

**UNIVERSIDAD DE MATANZAS
FACULTAD DE CIENCIAS TÉCNICAS
DEPARTAMENTO DE INFORMÁTICA**



**Sistema de Gestión de la Información en el Proceso de Matriculas al Programa de
Formación Doctoral de la Universidad de Matanzas
Ejercicio Profesional en opción al título de Ingeniero Informático.**

Autor: Alfredo David Morales Alfonso.

Tutor: Dr. Walfredo González Hernández .

Matanzas, 2023

Pensamiento

“El conocimiento no es una vasija que se llena sino un fuego que se enciende.”

Plutarco

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas, especialmente a la Facultad de Ciencias Técnicas a que hagan el uso que estimen pertinente de él. Y para que así conste, firmo la presente a los _____ días del mes de _____ del 2022.

Firma de Autor

Firma de Tutor

Opinión del Tutor

Agradecimientos

El camino de la formación profesional de un estudiante constituye un requerimiento a cumplir basado en tiempo, esfuerzo y confianza. La tarea de hacerse siempre va, en gran parte, con el trabajo realizado y el apoyo brindado por nuestros cercanos en estrecha sinceridad. Agradecer por este logro a un gigante como el Dr. Walfredo González Hernández, por sus enseñanzas, apoyo y en cuestiones extraprofesionales por ser como un amigo. A la mujer que hoy es mi esposa por nunca darse por vencida conmigo y estar siempre que yo no estaba para mí. A mis pocos amigos y el tiempo que se tomaron para levantarme siempre que consideraba este trayecto un camino opaco e imposible, en especial a Daniel Menéndez por contribuir en creces a mi preparación durante toda esta etapa. En especial agradecimiento a mis padres, por ser siempre faros y soporte durante toda mi vida y en especial en estos cuatro años de formación profesional. A la enseñanza superior en la Universidad de Matanzas por enseñar que cuando los recursos son pocos, el carácter propio forma, enseña y prepara para la vida.

Resumen

En la Universidad de Matanzas, el proceso de matrícula en el Programa de Formación Doctoral comienza cuando un solicitante desea matricular en un programa de Formación Doctoral en la Universidad de Matanzas. Este solicitante puede ser cubano o bien extranjero, y para cada aspecto se requiere una documentación distinta. El solicitante comienza el intercambio a través del taller, organizado por el jefe de línea y en cuya labor se emiten criterios que posteriormente son elevados al comité doctoral para su posterior aprobación. En este comité doctoral se procesa estos criterios y la decisión de este es elevada por el coordinador del programa a la comisión de grados de la Universidad. Esta comisión decide sobre la aprobación o no del tema y los tutores propuestos y en caso de aprobarse se realiza la matrícula, para la que es necesario entregar una documentación y se declara oficialmente como doctorando. Una vez declarado, se ubica formalmente la línea de investigación asociada a su tema y culmina el proceso de matrícula del aspirante, ahora doctorando.

Índice

Introducción	1
Capítulo 1. Marco Teórico Referencial	5
1.1 Descripción de la organización Objeto de Estudio	5
1.1.2 Descripción del negocio	6
1.1.3 Realización del proceso en la actualidad	8
1.2 Metodologías de desarrollo de software	8
1.2.1 Por qué seleccionar RUP?:.....	9
1.3 Definiciones Generales	9
1.3.1 Lenguaje de Programación	9
1.3.2 Aplicación Web	10
1.4 Tecnologías utilizadas para la construcción de Interfaces de Usuario (Frontend)	10
1.4.1 Lenguaje Estructural Utilizado.....	10
1.4.2 Lenguaje de Estilos Utilizado	10
1.4.3 Lenguaje de Implementación de Funcionalidades.....	11
1.4.4 Framework de desarrollo	11
1.5 Tecnologías utilizadas para la construcción de Modelos y Controladores (Backend).....	11
1.5.1 Entorno de Ejecución Utilizado	11
1.5.2 Framework de desarrollo	12
1.6 Gestión de Datos.....	12
1.6.1 Gestor de Bases de Datos	12
1.6.2 Administrador de Bases de Datos	13
1.7 Otras Herramientas de Desarrollo	13
Visual Studio Code es un editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. La elección principal para desarrolladores web y JavaScript, con extensiones para admitir casi cualquier lenguaje de programación.(Microsoft, s. f.).....	13
1.8 Algunas definiciones	13
1.8.1 Patrón Modelo-Vista-Controlador	13
1.9 Conclusiones Parciales del Capitulo.....	13
Capítulo 2: Análisis, diseño y desarrollo de la solución propuesta	14
2.1 Descripción de la solución	14
2.1.1 Herramientas y control de versiones	14
2.1.2 Roles de Usuario	15

2.2 Etapas de Desarrollo.....	15
2.2.1 Fase de Inicio	16
2.2.2 Fase de Elaboración	18
2.2.3 Fase de Implementación.....	30
2.2.4 Fase de Transición	53
2.3 Modelado de los Datos.....	53
2.3.1 Modelo Conceptual	54
2.3.2 Modelo Lógico	55
2.3.3 Modelo Físico	55
2.4 Análisis del costo del Software desarrollado.....	56
□ Análisis de los beneficios.....	62
2.5 Conclusiones Parciales del Capitulo.....	62
Capítulo 3: Validación de la solución propuesta.....	62
3.1 Medición de las variables de Investigación.....	62
3.1.1 Medición de las variables dependientes.....	63
3.2 Valoración de la solución obtenida y cumplimiento de la Hipótesis.....	64
3.3 Vistas del Software.....	65
3.4 Conclusiones Parciales del Capitulo.....	67
Conclusiones Generales.....	67
Recomendaciones	68
Bibliografía.....	68
Anexos	71

Introducción

En general, las Tecnologías de la Información y la Comunicación (TIC) han transformado la gestión de matrículas en el proceso doctoral, permitiendo una mayor eficiencia, transparencia y accesibilidad a la educación superior. Además, la tecnología ha permitido una mayor transparencia y eficacia en la gestión de la información de los estudiantes, facilitando la toma de decisiones y la planificación de recursos por parte de las universidades.

La formación doctoral es de suma importancia en el ámbito académico y profesional. Este proceso educativo avanzado proporciona a los estudiantes las herramientas necesarias para convertirse en expertos en su campo de estudio. Una de las principales razones por las que la formación doctoral es crucial es porque fomenta la investigación avanzada. Los estudiantes de doctorado tienen la oportunidad de llevar a cabo investigaciones originales y contribuir al conocimiento existente en su disciplina. Esto implica aprender a diseñar y llevar a cabo estudios, analizar datos y comunicar los resultados de manera efectiva.

Además, la formación doctoral es un catalizador para el desarrollo de habilidades. Durante este proceso, los estudiantes adquieren habilidades transferibles como el pensamiento crítico, la resolución de problemas, la comunicación oral y escrita, el liderazgo y la colaboración. Estas habilidades son altamente valoradas en diversos entornos profesionales y académicos. Un título de doctorado también abre puertas en términos de avance profesional. En muchos campos, es un requisito para acceder a oportunidades de alto nivel en la investigación, la docencia universitaria, la consultoría y la alta dirección. La formación doctoral demuestra un nivel avanzado de conocimientos y experiencia, lo que aumenta las perspectivas laborales.

Además, los doctores desempeñan un papel fundamental en el avance del conocimiento y la sociedad en general. Sus investigaciones y publicaciones contribuyen al desarrollo científico, la innovación tecnológica y la mejora de políticas y prácticas. En conclusión, la formación doctoral es esencial para el progreso académico y profesional. Proporciona una base sólida de conocimientos, promueve la investigación avanzada y el desarrollo de habilidades, y abre puertas a nuevas oportunidades. Si estás considerando embarcarte en un programa de doctorado, debes evaluar cuidadosamente tus objetivos y aspiraciones para aprovechar al máximo esta experiencia enriquecedora.

Entre algunas de las aplicaciones consultadas para el desarrollo de esta investigación se encuentra Banner, desarrollada por Ellucian. Ellucian, una empresa que se centra en software para la educación superior, proporciona Banner a la universidad (ver Anexo 1). También puede escuchar los términos “SSB (o Banner de autoservicio o Banner de

autoservicio)” o “Navegador de Aplicaciones (o App Nav)”); Estas son sólo dos formas diferentes que tiene Banner de presentar información al usuario. Generalmente, la mayoría de las personas trabajan en SSB cuando usan Banner, pero el personal y los profesores que tienen acceso de nivel superior a los datos de la universidad usarán Application Navigator.(OIT Support, 2022)

Otro de los antecedentes encontrados es PowerSchool (Ver anexo 2). PowerSchool es el sistema de información estudiantil de más rápido crecimiento y más utilizado, que atiende a decenas de millones de estudiantes en todo el mundo. PowerSchool Mobile mejora la participación de los padres y la responsabilidad de los estudiantes con un acceso fácil e instantáneo a la asistencia en tiempo real, tareas, puntajes, calificaciones y más. Los padres o tutores con varios estudiantes pueden alinear a todos los estudiantes en una sola cuenta, eliminando la necesidad de recordar diferentes cuentas de inicio de sesión y contraseñas para ver los detalles de los estudiantes(data.ai, s. f.)

En el caso del Proceso de Matriculas en el Programa de Formación Doctoral de la Universidad de Matanzas, durante la investigación, los estudiantes trabajan en su proyecto de tesis, bajo la supervisión de un tutor y un comité de tesis. Los Doctorandos o Aspirantes deben defender su tesis ante el comité de tesis para obtener su título doctoral. El proceso de formación doctoral en Cuba generalmente toma de tres a cinco años, dependiendo del campo de estudio, y se enfoca en la investigación y el desarrollo científico en áreas de alta prioridad para el país, como la salud, la agricultura y las energías renovables.

Estas etapas antes mencionadas parten de un primer momento, donde el aspirante realiza la presentación de su tema en el Programa de Formación Doctoral de la Universidad y lo defiende en talleres especializados para dicha labor. Una vez aprobado el tema, el aspirante está listo para hacer su solicitud de matrícula y comenzar su preparación como doctorando hasta convertirse en Doctor en Ciencias.

Sin embargo, en la Universidad de Matanzas, este proceso se realiza de manera manual y presencial, lo que conlleva el procesamiento de grandes volúmenes de documentos en períodos de tiempo cortos para otorgar los créditos por cada una de las actividades que realiza el aspirante dentro del programa. A partir de la problemática anterior surge el **problema de investigación** que da origen al presente trabajo de diploma:

¿Cómo gestionar la información asociada al proceso de matrícula al Programa de Formación Doctoral en la Universidad de Matanzas?

Este problema se enmarca en el **objeto de estudio**: Matrícula al Programa de Formación Doctoral.

Delimitándose como **campo de acción**: La informatización del proceso de matrícula al programa de Formación Doctoral en la Universidad de Matanzas`

Se plantea como **hipótesis** que, si se desarrolla una aplicación web para la gestión de la información del proceso de matrícula al Programa de Formación Doctoral, contribuirá a la persistencia de la información asociada a este proceso.

Objetivo general:

Desarrollar una aplicación web que contribuya a la persistencia en la gestión de la información del proceso de matrícula al Programa de Formación Doctoral en la Universidad de Matanzas

Objetivos Específicos:

- Determinación de los referentes teórico-metodológicos de la gestión de la información de los procesos de matrícula en programas de postgrado
- Análisis de la aplicación web que contribuya a la eficiencia en la gestión de la información del proceso de matrícula al Programa de Formación Doctoral en la Universidad de Matanzas
- Diseño de la aplicación web que contribuya a la eficiencia en la gestión de la información del proceso de matrícula al Programa de Formación Doctoral en la Universidad de Matanzas
- Implementación aplicación web que contribuya a la eficiencia en la gestión de la información del proceso de matrícula al Programa de Formación Doctoral en la Universidad de Matanzas
- Validación de la aplicación web que contribuya a la eficiencia en la gestión de la información del proceso de matrícula al Programa de Formación Doctoral en la Universidad de Matanzas.

Se definen como **variables** las expuestas a continuación:

- **Variable dependiente:** Persistencia de la información asociada al proceso descrito
- **Variable Independiente:** La aplicación web

Esta investigación de justicia basada en los siguientes aspectos:

Justificación Practica

- El software obtenido será utilizado por los miembros del departamento que conducen el programa de formación doctoral
- Se mejora la experiencia de usuario para los aspirantes que soliciten cursar un doctorado en la Universidad de Matanzas
- El software obtenido contiene un módulo de gestión de los miembros del departamento encargado, así como de cada una de las categorías docentes y especialidades.

Justificación Metodológica

- El resultado de la investigación es un software de gestión para el proceso de solicitud de un aspirante para convertirse en doctorando
- Puede ser empleado en investigaciones relacionadas con los doctorandos que realizan su preparación profesional en la Universidad de Matanzas
- La propuesta de otros módulos para la gestión de la formación de los doctorandos puede solicitar información al software en cuestión

Para el desarrollo de la investigación se emplearon los siguientes **métodos y técnicas**:

- **Métodos teóricos:**

Análisis Histórico-lógico: se empleó para comprender el devenir histórico de los programas de formación doctoral, así como las condiciones y antecedentes que favorecieron su surgimiento y posterior auge como objeto de estudio, basándose en una detallada revisión bibliográfica.

Enfoque de sistema: Proporcionó la orientación general de la aplicación web como un sistema de clases que se relacionan entre sí a partir de la comunicación que establecen entre ellas.

- **Métodos empíricos** abordados a través de las técnicas siguientes:

Entrevista: Se realizó para recopilar información a través de una planificada conversación junto a los especialistas de diversos temas del objeto de estudio, cuyo análisis posterior posibilita el avance en la investigación, incluyendo el adecuado levantamiento de los requisitos del negocio.

Como **resultados obtenidos** se tienen los siguientes:

- Mejora de la persistencia y estructuración en el proceso de matriculación: La aplicación web reduce el esfuerzo y los recursos necesarios para llevar a cabo el proceso de matriculación, ya que permitiría a los estudiantes completar el proceso en línea en lugar de tener que hacerlo de forma presencial.
- Reducción de errores en la matriculación: Al automatizar el proceso de matriculación, se reduce la posibilidad de errores humanos, lo que mejora la precisión y la integridad de los datos de matriculación
- Mayor satisfacción del estudiante: Al ofrecer una forma más sencilla y conveniente de matricularse, se aumenta la satisfacción del estudiante y, por lo tanto, mejora su experiencia general en el proceso doctoral.
- Mayor eficacia en la gestión de datos: La aplicación web facilita la gestión de los datos de matriculación, lo que permite un seguimiento más efectivo de los estudiantes y una mejor capacidad de análisis de los datos de matriculación.

En correspondencia con lo tratado anteriormente, el documento queda estructurado como sigue a continuación:

Capítulo I: Marco teórico referencial

Recoge los conceptos fundamentales asociados al dominio del problema. Realiza un estudio de las tendencias y tecnologías actuales sobre las que se apoya la propuesta y analiza las soluciones existentes.

Capítulo II: Análisis, diseño y desarrollo de la solución propuesta

Se expone la propuesta de solución al problema de investigación, la organización y disposición del proyecto mediante la presentación de una planificación inicial haciendo uso del marco de trabajo de desarrollo de software RUP. Es analizado y estudiado los beneficios obtenidos mediante la implementación del proyecto de software.

Capítulo III: Validación de la solución propuesta

Se hace el análisis de los resultados obtenidos. También se realizan las pruebas al software con el objetivo de entregarle al cliente un producto totalmente funcional, cumpliendo con todos los requisitos demandados por el mismo y satisfaciendo sus necesidades.

Finalmente, son presentadas las **Conclusiones y Recomendaciones** de la investigación para dejar el camino abierto a futuros estudios relacionados con la temática abordada.

De igual forma, quedan recogidas las **Bibliografías** utilizadas y **Anexos** que fueron necesarios para el desarrollo del trabajo de investigación y mejor entendimiento del mismo.

Capítulo 1. Marco Teórico Referencial

En el presente capítulo se aborda los fundamentos teóricos, antecedentes y métodos que posee la investigación. Son reflejados los principales conceptos que soportan el objeto de estudio para brindar una mejor comprensión del mismo. Es analizada la metodología de desarrollo escogida para garantizar la adecuada realización de la aplicación en correspondencia con las características del proyecto a implementar. También, se hace referencia a las distintas tecnologías y herramientas seleccionadas para el desarrollo del software.

1.1 Descripción de la organización Objeto de Estudio

El programa de formación doctoral de la Universidad de Matanzas se enfoca en fomentar la excelencia académica y la capacidad investigativa de los estudiantes. Les proporciona las herramientas teóricas y metodológicas necesarias para llevar a cabo investigaciones de alta calidad y desarrollar habilidades avanzadas en su campo de estudio.

Promueve la interdisciplinariedad y la colaboración entre estudiantes y profesores, fomentando un entorno propicio para el intercambio de conocimientos y la generación de ideas innovadoras. Los estudiantes de doctorado tienen la oportunidad de participar en proyectos de investigación conjuntos, seminarios,

conferencias y otras actividades académicas que enriquecen su formación y se preocupa por el desarrollo integral de los estudiantes, brindándoles apoyo y orientación en su proceso de investigación. Se les proporciona asesoramiento académico y supervisión por parte de profesores especializados, quienes los guían en la elaboración de sus tesis doctorales y en términos específicos pudiesen citarse otras responsabilidades como:

- Selección y admisión de estudiantes: El programa de formación doctoral lleva a cabo un proceso de selección riguroso para admitir a los **temas pertinentes, actuales y novedosos**. Se evalúan sus antecedentes académicos, habilidades de investigación y aptitudes de los futuros doctorandos para determinar su idoneidad para el programa.
- Diseño de planes de estudio: El programa desarrolla planes de estudio detallados y estructurados que brindan una base sólida de conocimientos en el área de estudio elegida. Estos planes incluyen cursos teóricos y prácticos que permiten a los estudiantes profundizar en su campo y adquirir habilidades avanzadas de investigación.
- Asesoramiento académico y supervisión: Los estudiantes de doctorado reciben asesoramiento académico y supervisión por parte de profesores experimentados y especializados en su área de investigación. Estos profesores los guían en la elaboración de sus proyectos de investigación, les brindan retroalimentación y los apoyan en la superación de obstáculos que puedan surgir durante el proceso.
- Desarrollo de habilidades de investigación: El programa de formación doctoral se enfoca en el desarrollo de habilidades de investigación avanzadas. Los estudiantes aprenden a formular preguntas de investigación pertinentes, diseñar metodologías adecuadas, recopilar y analizar datos, y comunicar los resultados de manera efectiva a través de publicaciones y presentaciones académicas.
- Participación en actividades académicas: El programa fomenta la participación activa de los estudiantes en conferencias, seminarios, talleres y otras actividades académicas. Esto les brinda la oportunidad de compartir sus investigaciones, recibir retroalimentación de expertos en su campo y establecer contactos con otros investigadores.

1.1.2 Descripción del negocio

El proceso de Matrículas en el Programa de Formación Doctoral de la Universidad de Matanzas comienza cuando un solicitante (o aspirante) decide presentar al Programa de Formación Doctoral de la Universidad de Matanzas su tema para convertirse en doctorando. Una vez presenta la documentación pertinente (ya sea cubano o

extranjero), este aspirante puede realizar intercambios con el programa para lograr la aprobación de su tema.

Este programa se encuentra formado por 3 líneas de investigación: **Formación de niños, adolescentes y jóvenes en condiciones de disfunción social y educativa, Formación de profesionales en la Educación Superior y Estudios sociales y humanísticos en la educación.** La línea de investigación es la encargada de conducir la formación del doctorando hasta la predefensa y por cada línea de investigación existen asociados doctores, doctorandos, comisiones evaluadoras y proyectos de investigación. Cada una de ellas posee un jefe de Línea y su función es la de organizar los talleres de aprobación, aprobar los criterios que se plantean en estos talleres y entregarlos al Comité Doctoral durante esta etapa.

Se describe como **Taller de Aprobación** al tribunal encargado de conducir las defensas que realiza el aspirante para la aprobación de su tema y está conformado por todos los miembros de la línea asociada a este tema. Este taller constituye el espacio de funcionamiento de la línea y sus criterios se rigen por un grupo de indicadores para decidir sobre la propuesta de aprobación o no del tema. Este compendio de criterios que se exponen en este espacio es aprobado por el Jefe de Línea y este se encarga de elevarlos al Comité Doctoral para su posterior procesamiento. Cabe destacar que un aspirante se presentara al Taller de Aprobación tantas veces se requiera hasta que se apruebe.

En el programa existe un **Comité Doctoral**, que es el encargado de realizar el compendio de evaluaciones con la propuesta de los tutores una vez el doctorando demuestra la solidez y pertinencia de su tema durante la defensa de este en el taller de aprobación. Este compendio de evaluaciones y propuesta de tutor (o tutores), es elevada por el coordinador del programa a la Comisión de Grados para su aprobación oficial.

La **Comisión de Grado** es la encargada de aprobar o no la propuesta emitida por el Comité Doctoral. Una vez aprobado el tema y los tutores se entrega la documentación pertinente para realizar la matricula del aspirante y se declara oficialmente como Doctorando. Luego es ubicado en la línea correspondiente, que fue encargada de realizar el seguimiento de las defensas durante el taller de aprobación.

En caso de no aprobarse el tema en el Comité Doctoral o por la Comisión de Grado de la Universidad, el coordinador del programa informa de las insuficiencias al aspirante y este debe realizar el proceso nuevamente hasta que se oficialice si condición de Doctorando

1.1.3 Realización del proceso en la actualidad

En la Universidad de Matanzas, este proceso se realiza de manera manual y presencial, en el cual se aglomeran un conjunto de dificultades a la hora de llevar la constancia de cada una de las actividades que realiza el aspirante dentro del programa. Por citar algunas se tienen las siguientes:

- Es un proceso tedioso y propenso a errores, ya que implica la manipulación manual de grandes cantidades de papel y datos, lo que aumenta la posibilidad de errores humanos.
- La gestión manual de la matrícula puede ser muy costosa en términos de esfuerzo y recursos, ya que implica la impresión y distribución de formularios, la recopilación y el procesamiento manual de la información de los estudiantes, y la realización de tareas administrativas repetitivas.
- La gestión del flujo del proceso, debido a su complejidad es difícil de comprender en un inicio, debido a que el aspirante debe estar en un constante intercambio presencial con el programa sin ser esta una necesidad obligatoria
- Puede restringir el acceso a la educación superior, ya que requiere que los estudiantes se presenten físicamente en las oficinas universitarias para realizar la matrícula, lo que puede ser difícil para aquellos que viven en áreas remotas o tienen dificultades para desplazarse.

1.2 Metodologías de desarrollo de software

Una metodología de desarrollo se puede definir como un conjunto de procedimientos que indican los roles quienes ejecutan las acciones de desarrollo, las actividades a ejecutar, así como los elementos esenciales a medir en cada proceso (González Hernández 2019). Para otros autores, es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a la producción de software (Zhi et al. 2023). En los procesos de desarrollo siempre se toman decisiones en cuanto al tipo de metodología a utilizar porque se dividen en dos grupos. Un primer grupo denominado metodologías tradicionales y un segundo llamado metodologías ágiles. Las primeras estructuran sus procesos a partir de un uso intensivo de la documentación de todo el proceso del ciclo de vida del proyecto, mientras que las ágiles dan mayor importancia a la capacidad de respuesta a los cambios.

Las metodologías ágiles preponderan la codificación que los planes y la documentación por lo que planifican iteraciones cortas donde participa el cliente como parte del equipo de desarrollo. Su visión del desarrollo de software está basada en el desarrollo rápido y funcional, que el proceso de desarrollo responda a los cambios a seguir estrictamente un plan. Los equipos evalúan el proyecto en reuniones regulares, llamadas sprints o iteraciones.

Metodologías tradicionales plantean controles continuos durante todo el proceso de desarrollo, la utilización de los artefactos adecuados y, sobre todo, a documentar la selección de herramientas y metodologías, así como todas las fases de desarrollo. Al ser utilizada en casos donde los proyectos son grandes e involucran a varias personas pues deben centrarse en la documentación como vía de comunicación entre los actores de cada proceso para cumplir con un plan de proyecto.

