3-0ubuntu1.1).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 4
root@UbuntuVPSTesis:~#
```

Figura 63. Instalación de Docker en VPS

Y con esto tendremos configurado e instalado todos los requerimientos para desplegar los microservicios en este VPS.

Configuración de VPS Atukhosting

Se tomó la decisión de usar un segundo VPS ya que el que tiene Digital Ocean no proporciona una base de datos en su servicio básico y esta de Atukhosting permite tener base de datos, pero no tiene soporte de Docker y ya se contaba con este VPS comprado,



en situaciones normales se podría contar el propio servicio de Digital Ocean o un solo VPS que cubra con las necesidades completas, pero en este caso vamos a configurar este para poder alojar las bases de datos para cada microservicio.

Lo primero aquí es poder crear el usuario que usaran las bases de datos para hacer el Login, con un nombre y contraseña.

| Usuarios | Acciones |
|---------------------|---|
| akcdlkck_anthonyx57 | Cambiar contraseña Cambiar el nombre Eliminar |

Figura 64. Usuario creado en VPS de base de datos

Con eso cuando creemos las bases de datos podremos anexar este usuario a esta y así conectarnos en nuestro gestor de base de datos para poder desplegar los backups de las bases de datos en estas, que es lo que se hará, dejar definidas todas las bases de datos que se usaran para los microservicios.

| Base de datos | Tamaño | Usuarios con privilegio |
|--------------------------------|---------|-------------------------|
| akcdlkck_ServicioAutenticacion | 7.77 MB | akcdlkck_anthonyx57 |
| akcdlkck_ServicioSustentacion | 7.51 MB | akcdlkck_anthonyx57 |
| akcdlkck_ServicioTesisControl | 7.51 MB | akcdlkck_anthonyx57 |
| akcdlkck_ServicioUsuarios | 7.51 MB | akcdlkck_anthonyx57 |

Figura 65. Base de datos preparadas en el VPS

4.4.3.2 Configuración de dominio FrontEnd con CloudFlare

Una vez comprado el dominio de go daddy lo que se hizo es transferir la DNS a clouflare de la siguiente manera.



Primero nos vamos a la página de go daddy , buscamos nuestro dominio y nos dirigimos a configuración avanzada dentro, seleccionamos Hostname, Dentro habrá una serie de instrucciones a seguir

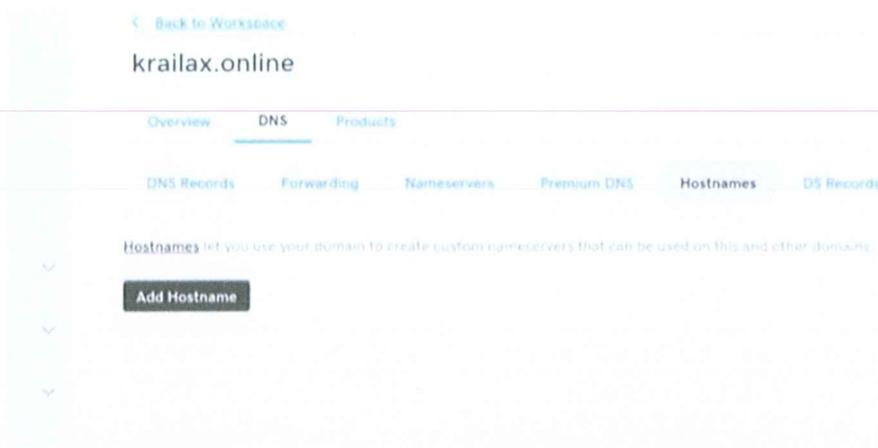


Figura 66. Configuración de Traslado del Dominio de Go daddy

Luego de seguir las instrucciones entramos a cloudflare y añadimos Hostname de go Daddy.

| <input type="checkbox"/> | Tipo | Nombre | Contenido | Estado de proxy | TTL | Acciones |
|--------------------------|-------|-----------------|------------------------------|---------------------------|------------|------------------------|
| <input type="checkbox"/> | CNAME | ._domainconnect | ._domainconnect.gd.domain... | Redirigido mediante proxy | Automático | Editar |

Figura 67. Configuración de Cloudflare para obtener el dominio trasladado.

4.4.3.3 Despliegue del software

4.4.3.3.1 Backend

Despliegue de las Base de datos



Por medio de Pgadmin nos conectamos al VPS que contiene las bases de datos que creamos anteriormente para poder desplegar un backup con la estructura de las entidades que tendrá cada una, entonces lo primero es comprobar que se pueda conectar a las distintas bases de datos que ya están creadas para así poder empezar a gestionar cada una de estas.

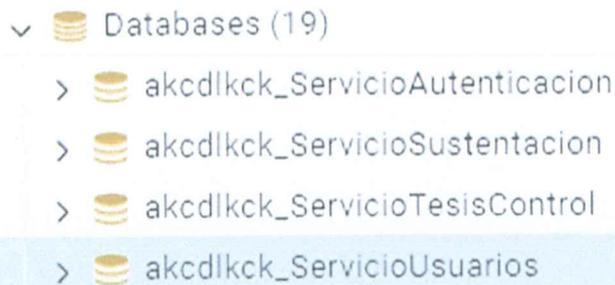


Figura 68. Conexión a bases de datos

Con la conexión ya hecha se puede ingresar el SQL de la estructura de las tablas o generar un backup de estas que solo lleve la estructura de las tablas mas no los datos si no son requeridos para poder acabar de desplegar la base de datos.

Probar la conexión las bases con los microservicios

Una vez hecho reestructurada la base de datos en el VPS se necesita probar que se pueda conectar los microservicios a esta antes de que se haga un build al proyecto y que este sea subido al VPS, en esta parte toca cambiar la configuración del proyecto en las variables de entorno para apuntar al base de datos remotas.

Levantar el contenedor NATs

Después de que en la planificación descargáramos la imagen del NATs y tuviera ya creado el contenedor, ahora lo primordial es encender el contenedor de Docker.



