

UNIVERSIDAD METROPOLITANA



FACULTAD DE INGENIERIAS

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

SEDE QUITO

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS DE INFORMACIÓN**

TEMA:

**IMPLEMENTACIÓN DE UN SISTEMA QUE GESTIONE LOS COBROS DE
MATRICULAS Y PENSIONES DE LOS ESTUDIANTES DE LA UNIDAD
EDUCATIVA "GENOVEVA GERMAN"**

AUTOR:

SEGUNDO VINICIO ZURITA SIMBAÑA

ASESOR:

ING, SYLVIA LLUMIQUINGA

QUITO – 2025

CERTIFICADO DEL TUTOR

Ing. SYLVIA LLUMIQUINGA QUISPE, en calidad de asesor del trabajo de graduación o titulación.

CERTIFICA

Que el trabajo de graduación ha sido desarrollado de manera autónoma por el señor **Zurita Simbaña Segundo Vinicio**, portador de la cédula de identidad No 172526508-4 con el tema: **IMPLEMENTACIÓN DE UN SISTEMA QUE GESTIONE LOS COBROS DE MATRICULAS Y PENSIONES DE LOS ESTUDIANTES DE LA UNIDAD EDUCATIVA "GENOVEVA GERMAN**, el mencionado señor ha cumplido con todos los requisitos legales exigidos.

Es todo cuanto puedo decir en honor a la verdad, facultando al interesado hacer uso de la presente.

Atentamente

Ing. Sylvia LLumiquinga Quispe MSc.

TUTOR DE TRABAJO DE TITULACIÓN

CERTIFICADO DE AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, **Zurita Simbaña Segundo Vinicio**, estudiante de la Universidad Metropolitana del Ecuador "UMET", Ingeniería en sistemas de información, declaro en forma libre y voluntaria que el presente trabajo de titulación que versa sobre: **IMPLEMENTACIÓN DE UN SISTEMA QUE GESTIONE LOS COBROS DE MATRICULAS Y PENSIONES DE LOS ESTUDIANTES DE LA UNIDAD EDUCATIVA "GENOVEVA GERMAN** y las experiencias vertidas en las misma, son autoría de la compareciente, las cuales se han realizado en base a la recopilación de bibliografía, consultas de internet y consultas de campo.

En consecuencia, asumo la responsabilidad de la originalidad de la misma y el ciudadano al referirme a las fuentes bibliográficas respectivas para fundamentar el contenido expuesto.

Atentamente,

Zurita Simbaña Segundo Vinicio

C.C. 1725265084

AUTOR

CESIÓN DE DERECHOS DEL AUTOR

Yo, **Zurita Simbaña Segundo Vinicio**, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación, **IMPLEMENTACIÓN DE UN SISTEMA QUE GESTIONE LOS COBROS DE MATRICULAS Y PENSIONES DE LOS ESTUDIANTES DE LA UNIDAD EDUCATIVA "GENOVEVA GERMAN**, modalidad Proyecto de Investigación de conformidad con el Art. 114 del CODIGO ORGANICO DE LA ECONOMIA SOCIAL DE LOS CONOCIMIETOS, CREATIVIDAD E INNOVACIÓN, cedo a favor de la Universidad Metropolitana del Ecuador una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Metropolitana del Ecuador para que realice la digitación y publicación de este trabajo de titulación en el repositorio virtual, de conformidad a los dispuesto en el Art. 144 de la ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Atentamente

Zurita Simbaña Segundo Vinicio

C.C. 1725265084

DEDICATORIA

Dedico este proyecto con todo mi corazón a mis padres Segundo y Verónica por el apoyo incondicional y por todos sus consejos, valores y principios que me han inculcado desde niño, a mi hermano David porque siempre ha estado conmigo apoyándome en los buenos y malos momentos. A mi novia Lorena por estar siempre en esos momentos difíciles brindándome su amor, paciencia y comprensión. A mi tía Miriam porque siempre está dispuesta a escucharme y ayudarme en cualquier momento. A mis abuelitos Alfredo y Rosario que siempre han sido un ejemplo de vida, gracias a todos por sus fuerzas y confianza puedo ver alcanzar una meta más, Hoy celebro este logro con ustedes, pues cada paso ha sido posible gracias a su amor y motivación.

AGRADECIMIENTO

Agradezco a mi Dios, a la Virgen del Quinche y al niño Jesús, porque su bondad no tiene fin, cuando caigo y me ponen a prueba, aprendo de mis errores y me doy cuenta de que me los ponen en frente mío para que mejore como ser humano y crezca de diversas maneras.

Un agradecimiento muy noble y sincero a mis queridos y amados padres Segundo y Verónica, quienes han sido las personas que siempre me apoyaron moral y económicamente en todo el transcurso de mi vida, dándome ejemplo de superación, humildad y sacrificio, por haberme forjado como la persona que soy en la actualidad.

A mi hermano David, por su apoyo incondicional durante mi camino hacia la titulación y enseñarme que si se puede.

A mi novia Lorena por todo su amor y confianza, por ser un pilar de fortaleza y superación, tu ayuda ha sido muy fundamental, este proyecto no fue fácil, pero estuviste motivándome y ayudándome hasta donde tus alcances lo permitían.

A mi tía Miriam y mis abuelitos Alfredo y Rosario que con sus regaños, consejos y sabiduría son un ejemplo claro en mi vida y me motivaron a seguir adelante en mis estudios.

Dar las gracias a mi noble Universidad Metropolitana del Ecuador, por ser la Institución donde me perfile para obtener conocimientos académicos y formarme como un profesional en la carrera de Ingeniería en Sistemas de información. De igual manera quiero agradecer a mi Tutora MSC. Sylvia LLumiquinga por haber aceptado ser la tutora de esta tesis, a pesar del tiempo y de la situación en que solicite su apoyo y que con su gran calidad humana y profesional dedico su tiempo a esta tesis.

Gracias.

ÍNDICE

| | |
|--|-------------|
| CERTIFICADO DEL TUTOR | II |
| CERTIFICADO DE AUTORÍA DEL TRABAJO DE TITULACIÓN | III |
| CESIÓN DE DERECHOS DEL AUTOR | IV |
| DEDICATORIA..... | V |
| AGRADECIMIENTO | VI |
| RESUMEN..... | XII |
| ABSTRACT | XIII |
| INTRODUCCIÓN | 1 |
| CAPITULO I | 9 |
| 1. FUNDAMENTACIÓN TEÓRICA..... | 9 |
| 1.1. Descripción de la Unidad Educativa Genoveva German | 9 |
| 1.1.1. Narración de la historia de la Unidad Educativa Genoveva German | 9 |
| 1.1.2. Visión..... | 10 |
| 1.1.3. Misión | 10 |
| 1.1.4. Principios Educativos | 10 |
| 1.1.5. Objetivo de estudio | 10 |
| 1.1.6. Flujo actual de procesos | 11 |
| 1.2. Análisis crítico de la ejecución de procesos | 15 |
| 1.3. Proceso a automatizar en la Unidad Educativa “Genoveva German” | 15 |
| 1.4. Sistemas automatizados existentes vinculados con el campo de acción... | 16 |
| 1.5. Metodologías y herramientas de desarrollo. | 17 |
| 1.5.1. Metodologías tradicionales para el desarrollo de software | 17 |
| 1.5.2. Metodología de RUP..... | 18 |
| 1.5.3. UML..... | 18 |
| 1.5.4. Herramientas de desarrollo..... | 19 |
| 1.5.5. C# (.NET 8) | 20 |
| 1.5.6. WinUI 3..... | 20 |
| 1.5.7. MySQL (MariaDB vía XAMPP) | 21 |
| 1.5.8. Entity Framework Core (Pomelo MySQL) | 21 |
| 1.5.9. Microsoft.Extensions.DependencyInjection | 22 |
| 1.5.10. PhpMyAdmin (XAMPP) | 22 |
| 1.5.11. SessionService | 23 |
| 1.5.12. NavigationView (WinUI 3)..... | 24 |

| | | |
|--------------------|---|----|
| 1.6. | Tendencias y tecnologías actual..... | 28 |
| 1.7. | Conclusiones del capítulo | 29 |
| CAPÍTULO II..... | | 31 |
| 2. | MODELO DE NEGOCIO, REQUISITOS FUNCIONALES Y MODELOS DEL SISTEMA..... | 31 |
| 2.1. | Modelo del Negocio y Requisitos | 31 |
| 2.2. | Diagrama de caso de uso del negocio | 33 |
| 2.2.1. | Actores y casos de uso del sistema..... | 33 |
| 2.2.2. | Requisitos..... | 34 |
| 2.3. | Requisitos Funcionales del sistema | 35 |
| 2.4. | Requisitos no funcionales..... | 38 |
| 2.5. | Especificación de casos de uso y diagramas de actividades | 38 |
| 2.6. | Modelo de Casos de uso del negocio | 40 |
| 2.6.1. | Actores del Negocio | 40 |
| 2.7. | Diagrama general de casos de uso del negocio | 44 |
| 2.7.1. | Modelo de objetos del negocio | 46 |
| 2.7.2. | Reglas del negocio..... | 48 |
| 2.8. | Conclusiones del capítulo | 50 |
| CAPITULO III | | 51 |
| 3. | DESCRIPCIÓN DE LA PROPUESTA DE SOLUCION | 51 |
| 3.1. | Introducción | 51 |
| 3.2. | Análisis del diseño del sistema..... | 51 |
| 3.2.1. | Modelo vista controlador (MVC) | 51 |
| 3.2.2. | Interacción de los componentes | 54 |
| 3.2.3. | Diagrama de clases del diseño..... | 54 |
| 3.2.4. | Diagrama de secuencia | 55 |
| 3.2.5. | Modelo lógico de la base de datos | 56 |
| 3.2.6. | Diagrama de componentes..... | 57 |
| 3.2.7. | Diagrama de despliegue | 59 |
| 3.3. | Conclusión | 60 |
| CAPÍTULO IV | | 62 |
| 4. | PRUEBA Y ANÁLISIS DE FACTIBILIDAD | 62 |
| 4.1. | Planificación basada en el método de estimación por puntos de caso de uso | 62 |
| 4.2. | Definición de factores de complejidad..... | 63 |
| 4.3. | Complejidad según tipo de actor | 63 |

| | |
|--|----|
| 4.3.1. Factores técnicos | 64 |
| 4.3.2. Factores ambientales..... | 65 |
| 4.3.3. Métricas para medir para realizar la estimación | 66 |
| 4.4. Porcentaje de actividades por ciclo de vida | 67 |
| 4.5. Casos de pruebas..... | 68 |
| 4.6. Análisis de Pruebas de Caja Negra..... | 70 |
| 4.7. Conclusiones del capítulo | 72 |
| CONCLUSIONES..... | 74 |
| RECOMENDACIONES | 76 |
| BIBLIOGRAFÍA | 77 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Flujo de proceso | 11 |
| Tabla 2 Comparación de Herramientas Informáticas para la Gestión de Cobros | 17 |
| Tabla 3 Comparativa de las Características de las Tecnología | 25 |
| Tabla 4 Comparación de metodologías aplicadas al proyecto | 29 |
| Tabla 5 Especificaciones de los actores del sistema | 34 |
| Tabla 6 Descripción del trabajador del negocio | 34 |
| Tabla 7 Módulo de autenticación y control de acceso | 35 |
| Tabla 8 Módulo de gestión de usuarios, roles y permisos | 36 |
| Tabla 9 Módulo académico | 36 |
| Tabla 10 Módulo financiero | 37 |
| Tabla 11 Módulo de reportes | 37 |
| Tabla 12 Casos de uso: Gestión de estudiantes | 39 |
| Tabla 13 Historia de Usuario 1 | 41 |
| Tabla 14 Historia de usuario 2 | 41 |
| Tabla 15 Historia de usuario 3 | 42 |
| Tabla 16 Historia de usuario 4 | 42 |
| Tabla 17 Product Backlog | 44 |
| Tabla 18 Objetos del negocio | 47 |
| Tabla 19 Descripción de Reglas del Negocio | 49 |
| Tabla 20 Factores de complejidad de requisitos | 63 |
| Tabla 21 Clasificación de actores según complejidad | 63 |
| Tabla 22 Factores Técnicos | 64 |
| Tabla 23 Factores ambientales del entorno de desarrollo | 65 |
| Tabla 24 Escala de ponderación | 66 |
| Tabla 25 Ciclo de vida | 67 |
| Tabla 26 Caso de prueba 1 | 69 |
| Tabla 27 Caso de pruebas 2 | 69 |
| Tabla 28 Caso de pruebas 3 | 70 |
| Tabla 29 Pruebas caja negra | 71 |

ÍNDICE DE ILUSTRACIONES

| | |
|--|----|
| Figura 1 Flujo de pago de matrícula | 12 |
| Figura 2 Componentes de un caso de uso | 32 |
| Figura 3 Diagrama caso de uso del negocio | 33 |
| Figura 4 Diagrama de caso de uso del sistema-autenticación y acceso a funciones | 46 |
| Figura 5 Diagrama de caso de uso- Gestión de usuarios | 46 |
| Figura 6 Diagrama de caso de uso- registro de estudiantes | 46 |
| Figura 8 Arquitectura del sistema basado en el patrón MVC | 53 |
| Figura 9 Diagrama de clases | 55 |
| Figura 10 Diagrama de secuencia de registro de pago..... | 56 |
| Figura 11 Modelo lógico de la base de datos | 57 |
| Figura 12 Diagrama de componentes..... | 58 |
| Figura 13 Diagrama de despliegue | 60 |

RESUMEN

En la Unidad Educativa Genoveva Germán, los procesos de recolección para la matrícula y la matrícula se realizaron manualmente utilizando hojas de cálculo y archivos físicos, lo que llevó a un flujo de trabajo desorganizado y propenso a errores, así como a un servicio lento. Este trabajo presenta el diseño e implementación de un sistema de escritorio que automatiza estos procesos la solución se implementó en C# (.NET 8) con WinUI 3 para la interfaz, MySQL (MariaDB a través de XAMPP) como sistema de gestión de bases de datos, y con Entity Framework Core (Pomelo MySQL) como el ORM utilizando inyección de dependencia para el control de sesiones. La arquitectura sigue el patrón MVVM y fue construida usando la metodología RUP, el proyecto también involucra la recopilación de requisitos, modelado UML, definición de casos de uso e historias de usuario, junto con pruebas funcionales (caja negra). El sistema integra autenticación y control de acceso; gestión de usuarios, estudiantes, cursos y clases paralelas; emisión y control de recibos de pago (matrícula y colegiatura); y reportes financieros. La evidencia muestra una reducción en errores, mejora en el tiempo de servicio, mayor seguridad de la información y trazabilidad, mejor control de la morosidad en los pagos y fortalecimiento de la toma de decisiones institucional.

Palabras clave: Sistema, Gestión de cobros, Matrículas y pensiones, Automatización, Unidad Educativa Genoveva Germán, RUP, MVC

ABSTRACT

At the Genoveva Germán Educational Unit, the collection processes for enrollment and registration were performed manually using spreadsheets and physical files, leading to a disorganized and error-prone workflow, as well as slow service. This work presents the design and implementation of a desktop system that automates these processes. The solution was implemented in C# (.NET 8) with WinUI 3 for the interface, MySQL (MariaDB via XAMPP) as the database management system, and Entity Framework Core (Pomelo MySQL) as the ORM using dependency injection for session control. The architecture follows the MVVM pattern and was built using the RUP methodology. The project also involves requirements gathering, UML modeling, use case and user story definition, along with functional (black-box) testing.

The system integrates authentication and access control; management of users, students, courses, and parallel classes; issuance and control of payment receipts (tuition and fees); and financial reporting. The evidence shows a reduction in errors, improved service time, greater information security and traceability, better control of late payments, and a strengthening of institutional decision-making.

Keywords: System, Collection Management, Tuition and Fees, Automation, Genoveva Germán Educational Unit, RUP, MVC

INTRODUCCIÓN

En la actualidad, la tecnología forma parte de la práctica en todos los aspectos de nuestra vida diaria, llegando a convertirse en un elemento indispensable. En el sector educativo, se ha consolidado como un recurso fundamental para facilitar el aprendizaje. Al mismo tiempo, la continua incorporación de sistemas de comunicación motiva a empresas, organizaciones, instituciones e industrias a automatizar sus operaciones, lo que les permite optimizar su rendimiento y eficacia laboral

La Unidad Educativa GENOVEVA GERMAN, ubicada en el Cantón Mejía, sector La Tesalia, hacienda Puichig, requiere implementar un sistema que automatice sus procesos internos. Una de las principales demandas es la gestión del cobro de matrículas y pensiones para todos los estudiantes.

La incorporación de esta automatización permitirá agilizar la respuesta, además de facilitar la presentación organizada y en línea de los informes diarios de recaudación de pensiones.

En la actualidad, la recolección de inscripciones y cuotas se realiza manualmente en el caso de una institución educativa, el uso de la tecnología para automatizar procesos comerciales va más allá de usar una computadora para revisar correos electrónicos y crear documentos de word significa que se puede instalar una computadora se puede capacitar para su operación y se puede utilizar a diario para hacer más fáciles las tareas tediosas que maneja un solo individuo en una institución educativa independientemente del rol desempeñado en la institución, (Narváez & Triviño, 2011)

Formulación del problema

La Unidad Educativa Genoveva Germán enfrenta importantes dificultades en la administración de la recaudación de pagos por inscripciones y mensualidades debido al uso de métodos manuales para registrar las transacciones y elaborar los reportes. En la actualidad, se emplean hojas de cálculo y registros físicos, lo que provoca desorden y dificulta la consulta rápida de la información, aumentando el riesgo de errores y omisiones. El manejo manual de los cobros y pagos pendientes genera demoras en la atención a los padres de familia, quienes deben esperar mientras el personal busca la información en archivos dispersos. Asimismo, la

actualización de tarifas y la elaboración de reportes sobre pagos realizados y pendientes es lenta, impidiendo un control eficaz y en tiempo real de las finanzas del centro educativo.

Esta problemática afecta la eficiencia administrativa y la calidad del servicio, lo que podría poner en riesgo el desarrollo y la sostenibilidad de la institución. Para solucionar estos inconvenientes, es fundamental implementar un sistema que automatice los procesos de cobro y gestión de datos, garantizando la integridad de la información, agilizando la atención a los padres de familia y facilitando la generación de informes precisos y oportunos.