Después de haber realizado una comparación entre las metodologías ágiles y las tradicionales, determina que en el caso de este proyecto es preferible el uso de metodologías pesadas porque se trata de un software que tiene gran complejidad y es parte de un proyecto en desarrollo que contiene 4 módulos. Dentro de las metodologías pesadas la más conocida es Rational Unified Process (RUP) (Bouihia and Bahaja 2019; Cvetković and Cvetković 2019; Ozkaya and Erata 2020).

1.2.1 ¿Por qué seleccionar RUP?:

Luego de analizar estas metodologías se escoge RUP por las siguientes razones:

- ✓ Teniendo en cuenta las características del equipo de desarrollo, el cual está conformado por 4 individuos, resulta apropiado la implementación de prácticas correspondientes a la metodología RUP, tales como la utilización intensiva de documentos.
- ✓ La aplicación que se desarrolla es por paquetes y guarda mucha relación con los procesos que realiza el cliente en su quehacer diario, pero por su tamaño, requiere una documentación exhaustiva.
- ✓ El proyecto está basado en la ley de procesos de formación doctoral recién aprobada por lo que su desarrollo tiene en cuenta procesos que no serán modificables al menos 5 años después de su despliegue.
- ✓ RUP permite que los desarrolladores puedan interactuar libremente entre ellos.

A raíz del análisis anteriormente realizado y teniendo en cuenta que el cliente es también tutor del proyecto, se puede afirmar que el sistema a desarrollar puede culminarse con resultados exitosos aplicando la metodología RUP.

1.3 Definiciones Generales

1.3.1 Lenguaje de Programación

Los lenguajes de programación son herramientas fundamentales en el desarrollo de software y aplicaciones, ya que permiten a los programadores comunicarse con las máquinas y crear instrucciones que las ejecuten. En la actualidad, los más utilizados son los lenguajes de alto nivel, caracterizados por ser los más cercanos a los lenguajes natural y matemático y tener mayor legibilidad, mayor facilidad de codificación (Luis Hernández Yáñez, 2014). Para el desarrollo de aplicaciones web, el modelo MVC (Modelo Vista-Controlador), es el más utilizado y se basa en la arquitectura o patrón

arquitectónico Cliente Servidor, donde los lenguajes que destacan en cada una de las partes se describen a continuación

1.3.2 Aplicación Web

Las aplicaciones se han convertido en parte de la vida cotidiana debido a su uso en smartphones y tablets, aunque este no está restringido solo a los dispositivos móviles, sino que también hay algunas aptas para ordenadores de escritorio. Mientras que una aplicación nativa es un tipo de software que se adapta a una plataforma determinada, una web app tiene lugar en el navegador web y guarda ciertas diferencias con respecto a las denominadas aplicaciones nativas.

Una web app (aplicación web en español) se basa en HTML, JavaScript o CSS. Puesto que se carga en el **servidor web** y se ejecuta en el navegador, no requiere **ninguna instalación**. Además, también se puede crear un acceso directo para ella en el escritorio del ordenador o en la pantalla de inicio de los terminales móviles mediante un marcador (lonos, 2019). No hay diferencias ente descargarse una aplicación y usar las aplicaciones web. Se comparte el mismo diseño y el cliente goza de la misma rapidez. Al aumentar la facilidad de uso, también lo hace la fidelización y el aumento de los seguidores (Zumbana Santamaria 2021).

1.4 Tecnologías utilizadas para la construcción de Interfaces de Usuario (Frontend)

1.4.1 Lenguaje Estructural Utilizado

HTML, (HyperText Markup Language) es el lenguaje estándar para la creación de documentos destinados a ser visualizados en un navegador web. Define la estructura y significado del contenido web, y es asistido por tecnologías como CSS y JavaScript. Este lenguaje ha evolucionado desde su creación en 1990, con la versión HTML5 ofreciendo nuevas etiquetas y características para crear páginas web más interactivas y dinámicas. Es esencial para cualquier carrera en el desarrollo web y su comprensión básica es crucial para trabajar en este campo

HTML es la base de la mayoría de las páginas web: es la forma en que les decimos a los navegadores que estructuren el contenido en títulos, encabezados, párrafos, imágenes, enlaces, listas, formularios, tablas y más.(Coppola Maria, s. f.)

1.4.2 Lenguaje de Estilos Utilizado

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en un lenguaje de marcado como HTML o XML (incluyendo dialectos como SVG, MathML o XHTML). CSS es una tecnología clave de la World Wide Web junto a HTML y JavaScript. Su objetivo principal es separar el contenido y la presentación, incluyendo la disposición, colores y fuentes

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.(Eguíluz Pérez, s. f.)

1.4.3 Lenguaje de Implementación de Funcionalidades

JavaScript es fundamental para el desarrollo web moderno, ya que permite agregar interactividad y dinamismo a las páginas web. Se utiliza comúnmente para crear interfaces de usuario, manejar eventos, y actualizar el contenido de la página sin tener que recargar el navegador

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios(Eguíluz Pérez, 2009)

1.4.4 Framework de desarrollo

Vue 3 es la última versión del popular framework de JavaScript Vue.js. Fue lanzado en septiembre de 2020 y se caracteriza por ser más rápido, más pequeño y más fácil de mantener que su predecesor, Vue 2.0. Además, Vue 3 introduce la API de composición, que proporciona una forma más organizada de estructurar la lógica de los componentes de la vista y promueve la reusabilidad y la reactividad. Vue 3 es ampliamente utilizado en el desarrollo web moderno y cuenta con una amplia comunidad y bibliotecas para facilitar el desarrollo de aplicaciones web

Tiene un paquete oficial para poder tener un sistema de rutas. El sistema de rutas llamado vue router, permite tener una página web SPA, es decir, todas las páginas están ya cargadas cuando el usuario entra en la página web de tal forma que cambiar de página es instantáneo y control sobre el componente, que pasa antes de que se cargue, que pasa justo al cargarse o al destruirse (Germán & Martinez, 2021).

1.5 Tecnologías utilizadas para la construcción de Modelos y Controladores (Backend)

1.5.1 Entorno de Ejecución Utilizado

Node.js Node.js es un entorno asíncrono basado en eventos, sin bloqueo, que utiliza el motor de JavaScript V8 de Google. Se utiliza para desarrollar aplicaciones que hacen un uso intensivo de la capacidad de ejecutar JavaScript tanto en el cliente como en el lado del servidor y, por lo tanto, se benefician de la reutilización del código y la falta de cambio de contexto. Es de código abierto y multiplataforma. Las aplicaciones Node.js están escritas en JavaScript puro y se pueden ejecutar dentro del entorno Node.js en Windows, Linux (Stack Overflow, s. f.).

Es fundamental para el desarrollo web moderno, ya que permite agregar interactividad y dinamismo a las páginas web de manera eficiente. Se utiliza comúnmente en aplicaciones web de alto tráfico, servidores de contenido estático y aplicaciones de una sola página (SPA). Además, proporciona una implementación de servidor HTTP integrada, lo que lo hace independiente de componentes de software de terceros como XAMPP

1.5.2 Framework de desarrollo

Express.js es un marco de aplicación web para Node.js que simplifica el proceso de creación de aplicaciones web y APIs. Express.js proporciona una interfaz sencilla para interactuar con el servidor HTTP de Node.js y facilita la organización de la funcionalidad de la aplicación con middleware y enrutamiento. Express.js es conocido por su flexibilidad y facilidad de uso, lo que lo convierte en uno de los marcos de aplicación web más populares para Node.js. Es ampliamente utilizado en el desarrollo web moderno y cuenta con una amplia comunidad y bibliotecas para facilitar el desarrollo de aplicaciones web

Permite desarrollar aplicaciones web en el servidor usando el mismo lenguaje que en el cliente: JavaScript. Basado en Node.js y alternativa a Django o Ruby on Rails. Este framework es libre y gratuito, muy popular. Desarrollado por TJ Holowaychuk en 2010. Vendido a StrongLoop, que actualmente pertenece a IBM (Universidad Rey Juan Carlos, 2022)

1.6 Gestión de Datos

1.6.1 Gestor de Bases de Datos

MariaDB es un manejador popular de MySQL creado por los desarrolladores originales de MySQL. Se desarrolló a partir de las preocupaciones relacionadas con la adquisición de MySQL por parte de Oracle. Ofrece soporte tanto para pequeñas tareas de procesamiento de datos y necesidades de la empresa. Su objetivo es ser una gota en el reemplazo para MySQL que sólo requiere una desinstalación sencilla de MySQL y una instalación de MariaDB. MariaDB ofrece las mismas características de MySQL (UNEWEB, s. f.)

Además, MariaDB ofrece una mayor eficiencia y un mejor rendimiento de la base de datos en comparación con MySQL. Sin embargo, también presenta desventajas, como problemas de compatibilidad con algunas aplicaciones y un soporte limitado debido a su naturaleza de código abierto. En general, MariaDB es una excelente opción para muchas aplicaciones web y sitios web, ofreciendo una versión mejorada de MySQL con numerosas ventajas significativas

1.6.2 Administrador de Bases de Datos

Navicat es un Administrador de Base de datos rápido, fiable y asequible. Especialmente diseñada para simplificar la gestión de bases de datos y reducir los costes de administración. Con una interfaz gráfica del usuario intuitiva, Navicat le proporciona una manera más fácil de gestionar, diseñar y manipular datos en MySQL, MariaDB, SQL Server, SQLite, Oracle y PostgreSQL.

Permite analizar y migrar datos entre bases de datos / esquemas, por lo que podrá asegurarse de que cada base de datos almacene la misma información. También, podrá obtener informes detallados y síntesis de la migración para asegurar una exitosa finalización y guardar de esta manera las configuraciones como un perfil para configurar una programación (DanySoft, 2014).

1.7 Otras Herramientas de Desarrollo

Visual Studio Code es un editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. La elección principal para desarrolladores web y JavaScript, con extensiones para admitir casi cualquier lenguaje de programación.(Microsoft, s. f.)

Tiene características muy particulares y ventajas como soporte para múltiples lenguajes de programación, Intelli-Sense, compatibilidad multiplataforma, repositorio y soporte web. Además, Visual Studio Code incluye un modelo de extensibilidad público que permite a los desarrolladores construir y usar extensiones, y personalizar su experiencia de edición-compilación-depuración

1.8 Algunas definiciones

1.8.1 Patrón Modelo-Vista-Controlador

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. A partir del uso de framework basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores(Romero & Díaz González, 2012)

1.9 Conclusiones Parciales del Capítulo

Fueron los puntos abarcados en este capítulo, la descripción de las características del modelo de negocio sobre el cual se conduce la investigación en cuestión. Se detallo la lógica que posee y la necesidad de llevar a cabo el diseño de una aplicación web para darle solución a los problemas descritos. Se abordaron los conceptos generales asociados a la investigación y la definición de las tecnologías a utilizar para el desarrollo

de la propuesta solución, así como la descripción y comparación de las metodologías y el por qué de seleccionar RUP para la conducción de este proceso. Se justifican la utilización de JavaScript como lenguaje predominante en el desarrollo de las funcionalidades, tanto del lado del cliente como del lado del servidor, el apoyo en frameworks minimalistas como Express y Vue, la selección de HTML y CSS para estructurar y estilar la aplicación a construir, la utilización de MariaDB como Gestor de bases de datos y Node.js como entorno de ejecución y a la vez, servidor web.

Capítulo 2: Análisis, diseño y desarrollo de la solución propuesta

En el presente capítulo, se describen los procesos que se llevarán a cabo en el desarrollo del software. Se detallan los elementos necesarios para la descripción de la solución propuesta a través de Requisitos Funcionales y se aplica la Metodología RUP (Rational Unified Process), con el objetivo de garantizar el diseño de un programa lo más ajustado posible, en un proceso de desarrollo de software estructurado y detallado que se adapta a las necesidades de cada organización y basado en un conjunto de fases y entregables específicos que se utilizan como referencia para el desarrollo del proyecto.

2.1 Descripción de la solución

El sistema a desarrollar, luego de abordar todas las cuestiones y referentes teóricos-metodológicos, será concebido para la persistencia y gestión en línea de los datos y procesos que se realizar en la matrícula de un aspirante al Programa de Formación Doctoral de la Universidad de Matanzas. Se podrá tener constancia de todos los talleres y sesiones con participación de los aspirantes, estadísticas por módulos, gestión del personal del programa, así como la constancia física y lógica de cada documento generado por el sistema o acreditado por los miembros implicados en todos los pasos del flujo de matrícula del aspirante en cuestión. También las notificaciones pertinentes sobre cada acción crítica realizada en el sistema será una de las virtudes de este sistema informático propuesto como solución a la problemática planteada. Cada una de las acciones descritas requiere de la autenticación del usuario en el sistema.

2.1.1 Herramientas y control de versiones

Descripción	Herramienta
Metodología de desarrollo	Metodología RUP (Rational Unified Process),
Lenguaje Funcional Base	JavaScript
Servidor Web y Entorno de Ejecución	Node.js 8.1
Gestor de base de datos	MariaDB 11.3
Framework-Frontend	Vue 3

Framework-Backend	Express 4.18
-------------------	--------------

Tabla 1: Herramientas Fuente: Elaboración del autor.

2.1.2 Roles de Usuario

Administrador del sistema	<ul style="list-style-type: none"> • Gestión de Usuarios • Gestión de Nomencladores • Ver trazas del Sistema
Jefe de Línea	<ul style="list-style-type: none"> • Gestión de Línea de Investigación • Gestión de Talleres
Coordinador del Programa	<ul style="list-style-type: none"> • Decisión sobre documentos de Matricula • Ratificación de criterios en Comité Doctoral
Comité Doctoral	<ul style="list-style-type: none"> • Decisión inicial sobre Tema de Investigación
Comisión de Grados	<ul style="list-style-type: none"> • Decisión final sobre Tema de Investigación • Decisión sobre propuesta de Tutores
Secretario de Comisión de Grados	<ul style="list-style-type: none"> • Ratificación de decisiones de Comisión Doctoral • Ratificación de decisión Sobre Documentos de Matricula •
Aspirante	<ul style="list-style-type: none"> • Gestión de documentos propios • Participación en el proceso de matricula • Consultar estado de documentos
Doctor	<ul style="list-style-type: none"> • Participación en evaluaciones a aspirantes

Tabla 2: Roles. Fuente: Elaboración del autor.

2.2 Etapas de Desarrollo

La metodología RUP (Rational Unified Process) es un proceso de desarrollo de software que se basa en un conjunto de metodologías adaptables al contexto y necesidades de cada organización. El Proceso unificado de Rational (RUP) define cuatro fases de un proyecto: principio, elaboración, construcción y transición. (IBM, 2021) .Cada fase tiene

actividades específicas y genera entregables que sirven como referencia para el desarrollo del proyecto. Además, se apoya en herramientas que automatizan grandes partes del proceso, como la creación y mantenimiento de artefactos y modelos, el modelado visual, la programación y el testing. RUP contiene muchas de las buenas prácticas en el desarrollo de software moderno de una forma que es adaptable a un amplio rango de proyectos y organizaciones.

2.2.1 Fase de Inicio

La Fase de Inicio tiene como objetivo definir y acordar el alcance del proyecto con los patrocinadores, establecer una arquitectura inicial del sistema

. Durante esta fase, se llevan a cabo las siguientes actividades:

1. **Identificación del alcance del proyecto:** Se establecen los límites del proyecto, incluyendo las funcionalidades que se abordarán (Casos de uso).
2. **Definición de la arquitectura inicial del sistema:** Se establece la estructura básica del sistema, incluyendo componentes y relaciones entre ellos.
3. **Identificación de los patrocinadores del proyecto:** Se identifican los patrocinadores del proyecto y se establece una comunicación efectiva con ellos.
4. **Establecimiento del equipo de trabajo:** Se establece el equipo de trabajo y se definen los roles y responsabilidades de cada miembro.

2.2.1.1 Alcance del Proyecto (Caso de Uso)

Los diagramas de casos de uso son útiles para identificar los requisitos de uso del sistema y para comprender cómo los usuarios interactúan con el sistema. Además, estos diagramas pueden utilizarse para identificar las funciones y la forma en que los usuarios interactúan con el sistema, lo que permite a los desarrolladores diseñar un sistema que satisfaga las necesidades de los usuarios

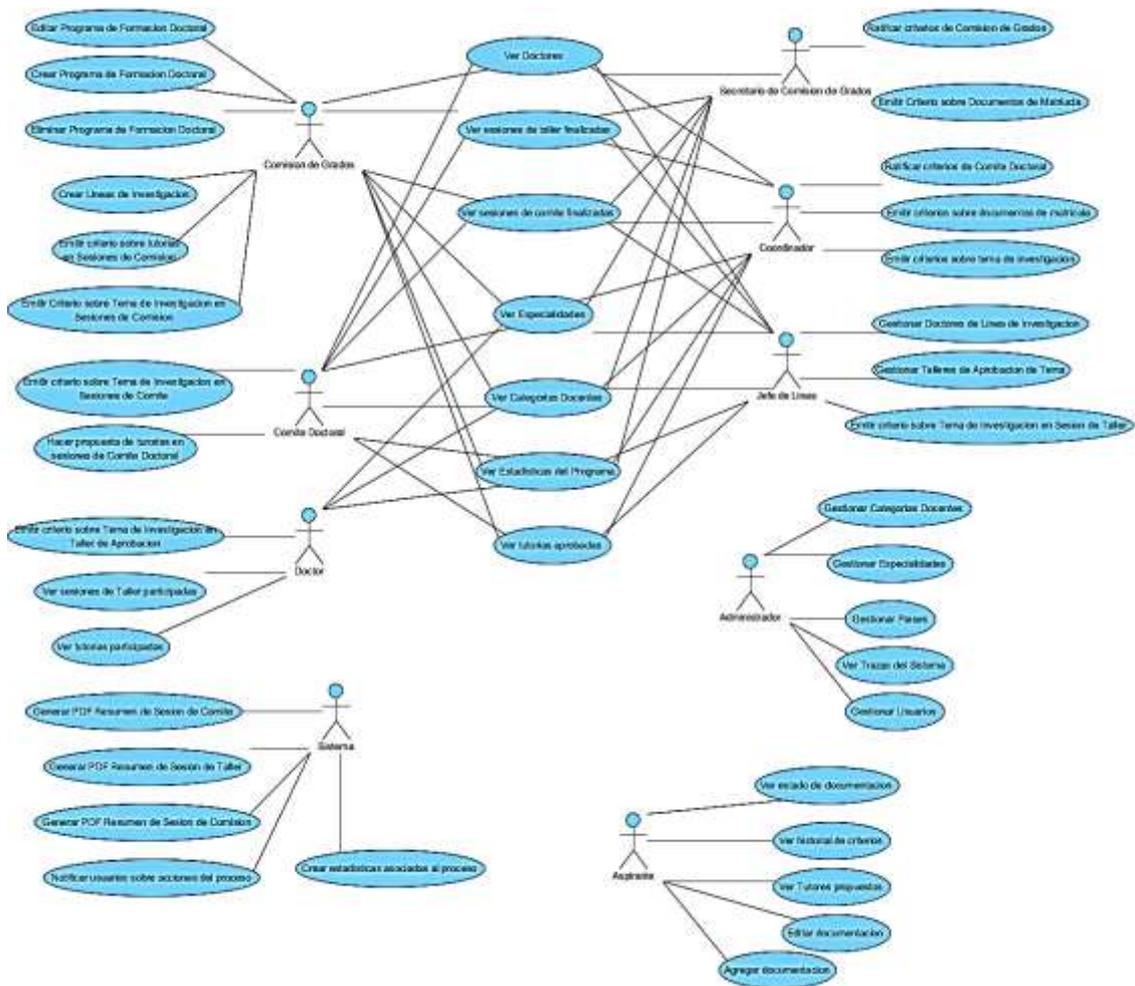


Ilustración 1: Diagrama de Casos de Uso. Fuente: Autor

2.2.1.2 Definición de la Arquitectura Inicial del Sistema

Para el desarrollo de este sistema se ha escogido el patrón arquitectónico MVC (Modelo-Vista-Controlador). Este patrón permite separar claramente los componentes de la aplicación, lo que facilita su mantenimiento y evolución. El modelo se encarga de representar la información y la lógica de negocio del sistema de gestión, mientras que la vista se encarga de mostrar la información al usuario, y el controlador actúa como intermediario entre el modelo y la vista, procesando las solicitudes del usuario y actualizando la información en el modelo. Además, este patrón permite una mayor flexibilidad en el diseño de la aplicación, lo que facilita la integración de nuevos componentes y funcionalidades en el futuro.

Para representar la arquitectura en un primer momento, se hace uso de los **Diagramas de Caja Negra**. Este se refiere a un sistema cuyo interior no puede ser descubierto, cuyos elementos internos son desconocidos y que sólo puede conocerse “por fuera”, a través de manipulaciones externas o de observación externa. Es utilizada para representar a los sistemas cuando no sabemos qué elementos o cosas componen al sistema o proceso, pero sabemos que a determinadas corresponden determinadas

salidas y con ello poder inducir, que a determinados estímulos, las variables funcionarán en cierto sentido.

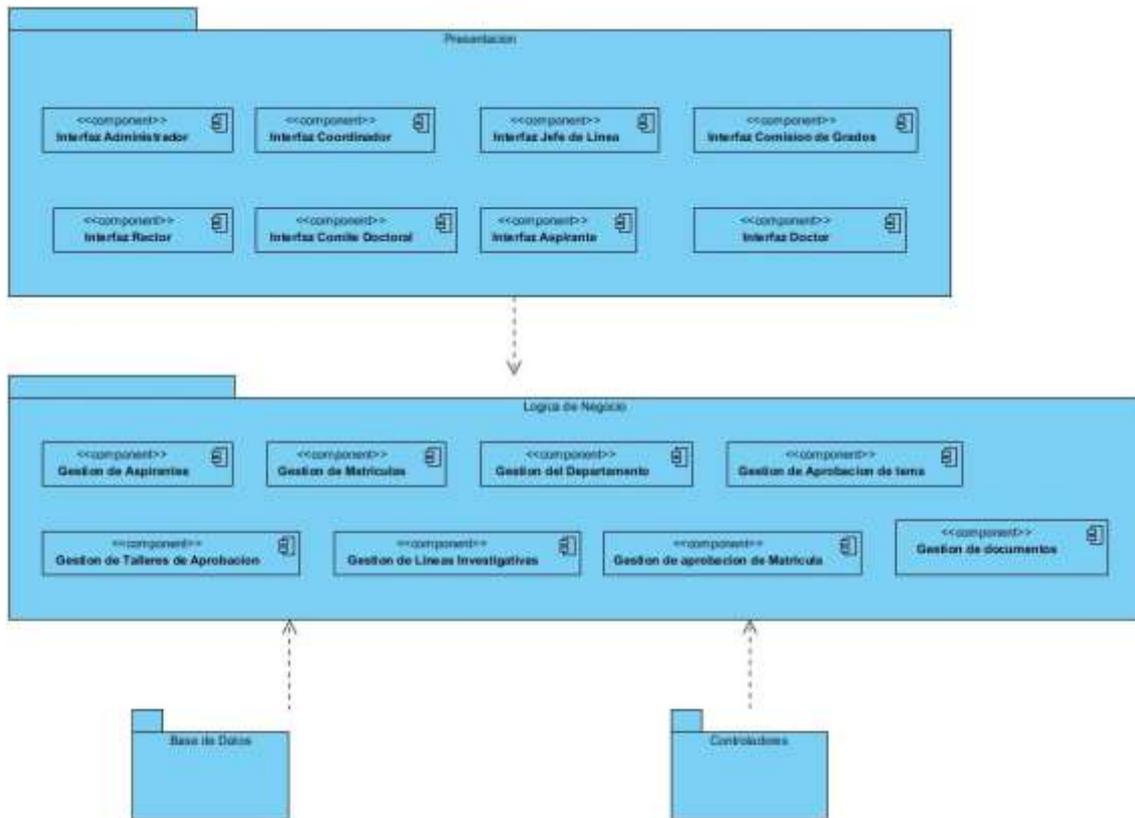


Ilustración 2 Diagrama de Componentes Caja Negra. Fuente: Visual Paradigm

2.2.1.3 Establecimiento del Equipo de Trabajo

Miembro	Cargo
Dr. Walfredo González Hernández	Jefe de proyecto
Alfredo David Morales Alfonso	Tester, Desarrollador

Tabla 3: Equipo de Trabajo. Fuente: Elaboración del autor.