```
root@UbuntuVPS:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
fd56fef495c7   nats     "/nats-server --conf..." 2 days ago    Up 2 days    4222/tcp, 6222/tcp, 8222/tcp       gre
at_gould
```

Figura 69. Servicio NATs en el contenedor levantado

Levantar los microservicios

Para el levantamiento de los microservicios se tiene que compilar los proyectos y una vez compilados enviarlos a los VPS por medio de conexión SSH o usando un GitHub en nuestro caso usando GitHub descargamos los archivos del repositorio para compilarlos en el VPS y poder ejecutarlos

Cuando se compile este solo deberíamos por consola ejecutar la instalación de dependencias con el npm install y luego podremos ejecutar el programa y esperar que este funcione de manera correcta.

```
root@UbuntuVPS:~/var/www/AutenticadorServicio# npm run start
serviciossuavice@0.0.1 start
  > nest start
  >
[00:00:00] 01/12/2025, 01:11:31 PM LOG [NestFactory] Starting Nest application...
[00:00:00] 01/12/2025, 01:11:31 PM WARN [ModuleMetadataFactory] The module "IntercalCoreModule" is taking 10.00ms to serialize, this may be caused by larger object
  statically assigned to the module. More details: https://github.com/nestjs/nest/blob/master/packages/core/ModuleMetadataFactory.ts#L12729-L12730
[00:00:00] 01/12/2025, 01:11:31 PM LOG [InstanceLoader] TypeOrmModule dependencies initialized *19ms
[00:00:00] 01/12/2025, 01:11:31 PM LOG [InstanceLoader] ormModule dependencies initialized *1ms
[00:00:00] 01/12/2025, 01:11:31 PM LOG [InstanceLoader] appModule dependencies initialized *1ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [InstanceLoader] ConfigModule dependencies initialized *1ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [InstanceLoader] TypeOrmModule dependencies initialized *17ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [InstanceLoader] TypeOrmModule dependencies initialized *1ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [InstanceLoader] TypeOrmModule dependencies initialized *4ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [InstanceLoader] TypeOrmModule dependencies initialized *4ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [InstanceLoader] RoleModule dependencies initialized *30ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [InstanceLoader] CareerModule dependencies initialized *4ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [InstanceLoader] UserModule dependencies initialized *4ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [InstanceLoader] AutenticadorModule dependencies initialized *1ms
[00:00:00] 01/12/2025, 01:11:32 PM LOG [NestMicroservice] Nest microservice successfully started *476ms
[00:00:00] 01/12/2025, 01:11:33 PM LOG [Servicio de Usuarios] Servicio de Autenticacion iniciado
```

Figura 70. Microservicio iniciado y corriendo en VPS



4.4.3.3.2 Front End

Primero de todo hay que subir el proyecto en el repositorio de GitHub

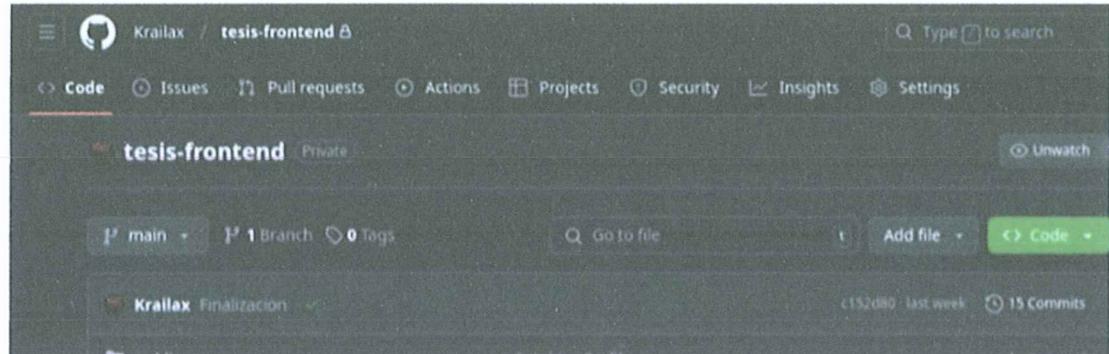


Figura 71. Repositorio de Github del Frontend

Luego de una vez subido el proyecto, nos dirigimos a la plataforma de Vercel e iniciamos sesión usando la cuenta de GitHub para que se nos vincule los proyecto que tenemos en este. Una vez dentro seleccionaremos a crear un nuevo proyecto.

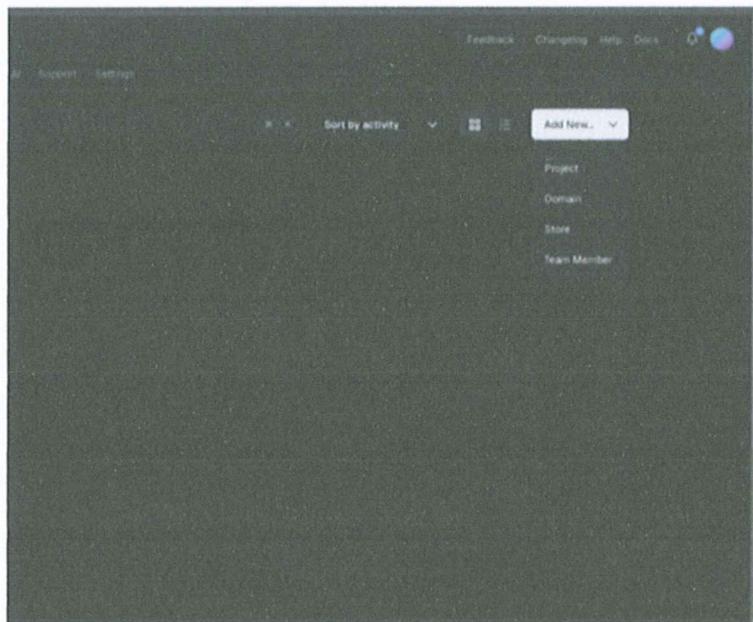


Figura 72. Plataforma de Vercel crear un nuevo proyecto



Una vez hecho eso, nos aparecerá todo nuestro proyecto de GitHub para desplegar seleccionamos es que necesitamos.