La principal pregunta que surge es: ¿Cómo mejorar la gestión y control de los cobros de matrícula y pensiones en la Unidad Educativa Genoveva Germán? Con la implementación de un sistema, se pretende optimizar los procesos administrativos, mejorar la organización y control de la información, y ofrecer un mejor servicio a los representantes y padres de familia.

Delimitación del problema

La situación problemática planteada afecta a la Unidad Educativa Genoveva Germán, ubicada en el Cantón Mejía, provincia de Pichincha. La situación dificulta la labor administrativa, especialmente para los empleados responsables del cobro de matrículas y pensiones, así como para los padres y representantes de los alumnos.

La carencia de un sistema automatizado para el manejo de estos cobros provoca desorden, pérdida de tiempo y errores en el control de la información financiera. Estas fallas afectan la eficiencia en los procesos administrativos y la calidad del servicio ofrecido a las familias, lo que repercute negativamente en la imagen de la institución.

Identificación de la línea de investigación.

Objeto de estudio

Gestión de cobros de matrículas y pensiones en instituciones educativas.

Campo de acción

Sistema de gestión de cobros y administración financiera basado en tecnologías web.

Objetivos

A partir del problema planteado en el texto Eanterior para la elaboración del proyecto se propone el siguiente objetivo general

Objetivo General

Implementar un sistema de gestión de cobros en matrículas y pensiones en la Unidad Educativa GENOVEVA GERMAN para la mejora de la eficiencia administrativa y el control financiero.

Objetivos Específicos

1. Analizar los requerimientos necesarios para la gestión de cobros en la Unidad Educativa.
2. Realizar el estudio del arte sobre las metodologías apropiadas y herramientas tecnológicas que permitan el desarrollo del software.
3. Diseñar la base de datos identificando los requisitos funcionales y no funcionales del sistema, los actores del sistema y los casos de uso.
4. Desarrollar el software para la gestión y control del cobro de las pensiones de los estudiantes de la Institución.
5. Validar el correcto funcionamiento del sistema, mediante pruebas de software como pruebas de caja negra u otras.

Justificación de la investigación

La Unidad Educativa Genoveva Germán enfrenta problemas en la gestión de los cobros de matrículas y pensiones, principalmente debido a la dependencia de procesos manuales y registros dispersos que dificultan la organización, actualización y seguimiento de los pagos. Este mal manejo de la información afecta tanto a los administradores de la institución como a los padres de familia, generando demoras en la atención y la posibilidad de errores en los registros financieros. Por lo tanto, la implementación de un sistema se justifica por las siguientes razones:

- Automatización de los cobros: El sistema permitirá registrar y controlar de manera eficiente los pagos de matrículas y pensiones, eliminando el uso de hojas de cálculo manuales y registros físicos.
- Actualización en tiempo real: Se actualizarán automáticamente los pagos

realizados y los pendientes, garantizando que la información esté siempre disponible y actualizada tanto para la administración como para los padres de familia.

- Generación de reportes automáticos: El reporte que informe acerca de los pagos, las deudas pendientes, así como la situación financiera de la institución tendrá que ser exacto. Con esta herramienta la administración tendrá más facilidad en la toma de decisiones en pro de la transparencia de la institución.
- Mejora en la atención al cliente: Al contar con la herramienta que automatiza el proceso de cobranza el tiempo de espera que habían incurrido los padres de familia y representantes se elimina parcialmente. Al mejorar la atención al cliente se fortalece el vínculo con la comunidad educativa.
- Facilita la toma de decisiones: La administración contará con información precisa y actualizada en tiempo real sobre los pagos lo que agiliza la toma de decisiones en la gestión financiera.

Con la implementación de este sistema, la Unidad Educativa Genoveva Germán podrá mejorar su eficiencia operativa, optimizar los recursos administrativos y brindar un mejor servicio a sus estudiantes y sus familias.

Viabilidad de la investigación

Viabilidad técnica

El sistema a desarrollar permitirá reducir los errores en los registros de los cobros de matrículas y pensiones, eliminando la duplicidad de datos y mejorando el control de los pagos. Utilizando tecnologías de código abierto como PHP, MySQL, JavaScript, CSS y XAMPP, se logrará una integración eficiente de la información, optimizando los procesos administrativos de la Unidad Educativa Genoveva Germán.

La elección de estas tecnologías no solo responde a su bajo costo y amplia disponibilidad, sino también a su alta compatibilidad con múltiples entornos operativos, lo que facilita la instalación y el despliegue del sistema sin requerir infraestructura especializada. Además, su arquitectura modular permitirá la escalabilidad del software, incorporando en el futuro nuevas funciones como control de asistencia, reportes avanzados o integración con pasarelas de pago, (Espinoza &

Mendez, 2022).

En términos de seguridad, MySQL ofrece mecanismos robustos de autenticación, cifrado y control de accesos, que garantizan la integridad y confidencialidad de la información financiera y académica (Roman Honores & Ormaza Andrade, 2022). Del mismo modo, el uso de tecnologías web ampliamente soportadas asegura que el mantenimiento y la actualización del sistema puedan ser ejecutados por personal técnico con conocimientos estándar, minimizando la dependencia de proveedores externos.

Viabilidad teórica

El marco teórico seleccionado sustenta el desarrollo del sistema, proporcionando los fundamentos necesarios para optimizar la gestión de cobros y mejorar la eficiencia administrativa. Las metodologías y herramientas de desarrollo empleadas permiten diseñar una solución adecuada a las necesidades de la institución, resolviendo de manera efectiva los problemas planteados en el proyecto.

Las metodologías de desarrollo adoptadas, como el Rational Unified Process (RUP) y el uso de UML para la modelación de procesos y estructuras, aportan una base sólida y ordenada para la planificación, diseño e implementación del sistema, (Puerto Moreno , 2020). Estas metodologías facilitan la identificación clara de los requisitos funcionales y no funcionales, así como la estructuración de casos de uso que reflejan fielmente las necesidades reales de la institución.

Asimismo, la elección de herramientas tecnológicas como C# (.NET 8), WinUI 3 y MySQL/MariaDB, junto con la aplicación de patrones de diseño como MVC y MVVM, responde a criterios de robustez, escalabilidad y mantenimiento a largo plazo. Esta selección no es arbitraria, sino que se apoya en estudios y documentación técnica que avalan su rendimiento y seguridad en entornos similares, (Camayo Ávila & Maita Diaz, 2025).

En consecuencia, la combinación de un marco teórico bien fundamentado, una metodología de desarrollo estructurada y herramientas tecnológicas modernas garantiza que el sistema propuesto no solo resolverá los problemas actuales, sino que también podrá adaptarse a futuras necesidades, asegurando su vigencia y relevancia en el tiempo.

Tipo de investigación

El procesamiento de este trabajo se desarrollará con una investigación descriptiva que permite documentar el contexto en el que se desarrollará el sistema, detallando características de los usuarios, procesos actuales y necesidades específicas de acuerdo a las normas del estudio de campo, se tuvo la oportunidad de trabajar en el lugar de investigación para obtener información necesaria que se realizan en el departamento de colecturía. Se utilizó un levantamiento bibliográfico en donde permitió describir y sustentar el detalle de los procesos utilizados y sustentar en el marco teórico, (Esquirol-Causa & Sánchez-Aldeguer, 2020).

Además, se complementó el enfoque descriptivo con elementos de investigación aplicada, orientados a transformar los hallazgos en una solución tecnológica concreta que resuelva los problemas identificados. Esto implicó no solo recopilar y analizar datos, sino también interpretarlos con base en referentes teóricos y experiencias previas documentadas en la literatura especializada. De esta manera, se garantiza que la propuesta final no sea únicamente un ejercicio académico, sino una herramienta funcional que pueda implementarse de forma efectiva en la Unidad Educativa Genoveva Germán, generando un impacto positivo en la gestión administrativa y financiera de la institución, (Viladas Jené, 2023).

Alcance del proyecto

El desarrollo del presente sistema busca solventar las necesidades de la Unidad Educativa GENOVEVA GERMAN referente al cobro de las pensiones que cancelan los estudiantes, generación de reportes y así lograr un manejo adecuado del dinero ingresado.

- La funcionalidad del sistema está definida mediante los siguientes módulos:
- Administrar usuarios
- Registro de Usuarios
- Inscripción de los pagos asociados a matrículas
- Inscripción y control de los pagos correspondientes a las pensiones.
- Reportes sobre los valores pendientes de cobro.

Beneficios de la investigación

La investigación permite sustentar el desarrollo de nuevos conocimientos que ayudan a la creación de un sistema para la gestión en cobros de matrícula y pensiones en la Unidad Educativa GENOVEVA GERMAN.

Para la realización e implementación del software se definieron todos los factores que influyen en su elaboración, aplicando y fortaleciendo el aprendizaje adquirido durante la formación en Ingeniería de Software. Este proceso integró conocimientos teóricos y prácticos en áreas como análisis de requisitos, modelado UML, diseño de bases de datos, patrones de arquitectura de software y metodologías de desarrollo, lo que permitió un trabajo profesional y alineado a las mejores prácticas del sector, (Madan & Teitge, 2021)

La implementación del software facilitará y ayudará a automatizar los procesos operativos, optimizando los recursos humanos, materiales y tecnológicos de la institución. Además, permitirá un mayor control y seguimiento de la información financiera, mejorando la exactitud de los registros, reduciendo los tiempos de respuesta y minimizando el margen de error. Estos beneficios se traducirán en una gestión más eficiente, una mejor atención a los padres de familia y un incremento en la transparencia y confiabilidad de los procesos administrativos.

Estructura del documento

El documento está organizado en cuatro capítulos:

En el capítulo 1 encontraremos los procesos, términos esenciales y conceptos claves relacionados con el trabajo de investigación. Donde se indagará los principales sistemas vinculados con la investigación y la necesidad educativa, utilizando herramientas y metodologías necesarias para la ejecución y desarrollo del sistema. Además, incluye el análisis del modelo, las reglas de negocio y la descripción detallada de las especificaciones funcionales y no funcionales importantes para abordar la problemática planteada.

En el capítulo II se lleva a cabo el análisis del negocio, definiendo tanto los requisitos funcionales como no funcionales del sistema. Asimismo, se identifican los actores involucrados y se detallan los casos de uso. Además, se presentan algunos de los diagramas de actividades más relevantes diseñados para el sistema.

En el capítulo III se expone con detalle la propuesta de solución para el proyecto. Se presentan los diagramas de clases y los principales diagramas de secuencia, además del diseño de la base de datos, donde el esquema entidad-relación es transformado al esquema relacional. Finalmente, se incluyen los aspectos clave sobre la implementación del sistema

En el capítulo IV se evalúa la viabilidad de la construcción del sistema utilizando el método de estimación de esfuerzo. Se detallan los pasos necesarios para su implementación y la preparación adecuada para su operación en la Unidad Educativa. Además, se presentan algunas de las interfaces gráficas más relevantes y se llevan a cabo pruebas de caja negra, pruebas funcionales, de validación de datos y de usabilidad para verificar la efectividad del sistema en condiciones reales de uso. Finalmente, el capítulo concluye con las consideraciones finales tras evidenciar los beneficios que aporta el desarrollo del sistema.

CAPITULO I

1. FUNDAMENTACIÓN TEÓRICA

El capítulo uno recoge la descripción del contexto en el desarrollo del sistema propuesto en la tesis, se detallarán los conceptos básicos y esenciales relacionados con la investigación. Además, se describe las metodologías, herramientas tecnológicas que se utilizaran en el desarrollo del proyecto.

1.1. Descripción de la Unidad Educativa Genoveva German

En este epígrafe se redacta las actividades que se realiza en la Unidad Educativa GENOVEVA GERMAN.

1.1.1. Narración de la historia de la Unidad Educativa Genoveva German

Según documentos existentes, la institución educativa empieza a funcionar el 20 de abril de 1978, bajo la denominación de Colegio Agropecuario, mediante acuerdo ministerial No 4566, publicado en el Registro Oficial No 598, del mes de junio del mismo año.

En 1979 inicia el colegio sus actividades con 25 alumnos, correctamente uniformados, siendo su primer Rector el Licenciado Jorge Encalada.

En el periodo lectivo 1979 - 1980 funciono en el segundo piso de una casa de construcción antigua situada frente al parque principal de Machachi, donde también funcionaba, por aquel tiempo, la escuela católica Mariano Negrete.

En el mes de julio del año de 1985, egresa la primera promoción de bachilleres, lo que constituye un gran acontecimiento para la sociedad de Machachi y el cantón Mejía. El comentario general era de alegría, positivismo y esperanza para los jóvenes que querían estudiar una profesión agrícola y pecuaria, como era su aspiración.

En el 2023 seguimos sembrando el mañana para las nuevas generaciones que quieren asumir a la agricultura y ganadería como la profesión.

La oferta educativa que ofreces es:

- Educación General Básica Superior (8vo. –9no. –10mo)
- Nivel: Bachillerato Técnico en Producción Agropecuaria (1ro. –2do. – 3ro)

1.1.2. Visión

Ser una institución técnica reconocida por su excelencia conforme a los estándares establecidos por el Ministerio de Educación, ofreciendo una formación integral, humanista, inclusiva e innovadora, fundamentada en principios y valores católicos.

1.1.3. Misión

Somos una institución Educativa Católica Diocesana con trayectoria, que ofrece una educación de alta calidad y calidez humana. Proporcionamos a nuestros estudiantes una formación integral, utilizando técnicas innovadoras, metodologías alternativas y docentes capacitados, con el objetivo de fortalecer la identidad ecuatoriana en los ámbitos agrícola y pecuario, así como promover los valores de la iglesia y la excelencia académica.

1.1.4. Principios Educativos

- **Respeto:** Para que exista un aprendizaje real y una convivencia positiva es necesario respetar las características, los intereses, las inteligencias múltiples y ritmos de desarrollo de cada niño y adolescente. Para ello, la comunidad, la familia y la institución educativa deben ofrecerles un espacio seguro.
- **Democracia:** Somos un Colegio con espacios para el ejercicio de los derechos humanos, promotores de la cultura de paz y de liderazgo bioético que fomenta la autonomía (capacidad de hacer de decidir), el pensamiento crítico y la responsabilidad individual.
- **Educación activa:** Los estudiantes participan activamente en su aprendizaje durante toda la jornada escolar, aplicando los conocimientos adquiridos de manera práctica e interactuando de manera continua.
- **Educación Comunitaria:** Los estudiantes participan activamente en su aprendizaje durante toda la jornada escolar, aplicando los conocimientos adquiridos de manera práctica e interactuando de manera continua.

1.1.5. Objetivo de estudio

La investigación permite sustentar el desarrollo de nuevos conocimientos que contribuyen a la creación de un sistema para la gestión de cobros de matrícula y

pensiones en la Unidad Educativa GENOVEVA GERMAN. Este trabajo no solo aporta una solución tecnológica a una problemática real, sino que también amplía el acervo académico en el campo de los sistemas de información aplicados al sector educativo.

Para la realización e implementación del software se definieron todos los factores que influyen en su elaboración, aplicando y fortaleciendo el aprendizaje adquirido durante la formación en Ingeniería de Software. Este proceso integró conocimientos teóricos y prácticos en áreas como análisis de requisitos, modelado UML, diseño de bases de datos, patrones de arquitectura de software y metodologías de desarrollo, lo que permitió un trabajo profesional y alineado a las mejores prácticas del sector.

La implementación del software facilitará y ayudará a automatizar los procesos operativos, optimizando los recursos humanos, materiales y tecnológicos de la institución. Además, permitirá un mayor control y seguimiento de la información financiera, mejorando la exactitud de los registros, reduciendo los tiempos de respuesta y minimizando el margen de error. Estos beneficios se traducirán en una gestión más eficiente, una mejor atención a los padres de familia y un incremento en la transparencia y confiabilidad de los procesos administrativos.

1.1.6. Flujo actual de procesos

Se describen a continuación los principales procesos que se desarrollan actualmente en el área de colectoría de la Unidad Educativa Genoveva Germán. Estos procesos se ejecutan de forma manual, lo que provoca pérdida de tiempo, desorganización y errores en la gestión de la información.

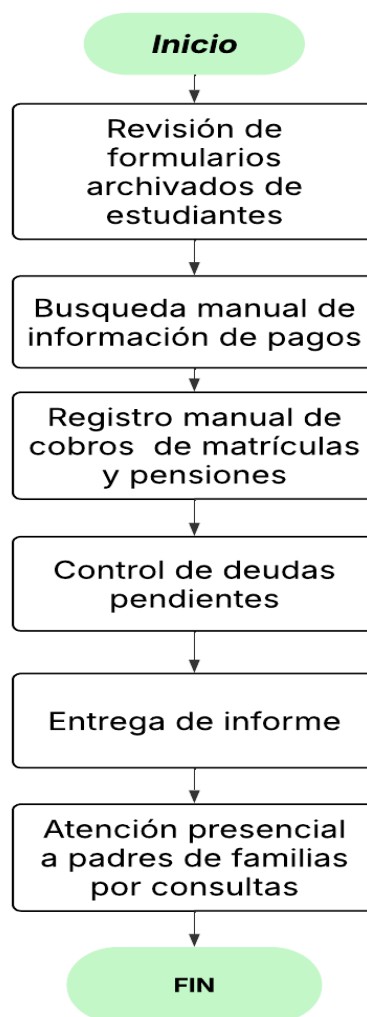
La colectora es el principal actor involucrado en estas actividades, que actualmente carecen de un sistema automatizado para su control y seguimiento. La siguiente tabla muestra los procesos identificados y los actores que intervienen en cada uno.

Tabla 1. Flujo de proceso

| No | PROCESO | Colectora | Representante/Padre de Familia | Rectora |
|----|---|-----------|--------------------------------|---------|
| 1 | Revisión de formularios archivados de estudiantes | X | | |

| | | | | |
|---|---|---|---|---|
| 2 | Búsqueda manual de información de pagos | X | X | |
| 3 | Registro manual de cobros de matrícula y pensiones | X | | |
| 4 | Control de deudas pendientes | X | | |
| 5 | Entrega de informe | X | | X |
| 6 | Atención presencial a padres de familia por consultas | X | X | |

Figura 1 Flujo de pago de matrícula



En esta imagen de flujo se describe el proceso presente sobre el manejo de cobros relacionados a matrículas y pensiones en la unidad educativa Genoveva Germán, evidenciando que todos los procesos aún se hacen de manera manual.