2.2.1.4 Patrocinadores del Proyecto

Patrocinador	Universidad de Matanzas

Tabla 4: Patrocinadores. Fuente: Elaboración del autor.

2.2.2 Fase de Elaboración

La Fase de Elaboración tiene como objetivo definir y acordar el alcance del proyecto con los patrocinadores o involucrados del proyecto, identificar los riesgos asociados al

proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores

Algunos aspectos clave de esta fase incluyen:

1. **Análisis de requisitos detallados del cliente y arquitectura del sistema:** Entender los requisitos del cliente y cómo se relacionan con la arquitectura del sistema.
2. **Modelado de negocio:** Familiarizarse con el funcionamiento de la empresa, sus procesos y cómo se relacionan con los requisitos del sistema.
3. **Diseño de la arquitectura del sistema:** Diseñar una arquitectura detallada que aborde los requisitos y la arquitectura inicial del sistema.
4. **Lista de casos de uso y descripción:** Identificar y describir los casos de uso principales que el sistema debe soportar.

2.2.2.1 Análisis de requisitos detallados del cliente y arquitectura del sistema

El diagrama de colaboración es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información mostrando cómo interactúan los objetos entre sí, es decir, con qué otros objetos tiene vínculos o intercambia mensajes un determinado objeto.

Un diagrama de colaboración muestra la misma información que un diagrama de secuencia, pero de forma diferente. En los diagramas de colaboración no existe una secuencia temporal en el eje vertical; es decir, la colocación de los mensajes en el diagrama no indica cuál es el orden en el que se suceden. Además, la colocación de los objetos es más flexible y permite mostrar de forma más clara cuáles son las colaboraciones entre ellos. En estos diagramas la comunicación entre objetos se denomina vínculo o enlace (*link*) y estará particularizada mediante los mensajes que intercambian (Cillero Manuel, s. f.)

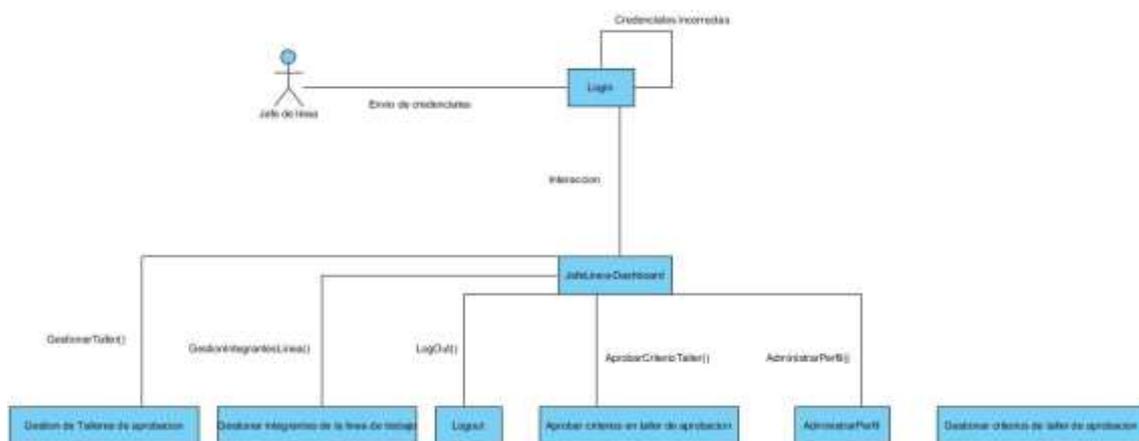


Ilustración 3. Diagrama de Colaboración para usuario Jefe de Línea. Fuente: Visual Paradigm

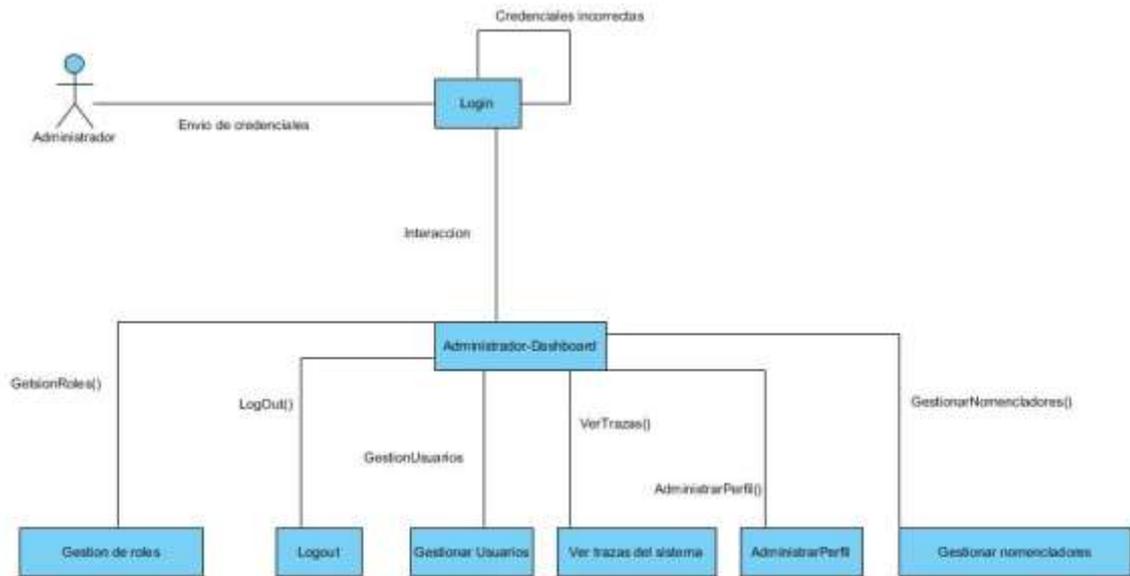


Ilustración 4 Diagrama de Colaboración para usuario Administrador. Fuente: Visual Paradigm

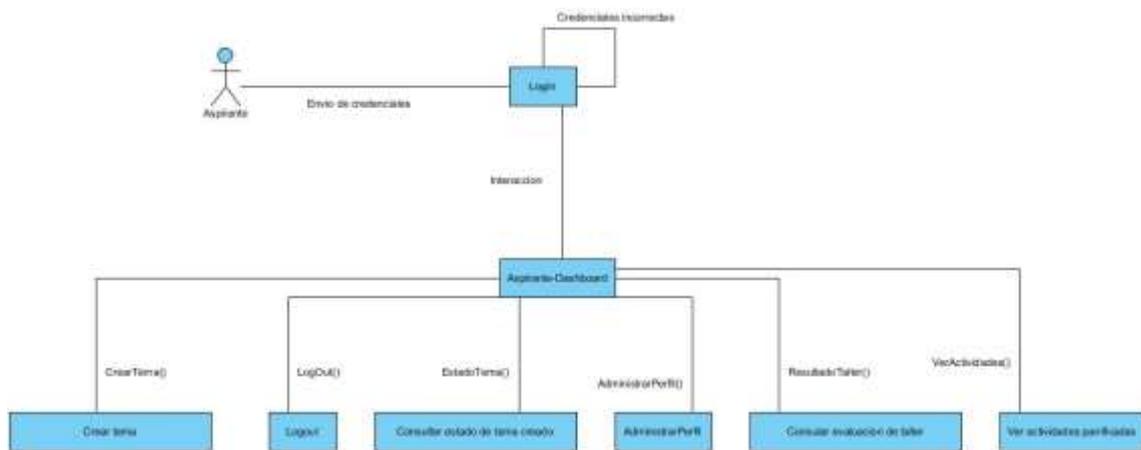


Ilustración 5. Diagrama de Colaboración para usuario Aspirante. Fuente: Visual Paradigm

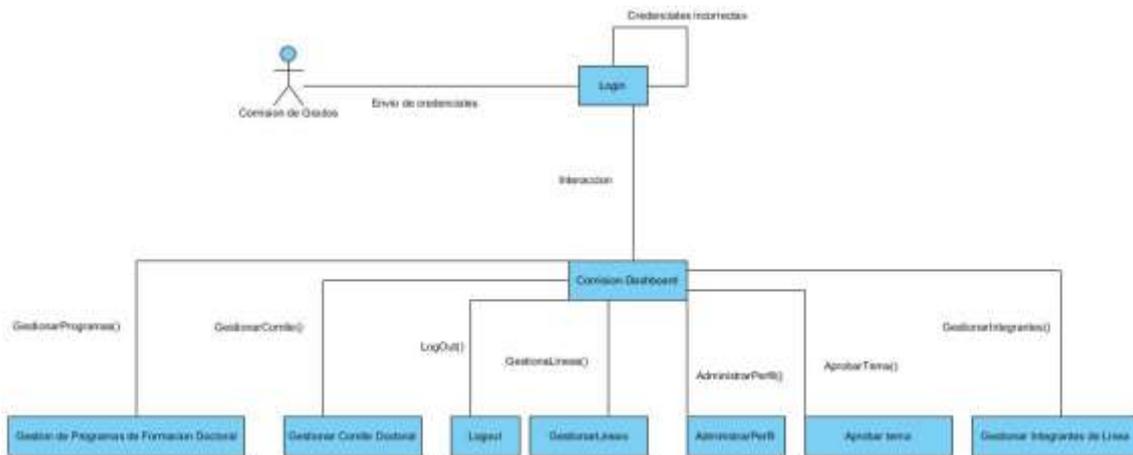


Ilustración 6. Diagrama de Colaboración para usuario Comisión de Grados. Fuente: Visual Paradigm

2.2.2.2 Modelo de Negocio

Luego de tener una descripción global del funcionamiento del negocio, se optó por la creación de un Diagrama BPMN (Business Process Model and Notation), para una comprensión detallada y grafica de cómo funciona el flujo de datos en el Proceso de Matriculas del Programa de Formación Doctoral de la Universidad de Matanzas

En un nivel elevado, BPMN está dirigido a participantes y otros interesados en un proceso de negocio con el fin de obtener conocimientos mediante una representación visual de los pasos fácil de entender. En un nivel más específico, se dirige a las personas que implementarán el proceso, brindando suficientes detalles para permitir una implementación precisa. Ofrece un lenguaje estándar y común para todos los interesados, sean técnicos o no

Su propósito es modelar formas de mejorar la eficiencia, representar nuevas circunstancias u obtener ventaja sobre la competencia. Este método también ha experimentado un empuje hacia la estandarización en los últimos años, y ahora su nombre es un poco diferente(Lucidchart, s. f.)

establecidos. Además, sirve como base para la implementación y las pruebas del sistema.

Los Diagramas de **Caja Blanca** son los adecuados para detallar la arquitectura de un sistema. Se utilizan para representar la arquitectura lógica de un sistema, mostrando las interfaces por las que los componentes se relacionan. Permite visualizar los detalles internos de los componentes y sus interacciones, lo que resulta útil para comprender el funcionamiento interno del sistema

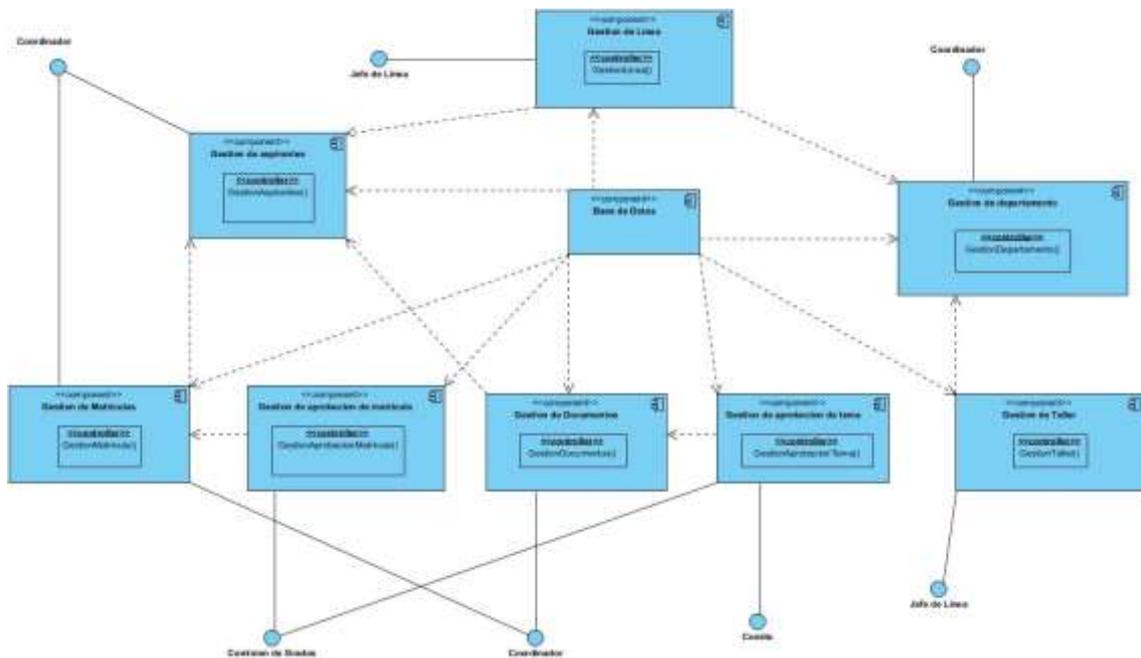


Ilustración 8. Diagrama de Componentes Caja Blanca. Fuente: Visual Paradigm

2.2.2.4 Lista de Casos de Uso y Descripción

Al explicar los casos de uso de un sistema, se busca detallar cómo interactúan los usuarios o actores con el sistema para lograr objetivos específicos. Los casos de uso son una herramienta fundamental en la ingeniería de requisitos y el desarrollo de software, ya que describen las interacciones entre el sistema y sus usuarios finales. Estos casos proporcionan una comprensión clara de las funcionalidades que el sistema debe ofrecer, centrándose en las necesidades y objetivos de los usuarios. Además, permiten visualizar de manera efectiva cómo se espera que el sistema responda a las acciones de los usuarios, lo que resulta crucial para el diseño y desarrollo exitoso de un sistema de software

- *Agregar Doctor*

Caso de Uso	Gestionar Usuarios
Actor	Administrador
Propósito	Agregar un doctor

Resumen	El caso de uso se inicia cuando el Administrador desea agregar un doctor al proceso de formación doctoral. Muestra una vista donde Administrador llena el formulario con los datos necesarios: nombre, categoría docente ,especialidad y carnet de identidad
Responsabilidades	R1-Permitir agregar un doctor
Precondiciones	Estar creado el rol Administrador
Postcondiciones	-Se agrega el doctor al sistema

Acciones del Actor	Respuesta del sistema
1-El Administrador selecciona "Doctores" en el Dashboard	2-El sistema solicita la vista de Doctor
	3-El sistema muestra vista de Doctor
4-El Administrador selecciona "Agregar Doctor"	
	5-El sistema muestra formulario para crear doctor.
6-El Administrador llena el formulario y confirma la acción	
	7-El sistema envía mensaje de éxito a la vista
	8-El sistema actualiza la vista con la nueva lista de doctores

Flujo alternativo

Si en el paso 6 la información es inválida, se envía un mensaje de error y la solicitud no es almacenada hasta que se corrija.

- *Gestionar Usuarios*

Caso de Uso	Gestionar Usuarios
--------------------	---------------------------

Actor	Administrador
Propósito	Eliminar un doctor
Resumen	El caso de uso se inicia cuando Administrador desea eliminar un doctor. Muestra una vista donde el Administrador puede ver a todos los doctores que se encuentran registrados en el sistema. Para eliminar a un doctor solo necesita presionar el botón "Eliminar" asociado al doctor
Responsabilidades	R1-Permitir eliminar un doctor
Precondiciones	Que se haya creado el rol Administrador Que se haya creado el Doctor
Postcondiciones	-Se elimina el doctor del sistema

Acciones del Actor	Respuesta del sistema
1-El Administrador selecciona "Doctor" en el Dashboard	2-El sistema solicita la vista de Doctor
	3-El sistema muestra vista de Doctor
4-El Administrador selecciona el doctor a eliminar presionando el botón "Eliminar".	
	5-El sistema muestra ventana de confirmación de la acción
6-El Administrador confirma la acción.	
	7-El sistema elimina el doctor requerido del modelo
	8-El sistema envía mensaje de éxito a la vista

	9-El sistema actualiza la vista con la nueva lista de doctores
--	--

Flujo alternativo

Si en el paso 6 no se confirma la acción, el sistema no elimina el doctor del sistema.

- *Gestionar Categoría Docente*

Caso de Uso	Gestionar Categorías Docentes
Actor	Administrador
Propósito	Editar una Categoría
Resumen	El caso de uso se inicia cuando el Administrador desea modificar los datos de alguna Categoría. Muestra una vista donde el jefe de departamento llena el formulario con los datos modificables.
Responsabilidades	R1-Permitir editar una categoría
Precondiciones	Que se haya creado el rol Administrador
Postcondiciones	-Se edita la categoría

Acciones del Actor	Respuesta del sistema
1-El Administrador selecciona "Categorías" en el Dashboard	2-El sistema solicita la vista de Categoría
	3-El sistema muestra vista de Categoría
4-El Administrador selecciona "Editar" asociado a la categoría que será editada	
	5-El sistema muestra formulario para editar la categoría.
6-El Administrador llena el formulario y confirma la acción	

	7-El sistema envía mensaje de éxito a la vista
	8-El sistema actualiza la vista con la nueva lista de doctores

Flujo alternativo

Si en el paso 6 la información es inválida, se envía un mensaje de error y la solicitud no es almacenada hasta que se corrija

- *Gestionar Línea de Investigación*

Caso de Uso	Gestionar Línea de Trabajo
Actor	Comisión de Grados
Propósito	Crear Línea de Trabajo
Resumen	El caso de uso se inicia cuando la Comisión de Grados desea crear una Línea de Trabajo . Muestra una vista donde llena el formulario con los datos necesarios para crear la Línea de Trabajo
Responsabilidades	R1-Permitir creación de Línea de Trabajo
Precondiciones	Que se haya creado el rol de Comisión de Grados
Postcondiciones	-Se crea la Línea de Trabajo

Acciones del Actor	Respuesta del sistema
1-La Comisión de Grados selecciona “ Línea de Trabajo ” en el Dashboard	2-El sistema solicita la vista de “ Línea de Trabajo ”
	3-El sistema muestra vista de Línea de Trabajo
4- La Comisión de Grados selecciona “ Crear nueva Línea de Trabajo ”	

	5-El sistema muestra formulario para crear la Línea de Trabajo .
6- La Comisión de Grados llena el formulario y confirma la acción	
	7-El sistema envía mensaje de éxito a la vista
	8-El sistema notifica a todos los miembros seleccionados al crear la Línea de Trabajo
	9-El sistema actualiza la vista con la nueva lista de Líneas de Trabajo

Flujo alternativo

Si en el paso 6 la información es invalida, se envía un mensaje de error y la solicitud no es almacenada hasta que se corrija

- *Gestionar Línea de Trabajo*

Caso de Uso	Gestionar Línea de Trabajo
Actor	Jefe de Línea
Propósito	Editar una Línea de Trabajo
Resumen	El caso de uso se inicia cuando el Jefe de Línea desea modificar los datos de alguna Línea de Trabajo. Muestra una vista donde Jefe de Línea llena selecciona los aspectos que desea editar de la Línea de Trabajo.
Responsabilidades	R1-Permitir editar una Línea de Trabajo
Precondiciones	Que se haya creado el rol de Jefe de Línea Que se haya creado la Línea de trabajo
Postcondiciones	-Se edita la Línea de Trabajo

Acciones del Actor	Respuesta del sistema
--------------------	-----------------------

1- El Jefe de Línea selecciona “Línea de Trabajo” en el Dashboard	2-El sistema solicita la vista de Línea de Trabajo
	3-El sistema muestra vista de Línea de Trabajo
4-El Jefe de Línea selecciona “Editar” asociado a la Línea de Trabajo que será editada.	
	5-El sistema muestra formulario con las opciones a editar de la Línea de Trabajo.
6-El Jefe de Línea selecciona los aspectos a editar y confirma la acción	
	7-El sistema envía mensaje de éxito a la vista
	8-El sistema actualiza la Línea de Trabajo con los nuevos aspectos modificados

- *Agregar Documentación*

• Caso de Uso	Agregar Documentación
Actor	Aspirante
Propósito	Agregar un nuevo documento
Resumen	El caso de uso se inicia cuando el Aspirante desea agregar un nuevo documento. El Aspirante va a una vista donde selecciona el documento y luego lo envía
Responsabilidades	R1-Permitir agregar un nuevo documento
Precondiciones	Que se haya creado el rol Aspirante
Postcondiciones	-Se crea un nuevo documento

Acciones del Actor	Respuesta del sistema
1- El aspirante selecciona “Estado de Documentos” en el Dashboard	2-El sistema solicita la vista de Estado de Documentos

	3-El sistema muestra vista de Estado de Documentos
4-El aspirante selecciona el apartado correspondiente al tipo de documento que subirá	
	5-El sistema muestra vista para agregar documento.
6-El Aspirante selecciona el documento y confirma la acción	
	7-El sistema envía mensaje de éxito a la vista
	8-El sistema guarda el documento

2.2.3 Fase de Implementación

Ahí es cuando se termina la construcción del proyecto, por eso tiene ese nombre. El principal objetivo es la elaboración del producto. Dado que el método se basa en el desarrollo de software, estamos hablando de crear códigos . Además, es en esta etapa que se realizan las primeras pruebas para que se prepare la base inicial para la etapa de transición.(Ortega, s. f.)

Esta fase esta complementada por los siguientes pasos:

1. **Implementación del sistema:** Desarrollar y construir el sistema utilizando la arquitectura y los requisitos definidos en las fases anteriores.
2. **Pruebas funcionales:** Realizar pruebas funcionales para validar que los componentes del sistema funcionen correctamente y cumplan con los requisitos.
3. **Pruebas de aceptación:** Realizar pruebas de aceptación para validar si un sistema cumple con los requisitos del usuario y si está listo para su implementación.
4. **Pruebas de Seguridad:**
5. **Preparación para el despliegue:** Preparar el sistema para su entrega al cliente, incluyendo la realización de demostraciones y la provisión de documentación

2.2.3.1 Implementación del Sistema

Se ha construido un sistema basado en la arquitectura Modelo-Vista-Controlador (MVC), la cual es un patrón de diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. El sistema se divide en tres componentes principales: el Modelo, que maneja los datos y la lógica de negocios; la Vista, que se encarga del diseño y la presentación; y el Controlador, que enruta

comandos a los modelos y vistas, y contiene la lógica que actualiza el modelo y/o vista en respuesta a las entradas de los usuarios de la aplicación.