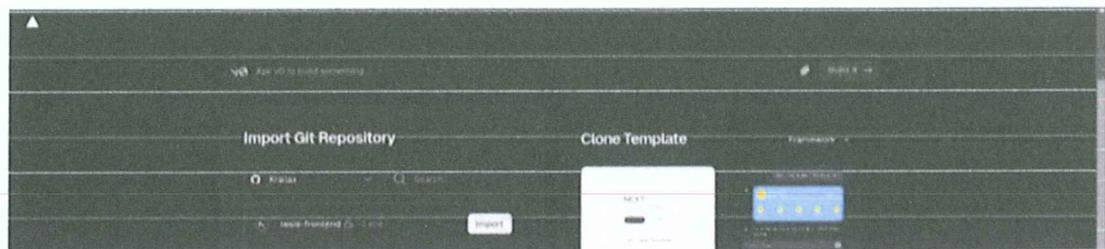


Figura 73. Seleccionando proyecto a desplegar

Luego aparecerán algunas configuraciones; en mi caso, las dejo por defecto y hago clic en el botón de despliegue



Figura 74. Configuración del despliegue del frontend

Allí solo nos tocará esperar hasta que el proyecto se despliegue correctamente. Para cambiar el dominio, hacemos clic donde dice "Domain" para que nos redirija a la página de configuración del dominio.

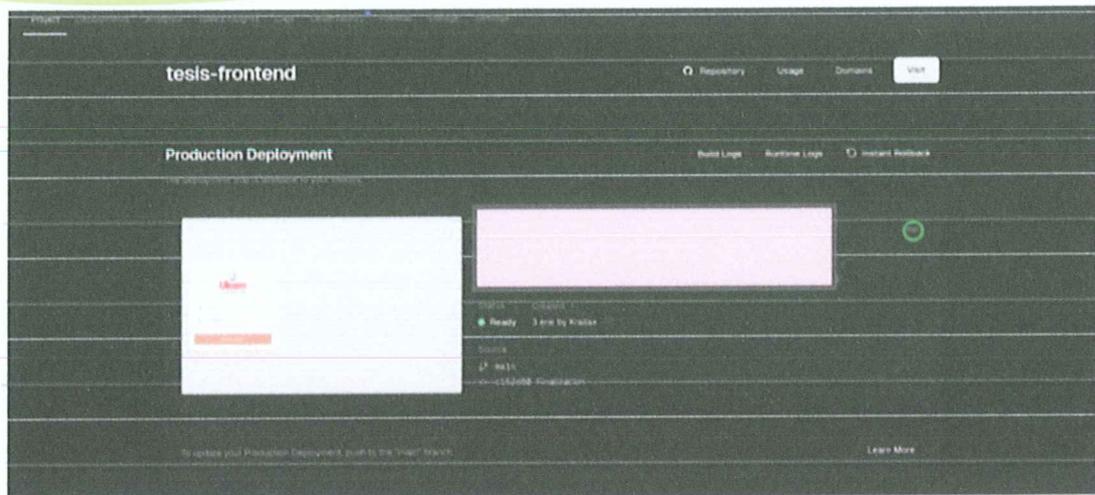


Figura 75. Despliegue de frontend Existoso

Una vez en la interfaz, aparecerá una serie de instrucciones a seguir. En mi caso, eliminé todo y agregué uno nuevo con el dominio adquirido previamente. Primero, nos pedirá que configuremos nuestro DNS de Cloudflare para que apunte a Vercel.

The screenshot shows the Cloudflare DNS records table. At the top, there is a search bar labeled 'Buscar registros DNS' and a button 'Agregar filtro'. Below the search bar is a table with the following columns: Tipo, Nombre, Contenido, Estado de proxy, TTL, and Acciones. The table contains four records, all of type CNAME.

| Tipo | Nombre | Contenido | Estado de proxy | TTL | Acciones |
|-------|-----------------|------------------------------|---------------------------|------------|----------|
| CNAME | ._domainconnect | ._domainconnect.gd.domain... | Redirigido mediante proxy | Automático | Editar |
| CNAME | portafolio | krailax.pages.dev | Redirigido mediante proxy | Automático | Editar |
| CNAME | tesis | cname.vercel-dns.com | Solo DNS | Automático | Editar |
| CNAME | www | cname.vercel-dns.com | Redirigido mediante proxy | Automático | Editar |

Figura 76. Configuración de CNAME del dominio para Vercel

Luego, podemos añadir el dominio a Vercel y esperar entre 10 minutos y 1 hora para que los cambios se apliquen y podamos acceder a la plataforma.

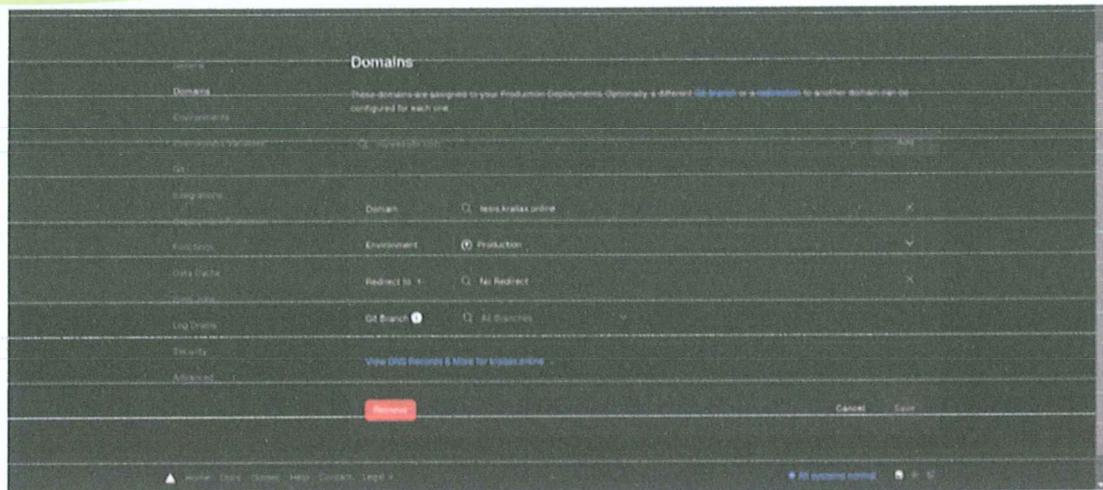


Figura 77. Configuración del dominio del sitio web en Vercel

Como resultado de todos los pasos, tenemos nuestra plataforma web desplegada en Vercel con un dominio personalizado.