La actividad se inicia con el examen de formularios con registros de estudiantes y la colectora revisando los archivos registrados en papel para ubicar los datos relevantes. Seguido a esto, se hace un registro de búsqueda de documentos, el cual se brindan los documentos en cuestión y de hojas de cálculo con la ayuda de los representantes.

Posterior a ello se hará el registro en libros de cobro para matrículas y pensiones. Esto va seguido de un control de deudas que se hace con documentos en papeles y archivos sin orden. Cuando la información se tiene ordenada en documentos con archivos, el colector realiza un cobro con el informe de saldos y moras el cual se paga y se reporta a la rectora. El proceso culmina con la atención en caja.

1. Revisión de formularios archivados de estudiantes

Proceso actual:

La colectora revisa los formularios físicos archivados de los estudiantes, los cuales contienen información crucial sobre los pagos de matrícula y pensiones. Estos formularios están organizados en carpetas físicas que son almacenadas en un archivo, lo que genera un proceso lento y propenso a errores. La colectora debe buscar manualmente en el archivo cada formulario cuando se necesita consultar la información de algún estudiante, lo que resulta en pérdida de tiempo y dificultad en el acceso rápido a los datos.

2. Búsqueda manual de información de pagos

Tanto la colectora como el representante/padre de familia se ven obligados a buscar la información sobre los pagos de matrícula y pensiones en registros físicos o en hojas de cálculo manualmente actualizadas. No existe un sistema centralizado para verificar rápidamente si un estudiante ha realizado el pago o si tiene alguna deuda pendiente. Este proceso es tedioso ya que la información está dispersa en diferentes lugares (archivos físicos y hojas de cálculo Excel), lo que genera demoras y eleva la probabilidad de incurrir en errores en la información revisada.

3. Registro manual de cobros de matrícula y pensiones

La colectora lleva a cabo el registro manual de los pagos de matrícula y pensiones. Esta tarea se realiza en hojas de cálculo, donde se ingresan los pagos conforme se van realizando. Sin embargo, el uso de estas hojas de cálculo aumenta el riesgo de errores humanos, como la duplicación de registros o la omisión de pagos, lo que afecta la integridad de los datos y la precisión de la información financiera. Además, los registros no son actualizados en tiempo real, lo que dificulta obtener una visión precisa de la situación financiera de la institución.

4. Control de deudas pendientes

La colectora es responsable de llevar un control de las deudas pendientes, pero este proceso se efectúa manualmente con el apoyo de hojas de cálculo o registros en papel. En ocasiones, los datos sobre los estudiantes que aún deben el pago de matrícula o pensión no están actualizados, ya que la información no se actualiza en tiempo real. Esto genera confusión, ya que los padres de familia o representantes deben esperar mientras la colectora busca en los registros físicos o electrónicos para verificar la deuda pendiente, lo que genera retrasos y frustración.

5. Entrega de informe

La colectora elabora los informes sobre los cobros realizados y las deudas pendientes de los estudiantes, lo cual se realiza de forma manual. Estos informes se entregan a la rectora y, en ocasiones, a los padres de familia/representantes para su consulta. Debido a que la información no se genera automáticamente y se debe procesar manualmente, la entrega de estos informes es demorada y puede no reflejar de manera exacta los pagos más recientes, lo que afecta la transparencia y la toma de decisiones oportunas.

6. Atención presencial a padres de familia por consultas

Los padres de familia/representantes acuden presencialmente a la institución para realizar consultas sobre los pagos de matrícula y pensiones, o para resolver dudas sobre las deudas pendientes. La persona responsable de la cobranza atiende a los padres revisando los registros manuales o las hojas de cálculo por lo que la información no está disponible de manera inmediata. Este procedimiento es lento ya que el tiempo de espera se incrementa mientras se localizan y comprueban los documentos, la falta de un sistema automatizado hace el proceso engorroso.

La falta de un sistema automatizado impide obtener información en tiempo real, dificulta la generación de informes y aumenta el tiempo de atención a los padres de familia, lo que afecta negativamente la calidad del servicio administrativo de la Unidad Educativa Genoveva Germán.

1.2. Análisis crítico de la ejecución de procesos

Colectora: Al ejecutar los procesos manuales en el área de colecturía de la Unidad Educativa Genoveva Germán se presentan varios inconvenientes que afectan la eficiencia y precisión del manejo de la información:

- La información relacionada con pagos y deudas está dispersa en documentos físicos, lo que dificulta el acceso rápido a los datos solicitados por padres de familia o autoridades
- La búsqueda manual en formularios archivados consume mucho tiempo, sobre todo cuando existen duplicados o se extravían documentos.
- Los registros de cobros se realizan de forma manual, aumentando el riesgo de errores, omisiones y pérdida de información financiera importante.
- La entrega de informes a las autoridades educativas no es oportuna, debido a que el procesamiento de datos es lento y requiere revisar múltiples archivos físicos.
- No se cuenta con un sistema centralizado que asegure la seguridad, disponibilidad y confiabilidad de la información registrada, lo que afecta la confianza en los procesos administrativos.

Estas dificultades muestran claramente la necesidad de implementar un sistema automatizado que permita optimizar el flujo de trabajo en la oficina de cobranza, centralizar la información y mejorar la atención a los padres y representantes.

1.3. Proceso a automatizar en la Unidad Educativa “Genoveva German”

Luego de haber estudiado todos los procesos que se realizan en la Unidad Educativa “GENOVEVA GERMAN” en el área de colecturía de manera manual, se procede a automatizar las actividades descritas y relacionadas con el campo de acción. La gestión de cobros de matrículas y pensiones inicia cuando el

representante/padre de familia se acerca a secretaria a inscribir al estudiante, momento en el cual la señorita secretaria ingresa los datos a una hoja de cálculo en Excel, para luego proceder a enviar la información a colectoría, la señorita colectora procede a revisar y verificar los datos del estudiante y realiza el cobro sea de la matrícula o de la pensión del estudiante.

Los procesos a automatizar son los siguientes:

- Administrar usuarios.
- Recopilar la información de los estudiantes de toda la unidad educativa.
- Registrar los pagos correspondientes a las matrículas.
- Registrar los pagos relacionados de pensiones.
- Emitir reportes que permitan tener un mejor control por parte de la colectora y autoridades de la institución en cada período.

La automatización de estos procesos permitirá centralizar la información en un único sistema, eliminando la dependencia de hojas de cálculo y documentos físicos que dificultan la consulta y actualización de datos. Asimismo, garantizará que cada transacción quede registrada de forma segura y en tiempo real, facilitando el seguimiento de pagos, la detección oportuna de deudas y la generación de reportes precisos que respalden la toma de decisiones administrativas y financieras.

1.4. Sistemas automatizados existentes vinculados con el campo de acción

Las herramientas, plataformas web y los sistemas automatizados se han convertido esenciales para administrar operaciones y servicios en múltiples sectores uno de ellos es el ámbito educativo. Ya que su necesidad es tener soluciones tecnológicas que agilicen de forma precisa y segura los procesos como la gestión de pagos, registros, informes y el control de la información, con el propósito de incrementar la eficiencia y administración educativa.

Considerando las necesidades de la Unidad Educativa Genoveva Germán, se llevó a cabo un análisis de diversos sistemas informáticos y plataformas web que brinden soluciones a las necesidades requeridas para llevar a cabo el cobro de matrículas y pensiones, así como la administración interna. A continuación, se muestra una tabla comparativa con algunas de las herramientas disponibles en el mercado.

Tabla 2 Comparación de Herramientas Informáticas para la Gestión de Cobros

| Programa | Licencia \$ | Soporte | Registro de Estudiantes | Actualización de Precios | Generación de Reportes | Control de Pagos | Generación de Recibos | Acceso Web para Padres |
|--------------------|-------------|-----------------|-------------------------|--------------------------|------------------------|------------------|-----------------------|------------------------|
| SIAGIE | Gratis | Soporte público | X | X | X | X | X | X |
| EduTrak | 200/año | 24h/soporte | X | X | X | X | X | X |
| Alexia | 500/año | 24h/soporte | X | X | X | X | X | X |
| Factumarket | 290/año | 5d/8h | X | X | X | X | X | X |
| Schoology | 300/año | 24h/soporte | X | X | X | X | X | |

Tras analizar los sistemas automatizados existentes, se observa que, aunque opciones como SIAGIE, EduTrak y Factumarket ofrecen funcionalidades útiles para la gestión de cobros y pagos, ninguno cubre completamente las necesidades específicas de la Unidad Educativa Genoveva Germán, como la actualización de precios o el control de proveedores. Además, los sistemas más completos como Alexia y Schoology son costosos y no están enfocados exclusivamente en la gestión de cobros.

Por lo tanto, se concluye que la implementación de un sistema propio sería la mejor opción para adaptar las funcionalidades a las necesidades específicas de la institución y optimizar los procesos administrativos de forma económica y eficiente.

1.5. Metodologías y herramientas de desarrollo.

1.5.1. Metodologías tradicionales para el desarrollo de software

Las metodologías tradicionales de desarrollo se caracterizan por un proceso disciplinado y estructurado, cuyo propósito es establecer eficazmente los requerimientos del cliente y obtener un software de calidad. La principal característica de estas metodologías es el énfasis en la planificación, la documentación, el control

del cronograma y la ejecución, así como en la asignación de roles y responsabilidades y la ejecución de actividades.

Las tecnologías tradicionales son más apropiadas para la ejecución de proyectos grandes, con requisitos estables que requieran procesos estructurales y documentación detallada, (Góngora Duarte, Morales Espitia, & Trujillo-González, 2022).

1.5.2. Metodología de RUP

RUP es una metodología diseñada para organizar y estructurar el desarrollo de software, que incluye un conjunto de acciones esenciales que tienen como objetivo transformar las necesidades del usuario en un sistema operativo y funcional. Este proceso hace referencia a los modelos en Cascada y por Componentes, lo cual es dirigido por casos de uso, centrado en la arquitectura, basados en aspectos esenciales para el desarrollo eficiente de software, (Kruchten, 2022).

Las metodologías tradicionales, como el Rational Unified Process (RUP), han demostrado ser eficaces en la aplicación de prácticas de gestión y técnicas para el desarrollo de software que no requiere escalabilidad. Sin embargo, es poco común encontrar este tipo de procesos en proyectos académicos. Por ello, han surgido diversas metodologías de desarrollo de software tradicionales que según el contexto y las necesidades específicas del cliente se adaptan para facilitar tanto el desarrollo como la implementación del software en el entorno para el cual fue creado, (Pérez A., 2023)

El objetivo es promover el uso de metodologías como RUP ya que mejoran el nivel de contacto entre el desarrollador y el usuario al mismo tiempo que proporcionan documentación ordenada y exhaustiva que puede servir como referencia del sistema en el futuro. Todo esto con el fin de optimizar el manejo de la tecnología y de los medios de desarrollo en el caso de RUP con el uso de la metodología se logra al final del ciclo un producto que resulta útil y que satisface tanto a la empresa de desarrollo de software como al usuario final quien espera un sistema terminado que cumpla con todos sus requerimientos de manera eficiente y en el tiempo pactado, (Sánchez Tudela, Molina Real, & Tudela-Rodríguez, 2025)

1.5.3. UML

UML es un lenguaje de modelado que permite representar, definir, desarrollar

y documentar diferentes componentes de un sistema de software desde varias perspectivas. Se puede utilizar en cualquier contexto de desarrollo y en todas las etapas del ciclo de vida además se puede aplicar en todos los dominios de interés y en todas las plataformas de implementación. También puede usarse en otras áreas, como la ingeniería de negocio y modelado de procesos gracias a los mecanismos de adaptación/extensión mediante perfiles, (Koc & Mert Erdoğan, 2021)

UML se emplea en la creación de distintos tipos de sistemas. El UML se puede utilizar para modelar sistemas de software, componentes de hardware y procesos dentro de organizaciones reales siempre cubre cualquier característica de un sistema en uno de sus diagramas y también se aplica en diferentes etapas de desarrollo de un sistema desde la especificación de análisis de requisitos hasta su finalización con la fase de prueba, (Ramírez Jiménez, Pulido-Hernández, Gómez Torres, & Rivera-Orozco, 2024)

1.5.4. Herramientas de desarrollo

En la elaboración del proyecto de titulación se emplea un aplicativo gratuito de código abierto que nos permite crear diversas herramientas de desarrollo para el diseño, la gestión, la construcción y otros procesos. Otras son herramientas con licencias de prueba. Los softwares utilizados son:

- C# (.NET 8)
- WinUI 3
- MySQL (MariaDB vía XAMPP)
- Entity Framework Core (Pomelo MySQL)
- Microsoft.Extensions.DependencyInjection
- phpMyAdmin (XAMPP)
- SessionService
- NavigationView (WinUI 3)

La selección de estas herramientas responde a criterios de compatibilidad, robustez y escalabilidad, permitiendo construir un sistema seguro y de alto rendimiento que se adapta a las necesidades de la Unidad Educativa Genoveva Germán. Cada tecnología fue elegida en función de su capacidad para integrarse

eficientemente con las demás, optimizar el flujo de trabajo de desarrollo y facilitar futuras mejoras o ampliaciones del sistema. Esta combinación tecnológica asegura que el producto final no solo cumpla con los requisitos actuales, sino que también pueda evolucionar conforme crezcan las demandas institucionales, (Villa Quishpe, Centeno Atavallo, Gavilánez Guanoluisa, & Cadena Moreano, 2023)

1.5.5. C# (.NET 8)

C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft, moderno, flexible y seguro. .NET lo convierte en un framework que permite desarrollar aplicaciones de escritorio, aplicaciones web, móviles y servicios en la nube. La nueva versión .NET 8, a pesar de haber sido lanzada recientemente, ya ofrece grandes avances en el desempeño, soporte de multiplataforma y optimización en la gestión de memoria de los recursos, lo que permite un desarrollo de aplicaciones mucho más rápidas y eficientes. Por si fuera poco, su ecosistema no es el único punto a favor ya que la comunidad de desarrolladores que .NET ofrece es verdaderamente robusta y activa, por lo que siempre existirá soporte y recursos por aprender y usar profesionalmente.

C# permite en este proyecto implementar los negocios y acceder a la base de datos, lo que permite mantener la estructuración y el mantenimiento del código. A diferencia de otros lenguajes, C# es compatible con MVC y MVVM, por lo que la separación de responsabilidades es mucho más eficiente y la escalabilidad del sistema es agigantada. Tanto como su ecosistema, C# cuenta con potentes nativos como Entity Framework Core y WinUI 3, lo que permite la construcción de un sistema completo, con control robusto en el flujo de los datos y la seguridad en el sistema es sublime, (Juárez López, Ramírez Jiménez, & Rivera Orozco, 2020)

1.5.6. WinUI 3

WinUI 3 es el marco moderno desarrollado por Microsoft para el desarrollo de interfaces gráficas de usuario (GUI) para aplicaciones de escritorio de Windows. Se fundamenta en el sistema Fluent Design y ofrece una experiencia visual atractiva y consistente que se adapta eficazmente a distintas resoluciones y dispositivos. Está desarrollado pensando en la integración con tecnologías modernas de .NET, lo que facilita la implementación del patrón MVVM junto con la inyección de dependencias para crear aplicaciones modulares. Además, viene con controles de interfaz

avanzados como NavigationView, listas dinámicas y formularios adaptativos, (Apiag, Cádiz, & Lincopinis, 2023)

Gracias a su enfoque moderno y su compatibilidad nativa con Windows 10 y 11, WinUI 3 ofrece un rendimiento optimizado y soporte para actualizaciones continuas de la interfaz, lo que asegura la vigencia tecnológica del sistema a lo largo del tiempo. Su arquitectura desacoplada permite que el diseño visual se mantenga independiente de la lógica de negocio, facilitando la personalización y el mantenimiento de la aplicación. Asimismo, su integración con herramientas como Visual Studio y GitHub acelera el ciclo de desarrollo, fomentando prácticas de trabajo colaborativas y una implementación más ágil de nuevas funcionalidades, (Hoefling, 2022)

1.5.7. MySQL (MariaDB vía XAMPP)

MySQL es un sistema de gestión para bases de datos relacionales que permite almacenar, organizar y manipular información de manera estructurada (RDBMS) de código abierto, es un sistema fácil de usar e integrar con otros lenguajes de programación, en este proyecto se utiliza en su variante MariaDB incluida en el paquete XAMPP lo que permite disponer de un entorno de desarrollo local rápido y accesible su uso extendido en sistemas de gestión académica se debe a su habilidad en el procesamiento de grandes volúmenes de datos y su compatibilidad con SQL, (Marzuki Shofi, Damar Nugraha, Anggraini, & Hakiem, 2023)

Considerando el uso anterior, se optó por usar MySQL/MariaDB en este sistema de gestión porque permite almacenar de forma segura y confiable, datos de estudiantes, matrículas, pagos y reportes. La base de datos se integra usando Entity Framework Core con el proveedor Pomelo MySQL lo que soporta el mapeo objeto relacional y ahorra en complejidad y mantiene el código a través del uso de consultas.

1.5.8. Entity Framework Core (Pomelo MySQL)

Entity Framework Core (EF Core) es un framework de mapeo objeto-relacional (ORM) de código abierto para .NET que permite a los desarrolladores interactuar con bases de datos mediante clases y objetos a nivel de OO en lugar de escribir consultas SQL esto ahorra tiempo y reduce la posibilidad de cometer errores. Además, aumenta la mantenibilidad. En este proyecto se emplea junto con el proveedor Pomelo MySQL, que es un conector especializado para MySQL y MariaDB, para obtener la mejor

potencia y total compatibilidad, (Durvas Jayaraman, 2025)

Con EF Core en este sistema, el desarrollo de las operaciones CRUD se vuelve más ágil y el código necesario para implementarlas es más limpio y compacto libre de repetición. También, su manejo de migraciones permite evolucionar el esquema de las bases de datos en espiral a medida que el sistema crece y esto se puede hacer de manera segura sin temor a perder datos o comprometer la integridad referencial.