Parte de la Aplicación	Cantidad
Vistas de Usuario	48
Modelos	28
Controladores	30

Tabla 5. Cantidad de partes de la aplicación. Fuente: Elaboracion del Autor

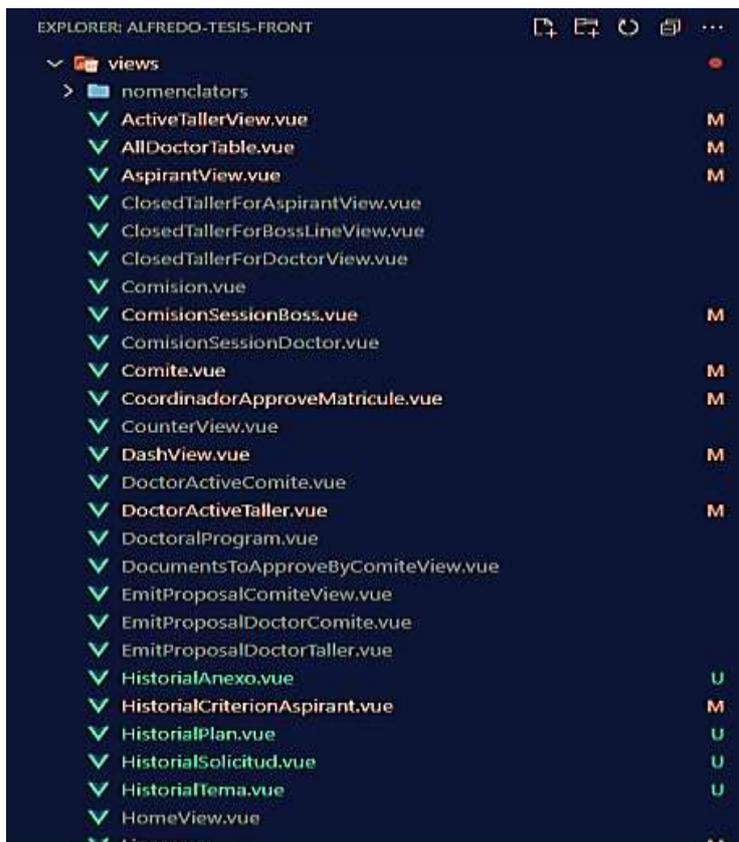


Ilustración 9. Vistas Construidas

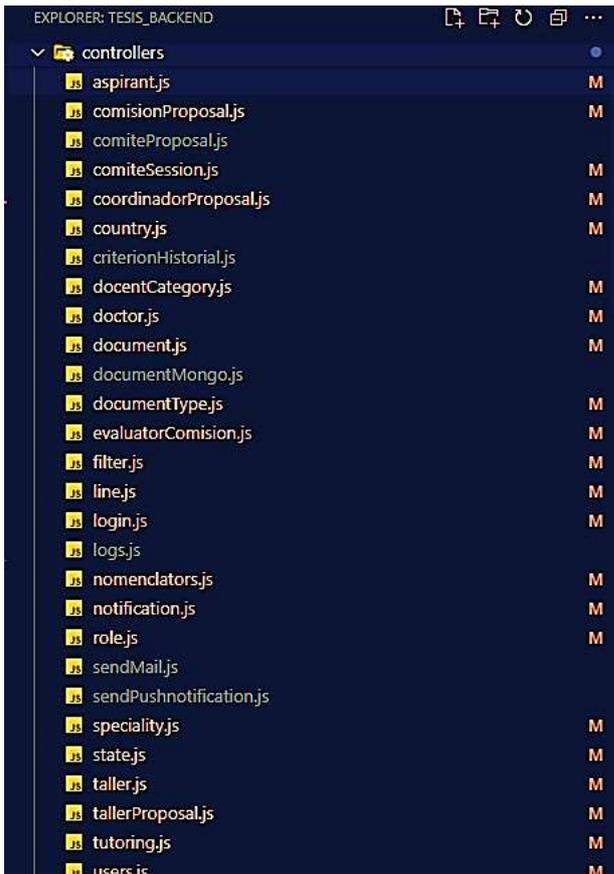


Ilustración 10. Controladores construidos

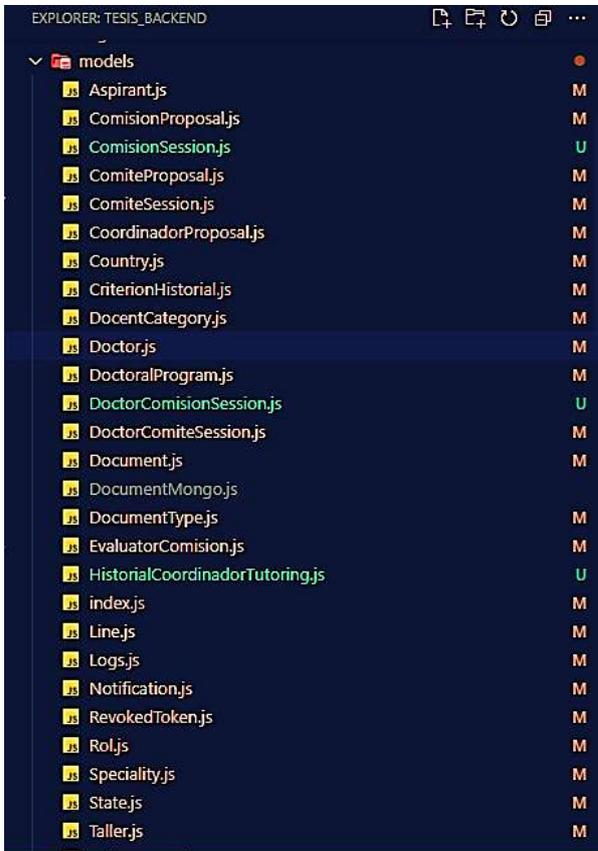


Ilustración 11. Modelos Construidos

2.2.3.2 Pruebas Funcionales

Las pruebas funcionales son un tipo de prueba que se centra en verificar que un sistema o componente de software cumple con las especificaciones y funcionalidades esperadas. Estas pruebas se enfocan en validar el comportamiento del software desde la perspectiva del usuario, y su objetivo es asegurar que el software realiza las funciones para las que fue diseñado de manera correcta

Al depender del desarrollo y configuración humana, los sistemas de software están sujetos a las modificaciones propias de las personas y, por tanto, a los errores cometidos por su realizador. De ahí surge la necesidad de crear pruebas de funcionalidad, que permitan detectar y anticipar escenarios de riesgo (Tamushi, 2022).

Herramienta utilizada para el Desarrollo de Pruebas Funcionales

Como se explicaba anteriormente, la aplicación web posee una arquitectura MVC(Modelo-Vista-Controlador), por lo que la comunicación entre la Vista y los Controladores se realizan a través de una AP I(Application Programming Interface)

La arquitectura de las API suele explicarse en términos de cliente y servidor. La aplicación que envía la solicitud se llama cliente, y la que envía la respuesta se llama servidor.(Amazon, s. f.)

Partiendo de esto, se decide utilizar para el desarrollo de las pruebas funcionales la herramienta **Postman**, que ofrece una variedad de funcionales para hacer test a las funcionalidades del sistema durante todo el proceso de desarrollo

Postman es una aplicación que nos permite realizar pruebas API. Es un cliente HTTP que nos da la posibilidad de testear 'HTTP requests' a través de una interfaz gráfica de usuario, por medio de la cual obtendremos diferentes tipos de respuesta que posteriormente deberán ser validados.(Encora, 2021)

Caso de Prueba	1
Requerimiento	Servidor corriendo en puerto 8000 y conexión a Base de Datos
Acción	Crear Nuevo Doctor
Tipo de Prueba	Funcional.
Hardware	Sistema de cómputo con un Procesador AMD-A12 - Memoria RAM de 8GB.
Software	<ul style="list-style-type: none">• Sistema Operativo: Windows 11• Base de Datos MariaDB –• Navegador: Google Chrome 119.0.60• Servidor Web: Node.js 18

Personal	Desarrollador
Caso de Prueba Funcional	1
Datos de Entrada	Nombre , Carnet de Identidad, Categoría Docente, Especialidad
Datos de Salida (Caso Exito)	<ul style="list-style-type: none"> • Mensaje: “Registro creado con éxito.”
Datos de Salida (Caso Caso Error)	<ul style="list-style-type: none"> • Objeto: Objeto con los campos erróneos y mensajes de validación

Tabla 6. Caso de Prueba 1. Fuente: Elaboracion del Autor

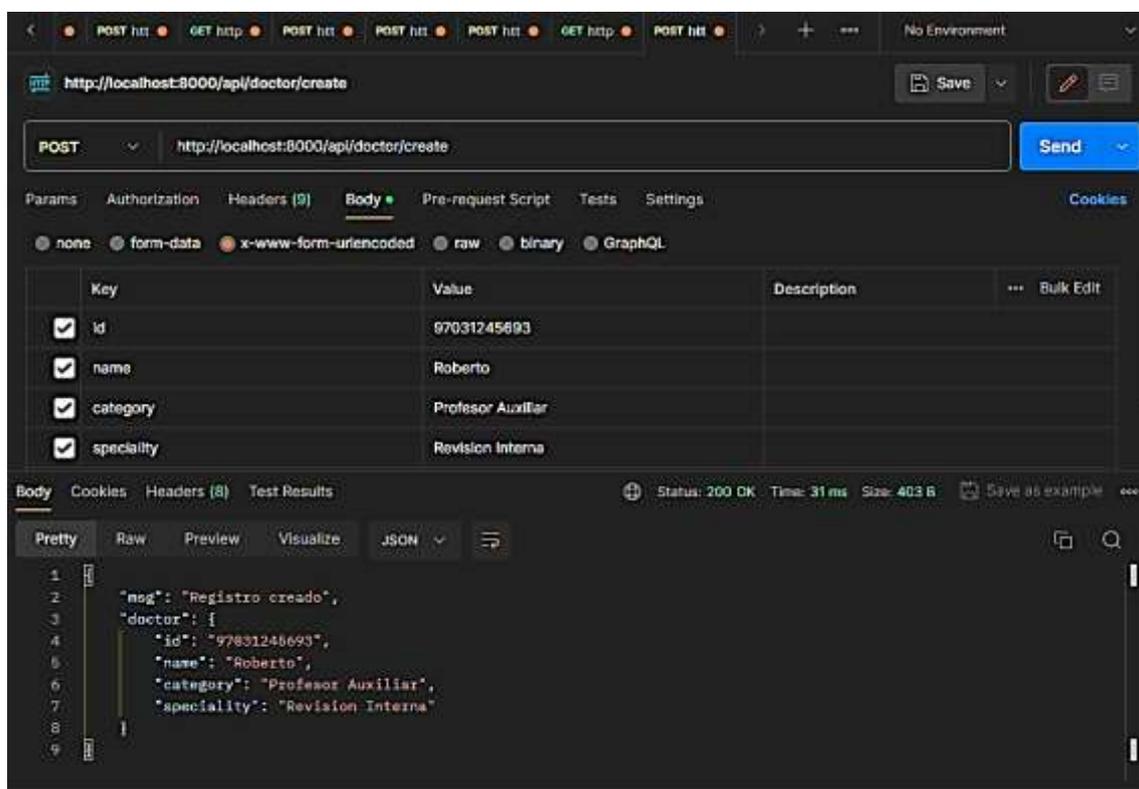


Ilustración 12. Caso de Prueba Funcional 1. Caso de Éxito

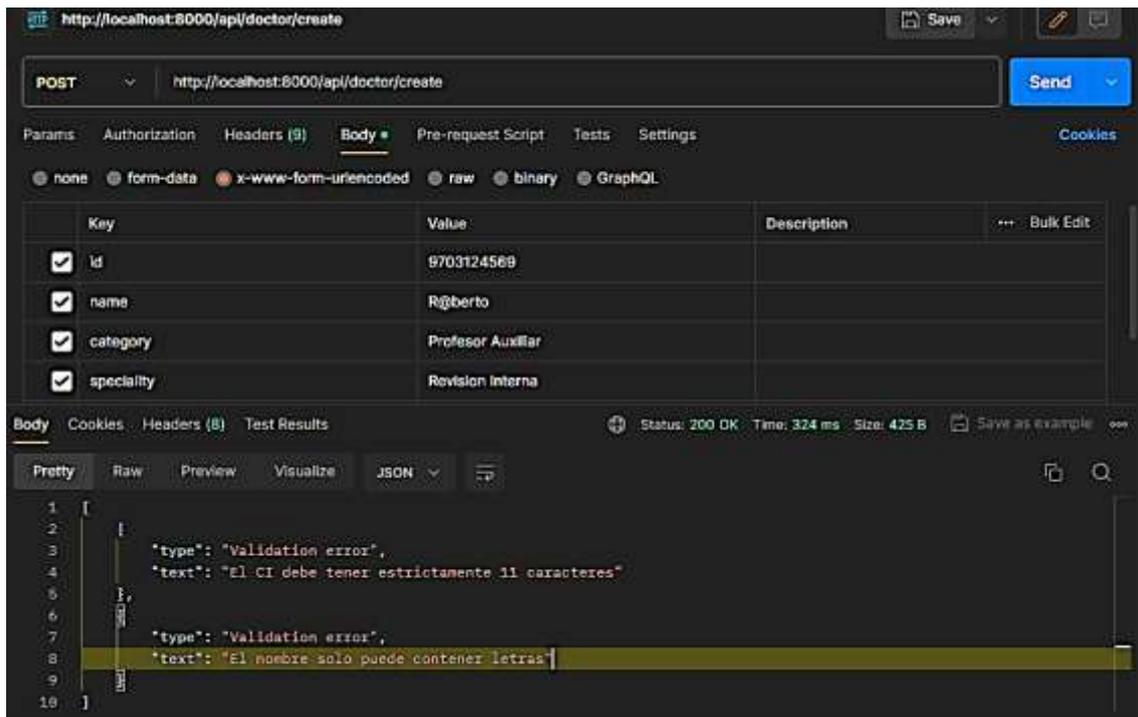


Ilustración 13. Caso de Prueba Funcional 1. Caso de Error

Caso de Prueba	2
Requerimiento	Servidor corriendo en puerto 8000 y conexión a Base de Datos
Acción	Eliminar Doctor
Tipo de Prueba	Funcional.
Hardware	Sistema de cómputo con un Procesador AMD-A12 - Memoria RAM de 8GB.
Software	<ul style="list-style-type: none"> • Sistema Operativo: Windows 11 • Base de Datos MariaDB – • Navegador: Google Chrome 119.0.60 • Servidor Web: Node.js 18
Personal	Desarrollador
Caso de Prueba Funcional	2
Datos de Entrada	Id
Datos de Salida (Caso Éxito)	<ul style="list-style-type: none"> • Mensaje: “Acción realizada con éxito.”
Datos de Salida (Caso Error)	<ul style="list-style-type: none"> • Mensaje: “No se encontró al doctor con ese id”

Tabla 7. Caso de Prueba 1. Fuente: Elaboracion del Autor

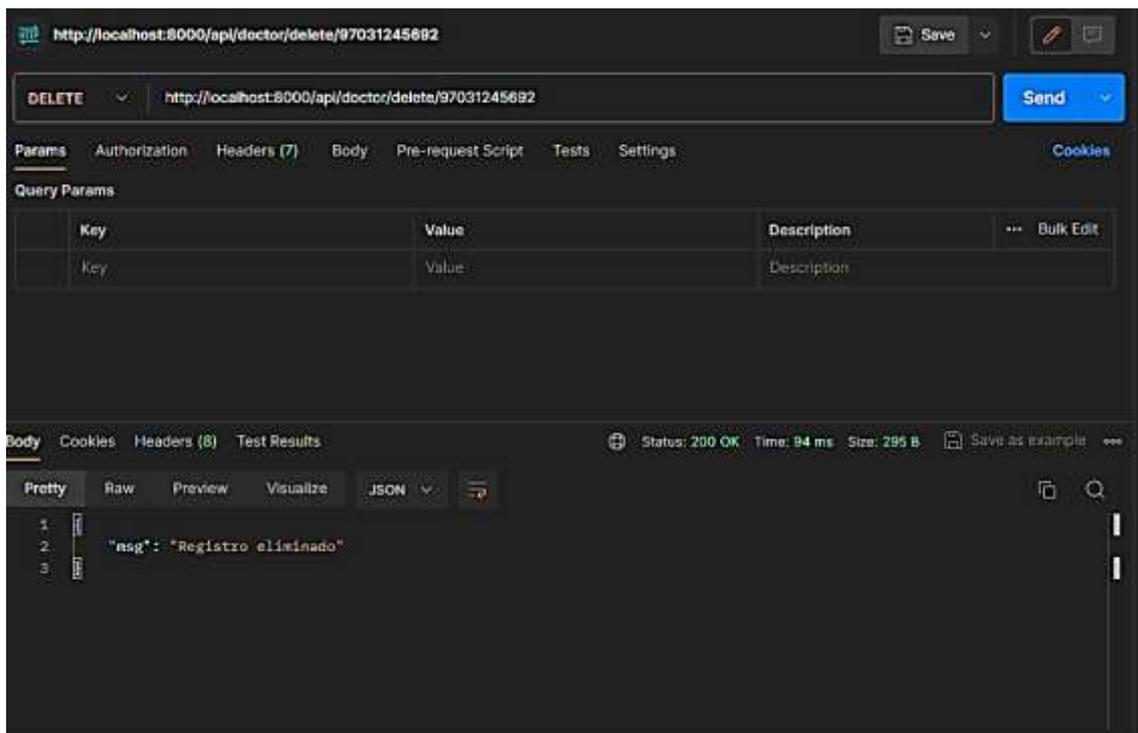


Ilustración 14 Caso de Prueba Funcional 2. Caso de Éxito

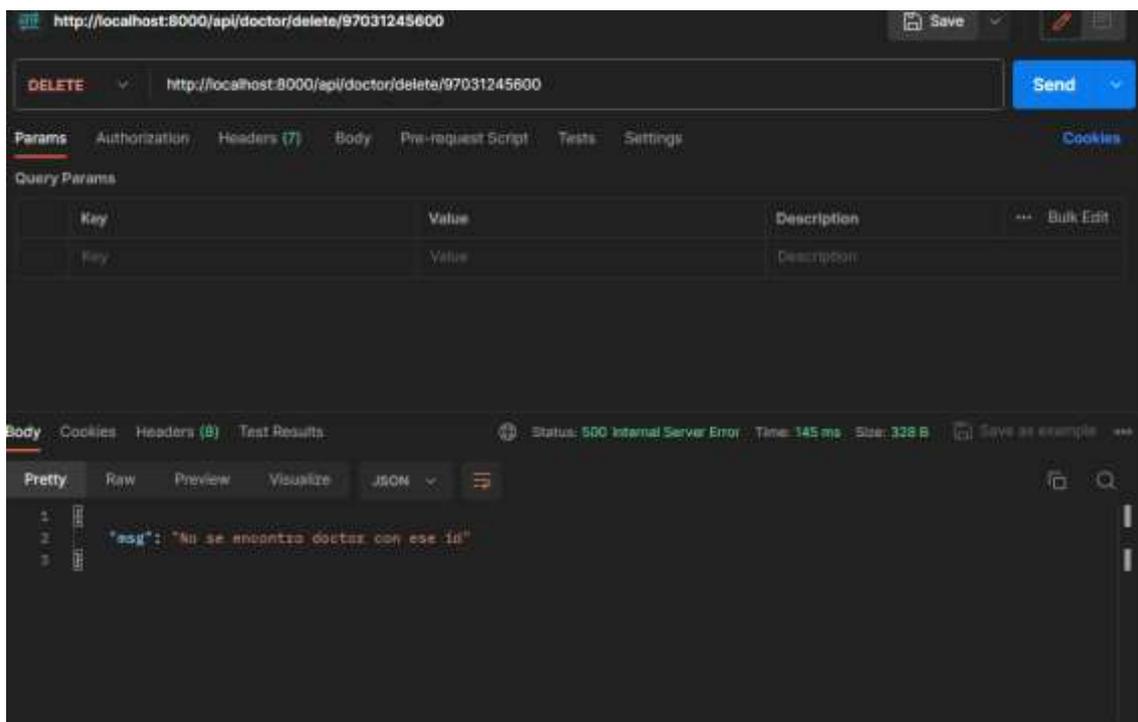


Ilustración 15. Caso de Prueba Funcional 2. Caso de Error

Caso de Prueba	3
Requerimiento	Servidor corriendo en puerto 8000 y conexión a Base de Datos
Acción	Editar Categoría Docente

Tipo de Prueba	Funcional.
Hardware	Sistema de cómputo con un Procesador AMD-A12 - Memoria RAM de 8GB.
Software	<ul style="list-style-type: none"> • Sistema Operativo: Windows 11 • Base de Datos MariaDB – • Navegador: Google Chrome 119.0.60 • Servidor Web: Node.js 18
Personal	Desarrollador
Caso de Prueba Funcional	3
Datos de Entrada	Identificador de Categoría Docente
Datos de Salida (Caso Éxito)	<ul style="list-style-type: none"> • Objeto: Devuelve un objeto con la nueva categoría editada
Datos de Salida (Caso de Error 1)	<ul style="list-style-type: none"> • Mensaje: “No se encontró la categoría con ese id”
Datos de Salida (Caso de Error 2)	<ul style="list-style-type: none"> • Mensaje: ”Campos Inválidos” • Objeto: Objeto con la descripción de las validaciones