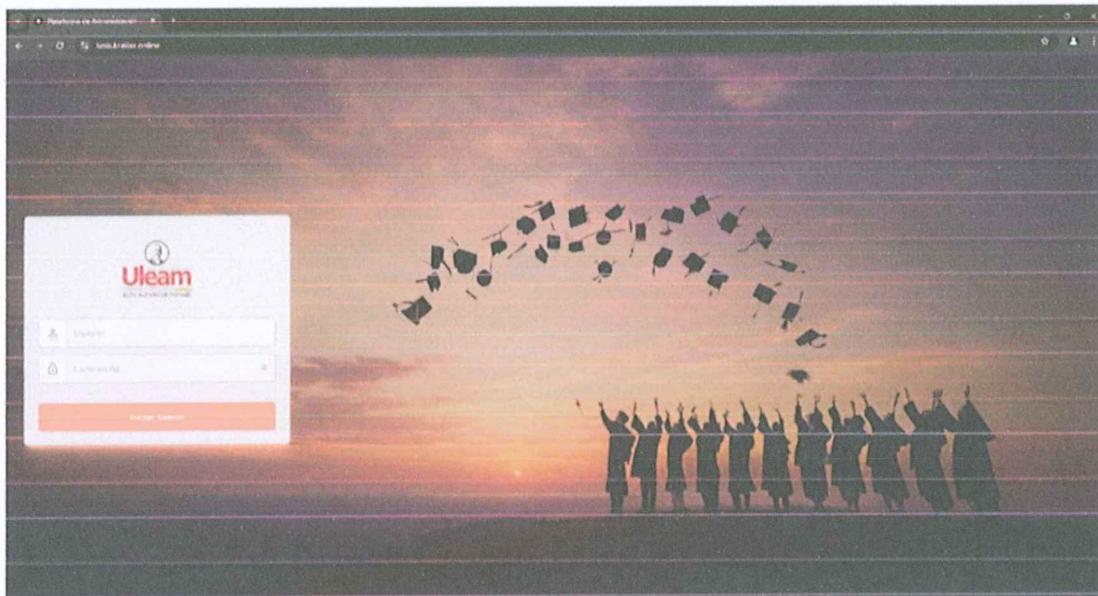


Figura 78. Configuración de Dominio Exitoso



4.5 Conclusión del Capitulo

Se logro desarrollar toda la metodología de trabajo planteada desde la fase de planificación, tomando en cuenta los datos obtenidos del capítulo tres para armar una arquitectura y un diseño que cumple con las necesidades, así después iniciar la fase de desarrollo donde se pudo completar de manera satisfactoria todos los puntos haciendo uso de los microservicios y las herramientas definidas anteriormente y gracias a estas poder someter el software en la ultima fase de pruebas para así comprobar como esta se comporta al pasarla a un entorno de producción con dos VPS donde se pudo comprobar el correcto comportamiento de la aplicación y como esta arquitectura de microservicios necesita ser desplegada.



Capítulo V



5. Evaluación de Resultados

5.1 Introducción

Después del desarrollo de la Aplicación Web para la gestión procesos académicos y documentos del proceso de titulación se realizaron pruebas de las funciones principales con las que cuenta esta aplicación las cuales se mostraran aquí adelante.

5.2 Presentación de resultados

Se realizaron pruebas de control para las funciones del software por cada tipo de rol presente dentro de la aplicación, cada uno con sus distintos privilegios y funciones.

5.2.1 Pruebas desde el usuario estudiante

En la tabla de a continuación de mostrar lo resultado de la prueba que tuvieron los usuarios tipo estudiante en si no tiene tantas funcionalidades más que revisar su tesis y subir documentos.

| ID | Nombre | Seguimiento | Resultado Esperados | Resultado Obtenidos |
|-----|---------------|---|--|---------------------|
| P-1 | Login | Visualización, ingreso de datos y pulsación de botón iniciar sesión | Ingresar lo datos e ingresar a la plataforma | Satisfactorio |
| P-2 | Home | Visualización de bienvenida | Visualicen | Satisfactorio |
| P-3 | Barra Lateral | Manejo de navegación | Redireccionamiento a otras páginas. | Satisfactorio |



| | | | | |
|-------|-----------------------------|---|--|---------------|
| P-3 | Perfil | Visualización de datos personales | Visualización | Satisfactorio |
| P-3.1 | Editar Información personal | Visualización de los campos de ingreso de datos y su relleno con la información del usuario | Ingreso los datos a su respectivo campo de formulario y guardo la información. | Excelente |
| P-4.1 | Tesis | Visualización y navegación a detalle de la tesis | Revisión de tesis y redireccionamiento de la pagina | Excelente |
| P-4.2 | Tesis | Detalles tesis y los detalles de sustentación. | Visualización | Excelente |
| P-4.3 | Tesis | Visualización del documento | Subida y descargar del documento de tesis | Excelente |
| P-5 | Cerrar Sesión | Salir de plataforma | Clic el botón cerrar sesión. | Satisfactorio |

TABLA 33. Pruebas realizadas en usuarios estudiante

5.2.2 Pruebas de usuario secretaria

En la tabla de a continuación de mostrar lo resultado de la prueba que tuvieron los usuarios tipo secretaria en la parte de administrar el sistema.

| ID | Nombre | Seguimiento | Resultado Esperados | Resultado Obtenidos |
|----|--------|-------------|---------------------|---------------------|
|----|--------|-------------|---------------------|---------------------|



| | | | | |
|---------------|-------------------|---|--|----------------------|
| P-1 | Login | Visualización, ingreso de datos y pulsación de botón iniciar sesión | Ingresar lo datos e ingresar a la plataforma | Satisfactorio |
| P-2 | Home | Visualización de bienvenida | Visualicen | Satisfactorio |
| P-3 | Barra Lateral | Manejo de navegación | Redireccionamiento a otras páginas. | Satisfactorio |
| P-3 | Perfil | Visualización de datos personales | Visualización | Satisfactorio |
| P-4 | Cerrar Sesión | Salir de plataforma | Clic el botón cerrar sesión. | Satisfactorio |
| P- 5. | Estudiante | Visualización de los estudiantes | Visualizar y pulsación de botón Activa o Desactivar al estudiante. | Satisfactorio |
| P- 5.1 | Estudiante | Creación o modificación de estudiante por medio del formulario | Primero pulsar botón crear o modificar y luego relleno de formulario y guarda información. | Satisfactorio |
| P- 5.2 | Estudiante | Redireccionamiento a los requisitos del estudiante | Pulsación al botón de redireccionamiento | Satisfactorio |
| P- 6. | Profesor | Visualización de los profesores | Visualizar y pulsación de botón Activa o Desactivar al profesor. | Satisfactorio |



| | | | | |
|---------------|------------------|--|--|----------------------|
| P- 6.1 | Profesor | Creación o modificación de profesor por medio del formulario | Primero pulsar botón crear o modificar y luego relleno de formulario y guarda información. | Satisfactorio |
| P- 6.2 | Profesor | Activación o Desactivación de roles como jurador y tutor | Pulsación de botón Activar o desactivar jurador y tutor | Satisfactorio |
| P-7.1 | Requisitos | Visualización de los requisitos y activación o desactivación | Visualizar, pulsar el botón de activa o desactivar requisito. | Satisfactorio |
| P-7.2 | Requisitos | Formulario de crear o modificar | Pulsar el botón correspondiente, rellenar el formulario y guardar información. | Satisfactorio |
| P-8.1 | Tesis Asignación | Visualización y activación o desactivación de tesis | Visualizar la tesis, correspondiendo a la situación el botón correspondiente activar o desactivar. | Satisfactorio |
| P-8.2 | Tesis Asignación | Formulario de crear o modificar | Pulsar el botón correspondiente, rellenar el | Satisfactorio |



formulario y
selecciona el tutor
correspondiente y
guardar
información.