1.5.9. Microsoft.Extensions.DependencyInjection

Microsoft.Extensions.DependencyInjection es la librería oficial de Microsoft para aplicar el patrón de inyección de dependencias en aplicaciones .NET. Este patrón de diseño permite desacoplar los distintos componentes de un sistema, lo que facilita la reutilización del código y hace que sea más sencillo probarlo. La inyección de dependencias es clave para desarrollar proyectos modulares y escalables, donde los servicios se registran y se resuelven de forma centralizada dentro de la aplicación, mejorando así su organización y mantenimiento, (Tan, Chen, & Jiao, 2023).

Dentro del contexto de este proyecto, se utiliza para registrar y gestionar servicios como SessionService, DbContext y los controladores de navegación y sus correspondientes repositorios. Esta técnica asegura que los componentes solo puedan acceder a las dependencias que son necesarias, mejorando la seguridad, la mantenibilidad y en general el rendimiento del sistema de la aplicación.

Además, su uso fomenta las buenas prácticas de arquitectura limpia, permitiendo que la lógica de negocio y la interfaz de usuario permanezcan independientes y fáciles de modificar sin afectar otras capas del sistema. Esta separación clara entre componentes facilita la implementación de pruebas unitarias y de integración, lo que contribuye a una mayor calidad del software y reduce el riesgo de errores en entornos de producción. Gracias a esta librería, el proyecto logra una estructura más ordenada y preparada para futuras ampliaciones o cambios tecnológicos.

1.5.10. PhpMyAdmin (XAMPP)

PhpMyAdmin es una herramienta de gestión visual para bases de datos MySQL y MariaDB, a la que se puede acceder a través de un navegador web ayuda a realizar tareas como crear, modificar y eliminar tablas, así como consultar y exportar datos. Es muy útil en entornos de desarrollo para la validación de datos y realizar

ajustes rápidos de forma gráfica, eliminando la necesidad de escribir comandos SQL.

En este proyecto, phpMyAdmin está integrado en este proyecto para las fases de desarrollo y prueba para asegurar que los datos se registren correctamente y refinar la estructura de la base de datos. Está integrado con XAMPP, lo que ayuda en una gestión más fluida del entorno local y, por lo tanto, acelera los ciclos de desarrollo y depuración del sistema. (Yusup Ali, Permana, & Achmad Kurniady, 2020)

1.5.11. SessionService

SessionService es un servicio creado e implementado en el proyecto, responsable de manejar el estado de la sesión para un usuario autenticado puede validar credenciales, mantener la información del usuario activa durante el transcurso de la aplicación y proporcionar restricciones de acceso al usuario según el rol asignado, (Annela, 2022)

El desarrollo de esta función garantiza la seguridad y el manejo óptimo de la sesión al bloquear el acceso no autorizado y hacer cumplir estrategias de autenticación estrictas. También trabaja con la lógica de navegación para redirigir automáticamente a un usuario al cerrar o expirar la sesión.

Adicionalmente, SessionService se encarga de registrar eventos relacionados con la actividad de cada sesión, como la hora de inicio, cierre y cualquier intento fallido de autenticación. Estos registros permiten generar reportes de auditoría y detectar patrones inusuales de acceso, contribuyendo así a la seguridad general del sistema. La capacidad de mantener un control detallado sobre el comportamiento de los usuarios facilita la detección temprana de posibles vulnerabilidades o accesos indebidos, (Honest, Patel, & Patel, 2013)

Otra ventaja importante de este servicio es su flexibilidad para integrarse con otras capas del sistema, como la base de datos y los módulos de control de permisos. Esto permite que cualquier cambio en los privilegios de un usuario se refleje en tiempo real durante su sesión activa, sin necesidad de cerrar y volver a iniciar. De esta manera, SessionService no solo asegura la continuidad de la experiencia del usuario, sino que también garantiza que las políticas de seguridad de la institución se apliquen de forma inmediata y efectiva.

1.5.12. NavigationView (WinUI 3)

NavigationView es un control que forma parte de los elementos de la interfaz de usuario en WinUI 3 que permite una implementación estructurada de menús de navegación laterales o superiores. Soporta íconos, etiquetas, submenús e incluso rutas dinámicas, lo que ayuda a organizar las diversas secciones de una aplicación. Su implementación correcta optimiza la experiencia del usuario al facilitar una navegación eficiente, (Leuthold, 2021)

En este proyecto, NavigationView constituye el núcleo de la interfaz visual, brindando acceso a módulos como Dashboard, Estudiantes, Matrículas, Pagos y Configuración. Su integración con C# y la lógica empresarial permite que los elementos del menú respondan de forma dinámica según las acciones y permisos del usuario autenticado.

Los criterios de rendimiento, compatibilidad, escalabilidad y mantenimiento justifican el uso de las herramientas de desarrollo seleccionadas para este proyecto. Ellas permiten la implementación de todas las funcionalidades, y la aplicación dos de los requerimientos no funcionales. El pair programming, C# y .NET 8 en conjunto con WinUI 3 permiten no solo crear una interfaz de usuario moderna, sino también una que sea responsive.

Junto con MySQL, Entity Framework Core facilita una base sólida en la gestión de datos mediante la aplicación de funcionalidades como inyección de dependencias y los componentes por ejemplo NavigationView, el modularidad y la organización del sistema es fortalecida. En una palabra, con todas estas tecnologías se crean soluciones robustas que, además de poder ser fácilmente ampliadas en el futuro, estas están en concordancia con las mejores prácticas de desarrollo, (Sánchez Pacheco, 2020)

Tabla 3 Comparativa de las Características de las Tecnología

| Tecnología | Tipo | Características | Ventajas | Desventajas |
|----------------------------------|--------------------------|---|---|--|
| C# (.NET 8) | Lenguaje de programación | Orientado a objetos, multiplataforma con .NET 8, integración nativa con herramientas Microsoft. | Alto rendimiento, sintaxis clara, gran comunidad de soporte. | Dependencia del ecosistema Microsoft, curva de aprendizaje inicial. |
| WinUI 3 | Framework UI | Framework moderno de Microsoft para interfaces de escritorio en Windows, basado en Fluent Design. | Interfaces modernas y responsivas, soporte MVVM, integración con .NET. | Solo disponible para Windows, curva de aprendizaje para principiantes. |
| MySQL (MariaDB vía XAMPP) | Base de datos relacional | RDBMS de código abierto, soporte SQL completo, alta compatibilidad. | Estable, rápido y ampliamente usado, integración con múltiples lenguajes. | Menos eficiente que bases NoSQL para datos no estructurados. |

| | | | | |
|---|----------------------------------|---|--|---|
| Entity Framework Core (Pomelo MySQL) | ORM | Mapeo objeto-relacional para .NET, soporte para migraciones y consultas LINQ. | Reduce código SQL, mejora mantenibilidad, integra validación de datos. | Rendimiento menor frente a consultas SQL optimizadas manualmente. |
| Microsoft.Extensions.DependencyInjection | Biblioteca .NET | Implementa inyección de dependencias en aplicaciones .NET. | Mejora modularidad, facilita pruebas y mantenimiento. | Puede añadir complejidad si no se estructura correctamente. |
| phpMyAdmin (XAMPP) | Herramienta de administración BD | Administración visual para MySQL/MariaDB vía navegador. | Fácil de usar, gestión centralizada, exportación/importación de datos. | No apto para entornos de alta concurrencia como solución principal. |
| SessionService | Servicio interno | Manejo de sesión de usuario, control de autenticación y permisos. | Seguridad en el acceso, validación de roles en tiempo real. | Debe programarse y mantenerse manualmente. |

| | | | | |
|-------------------------------------|---------------------|---|---|--|
| NavigationView (WinUI 3) | Control de interfaz | Componente de navegación lateral/superior con soporte para iconos y submenús. | Mejora usabilidad, navegación clara y estructurada. | Limitado a aplicaciones Windows con WinUI. |
|-------------------------------------|---------------------|---|---|--|

Fuente: (Wildermuth, 2022), (Kimmel, 2019), (Smith, 2023)

De acuerdo con el análisis de documentos que se han entregado, se puede afirmar que el sistema de gestión escolar desarrollado para la Unidad Educativa Genoveva Germán usa MySQL como sistema de gestión de bases de datos, lo que resulta conveniente en el caso de la escuela pues requiere el ingreso, manejo y el procesamiento de gran información como el registro de estudiantes el pago de matrículas y otros servicios administrativos. Además, se puede constatar que las estructuras que se MySQL que se han extraído se relacionan con las tablas que manejan la información de clave de los estudiantes, pago, usuario y otras entidades relacionadas con el sistema de gestión de cobros de matrículas y pensiones.

La base de datos MySQL es una de las más populares tecnologías, esto se debe a la eficiencia, estabilidad y la capacidad de mejorarse con la integración con otros lenguajes de programación y otros frameworks. De esta forma, se puede asegurar que la opción de base de datos se sigue incluyendo con estos requisitos en el sistema que se realizará.

Esto nos lleva a afirmar que algunas de las tecnologías más relevantes que se utilizarán en el proyecto son:

- MySQL (MariaDB via XAMPP) como gestor de base de datos.
- C# (.NET 8) y WinUI 3 para el diseño y la lógica de la aplicación de escritorio, esta última para la interfaz de usuario.
- Entity Framework Core (Pomelo MySQL) para interactuar con la base de datos utilizando mapeo objeto relacional.

La elección de estas tecnologías proporciona una base sólida para desarrollar un sistema eficiente, seguro, fácil de mantener y escalable, optimizando la gestión de cobros de matrículas y pensiones en la Unidad Educativa Genoveva Germán.

1.6. Tendencias y tecnologías actual

En el diseño de software para instituciones educativas, el tipo de metodología y tecnologías a utilizar determinan el alineamiento del sistema a las necesidades de las instituciones, su seguridad y robustez. Los sistemas se suelen construir utilizando metodologías ágiles, y su popularidad se imparte por su flexibilidad y rápida respuesta a los cambios. Las metodologías más clásicas, en cambio para obras que requieren más documentos, planificación y control de cada etapa, todavía son la mejor opción.

De la misma forma, el desarrollo del sistema para controlar el proceso de recaudación de matrículas y pensiones en la Unidad Educativa Genoveva Germán se sirvió de una metodología tradicional que es la más apropiada en este caso por su tendencia a usar enfoques iterativos. Este método se ocupa del control de los requerimientos, el diseño sistemático, y la verificación rigurosa de los estándares de calidad en cada etapa del ciclo. Con esto se garantiza que el sistema cumpla con los objetivos planteados durante su formulación y se nos acompaña desde su diseño inicial hasta la etapa de puesta en marcha.

Tabla 4 Comparación de metodologías aplicadas al proyecto

| Metodología Tradicional | Metodología Ágil |
|--|---|
| Planificación detallada y estructurada desde el inicio. | Planificación flexible y adaptativa en cada iteración. |
| Altamente documentada, lo que facilita el mantenimiento y futuras mejoras. | Documentación mínima, prioriza la entrega rápida. |
| Fases definidas y secuenciales. | Ciclos cortos de desarrollo sin fases formales. |
| Roles y responsabilidades bien establecidos. | Roles simplificados y colaborativos. |
| Ideal para proyectos con requisitos claros y estables. | Ideal para proyectos con requisitos cambiantes. |
| Mayor control de calidad en cada etapa. | Mayor rapidez en la entrega de incrementos funcionales. |
| Centrada en la arquitectura y casos de uso. | Centrada en la entrega de valor al cliente. |

1.7. Conclusiones del capítulo

Con un enfoque tradicional como este, se mantiene el control del desarrollo, la documentación y los procesos de calidad lo cual es crucial para un sistema que necesita ser institucionalmente estable y seguro durante un largo período de tiempo. Este enfoque de desarrollo sigue un conjunto lineal de fases, lo que facilita relativamente asegurar que cada requisito previo esté completamente validado antes de proceder al siguiente paso, disminuyendo las posibilidades de errores de producción. Además, este enfoque se alinea perfectamente con las tecnologías

utilizadas para el proyecto, que incluyen C# (.NET 8), WinUI 3 y Entity Framework Core con MySQL, manteniendo así el equilibrio entre la planificación metodológica definida y la arquitectura tecnológica para el sistema que gestiona la recolección de matrículas y tarifas en la Unidad Educativa Genoveva Germán.

CAPÍTULO II

2. MODELO DE NEGOCIO, REQUISITOS FUNCIONALES Y MODELOS DEL SISTEMA

En este capítulo se presentará el entorno que será objeto de automatización, utilizando diagramas de casos de uso de negocio la identificación de los actores involucrados y diagramas de actividades de negocio. Asimismo, se detallarán los requisitos funcionales considerando su nivel de complejidad, y se procederá a modelar y describir los casos de uso que conforman el sistema.






2.1. Modelo del Negocio y Requisitos

El sistema propuesto está orientado exclusivamente al uso del personal interno autorizado de la institución como el administrador, la colectora y la rectora. Aunque los padres de familia y estudiantes no tendrán acceso directo a la plataforma, sí será posible registrar estudiantes, asignarles sus respectivos pagos y generar comprobantes, como parte del flujo administrativo interno.

Este modelo de negocio contempla una transformación digital que abarca desde la inscripción del estudiante y la asignación de responsables financieros hasta el ingreso y control de pagos, emisión de reportes consolidados y validación de información por parte de las autoridades competentes. Para estructurar y comprender este proceso, se han identificado los actores principales los casos de uso vinculados a sus funciones así como los requisitos funcionales y no funcionales que guiarán el desarrollo del sistema.

En la tabla siguiente se detallan los estereotipos que serán aplicados en la elaboración del modelo de negocio.

Figura 2 Componentes de un caso de uso

| Componentes de un Caso de Uso | | |
|---|-------------|--|
| Figura | Nombre | Definición |
|  | Actor | Un actor en un caso de uso se define como una entidad externa que puede ser: personas, sistemas, que interactúan con el sistema para llevar a cabo un proceso o una función específica |
|  | Caso de Uso | Un caso de uso representa las distintas capacidades de un sistema. Cada caso de uso recibe un nombre y una descripción, la cual detalla el inicio y su final. Además, incluye una explicación de la funcionalidad que representan, junto con escenarios complementarios y requerimientos no funcionales. |
|  | Conectores | Los conectores en un caso de uso son relaciones visuales que indican como los actores y los casos de usos interactúan dentro de un sistema. Se representan en los diagramas de casos de uso utilizando líneas y flechas, permiten comprender mejor la estructura y el flujo de las |
|  | Extensión | entre casos de uso en la que uno extiende a otro. Es decir, el caso de uso extendido agrega o modifica el comportamiento de otro caso de uso bajo ciertas condiciones o circunstancias adicionales |
|  | Inclusión | Se utiliza para representar una relación entre casos de uso en la que un caso de uso incluye el comportamiento de otro caso de uso de manera obligatoria. |

Fuente: (Kimmel, 2019)

La figura presentada ilustra los principales componentes de un caso de uso en UML, los cuales permiten representar de manera visual la interacción entre un sistema y sus actores. En primer lugar, el actor se simboliza con una figura humana y representa a una entidad externa persona, sistema o proceso que interactúa con el sistema para cumplir una función específica. El caso de uso, representado por un óvalo, describe una capacidad o funcionalidad del sistema, incluyendo el inicio, desarrollo y final del proceso, así como los requisitos asociados.

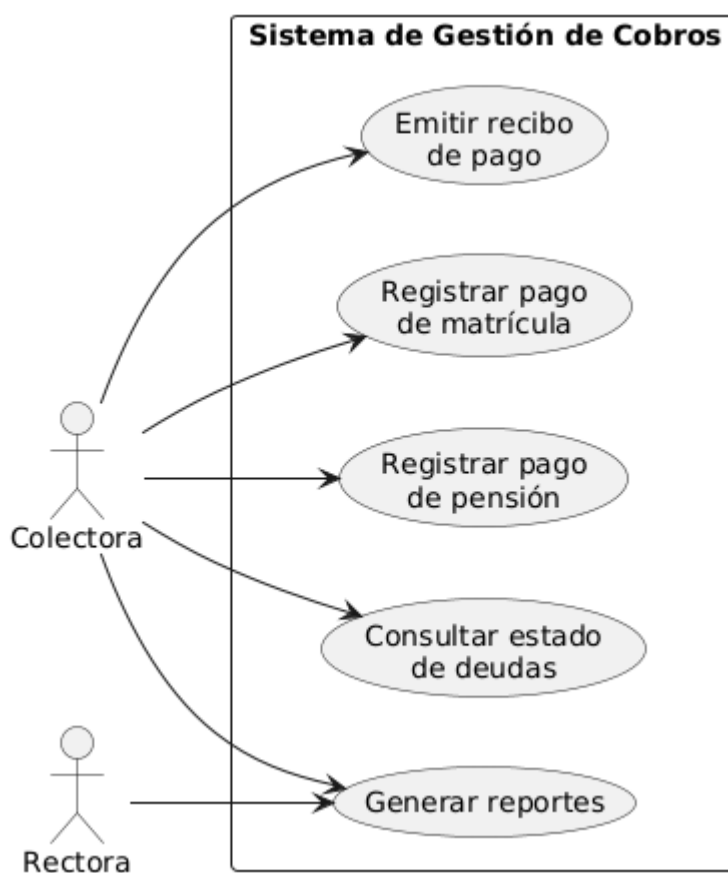
Los conectores son líneas que enlazan actores y casos de uso, mostrando la relación y el flujo de interacción. La relación de extensión (extends), mostrada con una flecha discontinua y el estereotipo correspondiente, indica que un caso de uso puede ampliar o modificar el comportamiento de otro bajo ciertas condiciones. Finalmente, la inclusión (include), también representada con una flecha discontinua

etiquetada, señala que un caso de uso incorpora de manera obligatoria la funcionalidad de otro. En conjunto, estos elementos conforman la base visual para comprender cómo se organizan y vinculan las funcionalidades dentro de un sistema.

2.2. Diagrama de caso de uso del negocio

En el siguiente diagrama se representa el caso de uso del negocio asociado a la gestión de cobros de matrículas y pensiones en la Unidad Educativa Genoveva Germán, mostrando la interacción entre el actor del negocio y el sistema en el proceso correspondiente.