Tabla 8. Caso de Prueba 1. Fuente: Elaboracion del Autor

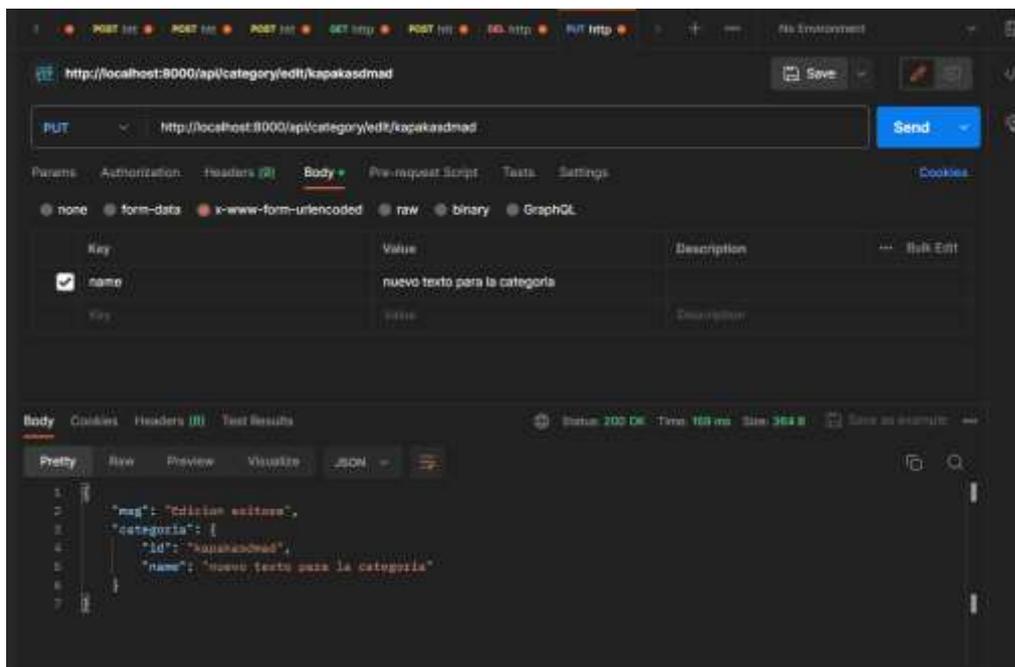


Ilustración 16 Caso de Prueba Funcional 3. Caso de Éxito

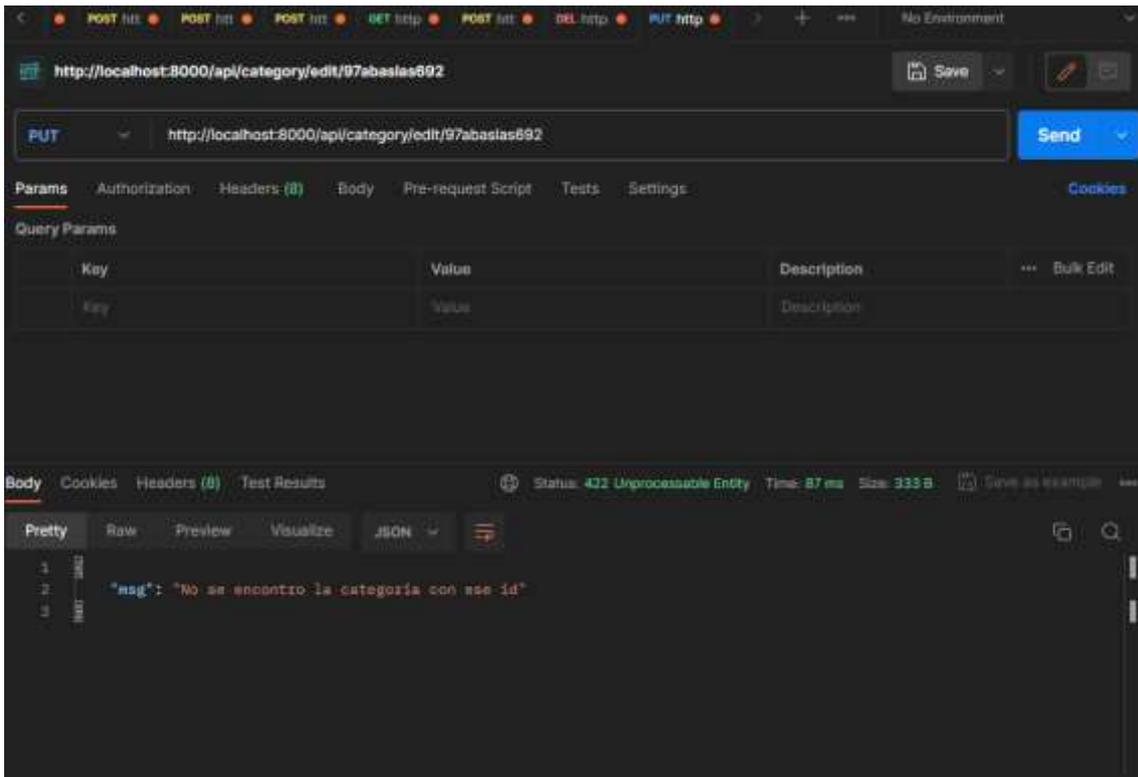


Ilustración 17. Caso de Prueba Funcional 3. Caso de Error 1

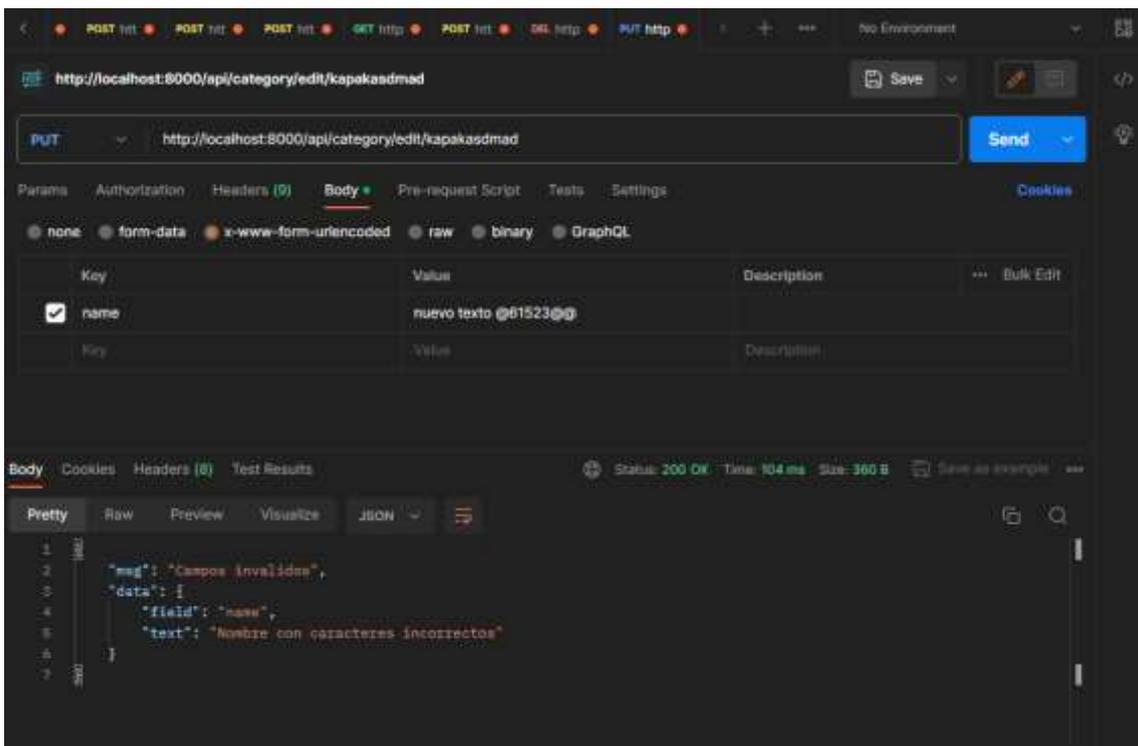


Ilustración 18. Caso de Prueba Funcional 3. Caso de Error 2

Caso de Prueba	4
Requerimiento	Servidor corriendo en puerto 8000 y conexión a Base de Datos
Acción	Crear Línea de Investigación

Tipo de Prueba	Funcional.
Hardware	Sistema de cómputo con un Procesador AMD-A12 - Memoria RAM de 8GB.
Software	<ul style="list-style-type: none"> • Sistema Operativo: Windows 11 • Base de Datos MariaDB – • Navegador: Google Chrome 119.0.60 • Servidor Web: Node.js 18
Personal	Desarrollador
Caso de Prueba Funcional	4
Datos de Entrada	Nombre de Línea de Investigación, Descripción, Jefe de Línea
Datos de Salida (Caso de Éxito)	<ul style="list-style-type: none"> • Objeto: Devuelve un objeto con la nueva Línea de Investigación
Datos de Salida (Caso de Error 1)	<ul style="list-style-type: none"> • Mensaje: "No se encontró el Jefe de Línea asignado entre los doctores registrados"
Datos de Salida (Caso de Error 2)	<ul style="list-style-type: none"> • Mensaje: "Campos Inválidos" • Objeto: Objeto con la descripción de las validaciones

Tabla 9. Caso de Prueba 1. Fuente: Elaboracion del Autor

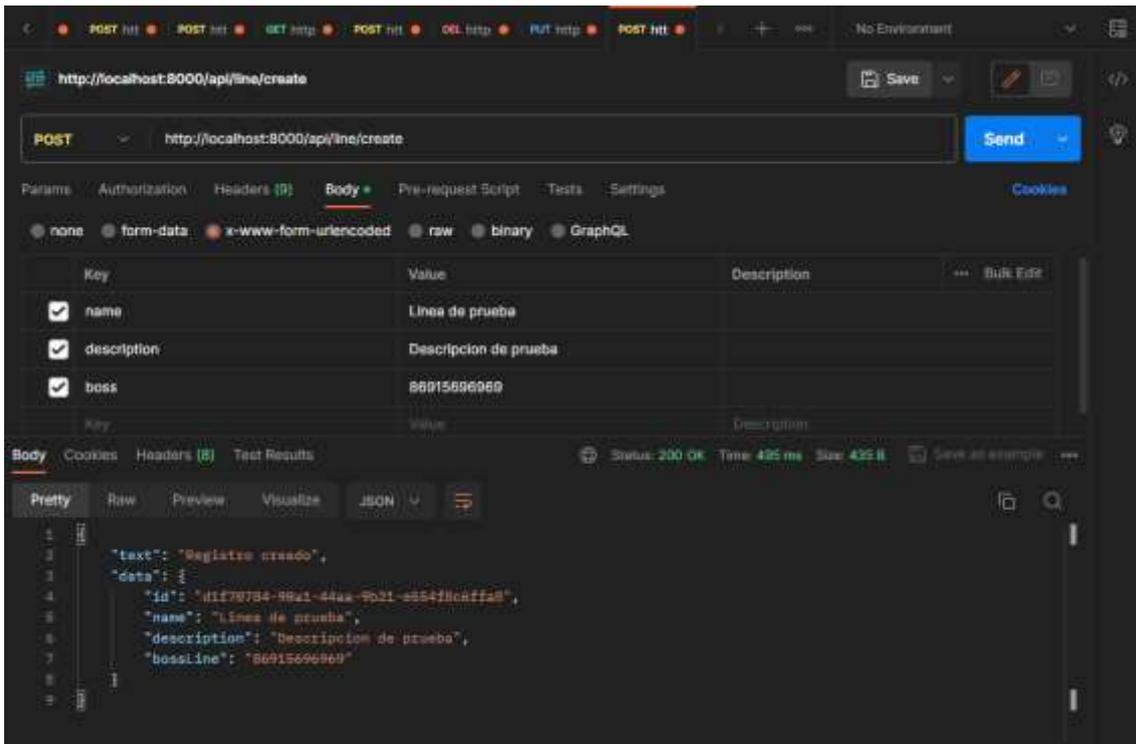


Ilustración 19. Caso de Prueba Funcional 4. Caso de Éxito

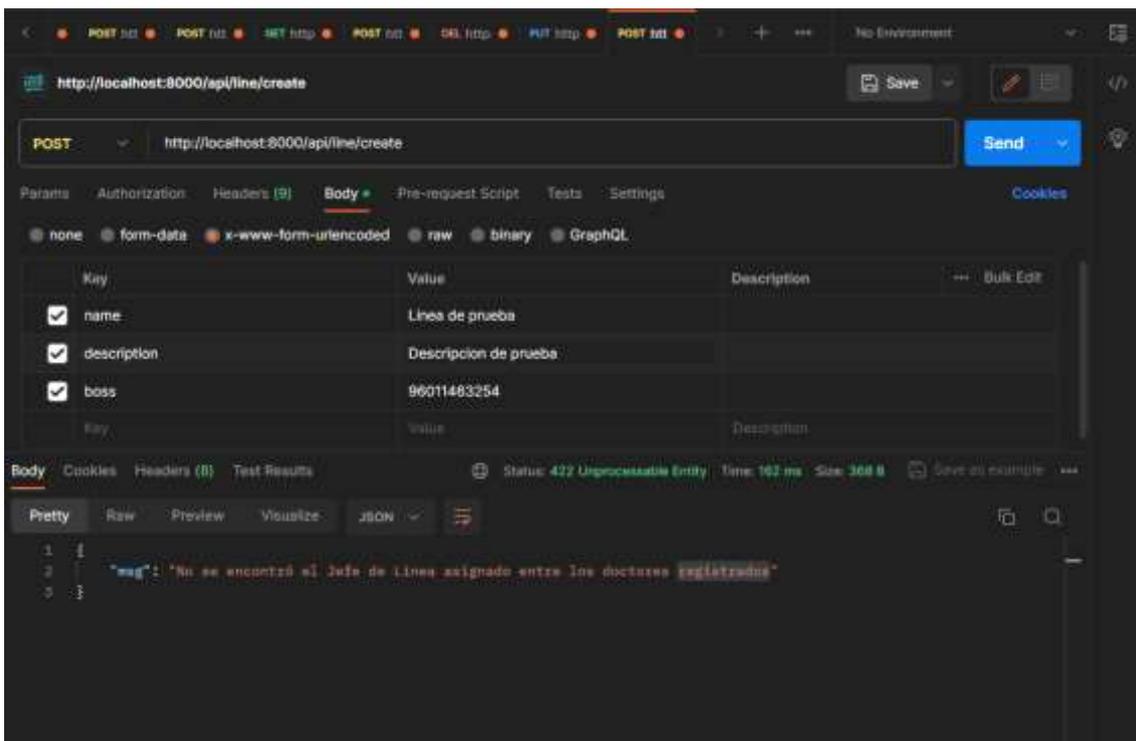


Ilustración 20. Caso de Prueba Funcional 4. Caso de Error 1

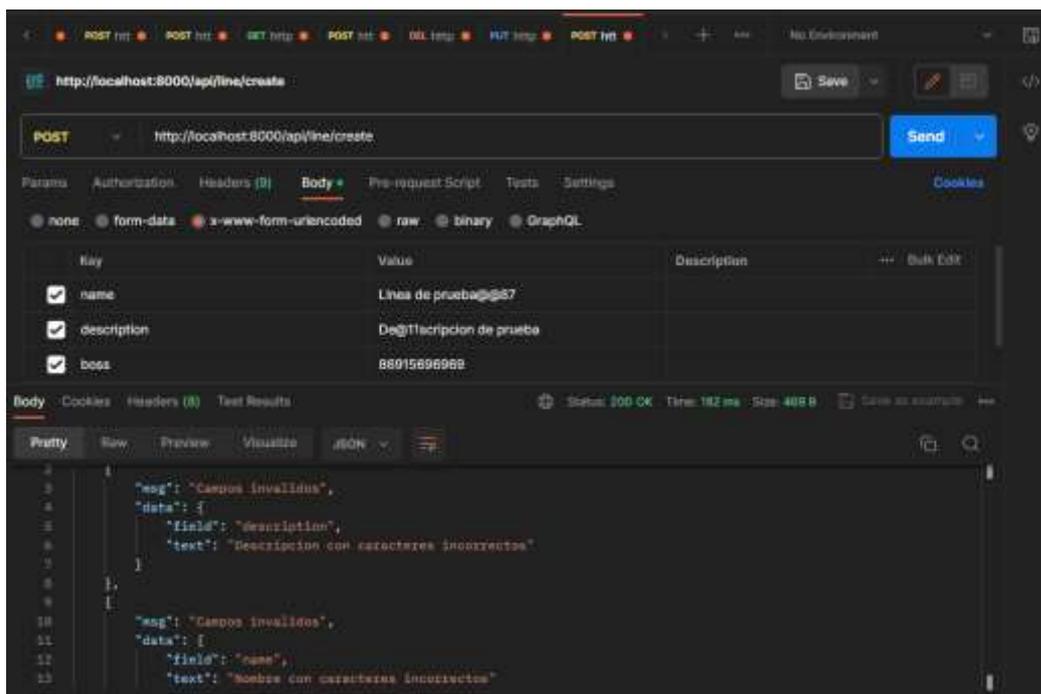


Ilustración 21. Caso de Prueba Funcional 4. Caso de Error 2

Caso de Prueba	5
Requerimiento	Servidor corriendo en puerto 8000 y conexión a Base de Datos
Acción	Agregar Documentación
Tipo de Prueba	Funcional.
Hardware	Sistema de cómputo con un Procesador AMD-A12 - Memoria RAM de 8GB.
Software	<ul style="list-style-type: none"> • Sistema Operativo: Windows 11 • Base de Datos MariaDB – • Navegador: Google Chrome 119.0.60 • Servidor Web: Node.js 18
Personal	Desarrollador
Caso de Prueba Funcional	5
Datos de Entrada	<ul style="list-style-type: none"> • Datos planos: Tema de Investigación, Descripción, Tipo de Documento, • Datos Binarios: Documento (Solo formato PDF)
Datos de Salida (Caso Éxito)	<ul style="list-style-type: none"> • Objeto: Devuelve un objeto con el documento creado
Datos de Salida (Caso de Error 1)	<ul style="list-style-type: none"> • Mensaje: “Extensión de Documento no soportada. Solo PDF”

Datos de Salida (Caso de Error 2)

- **Mensaje:** "Campos Inválidos"
- **Objeto:** Objeto con la descripción de las validaciones

Tabla 10. Caso de Prueba 1. Fuente: Elaboracion del Autor

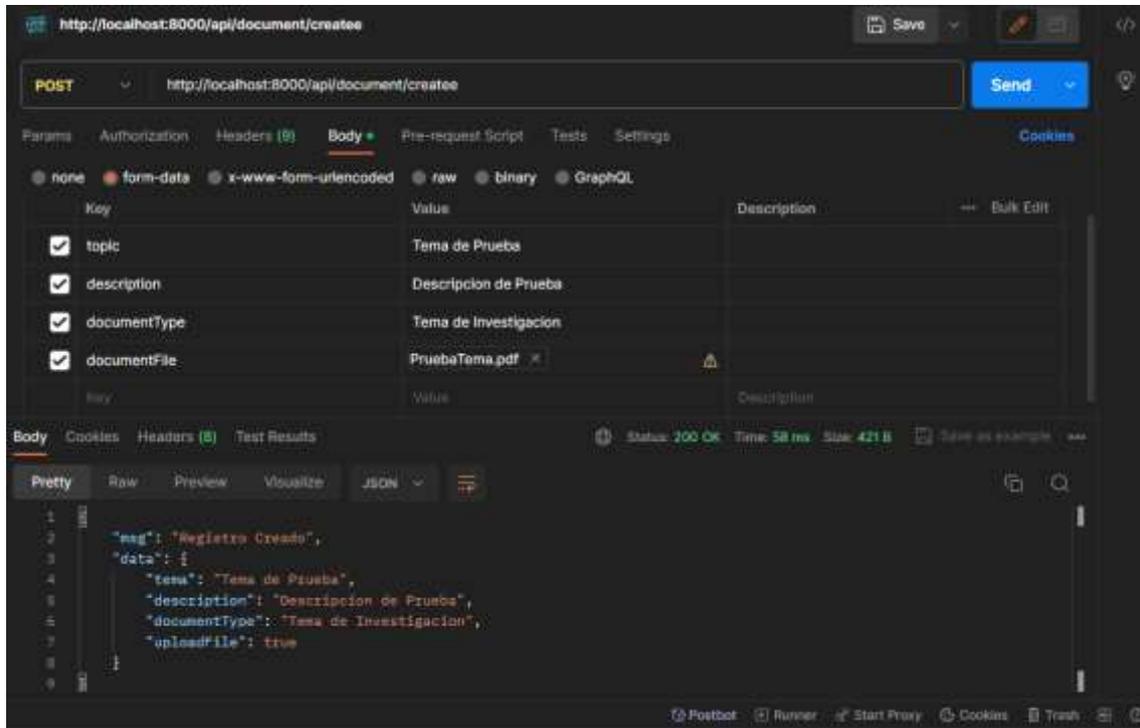


Ilustración 22. Caso de Prueba Funcional 5. Caso de Éxito

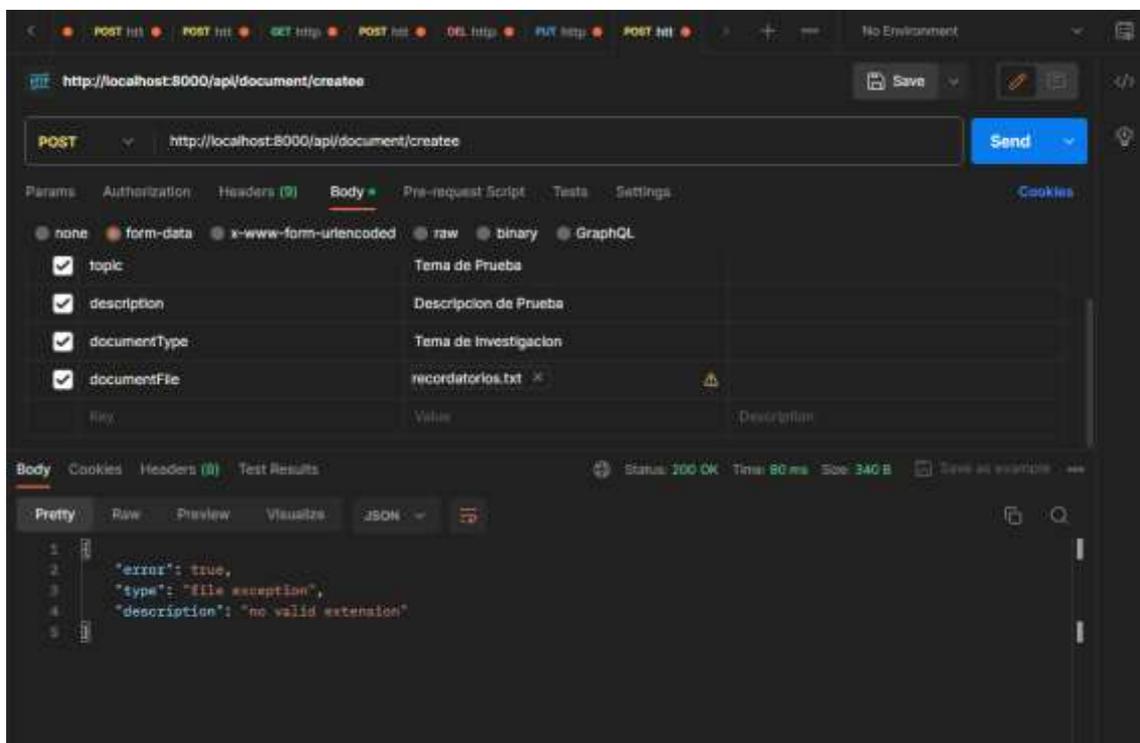


Ilustración 23. Caso de Prueba Funcional 5. Caso de Error 1

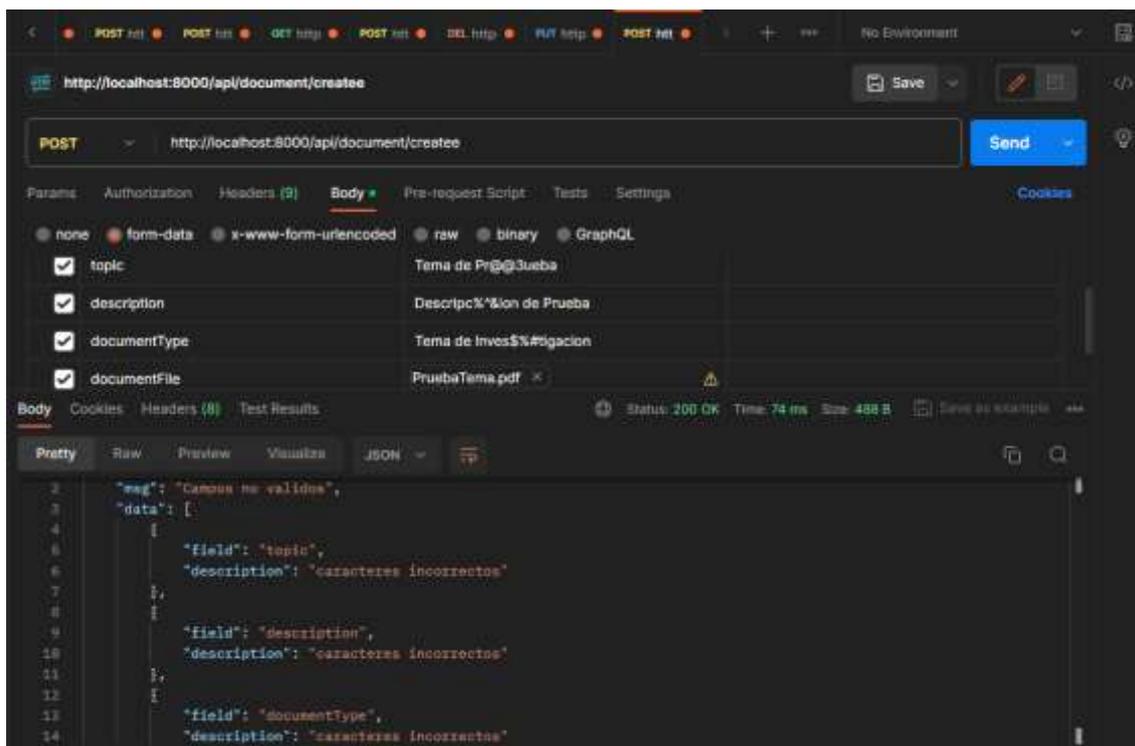


Ilustración 24. Caso de Prueba Funcional 5. Caso de Error 2

2.2.3.3 Pruebas de Aceptación

Las pruebas de aceptación se realizan para validar que el sistema cumple con los requisitos del usuario y está listo para su implementación. Estas pruebas son realizadas por el cliente o usuario final del sistema, y su objetivo es asegurar que el software cumple con las expectativas y necesidades del usuario.