TABLA 34. Pruebas realizadas en usuarios de tipo secretaria

5.2.3 Prueba desde el usuario profesor

En la tabla de a continuación demostrará lo resultado de la prueba que tuvieron los usuarios tipo profesor, en la actividades correspondiente dependiendo si son tutor o jurador.

| ID | Nombre | Seguimiento | Resultado | Resultado |
|-------|----------------|---|--|---------------|
| | | | Esperados | Obtenidos |
| P-1 | Login | Visualización, ingreso de datos y pulsación de botón iniciar sesión | Ingresar lo datos e ingresar a la plataforma | Satisfactorio |
| P-2 | Home | Visualización de bienvenida | Visualicen | Satisfactorio |
| P-3 | Barra Lateral | Manejo de navegación | Redireccionamiento a otras páginas. | Satisfactorio |
| P-3 | Perfil | Visualización de datos personales | Visualización | Satisfactorio |
| P-4 | Cerrar Sesión | Salir de plataforma | Clic el botón cerrar sesión. | Satisfactorio |
| P-5.1 | Tutor de Tesis | Visualización de tesis asignada como tutor | Visualización de tesis y | Satisfactorio |



| | | | | |
|--------------|------------------------|---|--|----------------------|
| | | | redireccionamiento a la tesis | |
| P-5.2 | Tutor de Tesis | Revisión de documento | Visualización de disponibilidad y descargar del pdf | Satisfactorio |
| P-6.1 | Jurado de Tesis | Visualización de tesis asignada como Jurado | Visualización de tesis y redireccionamiento a la tesis | Satisfactorio |
| P-6.2 | Jurado de Tesis | Revisión de documento | Visualización de disponibilidad y descargar del pdf | Satisfactorio |
| P-6.3 | Jurado de Tesis | Calificación de documento | Crear una calificación a la sustentación | Satisfactorio |

TABLA 35. Pruebas realizadas en usuarios profesor

5.2.4 Prueba desde el usuario administrador

En la tabla de a continuación de mostrar lo resultado de la prueba que tuvieron los usuarios tipo administrador.

| ID | Nombre | Seguimiento | Resultado Esperados | Resultado Obtenidos |
|------------|---------------|---|--|----------------------------|
| P-1 | Login | Visualización, ingreso de datos y pulsación de botón iniciar sesión | Ingresar lo datos e ingresar a la plataforma | Satisfactorio |
| P-2 | Home | Visualización de bienvenida | Visualicen | Satisfactorio |



| | | | | |
|---------------|-------------------|--|--|----------------------|
| P-3 | Barra Lateral | Manejo de navegación | Redireccionamiento a otras páginas. | Satisfactorio |
| P-3 | Perfil | Visualización de datos personales | Visualización | Satisfactorio |
| P-4 | Cerrar Sesión | Salir de plataforma | Clic el botón cerrar sesión. | Satisfactorio |
| P- 5. | Estudiante | Visualización de los estudiantes | Visualizar y pulsación de botón Activa o Desactivar al estudiante. | Satisfactorio |
| P- 5.1 | Estudiante | Creación o modificación de estudiante por medio del formulario | Primero pulsar botón crear o modificar y luego relleno de formulario y guarda información. | Satisfactorio |
| P- 5.2 | Estudiante | Redireccionamiento a los requisitos del estudiante | Pulsación al botón de redireccionamiento | Satisfactorio |
| P- 6. | Profesor | Visualización de los profesores | Visualizar y pulsación de botón Activa o Desactivar al profesor. | Satisfactorio |
| P- 6.1 | Profesor | Creación o modificación de profesor por medio del formulario | Primero pulsar botón crear o modificar y luego relleno de formulario y guarda información. | Satisfactorio |
| P- 6.2 | Profesor | Activación o Desactivación de roles como jurador y tutor | Pulsación de botón Activar o desactivar jurador y tutor | Satisfactorio |



| | | | | |
|--------|------------------|--|--|---------------|
| P-7.1 | Requisitos | Visualización de los requisitos y activación o desactivación | Visualizar, pulsar el botón de activa o desactivar requisito. | Satisfactorio |
| P-7.2 | Requisitos | Formulario de crear o modificar | Pulsar el botón correspondiente, rellenar el formulario y guardar información. | Satisfactorio |
| P-8.1 | Tesis Asignación | Visualización y activación o desactivación de tesis | Visualizar la tesis, correspondiendo a la situación el botón correspondiente activar o desactivar. | Satisfactorio |
| P-8.2 | Tesis Asignación | Formulario de crear o modificar | Pulsar el botón correspondiente, rellenar el formulario y selecciona el tutor correspondiente y guardar información. | Satisfactorio |
| P-9.1 | Rol | Visualización de los datos y botón de activación o desactivación | Visualizar los roles y el pulsar el botón correspondiente al a situación activación o desactivación de roles. | Satisfactorio |
| P-10.1 | Usuario | Visualización de los datos y botón de activación o desactivación | Visualizar los usuarios el pulsar el botón correspondiente al a situación activación o desactivación de roles. | Satisfactorio |
| P-10.2 | Usuario | Formulario de crear o modificar usuarios | Pulsar el botón correspondiente, | Satisfactorio |



| | | | | |
|--------|----------|--|--|---------------|
| | | | rellenar el formulario y selecciona el rol correspondiente y guardar información. | |
| P-11.1 | Carreras | Visualización de los datos y botón de activación o desactivación | Visualizar la carrera y el pulsar el botón correspondiente al a situación activación o desactivación de roles. | Satisfactorio |
| P-11.2 | Carreras | Formulario de crear o modificar carrera | Pulsar el botón correspondiente, rellenar el formulario y correspondiente y guardar información. | Satisfactorio |

TABLA 36. Pruebas realizadas en usuario administrador

5.3 Conclusión

En conclusión, el software de presento un rendimiento correcto en cada parte de la prueba ya que la mayoría de las respuestas a esta fue satisfactoria, además estas fueron hechas en los casos de cada usuario para poder comprobar que estas satisfagan las necesidades funcionales de cada uno de estos dando así una buena de saber que este esta cumpliendo con lo requerido



Capítulo VI



6. Conclusiones y Recomendaciones

6.1 Conclusiones

El desarrollo de la aplicación para la gestión de procesos académicos y documentos de titulación se pudo desarrollar de manera satisfactoria cumpliendo con los requerimientos del proceso de sustentación de tesis que este basado en una arquitectura puramente de microservicios usando tecnologías de comunicación como NATs y aplicando facultades del framework seleccionado para la Api como lo es NestJs y sus bibliotecas de microservicios, pudiendo también controlar los documentos habilitantes de los estudiantes para poder ingresar al proceso de sustentación.