Figura 3 Diagrama caso de uso del negocio



2.2.1. Actores y casos de uso del sistema

Se consideran actores del negocio a cualquier individuo, grupo u organización con los que la institución interactúa en el proceso de gestión de matrículas y pensiones.

Tabla 5 Especificaciones de los actores del sistema

| Actor del negocio | Función |
|--------------------------|---|
| Administrador | Encargado de la supervisión y control general de las operaciones financieras y administrativas de la institución. |
| Colectora | Responsable de la gestión de cobros de matrículas y pensiones, así como del registro de pagos y generación de reportes. |

Trabajador del negocio

El trabajador del negocio lleva a cabo ciertas tareas relevantes en el marco del proceso organizacional, aun cuando no siempre se relacione de forma concreta con el sistema automático. Su contribución es crucial para el entendimiento de las funciones que se van a sustentar, dado que brinda la información y la experiencia respecto a los métodos y procesos de la entidad, (Gómez Peñuela, Silva Labarca, & Ovalles Pabón, 2019)

Tabla 6 Descripción del trabajador del negocio

| Actor del negocio | Función |
|--------------------------|--|
| Secretaria | Gestiona la información administrativa relacionada con estudiantes y apoya en el control de documentación y registros. |

2.2.2. Requisitos

En el ámbito del desarrollo de software, la etapa de recolección de requisitos es vital en la garantía de la calidad y del producto final, así como en su construcción, siendo funcional. El objetivo es el de establecer, documentar y sistematizar las funcionalidades del sistema, con respecto a las expectativas de los usuarios y el contexto institucional en el que se utilizará.

Bajo este enfoque, el sistema permite cuyos resultados se integren en la solución que se elabore, definiendo los objetivos que se desee alcanzar, cumpliendo con el esquema de que se cumpla con todos los requisitos establecidos para una

respuesta funcional y de eficiencia a los procesos administrativos internos de la Unidad Educativa Genoveva Germán. El saber obtenido en esta fase se puede, en gran parte, reutilizar, lo que permite en gran medida, en otros sistemas, con el objetivo de disminuir los recursos, los costos de implementación, el tiempo y la escalabilidad del sistema.

Los requisitos del sistema se dividen en las siguientes dos categorías:

- Los requisitos sistemas funcionales, que describen acciones que son procesadas con los objetivos que el sistema puede realizar.
- Los requisitos sistemas no funcionales, que establecen condiciones de calidad, seguridad, rendimiento y otros requisitos que son a la solución.

A continuación, se presentan dichos requisitos junto a su clasificación, orden de prioridad y actores que se relacionan.

2.3. Requisitos Funcionales del sistema

En el caso de la Unidad Educativa Genoveva Germán, los procesos de matrícula y de gestión de pensiones que requieren ser automatizados para mejorar la eficiencia de la unidad escolar en el tratamiento de datos, en este caso, el sistema debe permitir al usuario ofrecer dichas funcionalidades. Todo el software descrito en los requerimientos se concibió para el uso exclusivo de la administración de la institución, (Páez Cárdenas, Arias Peralta, & Wanumen Silva, 2018)

Tabla 7 Módulo de autenticación y control de acceso

| ID | Requisito | Descripción | Prioridad | Actor |
|-----|--------------------|---|-----------|---------------|
| RF1 | Autenticar usuario | Valida las credenciales de acceso y asigna el perfil correspondiente. | Media | Administrador |

Nota: Esta tabla presenta los requisitos relacionados con el acceso seguro al sistema. La autenticación permite restringir el uso solo al personal autorizado de la institución.

Tabla 8 Módulo de gestión de usuarios, roles y permisos

| ID | Requisito | Descripción | Prioridad | Actor |
|-----------|--------------------|--|------------------|---------------|
| RF2 | Gestionar permisos | Crear, modificar, buscar y eliminar permisos del sistema. | Baja | Administrador |
| RF3 | Gestionar roles | Permitir la creación, edición y eliminación de roles, así como la asignación de los accesos correspondientes a cada uno. | Baja | Administrador |
| RF4 | Gestionar usuarios | Registrar y administrar cuentas internas (colectora, rectora). | Media | Administrador |

Nota: Este conjunto de requisitos permite al administrador controlar los niveles de acceso y las funciones disponibles para cada usuario del sistema, asegurando una correcta asignación de responsabilidades.

Tabla 9 Módulo académico

| ID | Requisito | Descripción | Prioridad | Actor |
|-----------|---|---|------------------|---------------|
| RF5 | Administrar la información de los estudiantes | Registro, la modificación, la consulta y la eliminación de sus datos. | Alta | Administrador |
| RF6 | Gestionar cursos y paralelos | Administración de divisiones | Baja | Administrador |

| | | | | |
|-----|--|---|-------|--------------|
| | | académicas internas. | | |
| RF7 | Administrar la información de los representantes | Registro y gestión de las personas responsables del manejo financiero de los estudiantes. | Media | Administrado |

Nota: Esta tabla agrupa las funcionalidades vinculadas a la estructura académica. Son esenciales para vincular los pagos con los estudiantes y sus responsables

Tabla 10 Módulo financiero

| ID | Requisito | Descripción | Prioridad | Actor |
|-----|--|---|-----------|-----------|
| RF8 | Registrar cobros | Ingreso de pagos de matrícula y pensión con validación y comprobante. | Alta | Colectora |
| RF9 | Gestionar los cobros mediante la actualización | Consulta o cancelación de los registros existentes. | Media | Colectora |

Nota: Los requisitos aquí listados forman parte del núcleo financiero del sistema, ya que permiten llevar un control estructurado y verificable de los ingresos por concepto de matrícula y pensiones.

Tabla 11 Módulo de reportes

| ID | Requisito | Descripción | Prioridad | Actor |
|------|------------------|--|-----------|-----------|
| RF10 | Generar reportes | Generación de reportes por estudiante, curso o período. Exportación en PDF. | Alta | Colectora |

| | | | | |
|------|--------------------|--|-------|---------|
| RF11 | Consultar reportes | Visualización en modo lectura de reportes financieros. | Media | Rectora |
|------|--------------------|--|-------|---------|

Nota: Esta tabla contiene los requisitos que permiten transformar los datos almacenados en información útil para la toma de decisiones. Los reportes son esenciales para el control institucional y la supervisión financiera.

2.4. Requisitos no funcionales

En el caso de la Unidad Educativa Genoveva Germán, los procesos de matrícula y de gestión de pensiones que requieren ser automatizados para mejorar la eficiencia de la unidad escolar en el tratamiento de datos, en este caso, el sistema debe permitir al usuario ofrecer dichas funcionalidades. Todo el software descrito en los requerimientos se concibió para el uso exclusivo de la administración de la institución (Villa Quishpe, Centeno Atavallo, Gavilánez Guanoluisa, & Cadena Moreano, 2023)

Además, las restricciones o requisitos no funcionales que están destinados a encuadrar el sistema dentro de un entorno de trabajo y que delimitan:

1. Capacidad mínima y máxima
2. Interfaces con otros sistemas
3. Recursos que se necesitan
4. Aspectos de seguridad
5. Aspectos de fiabilidad
6. Aspectos de mantenimiento
7. Aspectos de calidad

Estos requisitos tienen un origen más técnico y no tienen tanto interés para el cliente como lo tienen los funcionales, por lo tanto, parece que evidente que deban permanecer claramente separados en el modelo del sistema.

2.5. Especificación de casos de uso y diagramas de actividades

En esta sección se describen los principales casos de uso que conforman la interacción del personal administrativo con el sistema de gestión de matrículas y pensiones. Estos casos permiten apreciar con mayor claridad cómo se desarrollan las acciones dentro del sistema, las condiciones necesarias para ejecutarlas y los

participantes involucrados. El Lenguaje Unificado de Modelado (UML) funciona como un recurso para plasmar la lógica del sistema mediante diagramas de actividades, los cuales representan de forma ordenada y secuencial los procesos internos. A diferencia de los diagramas de flujo tradicionales, los diagramas de actividades muestran el carácter dinámico de las tareas, los momentos de decisión, la ejecución simultánea de acciones y las condiciones que regulan su avance.

Esta representación resulta fundamental para facilitar la comprensión entre el equipo de desarrollo y los usuarios institucionales, además de servir como base para la documentación técnica y pruebas funcionales (Iqbal, Al Azzoni, Allen, & Ullah Khan, 2020)

Tabla 12 Casos de uso: Gestión de estudiantes

| ID | Caso de Uso | Actor | Precondición |
|-----------|--------------------------------------|---------------|--|
| CU01 | Gestionar estudiantes | Administrador | El usuario debe haber iniciado sesión y tener privilegios administrativos. |
| CU02 | Registrar estudiante | Administrador | El estudiante debe ser nuevo y no estar registrado previamente. |
| CU03 | Modificar datos del estudiante | Administrador | El estudiante debe existir en el sistema. |
| CU04 | Eliminar estudiante | Administrador | El estudiante debe estar registrado y no tener pagos pendientes. |
| CU05 | Consultar información del estudiante | Administrador | El estudiante debe estar registrado y tener datos accesibles. |

Nota: Estos casos de uso reflejan el ciclo de vida de la información de los estudiantes dentro del sistema, desde su registro hasta la consulta y eliminación, garantizando control administrativo sobre cada acción

2.6. Modelo de Casos de uso del negocio

La Unidad Educativa Genoveva Germán enfrenta actualmente una serie de dificultades en el manejo manual de los procesos administrativos relacionados con el cobro de matrículas y pensiones. Esta situación ha generado desorganización, duplicidad de registros, demoras en la atención a los padres de familia y una falta de control eficiente sobre las finanzas institucionales. En este contexto, resulta fundamental comprender de forma detallada cómo se realizan actualmente estos procesos dentro de la institución, antes de plantear una solución tecnológica.

El modelo de casos de uso del negocio constituye una representación visual que permite identificar y detallar las actividades clave que se llevan a cabo en la institución en el área de gestión de cobros, así como los actores que intervienen y la forma en que interactúan. Esta herramienta resulta esencial para examinar el funcionamiento actual del sistema, detectar posibles deficiencias y establecer una base sólida para el diseño de un sistema automatizado que permita optimizar y perfeccionar dichos procesos.

A continuación, se describen los actores del negocio, los casos de uso asociados y el diagrama correspondiente.

2.6.1. Actores del Negocio

En el proceso de gestión de cobros de matrículas y pensiones dentro de la Unidad Educativa Genoveva Germán intervienen diversos actores institucionales que participan de manera directa en las actividades administrativas, financieras y académicas. Estos actores operan dentro de un flujo estructurado que requiere coordinación, validación de datos y control documental. A continuación, se describen los principales:

1. **Secretaria:** Responsable de registrar a los estudiantes al inicio de cada período lectivo. Su función consiste en recopilar y validar los datos académicos básicos del estudiante, y derivar la información correspondiente a colectoría para fines financieros.
2. **Colectora:** Encargada de ingresar, validar y gestionar los pagos por concepto de matrícula y pensiones. Además, genera reportes administrativos de ingresos y deudas y mantiene actualizada la información financiera institucional.

3. Rectora: Autoridad superior de la institución educativa. Supervisa el estado financiero mediante la revisión de reportes generados por colecturía y toma decisiones administrativas basadas en dichos informes. Su acceso al sistema es exclusivamente en modo lectura.

A continuación, se detallan los principales casos de uso del negocio identificados en la Unidad Educativa Genoveva Germán:

Tabla 13 Historia de Usuario 1

| Historia de Usuario | | | |
|--|---|----------------------|----------------------|
| Cód. | HU-01 | Usuario: | Secretaria |
| Descripción: | Como secretaria, quiero registrar la inscripción de los estudiantes para que puedan acceder a los servicios académicos y administrativos. | | |
| Sprint asignado: | 1 | Responsables: | Equipo de Desarrollo |
| Criterio de aceptación: | | | |
| <ul style="list-style-type: none"> • Inicia sesión correctamente. • Registra estudiantes nuevos con todos sus datos. • Los datos se almacenan correctamente y son visibles para la colectora. | | | |

Nota: La tabla contiene información detallada sobre el desarrollo de la Historia de Usuario, obtenida a través de reuniones y entrevistas con los involucrados en el proyecto. Elaboración y fuente propia.

Tabla 14 Historia de usuario 2

| Historia de Usuario | | | |
|--|---|----------------------|----------------------|
| Cód. | HU-02 | Usuario: | Colectora |
| Descripción: | Como colectora, quiero verificar los datos de inscripción enviados por la secretaria para validar que la información sea correcta antes de realizar cobros. | | |
| Sprint asignado: | 1 | Responsables: | Equipo de Desarrollo |
| Criterio de aceptación: | | | |
| <ul style="list-style-type: none"> • La colectora puede visualizar la lista de estudiantes inscritos. | | | |

- Puede acceder a los detalles de cada inscripción.
- Puede marcar como verificada o rechazada una inscripción

Nota: La tabla contiene información detallada sobre el desarrollo de la Historia de Usuario, obtenida a través de reuniones y entrevistas con los involucrados en el proyecto.

Tabla 15 Historia de usuario 3

| Historia de Usuario | | | |
|--------------------------------|---|----------------------|----------------------|
| Cód. | HU-03 | Usuario: | Colectora |
| Descripción: | Como colectora, quiero generar reportes de pagos y deudas para entregar información financiera actualizada a las autoridades. | | |
| Sprint asignado: | 1 | Responsables: | Equipo de Desarrollo |
| Criterio de aceptación: | <ul style="list-style-type: none"> • Puede filtrar los reportes por estudiante, curso o mes. • Ofrece la opción de visualizar y exportar los reportes en formato PDF • La información proporcionada es exacta y está actualizada | | |

Nota: La tabla contiene información detallada sobre el desarrollo de la Historia de Usuario, obtenida a través de reuniones y entrevistas con los involucrados en el proyecto

Tabla 16 Historia de usuario 4

| Historia de Usuario | | | |
|--------------------------------|---|----------------------|----------------------|
| Cód. | HU-04 | Usuario: | Rectora |
| Descripción: | Como rectora, quiero revisar los reportes de cobros para controlar el estado financiero de la institución. | | |
| Sprint asignado: | 2 | Responsables: | Equipo de Desarrollo |
| Criterio de aceptación: | <ul style="list-style-type: none"> • Puede acceder en modo lectura únicamente. • Consulta reportes por curso, paralelo o período. • La información está protegida contra modificaciones. | | |

Nota: La tabla contiene información detallada sobre el desarrollo de la Historia de Usuario, obtenida a través de reuniones y entrevistas con los involucrados en el proyecto.

Las historias de usuario presentadas en esta sección permiten definir de forma clara y estructurada las necesidades y expectativas de los distintos actores involucrados en los procesos de gestión de cobros dentro de la Unidad Educativa Genoveva Germán. Estas historias constituyen insumos esenciales para la organización de los sprints del proyecto y el diseño funcional del sistema, ya que establecen criterios claros de aceptación que deben cumplirse para considerar una funcionalidad como terminada.

Asimismo, este método orientado al usuario asegura que el desarrollo del sistema se adapte a situaciones reales del entorno institucional, optimizando la eficiencia operativa y la experiencia de los usuarios finales.

Product Backlog

Como parte del desarrollo del sistema para la gestión de cobros de matrícula y pensiones en la Unidad Educativa Genoveva Germán, se elaboró el Product Backlog a partir del levantamiento de requerimientos en entrevistas, observación, y procesos administrativos. Esta herramienta es un pilar clave dentro de la metodología ágil adoptada, pues permite organizar, priorizar y dar seguimiento a las funcionalidades imprescindibles para la implementación.

El Product Backlog está conformado por historias de usuario provenientes de la secretaria, la colecturía y la rectoría. Cada una de estas historias representa una necesidad particular dentro de los procesos, enfocándose en aspectos como la mejora del registro de estudiantes, el control de pagos, la generación de reportes y la gestión educativo-administrativa.

Estas historias han sido organizadas en sprints, lo que permite un desarrollo por etapas. Esta metodología facilita que el sistema se ajuste a las prioridades establecidas, favoreciendo la transparencia financiera, reduciendo errores administrativos y aumentando la eficiencia de los procesos.

Tabla 17 Product Backlog

| Código HU | Requerimiento | Sprint | Objetivo |
|-----------|--------------------------------------|--------|--|
| HU-01 | Registrar inscripción del estudiante | 1 | Permitir que la secretaria registre nuevos estudiantes en el sistema. |
| HU-02 | Verificar datos de inscripción | 1 | Permitir que la colectora valide la información de los estudiantes. |
| HU-03 | Registrar pago de matrícula | 2 | Permitir que la colectora registre el pago de matrícula. |
| HU-04 | Registrar pago de pensión | 2 | Permitir que la colectora registre los pagos mensuales de pensiones. |
| HU-05 | Generar reportes de cobros | 3 | Permitir que la colectora genere reportes detallados de pagos y deudas. |
| HU-06 | Revisar reportes financieros | 4 | Permitir que la rectora acceda a los reportes generados en modo lectura. |

2.7. Diagrama general de casos de uso del negocio

El diagrama de casos de uso del negocio representa de forma gráfica las interacciones clave entre los actores administrativos que participan en el proceso de gestión de cobros dentro de la Unidad Educativa Genoveva Germán. Este modelo permite visualizar cómo se ejecutan actualmente los procesos administrativos antes de su automatización, así como las responsabilidades asignadas a cada actor.

Actores identificados:

1. **Secretaria:** Encargada de ingresar los datos del estudiante para la inscripción inicial y derivarlos a colecturía.
2. **Colectora:** Responsable de registrar pagos de matrícula y pensiones, así como de generar reportes periódicos de cobros y estado financiero.
3. **Rectora:** Persona con autoridad que revisa los reportes generados para supervisar el estado financiero de la institución. Cuenta con acceso únicamente de lectura en el sistema.

Casos de uso representados:

1. Verificar datos de inscripción: Proceso llevado a cabo por la colectora y supervisado por la rectora, con el fin de confirmar la precisión de la información de la estudiante registrada por la secretaria.
2. Generar reporte de cobros: La colectora emite reportes periódicos que la rectora puede consultar.
3. Registrar pago de matrícula: Ejecutado por la colectora al momento de recibir el pago del representante.
4. Registrar pago de pensión: Similar al anterior, aplicado al pago mensual.
5. Consultar estado de cuenta: Permite al padre de familia acceder a la información de pagos realizados y pendientes.