Estas se llevan a cabo sobre la interfaz de usuario, y permiten de una manera ver cómo funciona la integración de las distintas partes del sistema al estar basado en una arquitectura como MVC. Para la documentación de estas pruebas, se utilizaron como referencia las pruebas funcionales descritas a los casos de uso propuestos, para demostrar la correspondencia de los resultados de las pruebas funcionales a nivel de código como los de las pruebas de aceptación a nivel de interfaz

Caso de Prueba	6
Prueba funcional en la que se basa	Prueba Funcional 5
Acción	Agregar Documentación
Tipo de Prueba	Aceptación.
Caso de Prueba de Aceptación	1

Observaciones	<ul style="list-style-type: none"> • Los datos binarios están validados desde el momento que selecciona el documento a enviar para que siempre sea un PDF (Nunca se dará el Caso de Error 1) • Los tipos de documento se encuentran contenidos en un SELECT en la interfaz para evitar que se envíen de manera incorrecta (Nunca se dará el Caso de Error 1)
----------------------	--

Tabla 11. Caso de Prueba 1. Fuente: Elaboracion del Autor

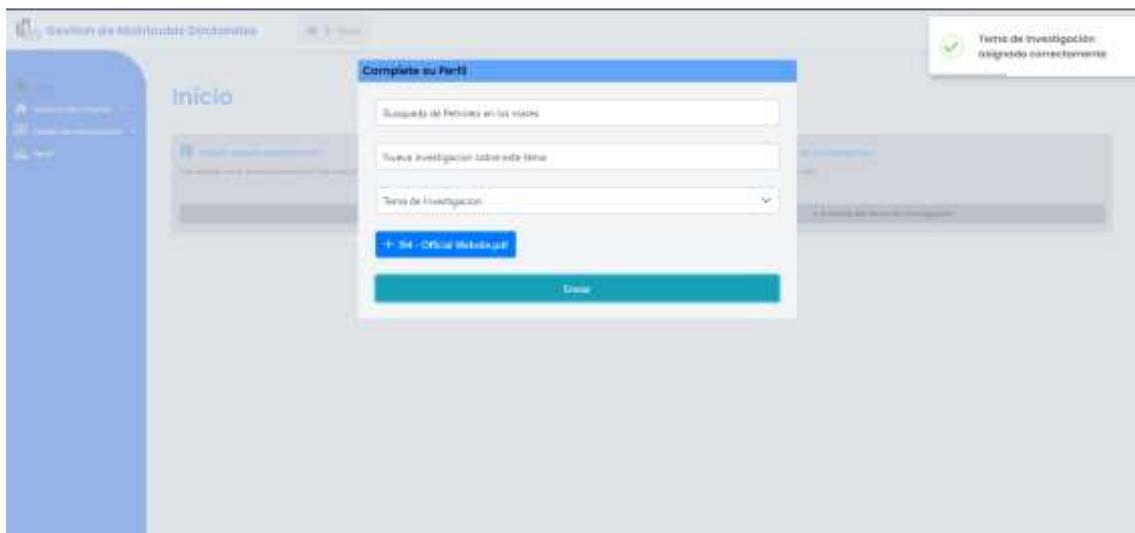


Ilustración 25. Caso de Prueba 6. Prueba de Aceptación . Caso de Éxito

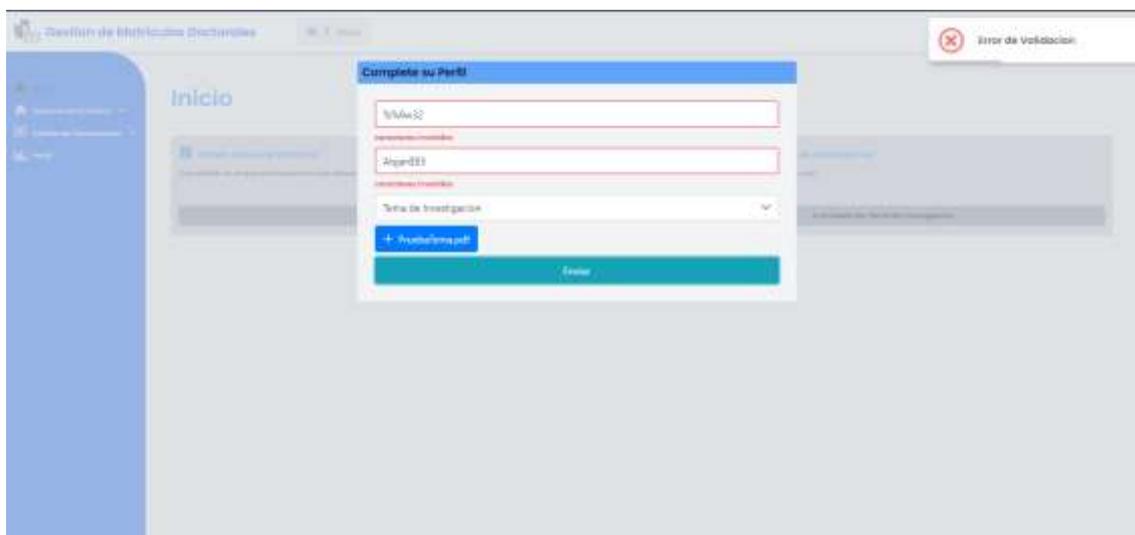


Ilustración 26. Caso de Prueba 6. Prueba de Aceptación .Caso de error 2

Caso de Prueba	7
Prueba funcional en la que se basa	Prueba Funcional 4
Acción	Crear nueva Línea de Investigación

Tipo de Prueba	Aceptación.
Caso de Prueba de Aceptación	2
Observaciones	<ul style="list-style-type: none"> Los doctores aptos para ser jefes de línea se encuentran contenidos en un SELECT en la interfaz para evitar que se envíen de manera incorrecta (Nunca se dará el Caso de Error 1)

Tabla 12. Caso de Prueba 1. Fuente: Elaboracion del Autor

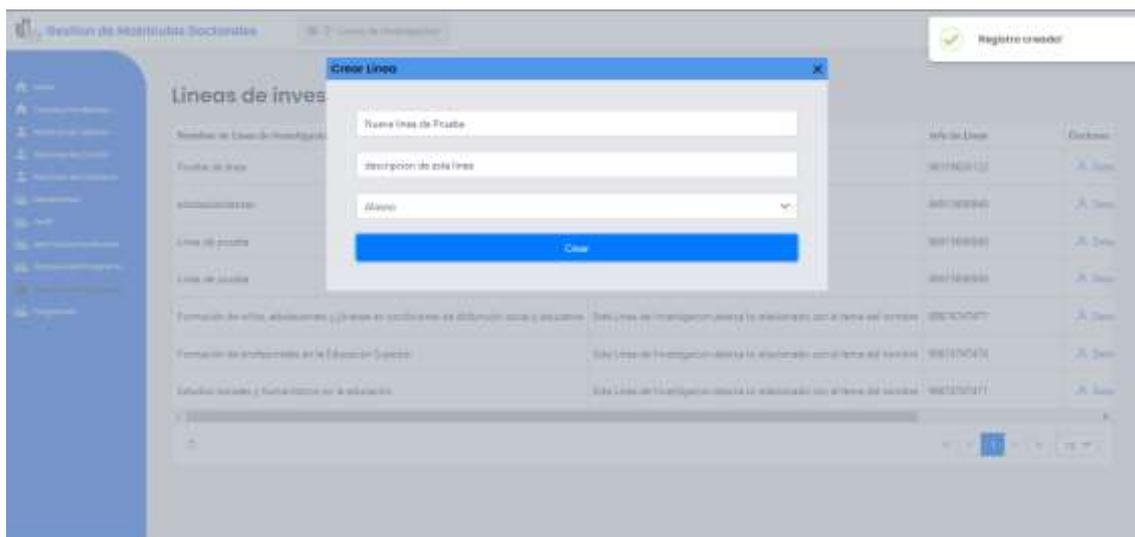


Ilustración 27. Caso de Prueba 7. Prueba de Aceptación. Caso de Éxito.

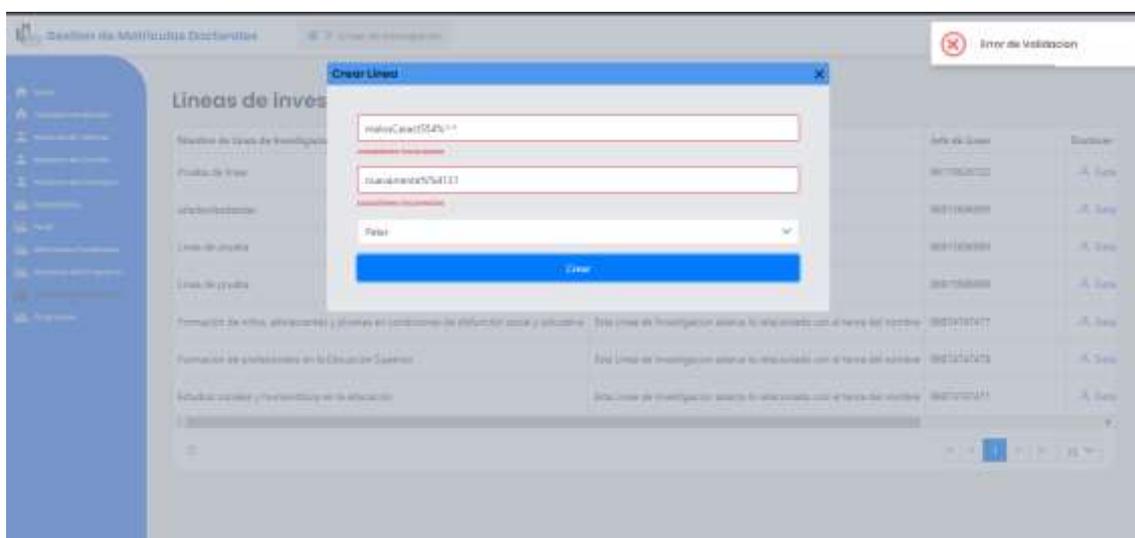


Ilustración 28. Caso de Prueba 7. Prueba de Aceptación. Caso de Error 2

Caso de Prueba	8
Prueba funcional en la que se basa	Prueba Funcional 1
Acción	Crear nuevo Doctor

Tipo de Prueba	Aceptación.
Caso de Prueba de Aceptación	3
Observaciones	<ul style="list-style-type: none"> Las Categorías Docentes se encuentran contenidas en un SELECT para evitar que estén se envíen de manera errónea

Tabla 13. Caso de Prueba 1. Fuente: Elaboracion del Autor

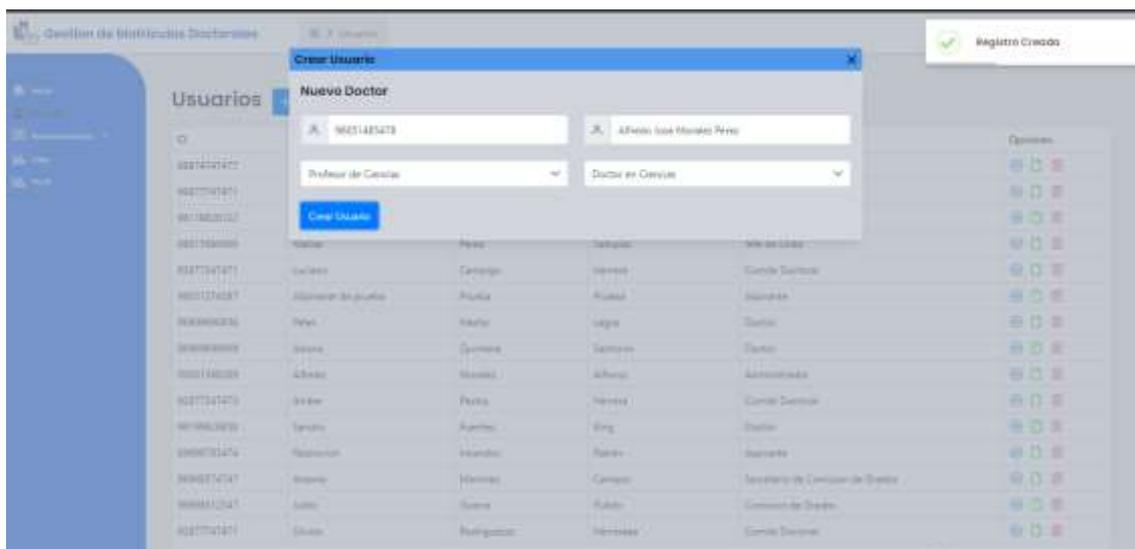


Ilustración 29. Caso de Prueba 8. Prueba de Aceptación. Caso de Éxito

Caso de Prueba	9
Prueba funcional en la que se basa	Prueba Funcional 2
Acción	Eliminar Doctor
Tipo de Prueba	Aceptación.
Caso de Prueba de Aceptación	4
Observaciones	<ul style="list-style-type: none"> El Id del doctor está asociado en la vista con el botón de eliminar, por lo que la acción siempre se realizara de manera satisfactoria (Nunca se dará el Caso de Error)

Tabla 14. Caso de Prueba 1. Fuente: Elaboracion del Autor

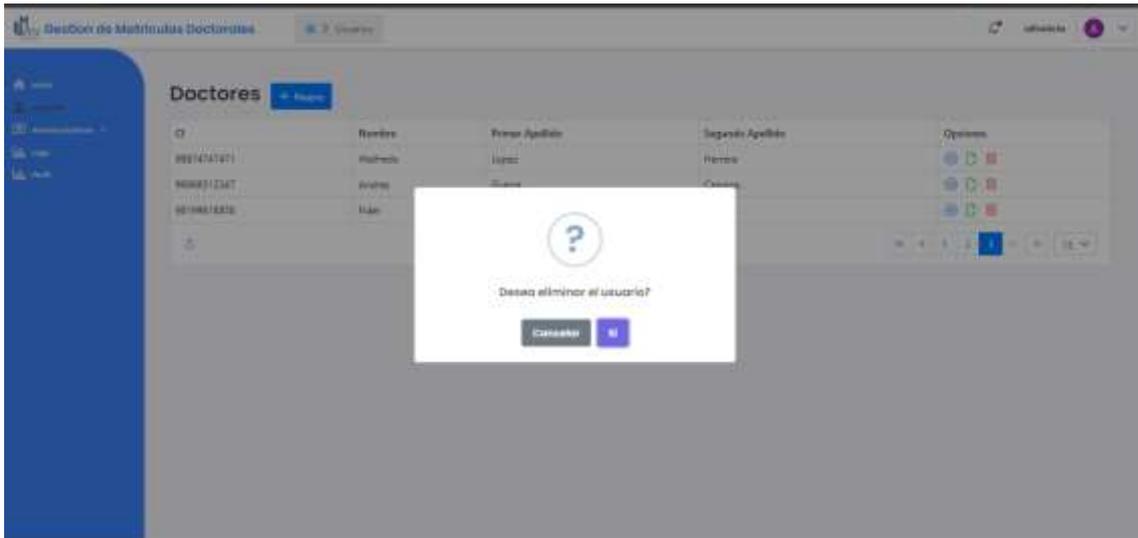


Ilustración 30. Caso de Prueba 9. Prueba de Aceptación. Caso de Éxito

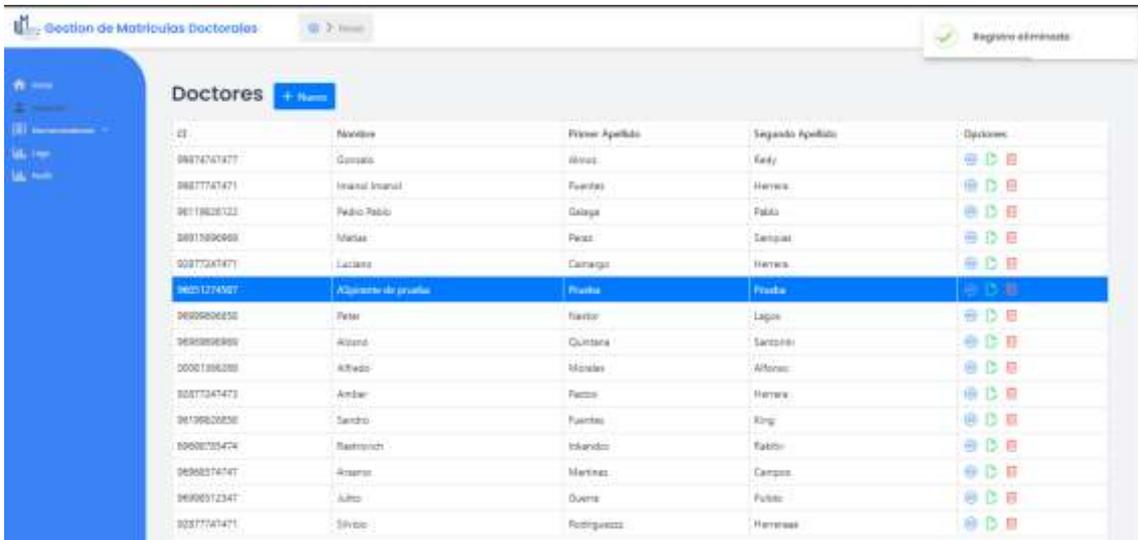


Ilustración 31. Caso de Prueba 9. Prueba de Aceptación. Caso de Éxito

Caso de Prueba	10
Prueba funcional en la que se basa	Prueba Funcional 3
Acción	Editar Categoría Docente
Tipo de Prueba	Aceptación.
Caso de Prueba de Aceptación	4
Observaciones	<ul style="list-style-type: none"> El Id de la Categoría Docente está asociado en la vista con el botón de Editar, por lo que la acción siempre se realizara de manera satisfactoria (Nunca se dará el Caso de Error 1)

Tabla 15. Caso de Prueba 1. Fuente: Elaboracion del Autor

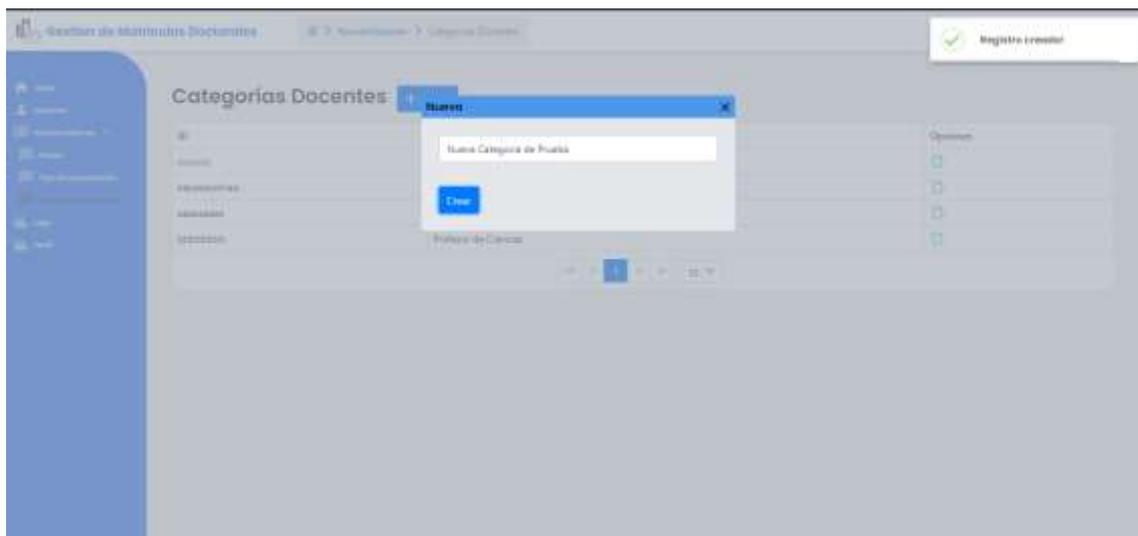


Ilustración 32. Caso de Prueba 10. Caso de Prueba de Aceptación. Caso de Éxito

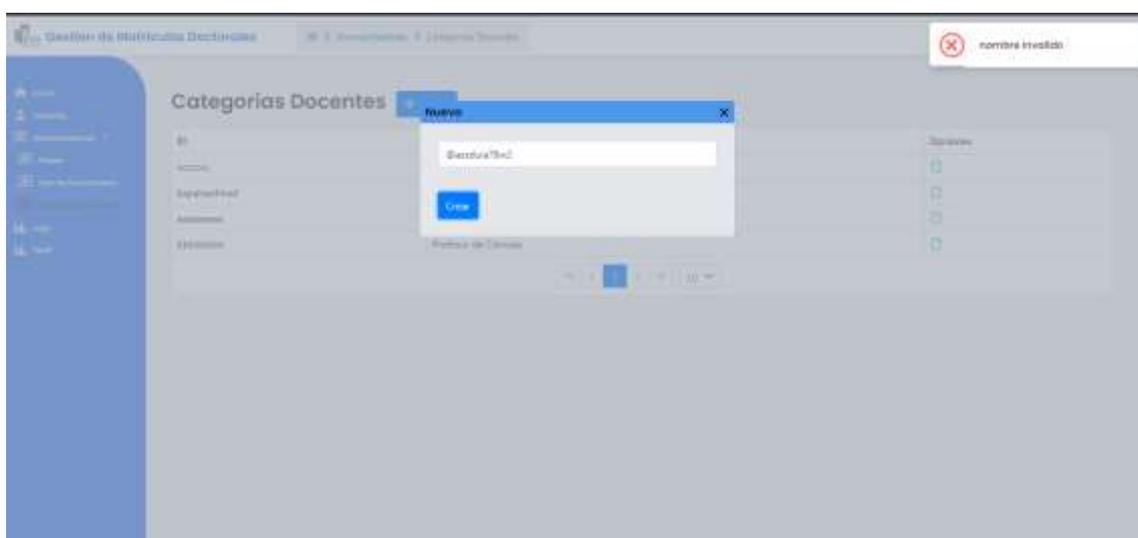


Ilustración 33. Caso de Prueba 10. Caso de Prueba de Aceptación. Caso de Error 2

2.2.3.4 Pruebas de Seguridad

Las pruebas de seguridad de aplicaciones (AST) son un proceso de identificación, análisis y corrección de las vulnerabilidades de seguridad de una aplicación web. Incluye probar la aplicación para detectar vulnerabilidades conocidas y examinar el código para detectar posibles problemas de seguridad. El proceso consiste en probar el código de la aplicación y su entorno para detectar fallos de seguridad y posibles vulnerabilidades. Una vez identificados, los problemas se abordan y solucionan. (Distillery, 2022)

Herramienta utilizada para el Desarrollo de Pruebas Funcionales

Al igual que en las pruebas funcionales, la herramienta escogida para hacer las pruebas de seguridad fue **Postman**, debido a que es un sistema con pocos usuarios y la seguridad está enfocada principalmente en la gestión de permisos y, al tener un flujo de

información tan estricto y extenso, se necesitan hacer pruebas más centralizadas y a nivel de código.

Una de las principales ventajas de esta herramienta es la configuración de encabezados de seguridad para hacer pruebas en los endpoints de la api y validar la seguridad y el control de acceso a través de la asignación de los tokens mediante estos encabezados

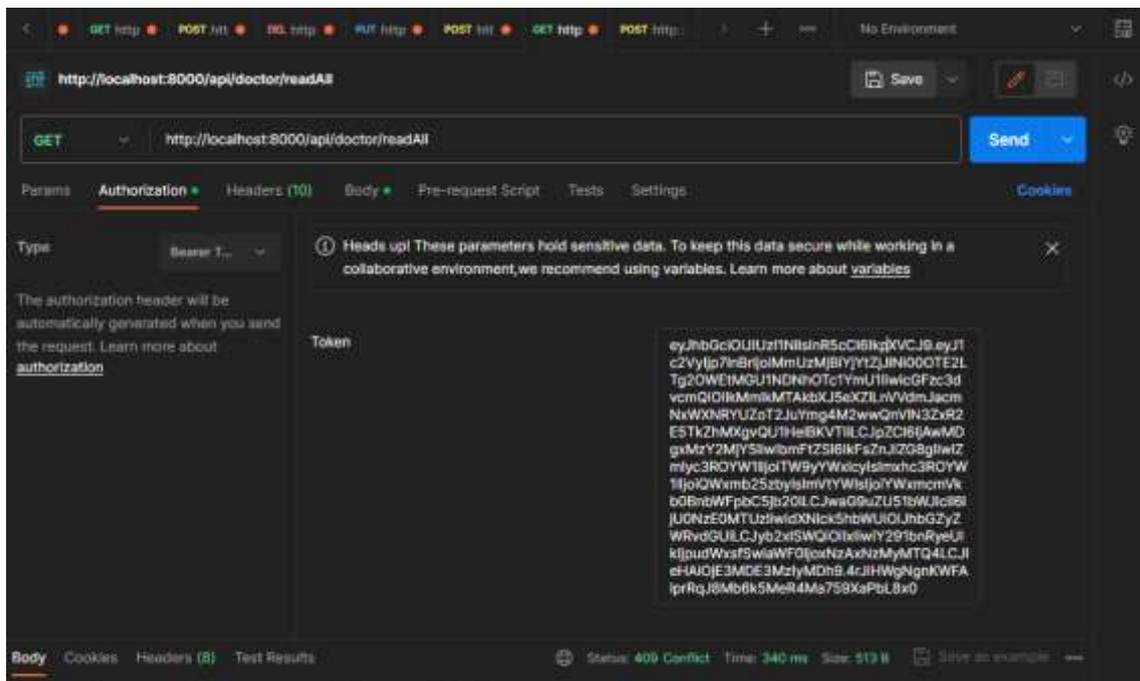
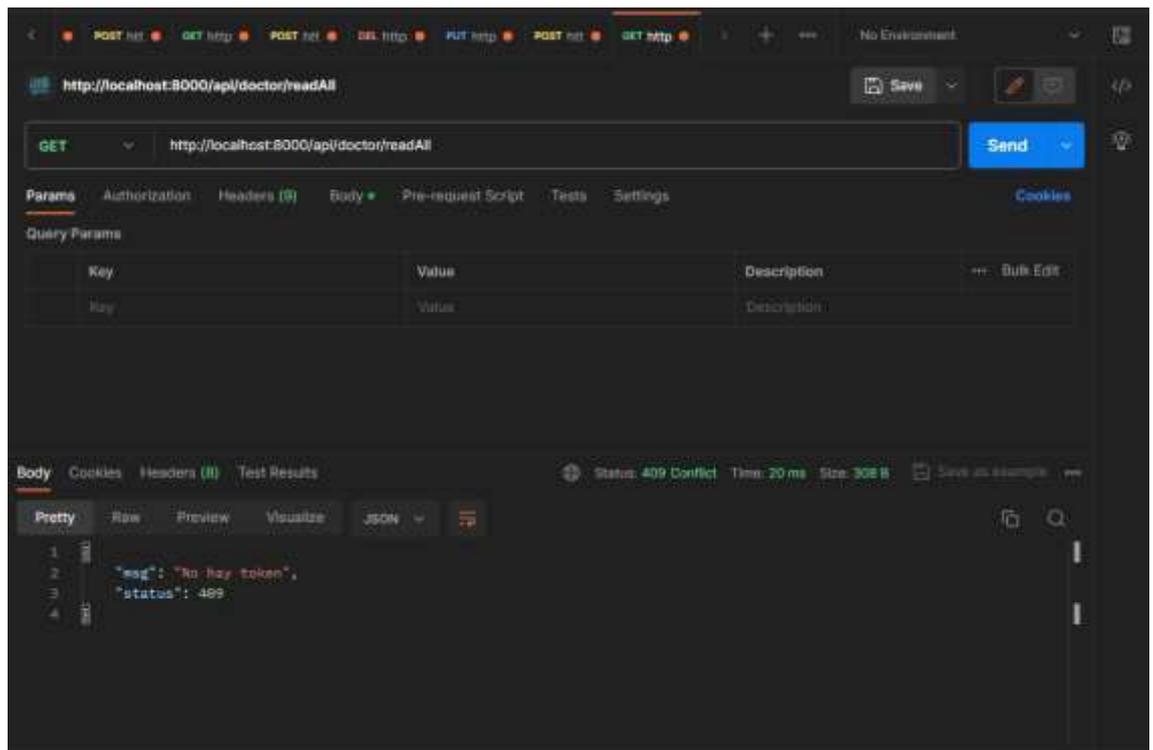


Ilustración 34. Configuración de encabezados de autorización en Postman

Se dividen las pruebas de seguridad en los siguientes aspectos para una mejor comprensión:



1.