Se hizo uso de herramientas para maquetar y definir la estructura lógica y de arquitectura para la aplicación tanto la parte visual de usuario como para la lógica y conexión de los microservicios, además gracias a los datos obtenidos en las entrevistas profesionales se pudieron definir los requerimiento necesarios e importantes para el software y así poder definir una arquitectura eficiente que se alinea con las necesidades de los actores.

Desarrollamos las distintas funcionalidades que necesita cada uno de los distintos usuarios objetivo de la aplicación con el uso de los microservicios desarrollados para así cubrir todas las necesidades identificadas anteriormente.

Se pudo implementar un despliegue de prueba para comprobar la factibilidad de hacerlo con una arquitectura de microservicios usando los servicios de VPS disponibles obteniendo una respuesta correcta en cuanto a su funcionamiento y la comunicación de estos en este entorno.



6.2 Recomendaciones

Al encargado del Proceso de Tesis de la FCVT de la ULEAM: Considerar expandir el sistema para poder controlar la fecha y los resultados de todas las sustentaciones de tesis para así pueda ser utilizado en todos los contextos y se tenga un mejor control del proceso.

Para el DIIT: Algunos errores posibles que se presenten pueden ser por la posible inconsistencia que se pueda generar en algún salto del flujo normal de datos en la aplicación ya que esta trabaja con más de una base de datos puede que se llegue a saltar alguna validación o que por alguna inserción fuera de la aplicación exista una inconsistencia en los datos generando errores de lectura.

Para Futuras tesis basadas en este proyecto: Si se va a usar una estructura similar a esta o este propio proyecto se recomienda la búsqueda de un proveedor VPS que pueda permitir levantar una estructura de comunicación NATS y tenga las suficientes capacidades para ejecutar más de un proyecto a la vez, ya que ciertos planes te brindan ciertas cosas que otros no, dejando a mitad el levantamiento de la aplicación.



7. Bibliografía

- Agung Nur Aprianto, A. S. (2024). *Analisis Kinerja Streaming RabbitMQ dan Nats untuk Komunikasi di Layanan Mikro.*
- Bejeck, B. (s.f.). *Kafka Streams in Action.*
- Bohora, P. (2021). *Design and Develop Decentralized Microservices Architecture with Docker Container.*
- Daniel López a, E. M. (2017). *Arquitectura de Software basada en Micro servicios para Desarrollo de Aplicaciones Web.*
- ELIZABETH, V. B. (2022). *SOFTWARE INFORMÁTICO PARA EL CONTROL Y REGISTROS DE TEMAS DE TITULACIÓN EN LA CARRERA TECNOLOGÍAS DE LA INFORMACIÓN.* Ipijapa.
- Gallego, M. T. (s.f.). *SCRUM.*
- ING. FRANKLIN JIMMY TÓALA ARIAS, M. E. (2022). *Software Informático Para El Control Y Registros De Temas De Titulación En La Carrera Tecnologías De La Información.*
- Ismael Domínguez Jiménez a, G. S.-B.-S. (2022). *Plataforma web de monitoreo del proceso de titulación del programa de Ingeniería de Software.*
- JACOBSON, R. M. (2017). *Comparativa y estudio de plataformas IoT.*
- Jessica D. Cambi Alvarado, J. A. (2006). *Trabajo de graduación previo a la obtención del título de Ingeniero de Sistemas.*



Joseph Saldaña Torres, R. Z. (2015). *SISTEMA WEB PARA LA GESTIÓN Y ADMINISTRACIÓN DE ANTEPROYECTOS Y TESIS DE GRADO.*

Karen Tatiana Gómez Suárez, R. A. (2018). *Un acercamiento a los microservicios.*

López Villacis Darío Fernando, S. S. (2021). *Aplicación Web Como Estrategia Informática Para La Gestión De Los Perfiles De Titulación De La Facultad De Ciencias Administrativas De La Universidad Técnica De Cotopaxi.*

Luis Alberto Casillas Santillán, M. G. (s.f.). *Bases de datos en MySQL .*

Martín, L. M. (2010). *Diseño y Construcción de Bases de Datos Distribuidas Heterogéneas sobre Oracle Y SQL Server.*

MERCHAN, J. J. (2020). *Aplicación Informatica Para Procesos De Titulacion De La Universidad Estatal Del Sur De Manabí.*

Mesa, E. G. (2006). *La digitalización de documentos, ¿amiga o enemiga?*

Navarrete, J. P. (s.f.). *MICROSERVICIOS.*

Perisé, P. M. (2020). *Arquitecturas de micro servicios sobre infraestructuras distribuidas heterogéneas: una prueba de concepto.*

Richardson, A. (2014). *RabbitQM Essentials.* Livery Place: Packt Publishing Ltd.