Casos de uso representados:

1. Registrar inscripción del estudiante: Realizado por la secretaria, permite ingresar los datos personales y académicos del alumno.
2. Verificar datos de inscripción: Acción ejecutada por la colectora para validar que la información ingresada por la secretaria esté completa y correcta.
3. Registrar pago de matrícula: Ejecutado por la colectora al momento de registrar el pago inicial del estudiante.
4. Registrar pagos mensuales de pensión: Realizado por la colectora mensualmente, vinculando el pago con el estudiante.
5. Generar reportes de cobros: La colectora genera informes periódicos sobre

pagos realizados, deudas pendientes y estados de cuenta.

6. Consultar reportes financieros: La rectora accede a los reportes generados para tomar decisiones informadas sobre la situación económica de la institución.

Figura 4 Diagrama de caso de uso del sistema-autenticación y acceso a funciones

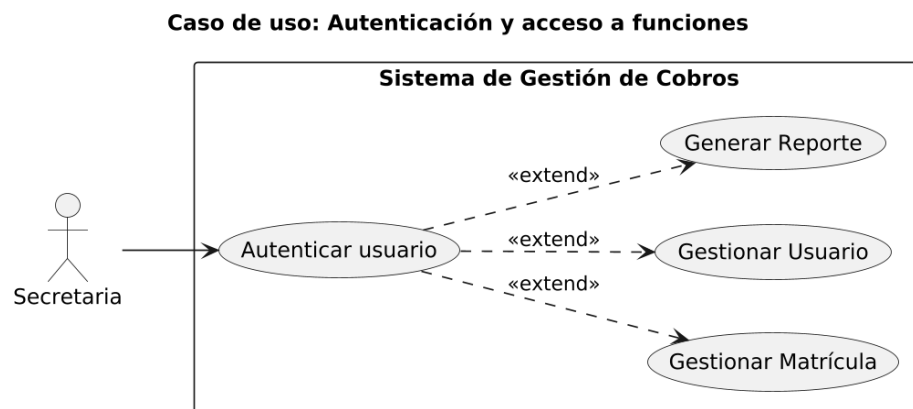


Figura 5 Diagrama de caso de uso- Gestión de usuarios

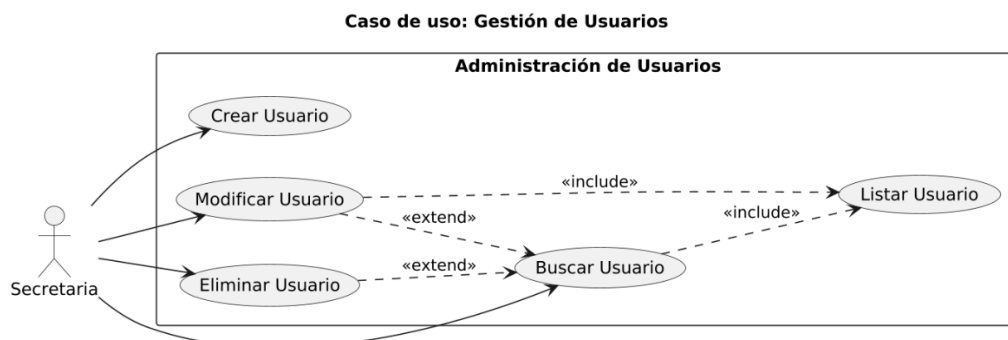
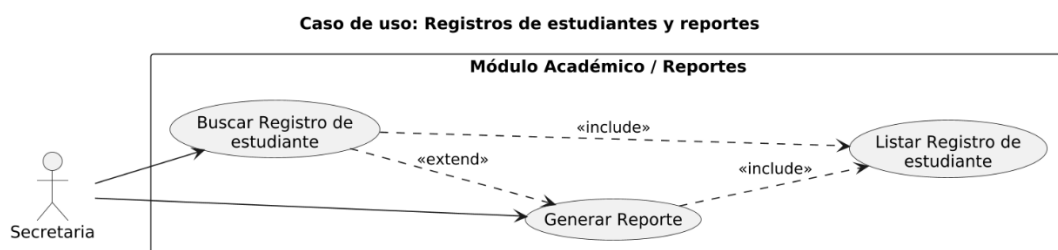


Figura 6 Diagrama de caso de uso- registro de estudiantes



2.7.1. Modelo de objetos del negocio

El modelo de objetos del negocio representa las entidades clave que forman parte de los procesos administrativos de la Unidad Educativa Genoveva Germán en

lo referente a la gestión de cobros. Este modelo permite identificar de forma estructurada los elementos principales que intervienen en la operación institucional, así como sus atributos y relaciones. Constituye la base conceptual sobre la que se construye el diseño de la base de datos y la lógica de funcionamiento del sistema.

Los objetos del negocio reflejan componentes reales del entorno educativo, tales como los estudiantes, los representantes, los pagos realizados y los reportes generados, entre otros. Esta abstracción facilita la comprensión del dominio del problema, mejora la comunicación entre el equipo de desarrollo y los usuarios, y contribuye a la correcta estructuración del sistema de información.

Objetos del Negocio Identificados

A continuación, se describen los objetos principales del negocio, sus características y su función dentro del sistema:

Tabla 18 Objetos del negocio

| Objeto de Negocio | Atributos Principales | Descripción |
|---|--|---|
| Estudiante | ID, nombres, apellidos, curso, paralelo, estado de matrícula | Persona inscrita en la institución, sobre la cual se gestionan los pagos. |
| Padre de familia / Representante | ID, nombres, apellidos, teléfono, correo, relación con el estudiante | Persona responsable del pago de la matrícula y pensiones del estudiante. |
| Matrícula | Código, valor, fecha de pago, estado (pagado/pendiente), estudiante_id | Registro correspondiente al pago único anual por concepto de matrícula. |
| Pensión | Código, mes, valor, fecha de pago, estado, estudiante_id | Registro de pagos mensuales realizados por el estudiante durante el año. |
| Usuario | Nombre de usuario, contraseña y rol (administrador, colectora, rectora, padre) | Representa a cada persona con acceso al sistema. |

| | | |
|-----------------------|--|---|
| Reporte | Código, tipo de reporte, período, fecha de generación, total recaudado, generador_id | Documento generado con información consolidada de cobros y deudas. |
| Curso/Paralelo | Código, nombre, nivel educativo | División académica en la que está inscrito el estudiante. |
| Rol | ID, nombre del rol, permisos asociados | Define las acciones permitidas para cada tipo de usuario en el sistema. |

Relación entre Objetos

Las relaciones más relevantes entre los objetos del negocio son:

- Un Padre de familia puede estar vinculado a uno o más Estudiantes.
- Un Estudiante puede tener una Matrícula y varias Pensiones asociadas.
- Los Pagos (matrículas y pensiones) se vinculan con el Estudiante correspondiente.
- Cada Reporte puede ser generado por un Usuario con rol de colectora.
- Los Usuarios se asocian a un Rol que define sus privilegios dentro del sistema.

2.7.2. Reglas del negocio

En el desarrollo del sistema para la gestión de cobros de matrícula y pensiones en la Unidad Educativa Genoveva Germán, es fundamental establecer un conjunto de reglas del negocio que definan el comportamiento esperado del sistema conforme a las políticas internas de la institución. Estas reglas son directrices operativas que determinan cómo deben ejecutarse ciertos procesos administrativos, garantizando la integridad, coherencia y seguridad de las operaciones.

La identificación de las reglas del negocio se basa en el análisis del flujo actual de trabajo, entrevistas con los usuarios del sistema (colectora, secretaria, rectora, padres de familia) y observación directa de las actividades realizadas en la institución. Su correcta implementación permite que el sistema no solo cumpla su función técnica, sino que también se adapte a las normativas institucionales y respalde la toma de decisiones basada en datos confiables y estructurados.

Tabla 19 Descripción de Reglas del Negocio

| Código | Regla del Negocio | Justificación |
|---------------|---|---|
| RN-01 | Un estudiante no puede pagar pensión si no ha cancelado la matrícula. | Garantiza que el proceso de inscripción esté completo antes del cobro mensual. |
| RN-02 | Solo la colectora puede registrar, modificar o eliminar pagos de matrícula o pensión. | Asegura control sobre los procesos financieros y evita accesos indebidos. |
| RN-03 | La generación de reportes de pagos está restringida exclusivamente a la colectora, mientras que la rectora únicamente tiene permiso para visualizarlos. | Mantiene la jerarquía de control y confidencialidad de la información financiera. |
| RN-04 | Un padre de familia únicamente puede acceder a la información correspondiente a su hijo o hija. | Protege la privacidad de los datos estudiantiles y financieros. |
| RN-05 | No está permitido registrar el pago de una pensión más de una vez en el mismo mes para un mismo estudiante. | Evita duplicidad de registros y asegura consistencia en la contabilidad. |
| RN-06 | Cada estudiante puede contar con una sola matrícula activa durante un periodo lectivo. | Controla el registro académico y evita inscripciones múltiples por error. |
| RN-07 | Cada usuario del sistema debe tener un rol único con permisos previamente definidos. | Determina las funciones y accesos disponibles según el perfil del usuario. |
| RN-08 | No se puede eliminar un estudiante que tenga deudas pendientes de matrícula o pensiones. | Impide pérdida de información financiera relevante antes de su resolución. |

| | | |
|--------------|--|--|
| RN-09 | El sistema debe registrar la fecha y hora exacta de cada transacción de pago. | Facilita la trazabilidad, auditoría y control de los movimientos financieros. |
| RN-10 | Todos los datos ingresados deben validarse antes de ser almacenados en la base de datos. | Previene errores en la captura de información y garantiza la integridad del sistema. |

2.8. Conclusiones del capítulo

En este capítulo se ha realizado un análisis detallado del modelo del negocio, identificando los procesos actuales relacionados con la gestión de cobros en la Unidad Educativa Genoveva Germán, así como los actores involucrados en dichos procedimientos. Se evidenció que el sistema actual, basado en registros manuales y procedimientos no automatizados, presenta deficiencias que afectan la eficiencia administrativa, la trazabilidad financiera y la toma de decisiones oportunas por parte de las autoridades.

A través del levantamiento de requisitos funcionales y no funcionales, así como la elaboración de historias de usuario, se definieron las necesidades operativas del sistema a desarrollar. Se organizaron los casos de uso y se representaron a través de diagramas UML las interacciones principales entre los actores de la institución, lo que facilitó una visualización clara de los flujos de trabajo que se mejorarán con la solución tecnológica propuesta.

Este análisis es fundamental para el diseño técnico del sistema, garantizando que su implementación futura atienda eficazmente las prioridades reales de la institución educativa. En el siguiente capítulo se abordará la arquitectura del sistema y su desarrollo progresivo, conforme a los principios de la metodología ágil adoptada.

CAPITULO III

3. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCION

3.1. Introducción

En este capítulo se presenta la propuesta de solución tecnológica para optimizar el proceso de gestión de cobros de matrícula y pensiones en la Unidad Educativa Genoveva Germán. La solución planteada consiste en el diseño e implementación de un sistema que permita automatizar los procesos administrativos asociados al registro de pagos, generación de reportes, control de deudas y manejo de usuarios, reemplazando el modelo actual basado en registros manuales y hojas de cálculo.

Esta iniciativa surge de la necesidad institucional de optimizar la eficiencia, precisión y transparencia en la gestión financiera, así como de ofrecer una mejor atención a los padres y representantes. La solución propuesta utiliza herramientas de desarrollo web de código abierto, lo que contribuye a reducir los costos de implementación y facilita la escalabilidad futura del sistema.

En este capítulo se describirán los componentes esenciales de la solución, incluyendo el diseño arquitectónico, la transformación del modelo entidad-relación al modelo relacional, los diagramas de clases, los diagramas de secuencia principales y las interfaces más destacadas.

Además, se justificará el uso de las tecnologías seleccionadas y se evidenciará cómo la propuesta responde a los requerimientos funcionales y no funcionales definidos previamente.

3.2. Análisis del diseño del sistema

3.2.1. Modelo vista controlador (MVC)

El desarrollo del sistema propuesto para la Unidad Educativa Genoveva Germán se basa en el patrón de arquitectura de software Modelo-Vista-Controlador (MVC), el cual permite una estructura modular, flexible y mantenible. Esta arquitectura ha sido ampliamente utilizada en el diseño de aplicaciones web modernas, ya que permite separar de manera clara las responsabilidades de cada componente del sistema, (Necula, 2024)

El patrón MVC divide la aplicación en tres componentes fundamentales:

- **Modelo:** Este componente se encarga de gestionar la lógica de datos y acceso a la información del sistema. En el caso del sistema de gestión de cobros, el modelo representa las entidades clave como estudiantes, padres de familia, matrículas, pensiones, reportes y usuarios, y está directamente vinculado con la base de datos MySQL. El modelo se encarga de procesar las peticiones que recibe del controlador y actualiza la información según las acciones llevadas a cabo por el usuario.
- **Vista:** La vista es la parte de la interfaz gráfica que interactúa con el usuario final, integrada por formularios, paneles de control y páginas visuales diseñadas para realizar acciones como registrar pagos, consultar reportes o revisar estados de cuenta. Estas vistas están personalizadas para ajustarse a diferentes perfiles de usuario, tales como administradores, colectora, padres de familia y rectora, mostrando únicamente las funciones que corresponden a cada rol.
- **Controlador:** El controlador actúa como conexión entre el modelo y la vista. Este componente se encarga de gestionar las solicitudes del usuario, tales como registrar pagos, consultar estados de cuenta o generar reportes. Para ello, ejecuta la lógica correspondiente, accede o actualiza la información en el modelo y finalmente envía la respuesta adecuada a la vista. Su función es fundamental para garantizar el cumplimiento de las reglas del negocio, como evitar que se realice el pago de una pensión sin haber completado previamente el pago de la matrícula.

El uso del patrón MVC en este proyecto permite:

1. Facilitar el mantenimiento del sistema, ya que las modificaciones en la lógica de negocio no afectan directamente a la presentación.
2. Mejorar la escalabilidad y organización del código, facilitando futuras ampliaciones o integración con nuevos módulos.
3. Asegurar una separación de responsabilidades clara, permitiendo al equipo de desarrollo trabajar de forma paralela en diferentes componentes.

Esta arquitectura se ha desarrollado empleando tecnologías como PHP, Laravel, JavaScript, HTML/CSS y MySQL, todas compatibles con el enfoque MVC. Gracias a esta estructura, el sistema propuesto ofrece una solución robusta y adaptable para las necesidades de gestión administrativa de la institución educativa.

Figura 7 Arquitectura del sistema basado en el patrón MVC



La figura muestra la arquitectura general del sistema para la gestión de cobros de matrícula y pensiones, estructurada bajo el modelo MVC. El usuario utiliza el navegador web (Vista) para interactuar con el sistema, enviando sus acciones al controlador de la aplicación. Este controlador se encarga de gestionar y coordinar los diferentes módulos funcionales, tales como la administración de estudiantes, matrículas, pagos y auditoría. Para llevar a cabo las operaciones necesarias, el controlador accede a la base de datos, garantizando así una clara división de responsabilidades que favorece el mantenimiento, la escalabilidad y la seguridad del sistema.

3.2.2. Interacción de los componentes

En el sistema propuesto para la gestión de cobros de matrícula y pensiones, los componentes del patrón Modelo Vista Controlador (MVC) interactúan de la siguiente manera:

- El usuario (administrador, colectora, padre de familia o rectora) realiza una acción en la interfaz, como registrar un pago o consultar un reporte.
- El controlador recibe esta acción y ejecuta la lógica correspondiente, como validar datos o acceder a registros.
- Luego, el controlador accede al modelo, donde se almacenan y gestionan los datos de estudiantes, matrículas, pensiones y reportes.
- Una vez procesados los datos, el controlador entrega el resultado a la vista, que se encarga de mostrar la información actualizada al usuario.
- El ciclo se repite con cada nueva interacción del usuario.

Este flujo es gestionado eficientemente mediante el framework Laravel, que implementa el patrón MVC de forma nativa y permite un desarrollo ágil, modular y seguro del sistema.

3.2.3. Diagrama de clases del diseño

Para comprender la estructura del sistema propuesto, es necesario identificar las clases que conforman su diseño. Según (Rodríguez Rodríguez & Peña Abreu, 2023) una clase representa “la definición de la estructura y el comportamiento de un conjunto de objetos que comparten el mismo patrón estructural y de comportamiento”.

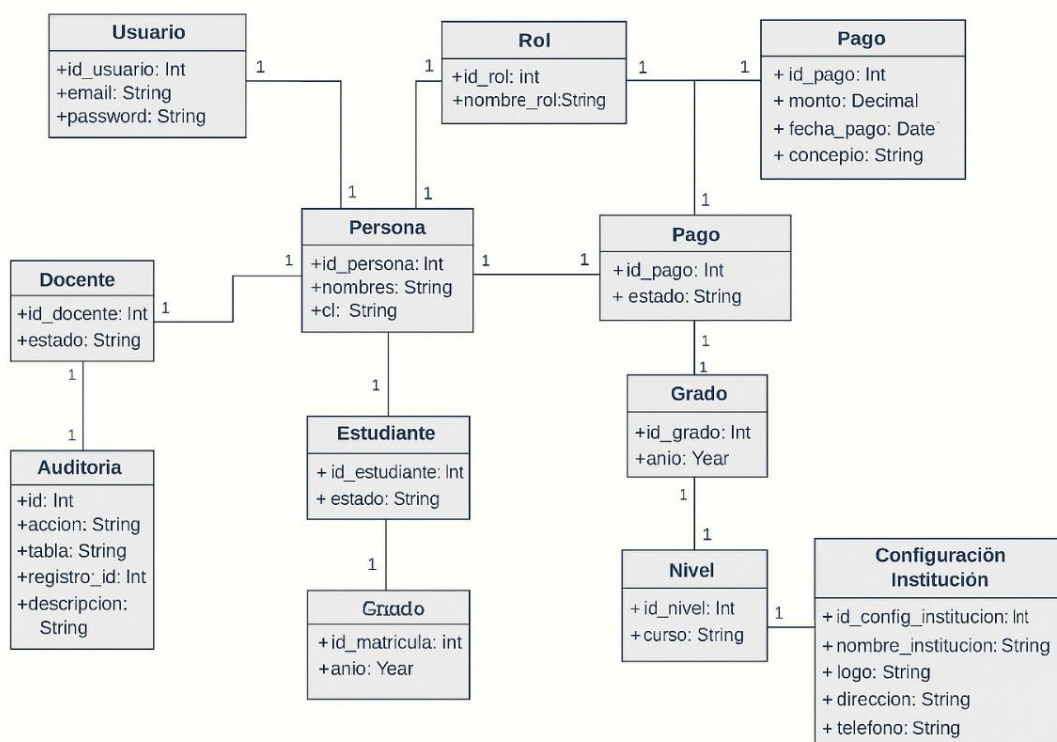
En este proyecto, el diagrama de clases se estructura según los procesos clave del sistema: administración de estudiantes, matrícula, pagos de pensiones, gestión de usuarios y creación de reportes. Cada clase contiene atributos y métodos que reflejan la lógica interna del sistema, siguiendo el patrón Modelo-Vista-Controlador (MVC).