Ilustración 36. Acción Obtener Doctores no permitida

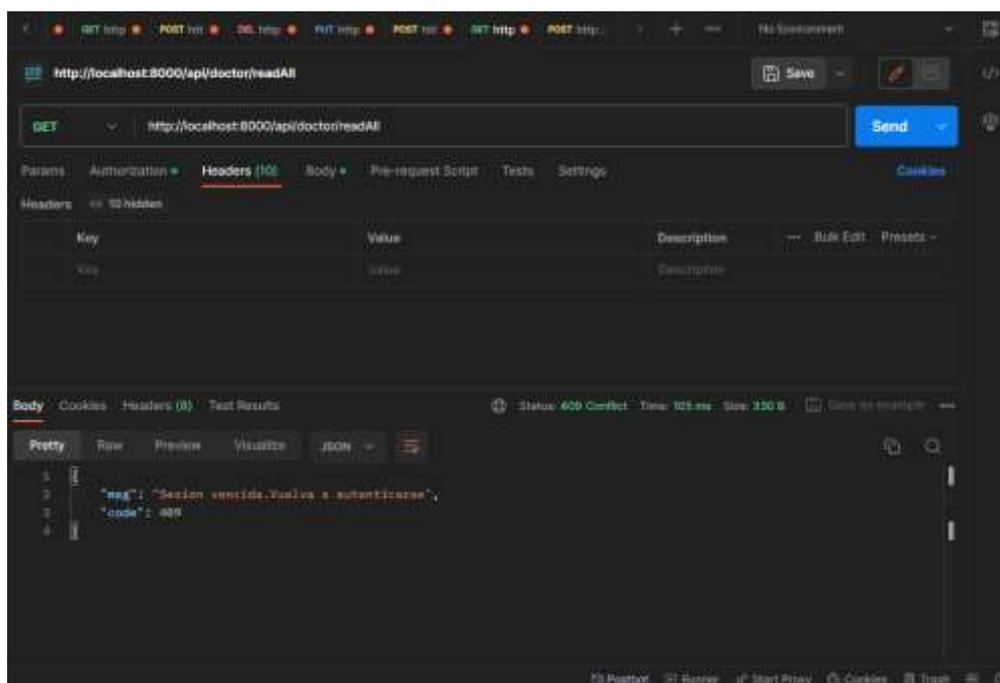


Ilustración 37. Sesión vencida

- 2. Control de Acciones por roles de bases de datos:** Las acciones realizadas por los usuarios, antes de llegar al controlador que la ejecuta y guarda los registros, es filtrada por un interceptor que interpreta el token y mediante las credenciales que posee, activa una sesión en el Backend con los privilegios necesarios. Con esto se logra un reforzamiento de la seguridad, donde los

permisos son a niveles de bases de datos y los mensajes de error los envía el mismo gestor de datos en cuestión, evitando inyecciones de código SQL.

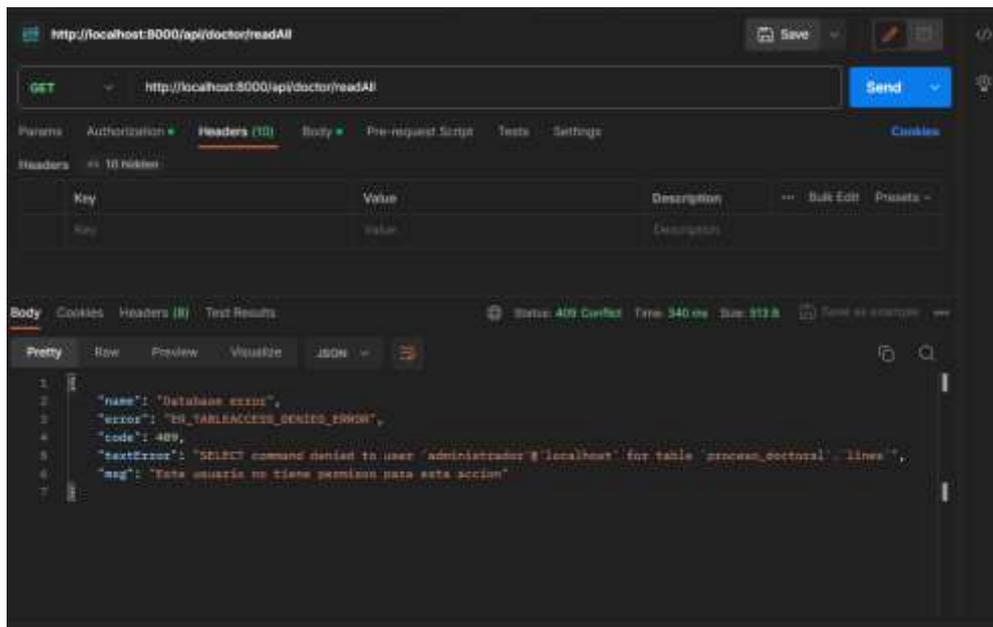


Ilustración 38. Mensaje SQL de acceso denegado a un recurso

Con este método, se obtiene un resultado satisfactorio en la implementación de la seguridad. El solicitar un recurso es validado dos veces antes de lograr llegar a la clase controladora y siempre que se realiza una petición, tanto el token como la sesión son validadas.

```
app.use("/api", accesControl, routingApp);
```

Ilustración 39. Interceptor de todas las peticiones del sistema

Este Access Control posee toda la lógica descrita anteriormente, y como puede evidenciarse, antes de llegar al routing de la app, se ejecuta toda la lógica de control de accesos y privilegios descrita anteriormente

2.2.3.5 Preparación para el despliegue

Para la maquetación de cómo será el despliegue, se utilizó la construcción de un Diagrama de Despliegue como herramienta de apoyo para su realización. Un diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, y el middleware que los conecta. Los diagramas de despliegue se utilizan normalmente para visualizar el hardware y el software físico de un sistema. Usándolo puedes entender cómo el sistema se desplegará físicamente en el hardware (Creately, 2022).

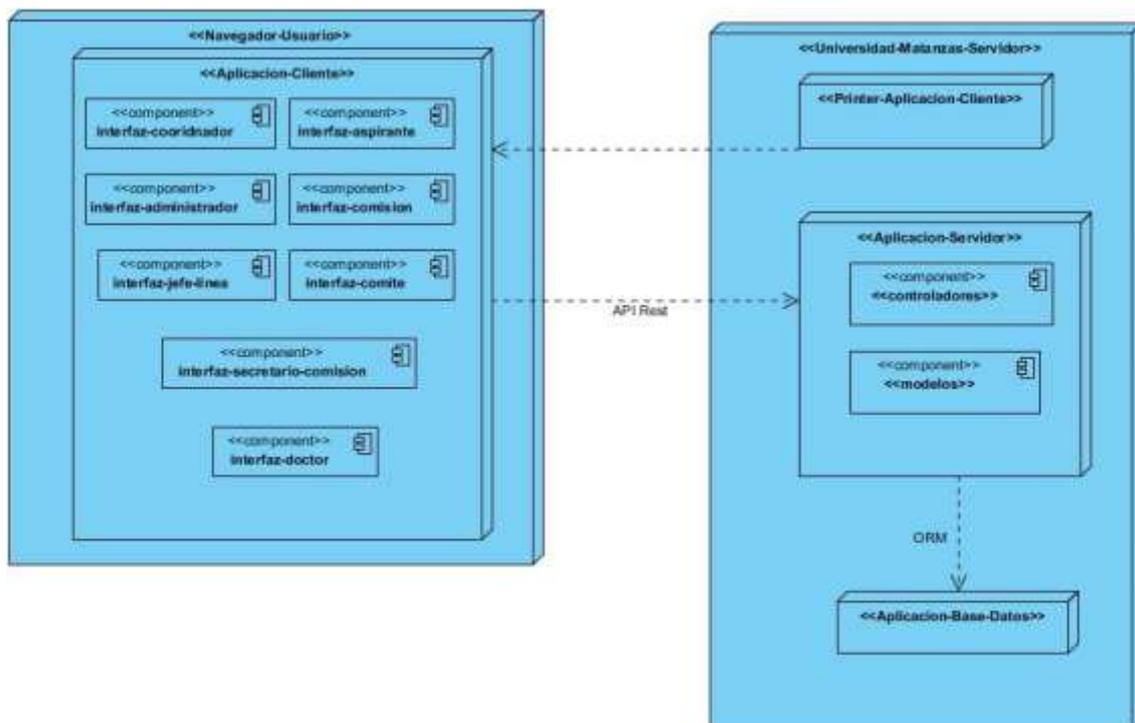


Ilustración 40 Diagrama de Despliegue

2.2.4 Fase de Transición

En la fase de transición el objetivo es garantizar que los requisitos se han cumplido, con la satisfacción de las partes interesadas. Esta fase a menudo se inicia con una versión beta de la aplicación. Otras actividades incluyen la preparación del ambiente, se completan, se identifican y corrigen defectos. La fase de transición termina con un cierre dedicado al aprendizaje de lecciones, las cuales quedan para futuros ciclos.

- Se libera el producto y se entrega al usuario para un uso real.
- Se incluyen tareas de marketing, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.
- Los manuales de usuario se completan y refinan con la información anterior.

En el caso de la fase de transición no será explicada pues el sistema aún se encuentra en el entorno de desarrollo y no se ha implantado. Se muestran los resultados obtenidos (información, datos, etc.) y se presentan los análisis correspondientes a dichos resultados a la luz de los referentes teóricos o investigaciones

2.3 Modelado de los Datos

Un modelo de datos es una representación visual de los elementos de datos de una empresa y de las conexiones entre ellos. Desempeña una función esencial al reunir a todos los segmentos de una empresa, incluyendo a los de TI, a los analistas de negocio y a los administradores, entre otros, para diseñar conjuntamente los sistemas de

2.3.2 Modelo Lógico

Esta fase incluye el perfeccionamiento de las entidades del sistema y empresariales de alto nivel de la fase conceptual en entidades lógicas más detalladas. Estas entidades lógicas y sus relaciones se pueden definir, opcionalmente, en un modelo lógico de datos mediante la utilización de los elementos de modelado del perfil UML para el diseño de bases de datos, como se describe en la Directriz: Modelo de datos. Este modelo lógico de datos forma parte del Producto de trabajo: Modelo de datos(Araneda, 2022)

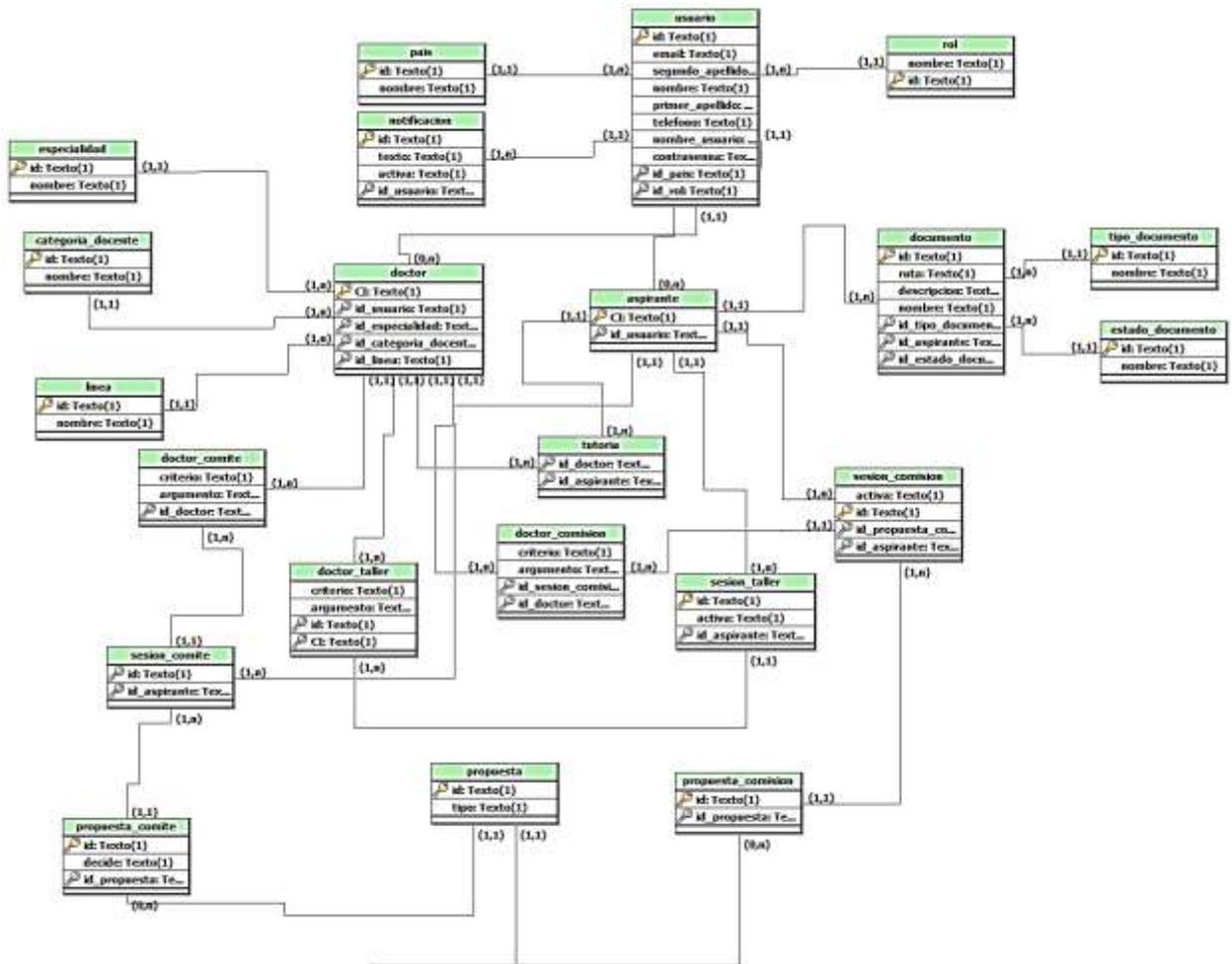


Ilustración 42 Modelo Lógico. Fuente: Autor

2.3.3 Modelo Físico

Esta fase incluye la transformación de los diseños de la clase lógica en diseños de tablas de bases de datos físicas detalladas y optimizadas. La fase física también incluye la correlación de los diseños de tablas de base de datos con espacios de tablas y con el componente de base de datos en el diseño de almacenamiento de bases de datos.(Araneda, 2022)

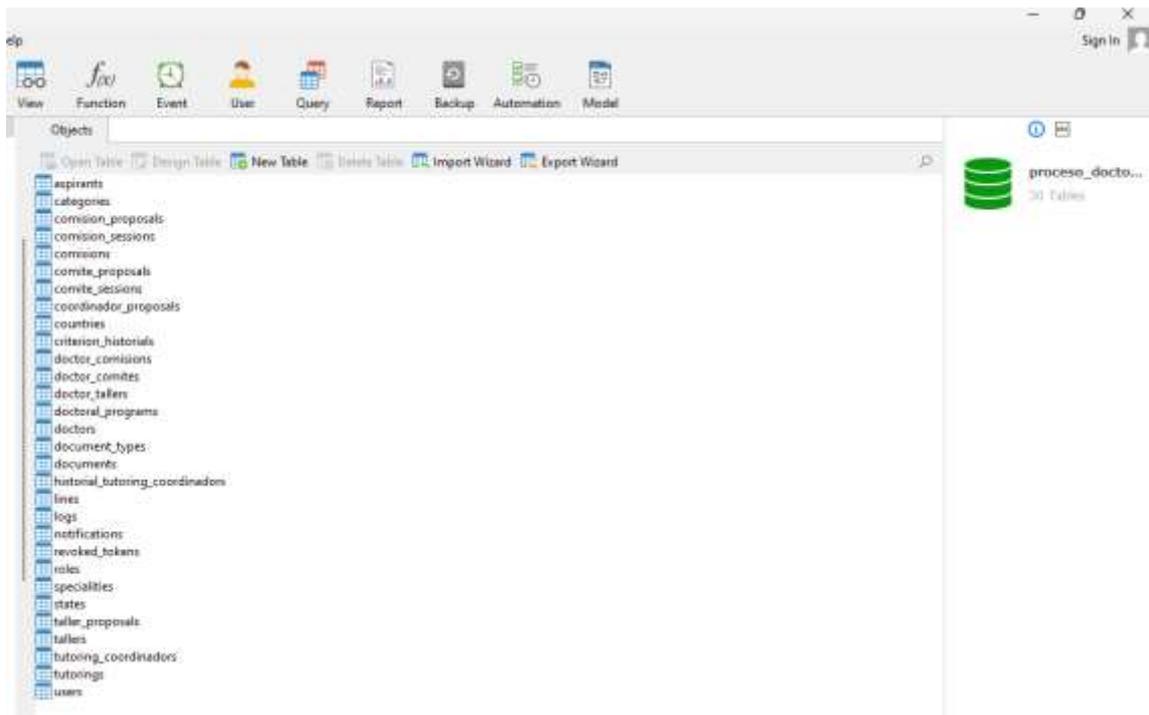


Ilustración 43. Modelo Físico de Datos. Fuente: Autor

2.4 Análisis del costo del Software desarrollado

El análisis de puntos de función es una técnica estándar para estimar el tamaño y la complejidad de un proyecto de desarrollo de software. Los puntos de función proporcionan una forma fiable de determinar el tamaño del software a partir de la definición de su alcance. Esta técnica se basa en el análisis de los requisitos del sistema, que se expresan como "puntos de función". Estos puntos se utilizan para estimar el coste de desarrollo y mantenimiento de un proyecto de software.

El principio de este enfoque es que software comprende un número de puntos de funciones que encajan dentro de uno de los cinco tipos de componentes. A cada punto función se le da un marcador (el multiplicador) dependiendo de su grado de complejidad. (Praxis, s. f.)

Componente	Acrónimo
Entrada	i
Salida	o
Consulta	q
Externos	if
Internos	ef

Tabla 16. Componentes de Medicion. Fuente: Elaboracion del Autor

Se definió el peso para cada requisito funcional del sistema, dando comienzo a la definición de los atributos necesarios para obtener el costo del software desarrollado

Requisitos	Componentes	Peso	Peso(Valor)
1-Sistema de autentificado	Consulta	Medio	4
2- Crear Doctor	Entrada	Medio	4
3- Eliminar Doctor	Entrada	Medio	4
4- Editar Doctor	Entrada	Medio	4
5- Listar Doctor	Salida	Medio	4
6-Crear Categoría Docente	Entrada	Bajo	3
7-Eliminar Categoría Docente	Entrada	Bajo	3
8-Editar Categoría Docente	Entrada	Bajo	3
9-Obtener Categoría Docente	Salida	Bajo	3
10-Crear Especialidad	Entrada	Bajo	3
11-Eliminar Especialidad	Entrada	Bajo	3
12-Editar Especialidad	Entrada	Bajo	3
13-Obtener Especialidad	Salida	Bajo	3
14-Ver Estado de Documentación	Entrada	Bajo	3
15-Ver Historial de Criterios	Salida	Bajo	3
16-Editar Documentación	Entrada	Medio	4
17-Agregar Documentación	Entrada	Medio	4
18-Generar PDF Sesión Comité	Salida	Bajo	3
19-Generar PDF Taller	Entrada	Bajo	3
20-Generar PDF Sesión Comisión	Entrada	Bajo	3
21-Notificar Usuarios	Entrada	Bajo	3
22-Crear Estadísticas asociadas al proceso	Salida	Bajo	3

23-Emitir criterio sobre tema de investigación en Sesión de Comité	Entrada	Bajo	3
24-Hacer propuestas de tutorías en Sesión de Comité	Salida	Bajo	3
25-Ver Sesiones de Taller	Salida	Bajo	3
26-Ver Sesiones de Comite	Salida	Bajo	3
27-Ver Sesiones de Comisión	Salida	Bajo	3
28-Ver Tutorías Aprobadas	Salida	Bajo	3
29-Emitir criterio sobre tema de investigación en Sesión de Taller	Entrada	Medio	4
30-Ver Tutorías participadas	Salida	Bajo	3
31-Crear Programa de Formación Doctoral	Entrada	Medio	4
32-Editar Programa de Formación Doctoral	Entrada	Medio	4
33-Eliminar Programa de Formación Doctoral	Entrada	Medio	4
34-Crear Línea de Investigación	Entrada	Medio	4
35-Emitir criterio sobre Tema de Investigación en Sesión de Comisión	Entrada	Medio	4
36-Emitir criterio sobre Tutorías en Sesión de Comisión	Entrada	Medio	4
37-Ratificar criterios de Comisión de Grados	Entrada	Medio	4

38-Emitir criterios sobre Documentos de Matricula	Entrada	Medio	4
39-Ratificar criterios de Comité Doctoral	Entrada	Medio	4
40-Crear Taller	Entrada	Medio	4
41-Editar Taller	Entrada	Medio	4
42-Eliminar Taller	Entrada	Medio	4
43-Crear País	Entrada	Medio	4
44-Editar País	Entrada	Medio	4
45-Obtener País	Salida	Bajo	3
46-Ver trazas del Sistema	Salida	Bajo	3
47-Crear Usuario	Entrada	Medio	4
48-Editar Usuario	Entrada	Medio	4
49-Obtener Usuario	Salida	Bajo	3
50-Eliminar Usuario	Entrada	Medio	4

Tabla 17. Peso de Requerimientos Funcionales. Fuente: Elaboracion del Autor

Cálculo de los puntos de función sin ajustar.

$$PFSA = PFTi + PFTo + PFTq + PFTif + PFTef$$

$$PFSA = 34 * 4 + 15 * 3 + 1 * 4 + 0 + 0$$

PFSA= 185

Obtención de ACT

N° de factor	Factor	Valor 0...5
1	Comunicación de Datos	1
2	Proceso Distribuido	1
3	Objetivos de Rendimiento	1
4	Configuración de Explotación Compartida	1
5	Tasa de Transacciones	1
6	Entrada de Datos en Línea	1
7	Eficiencia con el Usuario Final	1
8	Actualizaciones en Línea	0
9	Lógica de Proceso Interno Compleja	0

10	Reusabilidad del Código	1
11	Conversión e Instalación Completadas	1
12	Facilidad de Operación	1
13	Instalaciones Múltiples	0
14	Facilidad de Cambio	2
	Ajuste de Complejidad Técnica (ACT)	12

Tabla 18. Ajuste de complejidad técnica. Fuente: Elaboración del Autor

- **Cálculo de los puntos de función ajustados.**

$$PFA = PFSA * [0.65 + (0.01 * ACT)]$$

$$PFA = 185 * [0.65 + (0.01 * ACT)]$$

$$PFA = 185 * [0.65 + (0.01 * 12)]$$

$$PFA = 142.45$$

- **Cálculo de las Líneas por PFA**

$$PFA = 1000 * (KLOC / 100)$$

$$PFA * 100 = 1000 * KLOC$$

$$142.45 * 100 = 1000 * KLOC$$

$$14245 = 1000 * KLOC$$

$$KLOC = 14245 / 1000$$

$$KLOC = 14.24$$

- **Cálculo de las líneas de código por punto de función**

$$\text{Líneas de código} = PFA * (\text{Líneas por PF})$$

$$\text{Líneas de código} = PFA * 14$$

$$\text{Líneas de código} = 1994.3$$

- **Cálculo del esfuerzo.**

$$\text{Esfuerzo horas/persona} = PFA / [1/8 \text{ persona/hora}]$$

$$\text{Esfuerzo horas/persona} = 142.45 / [1/8]$$

$$\text{Esfuerzo horas/persona} = 1139.6$$

- **Cálculo de duración del proyecto.**

$$\text{Horas laborables al mes} = (26 \text{ días} * 8 \text{ horas}) = 208$$

$$\text{Miembro} = 1$$

$$\text{Duración en meses} = (\text{Esfuerzo horas/persona}) / \text{Horas laborables al mes}$$

$$\text{Duración en meses} = 1139.6 / 208$$

$$\text{Duración en meses} = 6 \text{ meses}$$

- **Costo Total del Desarrollo del Software**

$$\text{Costo Total} = \text{suelo de 1 participante} * \text{cantidad de participantes} * \text{DM}$$

$$\text{Costo Total} = 3500 * 1 * 6 = 21000$$

- **Análisis de los beneficios**

El análisis del costo en comparación con los beneficios descritos a lo largo de este trabajo lleva a la conclusión de que los aspectos positivos superan a los costos. Además, es importante señalar que, para esta institución, el software resultó ser gratuito, ya que es el resultado del trabajo de diploma del autor.