RIVILLA, J. E. (2017). *Implementación De Sistema Web Para La Gestión Y Control De Los Procesos De La Unidad De Titulación De La Carrera De Ingeniería En Sistemas De La Universidad Salesiana.*

Sachdeva, S. (2016). *Scrum Methodology.*

SÁNCHEZ, C. A. (2023). *Analisis Comparativo De Los Sistemas De Gestion De Base De Datos Mysql Y Oracle.*



Alberto, L., Santillán, C., Gibert, M., Óscar, G., & Mora, P. (s. f.). *Bases de datos en MySQL*.

Alfonso Zapata, V., Giovany Palacios, A., Alberto Collazos, C., Muñoz-Arteaga, J., Álvarez-Rodríguez, F. J., Silva-Sprock, A., de artículo, C., Ieee, estilo, Zapata, V., Palacios, A., Collazos, C., Muñoz-Arteaga, J., Álvarez-Rodríguez, F., & Silva, A.

(2014). *PATRONES DE MULTICULTURALIDAD PARA EL DISEÑO DE APLICACIONES WEB MULTICULTURALISM PATTERNS FOR WEB APPLICATION DESIGN* Colombia (Vol. 11, Número 11).

Arias-Gonzales, J. (2021). *DISEÑO Y METODOLOGÍA DE LA INVESTIGACIÓN*. <https://www.researchgate.net/publication/352157132>

Belloch, C. (2002). *Las Tecnologías de la Información y Comunicación en el aprendizaje*.

http://www.clubcultura.com/clubliteratura/clubescritores/sampedro/miradas_global.htm

Bernardis, E., Bernardis, H., Berón, M. M., Riesco, D. E., Rangel Henriques, P., & Joao Pereira, M. V. (s. f.). *Técnicas para Incrementar la Seguridad en Web Services basados en WSDL*.

Camilo Gutiérrez, J. R., & Pablo Ramírez, J. J. (2009). *Aplicabilidad de las metodologías de diseño de producto en el desarrollo y creación de páginas web y diseños gráficos*.

Cardacci, D. G. (2015). *ARQUITECTURA DE SOFTWARE ACADÉMICA*. www.cema.edu.ar/publicaciones/doc_trabajo.html



Cárdenas Sánchez, B. C., & Olarte Rojas, C. A. (2022). Análisis de seguridad entre microservicios con Amazon Web Service. *Revista Logos Ciencia & Tecnología*, 14(2), 42-52. <https://doi.org/10.22335/rict.v14i2.1546>

Corbera, F., Alejandro, N., & Gallego, D. (2008). *MODELOS AVANZADOS DE BASES DE DATOS*.

Cristiá, M. (2008). *Introducción a la Arquitectura de Software Testing software from set-based specifications View project*.
<https://www.researchgate.net/publication/251932352>

Curin, B. (2022). *Micro FrontEnd estrategias en el desarrollo de componentes reutilizables*.

Del Carmen, M., & Rivera, G. (2004). *LAS BASES DE DATOS. IMPORTANCIA Y APLICACIÓN EN EDUCACIÓN*.

Gibert, M., Oscar, G., & Mora, P. (2004). *Bases de datos en PostgreSQL*.

Gómez Enciso, E., & Porras Flores, E. E. (2018). Modelo de evaluación de seguridad para transmitir datos usando Web Services. *Industrial Data*, 21(1), 123-132.
<https://doi.org/10.15381/idata.v21i1.14927>

Guerrero, C. A., Suárez, J. M., & Gutiérrez, L. E. (2013). Patrones de diseño GOF (the gang of four) en el contexto de procesos de desarrollo de aplicaciones orientadas a la web. *Informacion Tecnologica*, 24(3), 103-114.
<https://doi.org/10.4067/S0718-07642013000300012>

Jesús Tramullas. (2004). *Diseño y arquitectura de información para el web*.

Karen Monica Hernández Guzmán. (2023). *Adaptación de un patrón de software en seguridad a la arquitectura de un Microservicio*.



Lluís Codina, & Cristòfol Rovira. (2007). *Introducción al diseño de páginas web con estándares y programas open source: Tutorial de XHTML, CSS y NVU*.

López, P., Patricia, H., & Uk, A. (2016). *Comparación del desempeño de los Sistemas*.

Magdalena, M., & Tijerina, H. (2021). *Modelado Funcional de Contenedores Virtuales Docker*.

Mamani Rodríguez, Z. E., Del Pino Rodríguez, L., & Gonzales Suarez, J. C. (2020). Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua. *Industrial Data*, 23(2), 141-149.

<https://doi.org/10.15381/idata.v23i2.17278>

María José García García, Juan José, Escribano Otero, Luis Fernández Sanz, & Pedro J. Lara. (s. f.). *Predicción del tiempo de carga en el diseño de páginas web*.

Montávez Sánchez, & Matías. (s. f.). *MONTAVEZ SANCHEZ, MATIAS_INFORMATICA_TFM*.