Estas clases reflejan la relación entre entidades clave como Estudiante, Representante, Matrícula, Pensión, Reporte y Usuario, permitiendo una estructura clara, reutilizable y escalable en el desarrollo de la aplicación.

Además, el diagrama de clases facilita la visualización de las dependencias y asociaciones entre los diferentes componentes, lo que permite identificar con

precisión cómo interactúan los objetos dentro del sistema. Esta representación no solo sirve como guía para la codificación, sino que también ayuda en la detección temprana de posibles redundancias o inconsistencias en el diseño. Al mantener una estructura coherente y modular, se garantiza que futuras ampliaciones o modificaciones puedan realizarse de manera eficiente, asegurando la mantenibilidad y adaptabilidad del sistema frente a nuevas necesidades o cambios en los procesos institucionales, (Aleman & Thomas Sosa, 2024)

Figura 8 Diagrama de clases



3.2.4. Diagrama de secuencia

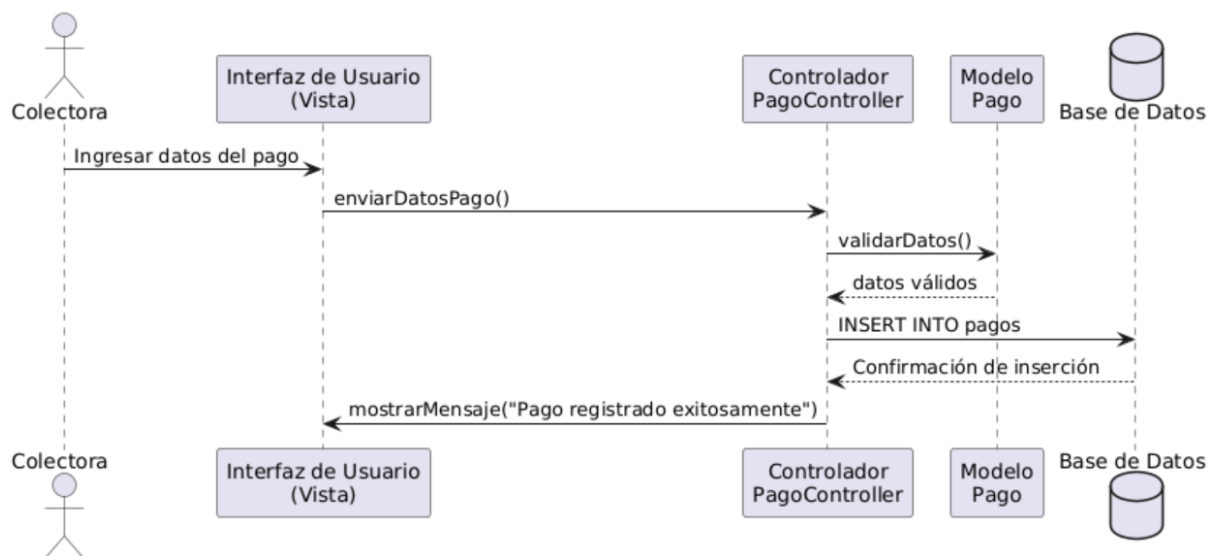
El **diagrama de secuencia** se construye en función de los procesos previamente definidos en el diseño del sistema propuesto para la Unidad Educativa Genoveva Germán. Esta herramienta permite representar gráficamente la interacción temporal entre los diferentes componentes del sistema, tales como interfaces de usuario, controladores y la base de datos, en la ejecución de funcionalidades clave como el registro de pagos (matrícula y pensiones) la consulta de estados de cuenta la generación de reportes financieros y la gestión de usuarios administrativos.

El flujo de mensajes ilustrado en este diagrama refleja cómo el sistema

responde ante las acciones de los actores institucionales (colectora, administrador y rectora) asegurando el cumplimiento de las reglas del negocio bajo la estructura del patrón arquitectónico Modelo-Vista-Controlador (MVC). Este modelo promueve una separación lógica entre los componentes, facilitando la escalabilidad, el mantenimiento del sistema y la comprensión del flujo interno de procesos.

Asimismo, el diagrama de secuencia permite validar el diseño funcional del sistema verificar la sincronización entre los módulos e identificar posibles puntos de mejora antes de la implementación final.

Figura 9 Diagrama de secuencia de registro de pago



La figura muestra el flujo de interacción entre los componentes del sistema durante el proceso de registro de un pago de matrícula por parte de la colectora. Se representa la comunicación entre la interfaz de usuario, el controlador PagoController, el modelo Pago y la base de datos, siguiendo el patrón arquitectónico Modelo Vista Controlador, este diagrama permite visualizar el orden lógico de los mensajes asegurando la validez de los datos antes de su almacenamiento definitivo y la retroalimentación correspondiente al usuario.

3.2.5. Modelo lógico de la base de datos

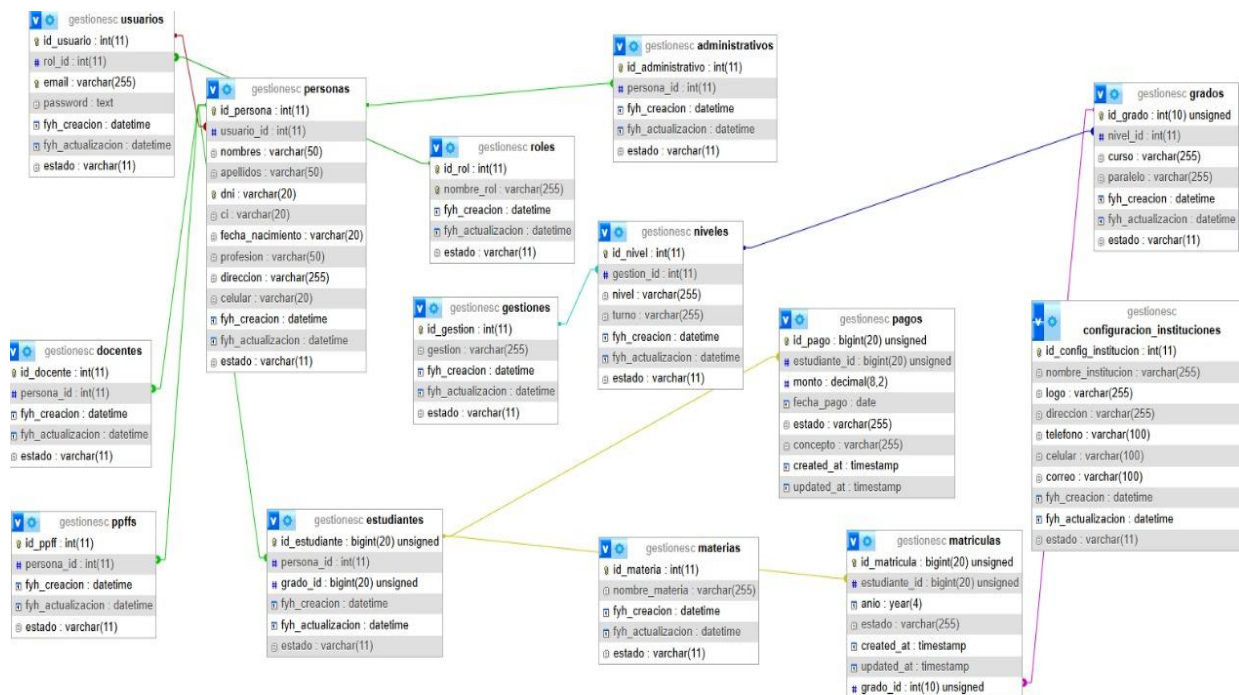
El modelo lógico describe las entidades y relaciones necesarias para la operación del sistema. Incluye:

- Relaciones uno a muchos entre Representante y Estudiantes.
- Asociación entre Estudiantes y sus Pagos.

- Asociación entre Pagos y Reportes.
- Asociación de Usuarios con Roles y Permisos.

Este modelo se construyó usando un enfoque relacional para asegurar la integridad referencial y la eficiencia del acceso a los datos.

Figura 10 Modelo lógico de la base de datos



El diagrama entidad-relación presentado refleja la estructura de la base de datos del sistema, evidenciando la interconexión entre las distintas tablas que conforman la gestión académica, administrativa y financiera. Cada entidad está diseñada para cumplir una función específica, desde la administración de usuarios y roles, hasta el registro de pagos, matrículas y configuraciones institucionales. Las relaciones entre tablas permiten mantener la integridad referencial y optimizar las operaciones CRUD, garantizando consistencia en la información almacenada. Esta organización lógica favorece la escalabilidad y facilita la implementación de nuevas funcionalidades sin comprometer la estructura existente.

3.2.6. Diagrama de componentes

El sistema está compuesto por los siguientes módulos:

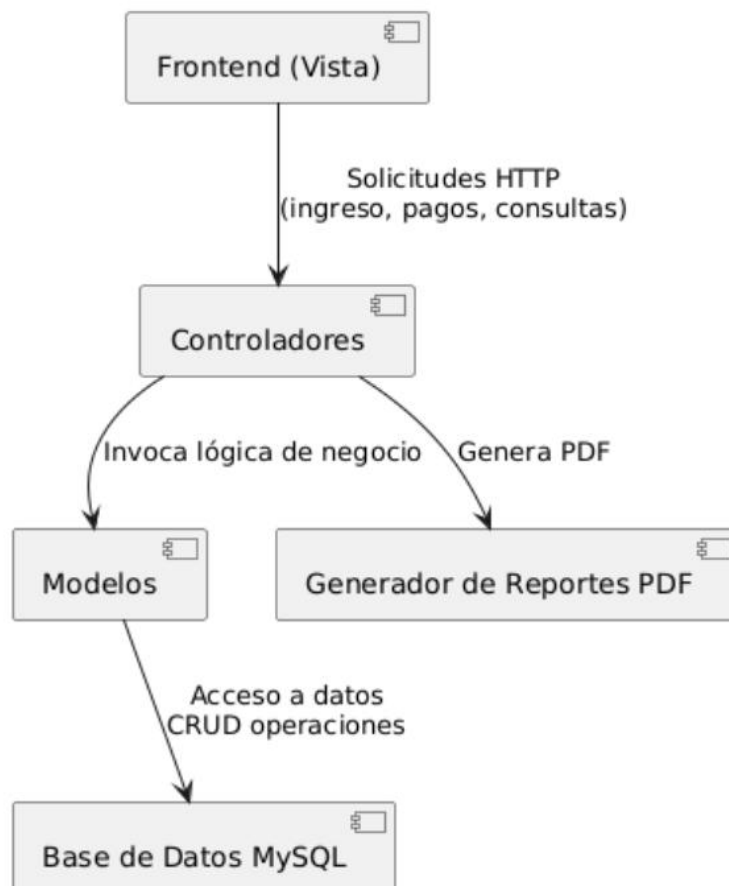
- Frontend (Vista): Interfaz desarrollada en HTML, CSS y JavaScript.
- Controladores: Gestionan peticiones y respuestas del usuario.

- Modelos: Encapsulan la lógica de negocio.
- Base de datos MySQL: Almacena información del sistema.
- Generador de reportes: Módulo para exportar datos en PDF.

Cada componente trabaja de forma independiente bajo Laravel, lo cual permite modificar o ampliar funcionalidades sin afectar el sistema completo.

Este diseño modular garantiza una alta cohesión y bajo acoplamiento entre los distintos elementos del sistema, lo que facilita su mantenimiento, pruebas y escalabilidad a largo plazo. Además, al implementar la arquitectura MVC en conjunto con el generador de reportes, se logra una separación clara entre la presentación, la lógica de negocio y la gestión de datos. Esto no solo optimiza el rendimiento y la organización interna del código, sino que también permite que futuras integraciones, como nuevos formatos de exportación o cambios en la interfaz, puedan realizarse de manera ágil sin comprometer la estabilidad del resto de componentes.

Figura 11 Diagrama de componentes



La imagen representa el flujo de funcionamiento interno del sistema web bajo

el patrón de arquitectura MVC (Modelo-Vista-Controlador) implementado en Laravel. En la parte superior, el Frontend (Vista) recibe las interacciones del usuario y envía solicitudes HTTP relacionadas con ingreso de datos, pagos y consultas.

Estas solicitudes son procesadas por los Controladores, que se encargan de invocar la lógica de negocio a través de los Modelos, los cuales gestionan el acceso y las operaciones CRUD (crear, leer, actualizar, eliminar) sobre la Base de Datos MySQL.

Paralelamente, los controladores pueden interactuar con el Generador de Reportes PDF, que transforma la información procesada en documentos PDF listos para ser entregados al usuario. Este esquema refleja la separación de responsabilidades y la comunicación ordenada entre los componentes del sistema, optimizando el mantenimiento y escalabilidad del software.

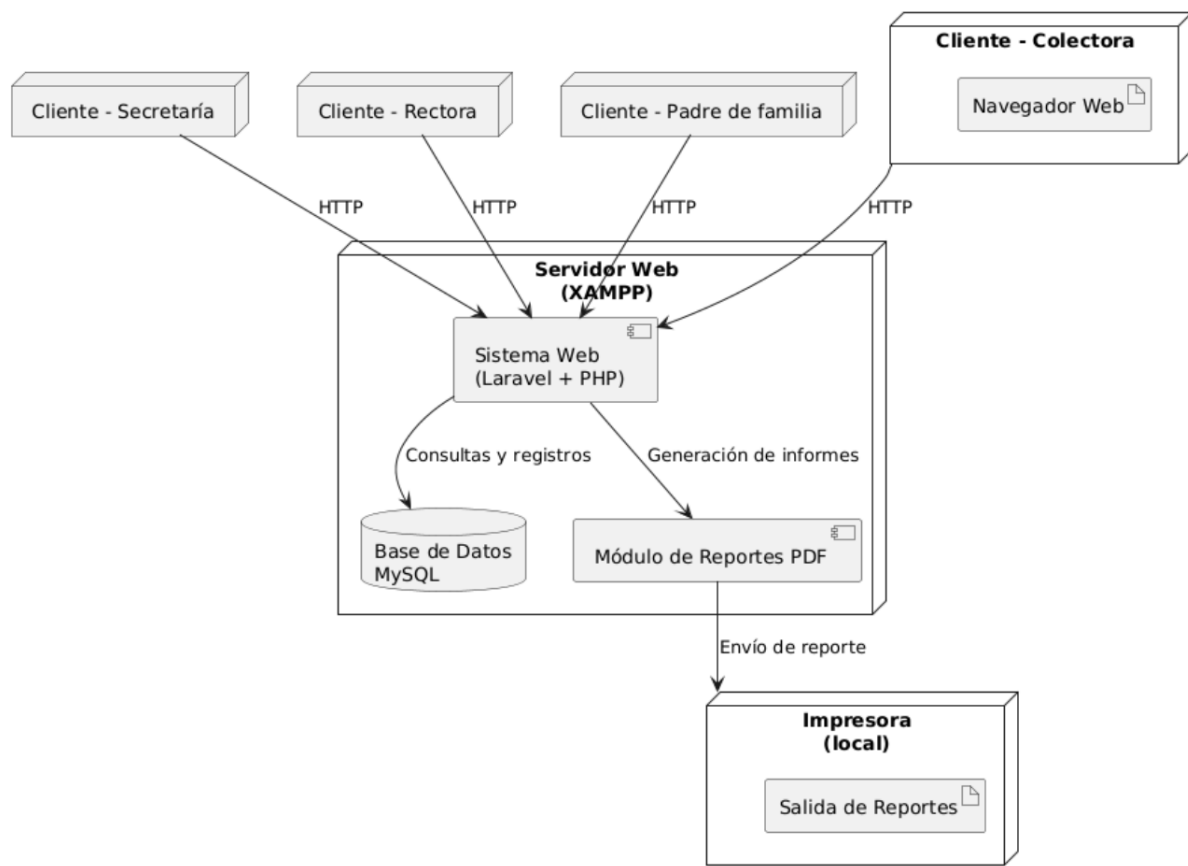
3.2.7. Diagrama de despliegue

El sistema se implementa en una arquitectura cliente-servidor local, compuesta por:

- Servidor web (XAMPP): Aloja el sistema y la base de datos.
- Clientes (estaciones de trabajo): Colectora, secretaría, rectora y padres acceden desde navegadores web.
- Base de datos: Integrada en el mismo servidor local.
- Dispositivo de salida: Impresora conectada para generación de reportes físicos.

Esta configuración permite una implementación económica, accesible y funcional para los recursos actuales de la institución.

Figura 12 Diagrama de despliegue



La figura muestra la arquitectura general del sistema web para la gestión de cobros de matrículas y pensiones, implementado sobre un servidor web con XAMPP que integra el framework Laravel con PHP. Los distintos clientes Secretaría, Rectora, Padres de familia y Colectora acceden al sistema mediante el protocolo HTTP a través de un navegador web. El sistema central se conecta a una base de datos MySQL para realizar consultas y registros de información, y dispone de un módulo de generación de reportes en PDF para elaborar informes financieros y administrativos. Estos reportes pueden ser enviados a una impresora local para su salida física, lo que facilita la distribución y respaldo documental. Esta arquitectura asegura la centralización de la información, el acceso simultáneo por diferentes perfiles de usuario y la generación eficiente de reportes en tiempo real.