2.5 Conclusiones Parciales del Capítulo

En conclusión, el desarrollo de una investigación con RUP que incluye pruebas de seguridad, funcionales y de aceptación, junto con el uso de diagramas de caso de uso, de despliegue y de componentes, proporciona una base sólida para el desarrollo de software. Al aplicar el análisis de puntos de función en este contexto, se puede estimar el tamaño y la complejidad del software, lo que a su vez permite una planificación más precisa del proyecto. Además, el enfoque disciplinado de RUP, combinado con las pruebas exhaustivas y los diagramas detallados, facilita la identificación temprana de posibles problemas y la toma de decisiones fundamentadas a lo largo del ciclo de vida del desarrollo del software. En este sentido, el uso de RUP y el análisis de puntos de función ofrecen una herramienta valiosa para estimar y gestionar el desarrollo de software de manera efectiva y eficiente.

Capítulo 3: Validación de la solución propuesta

3.1 Medición de las variables de Investigación

Se definieron durante la elaboración del Marco Teórico de la Investigación las siguientes variables:

- Variables dependientes:
 1. Estructuración de la información asociada al proceso
 2. Persistencia de la información asociada al proceso
- Variables independientes
 1. La aplicación web

Mediante una medición cualitativa de estas variables, se puede definir en que medida se cumple con el cumplimiento de la hipótesis planteada en el Marco Teórico de la Investigación y verificar si los resultados obtenidos son satisfactorios. Cabe destacar que se decidió medir las variables de esta manera, por ser el la persistencia el elemento clave de la hipótesis planteada y a si vez, el objetivo del desarrollo del software en cuestión.

3.1.1 Medición de las variables dependientes

Variable	Aspectos a medir cualitativamente
Estructuración de la información asociada al proceso	<ul style="list-style-type: none"> • La información guardada de forma empírica podría sufrir daños naturales • El proceso de matrícula es complejo y no cuenta con soporte gráfico para ayudar a su comprensión • Las distintas etapas de evaluación requieren de la asistencia personal de las partes • La participación humana directa a la hora de desarrollar el flujo del negocio puede retrasar el proceso de matrículas, debido a que hay muchas partes involucradas para su realización.
Persistencia de la información asociada al proceso	<ul style="list-style-type: none"> • No consta un registro del personal en soporte digital para evitar errores humanos • El proceso de matrículas no está almacenado en forma de datos dinámicos o consultables desde cualquier parte • La parte solicitante no tiene acceso a los señalamientos hechos o documentos entregados en caso de pérdida de estos

Tabla 19 . Medición de las variables dependientes respecto al desarrollo del proceso actual sin el Software

Variable	Aspectos a medir cualitativamente
Estructuración de la información asociada al proceso	<ul style="list-style-type: none"> • La información guardada en forma de datos dinámicos mejora la estructuración de la información asociada al programa y a la gestión del proceso del Aspirante • La comprensión del proceso de matrículas se simplifica a través del desarrollo de una interfaz atractiva y funcional • La utilización de la herramienta creada permite llevar el proceso de matrículas desde cualquier lugar con acceso a internet • No existe participación humana directa y la comunicación entre una etapa del proceso y otra se realiza a través de notificaciones
Persistencia de la información asociada al proceso	<ul style="list-style-type: none"> • Los registros creados son validados de manera correcta para impedir el riesgo de errores y corrupción de la información • Ambas partes pueden consultar el registro asociado a su rol en cualquier momento que desee

- | | |
|--|---|
| | <ul style="list-style-type: none">• Los aspirantes pueden consultar el estado de su proceso, obtener documentos enviados y solicitar los suyos para hacer correcciones en tiempo real |
|--|---|

Tabla 20. Medición de las variables dependientes respecto al desarrollo del proceso actual con el Software

La medición cualitativa realizada define como el desarrollo del sistema cumple con los aspectos a medir en cuanto al desenvolvimiento del proceso, arrojando criterios satisfactorios, mejorando la experiencia de los usuarios de ambas partes y la persistencia de los registros asociados.

3.2 Valoración de la solución obtenida y cumplimiento de la Hipótesis

Luego de la medición realizada a las variables definidas anteriormente, se tiene como valoración de la solución obtenida la siguiente:

El sistema creado para la Gestión del Proceso de Matrículas del Programa de Formación Doctoral de la Universidad de Matanzas, cumple en aspectos de experiencia de usuario satisfactoriamente, al poseer una interfaz amigable y la posibilidad de consulta de la información en tiempo real, permite llevar a cabo el proceso desde cualquier parte. Los criterios emitidos sobre la documentación de la parte evaluada son notificados a esta de manera automática y la gestión de la información de los miembros del programa se realiza de manera efectiva y persistente. La omisión directa de la participación humana en el flujo de negocio aumenta la eficiencia en términos de esfuerzo del personal que realiza las evaluaciones y la constancia de todos los registros y participaciones en los distintos eventos que se realizan en este, posibilita llevar un registro de cada iteración con toda la información asociada a este. Además, al estar desarrollado sobre la base del funcionamiento del proceso de manera remota, ahorra a su desarrollo costos en cuestiones de tiempo de desplazamiento. Los resultados de las pruebas realizadas arrojaron datos satisfactorios, por lo que el producto es usable, seguro y funcional

3.3 Vistas del Software

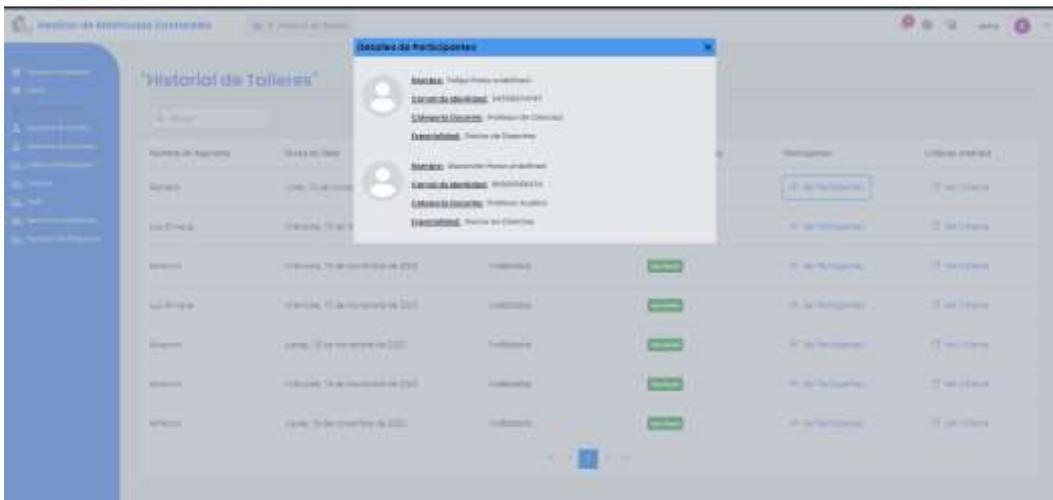


Ilustración 44 Vista de Historial de Talleres

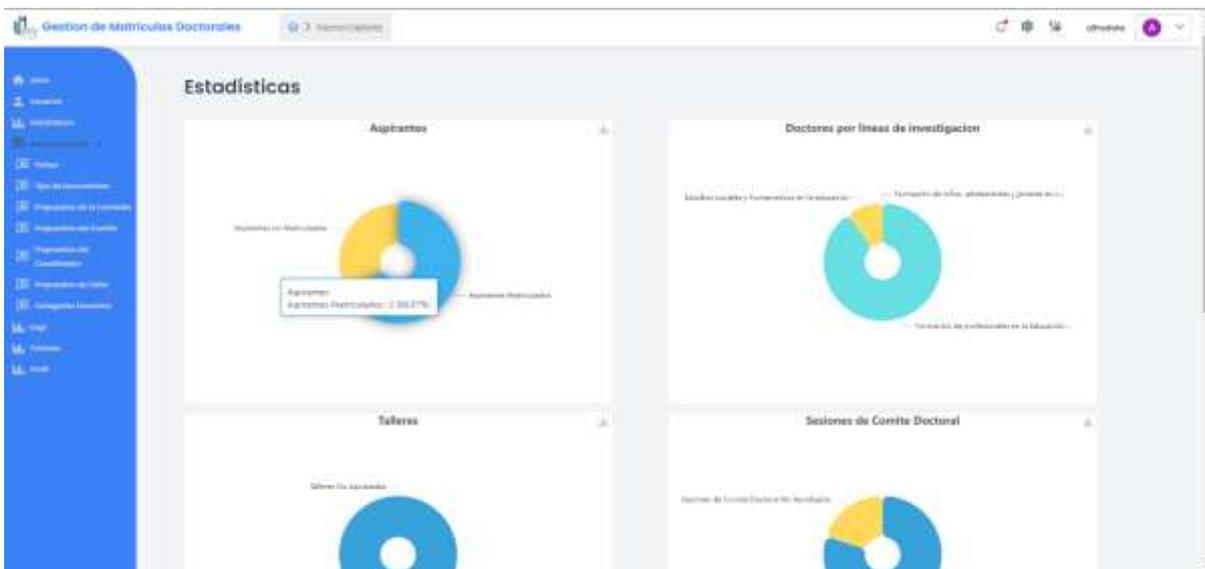


Ilustración 45. Vista de Estadísticas con datos asociados al proceso



Ilustración 46. Vista de Estado Real de documentos de Aspirante

ID	Nombre	Primer Apellido	Segundo Apellido	Rol	Opciones
5555555555	Alberto	E	Militar	Aspirante	[Icons]
1111111114	Rosalinda	Perez	Perez	Leña de Litro	[Icons]
3474747474	Alfonso	alfonso	alfonso	Aspirante	[Icons]
9999999999	Alberto	Militar	Rodriguez	Doctor	[Icons]
8888888888	Felipe	Perez	Rosari	Cuenta Doctoral	[Icons]
8888888888	Guadalupe	Guadalupe	Guadalupe	Aspirante	[Icons]
0000198428	Alfredo	Muñoz	Alfonso	Administrador	[Icons]
4444444457	Jose	Norales	Perez	Doctor	[Icons]
8888888888	Bárbara	Sánchez	Nolas	Cuenta Doctoral	[Icons]
8888888888	Alonso	Valderrama	Cruzado	Aspirante	[Icons]
7777777776	Edmundo	Edmundo	Edmundo	Aspirante	[Icons]
1111111112	Guadalupe	Guadalupe	Guadalupe	Aspirante	[Icons]
0000000000	Isabel	Gonzalez	Alejandro	Cuenta Doctoral	[Icons]
2222222222	Esteban	delgado	delgado	Aspirante	[Icons]

Ilustración 47. Vista de Historial de Usuarios del Sistema

Ilustración 48. Vista de Taller Activo

Ilustración 49. Vista de detalles de Sesión de Comisión de Grados para Aspirante.

3.4 Conclusiones Parciales del Capitulo

En este capítulo se abordó la medición cualitativa de las variables definidas en la hipótesis, así como la utilización de los resultados de esta en la formulación de la valoración de cómo cumple el software obtenido con lo planteado en la hipótesis. Para esta, se utilizaron referencias a las distintas pruebas realizadas y documentadas en el capítulo 3 y se mostraron una parte de las interfaces de usuario del sistema desarrollado. La valoración hecha es satisfactoria y queda demostrado que el producto cumple con las exigencias y requerimientos planteados desde el inicio de su desarrollo.

Conclusiones Generales

A lo largo de la investigación se describen los pasos para darle cumplimiento al objetivo de desarrollar una herramienta que permita gestionar información en el Proceso de Matriculas al Programa de Formación Doctoral en la Universidad de Matanzas.

Una vez finalizado el trabajo se plantean las siguientes conclusiones

- Se determinaron los referentes teórico-metodológicos de la gestión de la información de los procesos de matrícula en programas de postgrado a través de la investigación de las ventajas que ofrecen las aplicaciones que existen para este fin, como Banner y PowerSchool
- Se logró realizar un análisis de la viabilidad de crear una aplicación web que mejorara el proceso de matriculas al Programa de Formación Doctoral de la Universidad de Matanzas
- Utilizando las herramientas descritas y la metodología RUP, se logró hacer un diseño de la aplicación web para gestionar el proceso en cuestión y organizar el trabajo para lograr un resultado satisfactorio
- Utilizando las tecnologías descritas se logró la construcción de una aplicación web para mejorar el proceso de matrículas al Programa de Formación Doctoral de la Universidad de Matanzas y siguiendo las fases de la metodología RUP se gestionó el proceso de implementación de manera exitosa
- Las pruebas realizadas al sistema construido evidenciaron la validación del producto final, demostrando ser seguro, usable y funcional para gestionar la información del proceso de gestión de matriculas al Programa de Formación Doctoral de la Universidad de Matanzas

En resumen, se puede afirmar que el software creado cumple con las expectativas en términos de funcionalidad y calidad. Se presenta como una herramienta atractiva, fácil de usar, segura e intuitiva al trabajar con la información del Programa de Formación Doctoral en la Universidad de Matanzas.

Recomendaciones

Luego de la obtención de todas las facilidades y mejoras que ofrece este producto al proceso de Gestión de Matriculas al Programa de Formación Doctoral de la Universidad de Matanzas, se proponen las siguientes recomendaciones:

- Diseño de un logotipo atractivo y representativo para la aplicación
- Incorporar un sistema de notificaciones vía email que sea válido para Cuba
- Expandir el modulo de gestión de gráficos para otros datos de gestión

Bibliografía

- Amazon. (s. f.). *¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS*. Recuperado 28 de noviembre de 2023, de <https://aws.amazon.com/es/what-is/api/>
- Araneda, P. (2022). *Base de Datos*. <https://bookdown.org/paranedagarcia/database/>
- Cillero Manuel. (s. f.). *Diagrama de colaboración - manuel.cillero.es*. Recuperado 27 de noviembre de 2023, de <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-interaccion/diagrama-de-colaboracion/>
- Coppola Maria. (s. f.). *Qué es HTML y cómo utilizarlo (guía para principiantes)*. Recuperado 23 de noviembre de 2023, de <https://blog.hubspot.es/website/html>
- Creately. (2022). *Tutorial de Diagrama de Despliegue | ¿Qué es un Diagrama de Despliegue*. <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>
- DanySoft. (2014). *Navicat | Administración de Bases de Datos*. <https://www.danysoft.com/navicat-administracion-de-bases-de-datos/>
- data.ai. (s. f.). *PowerSchool Mobile's Play Store App Ranking & Store Data from data.ai*. Recuperado 30 de noviembre de 2023, de <https://www.data.ai/es/apps/google-play/app/20600004726239/?alreadyMoreDescription=true>
- Distillery. (2022). *¿Qué son las pruebas de seguridad de las aplicaciones y cómo funcionan? - Distillery*. <https://distillery.com/es/blog/que-son-las-pruebas-de-seguridad-de-las-aplicaciones-y-como-funcionan/>
- Eguíluz Pérez, J. (s. f.). *Introducción a CSS*. www.librosweb.es
- Eguíluz Pérez, J. (2009). *Introducción a JavaScript*. www.librosweb.es
- Encora. (2021). *Cómo realizar pruebas automatizadas con Postman*. <https://www.encora.com/es/blog/como-realizar-pruebas-automatizadas-con-postman>
- Escuela de Postgrado de Ingeniería y Arquitectura de Madrid. (2020). *Aplicaciones Web: Características, Ventajas y Ejemplos*. <https://postgradoingenieria.com/que-son-aplicaciones-web/>
- Germán, H., & Martínez, C. (2021). *VUE (CONSTRUCCIÓN DE INTERFACES)*.

- IBM. (2021). *Planificación de proyectos - Documentación de IBM*.
<https://www.ibm.com/docs/es/rational-clearquest/9.0.1?topic=settings-project-planning>
- Bouihia, B., & Bahaja, M. (2019). *An UML to OWL based approach for extracting Moodle's Ont* *Second International Conference on Intelligent Computing in Data Sciences (ICDS 2018) Second International Conference on Intelligent Computing in Data Sciences (ICDS 2018)*,
- Cvetković, J., & Cvetković, M. (2019). *Evaluation of UML diagrams for test cases generation: Case study on depression of internet addiction*. *Physica A*, 525, 1351–1359
<https://doi.org/10.1016/j.physa.2019.03.101>
- Ionos. (2019). *¿Qué es una web app y qué clases hay?* - IONOS.
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-una-web-app-y-que-clases-hay/>
- Lucidchart. (s. f.). *¿Qué es la notación de modelado de procesos de negocio?* | Lucidchart. Recuperado 27 de noviembre de 2023, de <https://www.lucidchart.com/pages/es/que-es-la-notacion-de-modelado-de-procesos-de-negocio>
- Luis Hernández Yáñez. (2014). *Fundamentos de la Programación*.
- Microsoft. (s. f.). *Visual Studio: IDE y Editor de código para desarrolladores de software y Teams*. Recuperado 24 de noviembre de 2023, de <https://visualstudio.microsoft.com/es/>
- OIT Support. (2022). *What Is Banner?* <https://support.stedwards.edu/s/article/what-is-banner>
- Ortega, L. (s. f.). *Metodología RUP: ¿Qué es, cuál es su objetivo y cómo se utiliza?* Recuperado 28 de noviembre de 2023, de <https://lean-management.site/rup/>
- Paus, L. (2015). *Cómo auditar la seguridad de tu sitio web con Vega*. <https://www.welivesecurity.com/la-es/2015/03/03/como-auditar-la-seguridad-sitio-web-vega/>
- Praxis. (s. f.). *Análisis de punto función - Praxis Framework*. Recuperado 29 de noviembre de 2023, de <https://www.praxisframework.org/es/library/function-point-analysis>
- Romero, Y. F., & Díaz González, Y. (2012). *Patrón Modelo-Vista-Controlador*. *Revista Telemática*, 11(1), 47–57. <http://revistatelematica.cujae.edu.cu/index.php/tele>
- Stack Overflow. (s. f.). *Node.js #node.js*.
- Tamushi. (2022). *Pruebas funcionales de software que debes considerar*. <https://www.testingit.com.mx/blog/pruebas-funcionales-de-software>

UNEWEB. (s. f.). MARIADB NIVEL I.

Universidad Rey Juan Carlos. (2022). Express.js.

BOUIHIA, B. AND M. BAHAJA 2019. An UML to OWL based approach for extracting Moodle's Ont. In *Proceedings of the Second International Conference on Intelligent Computing in Data Sciences (ICDS 2018) Second International Conference on Intelligent Computing in Data Sciences (ICDS 2018)* 2019 *Procedia Computer*, 313-322.

CVETKOVIĆ, J. AND M. CVETKOVIĆ Evaluation of UML diagrams for test cases generation: Case study on depression of internet addiction. *Physica A*, 2019, 525, 1351–1359

GONZÁLEZ HERNÁNDEZ, W. La competencia depurar sistemas informáticos en la Carrera de Ingeniería Informática de la Universidad de Matanzas. *ReiDoCrea*, 2019, 8(11), 119-132.

OZKAYA, M. AND F. ERATA A survey on the practical use of UML for different software architecture viewpoints. *Information and Software Technology*, 2020, 121, 1-27.

ZHI, Q., L. GONG, J. REN, M. LIU, et al. Element quality indicator: A quality assessment and defect detection method for software requirement specification. *Heliyon*, 2023/05/01/ 2023, 9(5), e16469.

ZUMBANA SANTAMARIA, J. A. El marketing digital para la reactivación del turismo cultural durante el covid 19. *Universidad Técnica de Ambato*, 2021.

Anexos

Anexo 1

The screenshot shows the Ellucian recruitment dashboard. At the top, there is a navigation bar with the Ellucian logo and a search bar. Below the navigation bar, there are several tabs: Home, Sales, Perform, Career, Recruit, IT, Content, Reports, Admin, Analytics, and Org Chart. The main content area is titled 'Candidates' and features a summary section with four large numbers: 11 TOTAL CANDIDATES, 2 INTERVIEW, 3 OFFER LETTER, and 6 REVIEW. Below this summary, there is a table of candidates with columns for Candidate, Location, Requirement, Status, Type, Source, and Role. The table lists three candidates: Rick Hubbard, Wanda Medina, and Todd Ramms. Each candidate entry includes their name, contact information, location, requirement, status, and a 'View Feedback' button.

Ilustración 50 . Software Banner

Anexo 2

The screenshot shows the PowerSchool SIS dashboard. At the top, there is a navigation bar with the PowerSchool logo and a search bar. Below the navigation bar, there are several tabs: Home, Attendance, Enrollment Summary, Health Management, Importing & Exporting, Incident Management, Master Schedule, Search Attachments, Special Functions, Special Programs, Teacher Schedules, Reports, System Reports, ReportWorks, People, Student Search, Staff Search, Contact Search, Enroll Student, Create Self, Create Contact, Setup, School, System. The main content area is titled 'Start Page' and features a search bar with 'Students' selected. Below the search bar, there is a table of student search results with columns for Student, Student Number, Grade Level, and Date of Birth. The table lists ten students: Thomas, Angel BD, Thomas, Anli CB, Thomas, Ireland BD, Thomas, Jalen CB, Thomas, Lauryn CC, Thomas, Mia BA, Thomas, Nyaha BA, Thomey, Taki GC, and Thompson, Jared BC. To the right of the search results, there is a 'Quick Data' section with a donut chart showing 'Attendance Taken' at 88% and a bar chart showing 'At Risk' students.

Ilustración 51. Software PowerSchool