Paloma Díaz Susana Montero Ignacio Aedo, M. (2003). *Ingeniería de la web y patrones de diseño Coordinadores*. www.pearsoneducacion.com

Sachdeva, S. (2016). Scrum Methodology. *International Journal Of Engineering And Computer Science*. <https://doi.org/10.18535/ijecs/v5i6.11>

Ugarte Palacin, F. M., & Salvatierra Melgar, Á. (2020). Los procedimientos empíricos en el proceso de titulación profesional en la universidad pública. *Revista Scientific*, 5(16), 82-98. <https://doi.org/10.29394/scientific.issn.2542-2987.2020.5.16.4.82-98>



- Vera-Rivera, F. H., Mauricio, C., Cuevas, G., Astudillo, H., Co, F. E., & Co, M. G. E. (2019). *Desarrollo de aplicaciones basadas en microservicios: tendencias y desafíos de investigación.*
- Vilajosana, X., Leandro, G., & Moldes, N. (s. f.). *Arquitectura de aplicaciones web.*
- Villalobos, G. M., Darío, G., Sánchez, C., Alberto, D., & Gutiérrez, B. (2010). DISEÑO DE FRAMEWORK WEB PARA EL DESARROLLO DINÁMICO DE APLICACIONES. *Scientia et Technica Año XVI, 44.*
- Yusef Hassan, & Francisco J. Martín. (2007). *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información.*



8. Anexo

| | | |
|---|--|---|
| <p>Manual de Usuario - 2024</p> <p>Plataforma de Administración de Sustentación</p> <p>Creado por: Indulgencia Acosta Loiz</p> | <p>Índice</p> <p>1.1. Introducción 4</p> <p>1.2. Propósito del Manual 4</p> <p>1.3. Alcance del Manual 5</p> <p>1.4. Tipos de usuarios 5</p> <p>2. Funcionalidades de Administración y Soporte</p> <p>2.1. Creación de usuarios 4</p> <p>2.1.1. Crear usuarios 4</p> <p>2.1.2. Modificar usuarios 6</p> <p>2.1.3. Habilitar y Deshabilitar usuarios 7</p> <p>2.2. Correos 8</p> <p>2.2.1. Crear correos 8</p> <p>2.2.2. Modificar correos 8</p> <p>2.2.3. Habilitar y Deshabilitar correos 8</p> <p>2.3. Roles 9</p> <p>2.3.1. Habilitar y Deshabilitar usuarios 9</p> <p>2.4. Reservas 10</p> <p>2.4.1. Crear reservas 10</p> <p>2.4.2. Modificar reservas 10</p> <p>2.4.3. Habilitar y Deshabilitar reservas 11</p> <p>2.5. Calendario 11</p> <p>2.5.1. Crear actividades 11</p> <p>2.5.2. Modificar actividades 11</p> <p>2.5.3. Habilitar y Deshabilitar actividades 12</p> | <p>2.5.4. Habilitar y Deshabilitar roles de administración del estudiante 12</p> <p>2.6. Perfil de usuario 12</p> <p>2.6.1. Crear perfil de usuario 12</p> <p>2.6.2. Modificar perfil de usuario 12</p> <p>2.6.3. Habilitar y Deshabilitar perfil de usuario 12</p> <p>2.6.4. Habilitar y Deshabilitar perfil de usuario como profesor 13</p> <p>2.6.5. Habilitar y Deshabilitar perfil de usuario como estudiante 13</p> <p>2.7. Tesis 13</p> <p>2.7.1. Crear tesis 14</p> <p>2.7.2. Modificar tesis 14</p> <p>2.7.3. Habilitar y Deshabilitar tesis 14</p> <p>2.7.4. Asignar Defensor a la tesis 14</p> <p>2.7.5. Eliminar asignación de defensor a la tesis 14</p> <p>2.8. Subseminarios 14</p> <p>2.8.1. Crear subseminarios 14</p> <p>2.8.2. Habilitar y Deshabilitar subseminarios 14</p> <p>2.8.3. Asignar Asesor a la sustentación 14</p> <p>2.8.4. Eliminar Asesor de la sustentación 14</p> <p>3. Funcionalidades de Profesor 14</p> <p>3.1. Documentos 14</p> <p>3.1.1. Descargar Documentos 14</p> <p>3.2. Asignación de Tesis 15</p> <p>3.2.1. Calificar Tesis 15</p> <p>4. Funcionalidades de Estudiante 15</p> <p>4.1. Perfil 15</p> <p>4.2. Documentos 15</p> |
|---|--|---|

Figura 79. Manual de usuario

Link del manual de usuario:

- https://uleam-my.sharepoint.com/:w:/r/personal/e1316215050_live_uleam_edu_ec/Documents/Tesis%20Documentos/Manual%20de%20Usuario.docx?d=wf2c84119bbae4aa4bb7fe5e920ad3794&csf=1&web=1&e=ca7ppY

Entrevista a encargado de proceso de titulación

Entrevistado: JOHN ANTONIO CEVALLOS MACIAS

1. ¿Cómo se lleva a cabo el proceso de presentación de tesis en su totalidad, desde la inscripción del estudiante hasta la entrega final de la calificación? ¿Cuáles son las etapas clave y qué acciones realiza cada actor en cada una de ellas?



2. ¿Cuáles son todos los actores que intervienen en el proceso de presentación de tesis, y qué roles desempeñan cada uno de ellos (estudiantes, jurados, coordinadores, administrativos, etc.)?

3. ¿Qué tipo de documentos deben presentar los estudiantes para la sustentación de su tesis, y qué tipo de documentación es generada por los jurados después de la evaluación (como actas, comentarios, calificaciones)?

4. ¿Cómo se pueden dividir los subprocesos de la presentación de tesis en etapas más específicas? Ejemplos: inscripción, revisión preliminar de la tesis, asignación de jurado, programación de la sustentación, evaluación y entrega de resultados.

5. ¿Existen tiempos establecidos o recomendaciones para cada fase del proceso de presentación de tesis? ¿Cuáles son los tiempos estándar para la carga de la tesis, la asignación de jurados, la revisión, y la entrega de calificaciones?

6. ¿En caso de retrasos o problemas inesperados (como la ausencia de un jurado o dificultades técnicas), ¿existen protocolos para extender los tiempos establecidos o para ajustar las fechas de entrega?

7. ¿Cómo se lleva a cabo el proceso de selección de los jurados para la sustentación de tesis? ¿Qué criterios se toman en cuenta (experiencia, especialización en el tema, disponibilidad)?

8. ¿Cuántos jurados participan generalmente en la evaluación de una tesis? ¿Este número varía según el tipo de tesis o el área de conocimiento?



9. **¿Cómo se organiza la comunicación entre los jurados y los estudiantes durante el proceso de revisión de la tesis? ¿Los jurados pueden interactuar con los estudiantes para hacer preguntas o sugerencias antes de la sustentación?**

10. **¿Existe algún miembro especial dentro del jurado, como un presidente o moderador, que desempeñe un rol adicional o tenga más responsabilidades que los otros jurados? ¿Cómo influye este rol en el proceso?**

11. **¿El miembro especial tiene algún poder adicional en caso de desacuerdos entre los otros jurados o en la toma de decisiones finales?**

Entrevista a secretaria encargada de los requisitos de sustentación

Entrevistado: Maria Elena Garcia Velez

1. **¿Cuánto tiempo toma subir la información de datos generales y cómo se hace este proceso?**

2. **¿Cuánto tiempo tienen los estudiantes para entregar esta información?**

3. **¿Cómo se organiza la recepción de documentos?**

4. **¿Qué pasa si un estudiante cumple con todo?**

5. **¿Qué documentos son necesarios para el registro del título?**

6. **¿Qué pasa si un estudiante no cumple con los requisitos?**

7. **¿Qué pasa antes y después de la sustentación?**