3.3. Conclusión

El diseño del sistema propuesto para la Unidad Educativa Genoveva Germán permite optimizar de manera significativa los procesos de cobro de matrícula y pensiones. A través del uso del patrón MVC, una base de datos estructurada y

herramientas de desarrollo modernas, se garantiza la escalabilidad, seguridad y eficiencia del sistema. La propuesta representa un avance tecnológico para la institución, alineado con sus necesidades administrativas y operativas.

CAPÍTULO IV

4. PRUEBA Y ANÁLISIS DE FACTIBILIDAD

En el presente capítulo se detalla la estimación del esfuerzo y el análisis de viabilidad técnica para la implementación del sistema de gestión de cobros de matrícula y pensiones, desarrollado para la Unidad Educativa Genoveva Germán. El objetivo es validar que el prototipo cumple con los requerimientos establecidos, se comporta correctamente bajo distintos escenarios de uso y es factible técnica y funcionalmente dentro del contexto de la institución.

Según Terán Bustamante, Dávila Aragón y Castañón Ibarra (2019), al seleccionar una nueva tecnología es necesario comprender detalladamente el entorno y las aplicaciones existentes, a fin de medir su impacto e integración. Una planificación estratégica adecuada aumenta significativamente la probabilidad de éxito de la solución tecnológica implementada.

4.1. Planificación basada en el método de estimación por puntos de caso de uso

Para realizar la planificación del esfuerzo requerido, se utiliza el método de estimación por puntos de casos de uso propuesto por Karen (1993), el cual se basa en medir la funcionalidad expuesta al usuario mediante casos de uso, sin depender de la tecnología utilizada.

Este método cumple con los siguientes objetivos:

- Medir independientemente de la tecnología utilizada.
- Proporcionar una métrica del tamaño del software.
- Estimar de forma preliminar el esfuerzo de desarrollo.

Este método resulta especialmente útil en las fases iniciales del proyecto, ya que permite obtener una estimación objetiva del alcance funcional del sistema antes de definir con detalle su arquitectura o las herramientas específicas que se emplearán en su construcción. Al centrarse en los casos de uso como unidad de medida, se logra una aproximación más cercana a la perspectiva del usuario final, considerando únicamente las funcionalidades que este podrá utilizar. Además, la aplicación de esta técnica facilita la comparación con proyectos anteriores de naturaleza similar, lo que

contribuye a ajustar los cálculos de esfuerzo y tiempo de desarrollo, optimizando la planificación de recursos humanos, plazos y costos asociados.

4.2. Definición de factores de complejidad

Para estimar la complejidad funcional del sistema, se consideran los siguientes niveles:

Tabla 20 Factores de complejidad de requisitos

| ID | Complejidad del requisito | Peso |
|----|---------------------------|------|
| C1 | Alta | 3 |
| C2 | Media | 2 |
| C3 | Baja | 1 |

Los módulos de matrículas, pagos, y reportes presentan complejidad alta, debido al procesamiento de datos y validaciones. Las interfaces de visualización o consulta se consideran de complejidad media o baja.

4.3. Complejidad según tipo de actor

Tabla 21 Clasificación de actores según complejidad

| ID | Complejidad | Peso | Criterio de clasificación |
|----|-------------|------|--|
| A1 | Alta | 0,5 | Usuario humano a través de GUI |
| A2 | Media | 0,3 | Interfaz tipo consola o protocolo |
| A3 | Baja | 0 | Comunicación por API o sistema externo |

Los actores que interactúan con el sistema son: colectora, rectora, secretaria y padres de familia, todos mediante una interfaz web gráfica, por lo tanto, se clasifican como actores tipo A1 (alta complejidad).

4.3.1. Factores técnicos

Los factores técnicos influyen directamente en el esfuerzo de desarrollo, especialmente en la implementación. A continuación, se detallan aquellos relevantes al sistema propuesto:

Tabla 22 Factores Técnicos

| ID | Factor Técnico | Peso (0,5–2) | Valor (0 o 1) | Justificación |
|-----------|------------------------------------|---------------------|----------------------|---|
| FT1 | Sistema distribuido | 2 | 1 | Arquitectura cliente-servidor |
| FT2 | Rendimiento y concurrencia | 0.5 | 1 | Acceso simultáneo de usuarios |
| FT3 | Complejidad de diseño gráfico | 0.5 | 1 | Interfaces amigables con validaciones visuales |
| FT4 | Algoritmos y procesamiento interno | 0.5 | 1 | Procesos de cálculo de valores, estados de pago |
| FT5 | Seguridad | 1 | 1 | Control de acceso con roles y autenticación |
| FT6 | Volumen de validaciones | 1 | 1 | Validación de montos, fechas, duplicados |
| FT7 | Reutilización | 1 | 1 | Reutilización de componentes en Laravel |
| FT8 | Utilización de estándares | 0.5 | 1 | Buenas prácticas de desarrollo y estándares web |

| | | | | |
|------|---------------------------------------|-----|---|--|
| FT9 | Integración con dispositivos externos | 0.5 | 0 | No se conecta con hardware externo |
| FT10 | Utilización de librerías externas | 1 | 1 | Se emplea Laravel DomPDF, DataTables, etc. |
| FT11 | Interoperabilidad | 0.5 | 1 | Acceso multiplataforma vía navegador |
| FT12 | Entrenamiento especial de usuario | 0.5 | 0 | No requiere capacitación avanzada |

El sistema presenta una complejidad técnica moderada a alta, lo cual exige una planificación adecuada, pero se compensa con la facilidad de uso y el modularidad del framework Laravel.

4.3.2. Factores ambientales

Estos factores evalúan el entorno del equipo desarrollador y condiciones organizacionales.

Tabla 23 Factores ambientales del entorno de desarrollo

| ID | Factor ambiental | Peso |
|----|--|------|
| F1 | Familiaridad con Laravel (en vez de RUP) | 1.5 |
| F2 | Experiencia en sistemas educativos | 0.5 |
| F3 | Experiencia con orientación a objetos | 1 |
| F4 | Capacidades de análisis | 0.5 |
| F5 | Motivación del equipo | 1 |
| F6 | Estabilidad de requisitos | 2 |
| F7 | Personal a medio tiempo | -1 |
| F8 | Lenguaje complejo | -1 |

| | | |
|-----|---|----|
| F9 | Equipo distribuido geográficamente | -1 |
| F10 | Dificultad del lenguaje de programación | -1 |

La Tabla 24 presenta la escala de ponderación utilizada para evaluar el impacto de los diferentes factores técnicos y ambientales en el desarrollo del sistema. Esta escala asigna un valor numérico según el grado de influencia de cada factor: un valor de 3 indica la ausencia del factor, representando el mejor escenario posible; un valor de 2 corresponde a una influencia parcial; y un valor de 1 refleja una influencia total, considerada como el peor caso. Esta clasificación permite cuantificar de manera objetiva las condiciones del entorno y su efecto sobre el proyecto, facilitando el cálculo posterior de métricas como el Factor de Complejidad Técnica y el Factor Ambiental.

Tabla 24 Escala de ponderación

| Valor | Descripción |
|-------|----------------------------------|
| 3 | Ausencia del factor (mejor caso) |
| 2 | Influencia parcial |
| 1 | Influencia total (peor caso) |

4.3.3. Métricas para medir para realizar la estimación

Las métricas a evaluar son:

1. Puntos de Requisitos Sin Ajustar (PRSA).
2. Factor de Peso del Actor Sin Ajustar (FPASA).
3. Factor de Peso de Requisitos Sin Ajustar (FPRSA).
 - Puntos de Requisitos Sin Ajustar (PRSA): $PRSA = FPASA + FPRSA$
 - Factor de Peso del Actor Sin Ajustar (FPASA): $FPASA = \sum_{i=1}^n \text{Peso}_i * \text{Actor}_i$
 - Factor de Peso de Requisitos Sin Ajustar (FPRSA): $FPRSA = \text{Peso}$
 - Factor de Complejidad Técnica (FCT): $FCT = 0,6 + (\sum_{i=1}^n \text{Peso}_i * \text{Valor}_i) / 100$
 - Factor Ambiental (FA): $FA = 1,4 - (0,03 * \sum_{i=1}^n \text{Peso}_i * \text{Valor}_i)$

- Puntos de Requisitos Ajustado (PRA): $PRA = PRSA * FTC * FA$
- El esfuerzo estimado en horas persona: $E = PRA * HH$
- $HH = 10$ horas

Las métricas descritas permiten establecer un cálculo preciso del tamaño y complejidad del sistema, así como del esfuerzo requerido para su desarrollo. A través de la determinación de los Puntos de Requisitos Sin Ajustar (PRSA) y la incorporación de factores como el Factor de Complejidad Técnica (FCT) y el Factor Ambiental (FA), se logra obtener los Puntos de Requisitos Ajustados (PRA), que representan una medida realista del alcance del proyecto.

Con esta información, se calcula el esfuerzo estimado en horas persona, utilizando la equivalencia definida de 10 horas por punto de requisito, lo que permite planificar de forma más eficiente los recursos humanos, los tiempos de desarrollo y la asignación de tareas, asegurando así un mejor control y seguimiento del avance del proyecto.

4.4. Porcentaje de actividades por ciclo de vida

Para asegurar una correcta planificación del desarrollo del sistema de gestión de cobros de matrícula y pensiones, es fundamental distribuir el esfuerzo en cada una de las fases del ciclo de vida del software. Esta planificación permite identificar el porcentaje de tiempo estimado que se invertirá en actividades clave como gestión, análisis, diseño, implementación, pruebas y despliegue.

La siguiente tabla presenta la estimación porcentual basada en un enfoque tradicional de desarrollo, considerando las características y necesidades específicas del sistema propuesto para la Unidad Educativa Genoveva Germán. Esta distribución servirá de guía para la asignación eficiente de recursos y para asegurar que cada fase reciba la atención adecuada para alcanzar un producto final funcional y de calidad.

Tabla 25 Ciclo de vida

| Fase | Porcentaje |
|------------------------|------------|
| Actividades de gestión | 20% |
| Modelado del negocio | 10% |
| Requerimientos | 10% |

| | |
|-------------------|-----|
| Análisis y diseño | 10% |
| Implementación | 30% |
| Pruebas | 10% |
| Despliegue | 10% |

La Tabla 25 muestra la distribución porcentual de las fases que conforman el ciclo de vida del desarrollo del sistema, evidenciando que la mayor dedicación se centró en la fase de implementación con un 30 %, seguida de las actividades de gestión con un 20 %. Las demás etapas, como el modelado del negocio, levantamiento de requerimientos, análisis y diseño, pruebas y despliegue, mantuvieron una asignación equitativa del 10 % cada una. Esta distribución refleja un enfoque equilibrado que prioriza la construcción y puesta en marcha del sistema, sin descuidar las fases iniciales de planificación ni las finales de validación y entrega, asegurando así la calidad y operatividad del producto final.

4.5. Casos de pruebas

Una vez concluido el desarrollo del sistema para la Unidad Educativa Genoveva Germán, se procedió a realizar casos de prueba que permiten validar su funcionalidad, estabilidad y cumplimiento de los requisitos establecidos. Estas pruebas son fundamentales para asegurar que el sistema responde correctamente ante diferentes escenarios de uso real, garantizando así su calidad y fiabilidad antes de su implementación definitiva.

Los casos de prueba se han organizado por módulos principales del sistema: gestión de pagos, gestión de usuarios y generación de reportes. Para cada uno, se especifica el requisito evaluado, los datos de entrada esperados, los resultados previstos y los resultados obtenidos tras la ejecución, así como el estado final de aprobación.

La Tabla 26 describe el Caso de Prueba 1, enfocado en comprobar la funcionalidad de registro de pagos mensuales dentro del sistema. La entrada esperada consistió en la cédula del estudiante, el valor a cancelar y la fecha del pago. El resultado esperado era la confirmación del registro y la visualización inmediata del pago en el sistema. El resultado obtenido confirmó que el pago fue registrado

correctamente, cumpliendo con las validaciones y procesos establecidos. El estado final fue Aprobado, lo que evidencia que el sistema gestiona de manera precisa y eficiente los pagos mensuales, garantizando la actualización oportuna de la información financiera.

Tabla 26 Caso de prueba 1

| Campo | Valor |
|--------------------|--------------------------------------|
| ID del caso | CP01 |
| Requisito | Registrar pagos mensuales |
| Entrada esperada | Cédula del estudiante, valor, fecha |
| Resultado esperado | Confirmación y visualización de pago |
| Resultado obtenido | Pago registrado correctamente |
| Estado | Aprobado |

La Tabla 27 presenta el Caso de Prueba 2 (CP02), cuyo propósito fue validar el registro de un usuario asignándole un rol específico dentro del sistema. La entrada esperada incluyó los datos necesarios para su creación: nombre, correo electrónico, contraseña y el rol correspondiente. El resultado esperado consistía en que el usuario quedara registrado correctamente y contara con los permisos y accesos asociados a su rol. El resultado obtenido confirmó que el sistema asignó el acceso de acuerdo al rol definido, cumpliendo así con la especificación funcional. El estado final fue Aprobado, demostrando que el módulo de gestión de usuarios y roles opera de manera correcta y confiable.

Tabla 27 Caso de pruebas 2

| Campo | Valor |
|------------------|--------------------------------------|
| ID del caso | CP02 |
| Requisito | Registrar usuario con rol específico |
| Entrada esperada | Nombre, correo, contraseña, rol |

| | |
|--------------------|--|
| Resultado esperado | Usuario registrado con acceso adecuado |
| Resultado obtenido | Acceso según rol |
| Estado | Aprobado |

La Tabla 28 describe el Caso de Prueba 3 (CP03), orientado a verificar la funcionalidad de generación de reportes de pagos por estudiante. En esta prueba, la entrada esperada fue el nombre de un estudiante registrado en el sistema, con el objetivo de obtener un archivo PDF que contuviera la información detallada de sus pagos.

El resultado esperado consistía en la correcta generación de dicho documento, y el resultado obtenido confirmó que el sistema descargó el archivo PDF de forma satisfactoria, con los datos correspondientes. El estado final fue Aprobado, lo que evidencia que la funcionalidad cumple con los requisitos establecidos y garantiza la correcta emisión de reportes individuales para control administrativo y financiero.

Tabla 28 Caso de pruebas 3

| Campo | Valor |
|--------------------|---|
| ID del caso | CP03 |
| Requisito | Generar reporte de pagos por estudiante |
| Entrada esperada | Nombre del estudiante |
| Resultado esperado | PDF generado con información de pagos |
| Resultado obtenido | Archivo PDF descargado |
| Estado | Aprobado |

4.6. Análisis de Pruebas de Caja Negra

El sistema propuesto fue evaluado mediante pruebas de caja negra, enfocadas en validar la funcionalidad del software desde la perspectiva del usuario final, sin considerar el código fuente interno. Esta técnica permitió verificar si el sistema respondía adecuadamente ante diversas entradas y escenarios definidos en los

requisitos funcionales.

Tabla 29 Pruebas caja negra

| Nº | Funcionalidad evaluada | Usuario responsable | Entrada de prueba | Resultado esperado | Resultado obtenido | Estado |
|-----------|----------------------------------|----------------------------|---|--------------------------------------|-------------------------------|---------------|
| 1 | Registro de estudiantes | Secretaria | Datos completos y válidos | Registro exitoso | Registro exitoso | Aprobado |
| 2 | Registro de estudiantes | Secretaria | Campos vacíos o inválidos | Mensaje de error por validación | Error detectado correctamente | Aprobado |
| 3 | Verificación de inscripción | Colectora | Estudiante previamente registrado | Verificación exitosa | Datos verificados | Aprobado |
| 4 | Verificación de inscripción | Colectora | Estudiante no registrado | Rechazo del proceso con mensaje | Mensaje mostrado | Aprobado |
| 5 | Registro de pago de matrícula | Colectora | Datos válidos con estudiante verificado | Pago registrado correctamente | Registro exitoso | Aprobado |
| 6 | Registro de pago de pensión | Colectora | Datos válidos y mes disponible | Registro exitoso | Pago registrado | Aprobado |
| 7 | Generación de reportes de cobros | Colectora | Filtro por mes y paralelo | Reporte generado y disponible en PDF | Reporte correcto | Aprobado |
| 8 | Revisión de reportes financieros | Rectora | Acceso de solo lectura a | Visualización correcta, sin | Datos accesibles | Aprobado |

| | | | | | | |
|----|--------------------------------|--------------------|--|---|----------------------|----------|
| | | reportes generados | opción de modificar | | | |
| 9 | Gestión de usuarios y roles | Administrador | Creación y asignación de roles con datos válidos | Roles creados y aplicados con éxito | Gestión correcta | Aprobado |
| 10 | Gestión de errores del sistema | Todos | Entradas duplicadas, vacías o inconsistentes | Mensaje de error claro y bloqueo de acción incorrecta | Validación funcional | Aprobado |

La tabla presentada resume los resultados de las pruebas funcionales realizadas al sistema de gestión de cobros de matrículas y pensiones, evaluando diez funcionalidades clave en diferentes perfiles de usuario, como Secretaria, Colectora, Rectora y Administrador. Cada caso de prueba contempló entradas válidas e inválidas para verificar la correcta validación de datos, la ejecución de procesos y la generación de reportes. Los resultados obtenidos coincidieron con los resultados esperados en todos los escenarios, confirmando el correcto funcionamiento de registros, verificaciones, pagos, reportes, gestión de usuarios y manejo de errores. En consecuencia, todas las pruebas fueron aprobadas, evidenciando que el sistema cumple con los requisitos establecidos y garantiza su operatividad en condiciones reales de uso.

4.7. Conclusiones del capítulo

La validación del sistema desarrollado para la gestión de cobros en la Unidad Educativa Genoveva Germán demuestra que la solución propuesta es técnicamente viable, funcional y adecuada al entorno institucional. La estimación del esfuerzo mediante puntos de caso de uso, el análisis detallado de factores técnicos y ambientales, la planificación equilibrada del ciclo de vida del software, así como la ejecución de pruebas funcionales y de caja negra, evidencian que el sistema cumple con los requerimientos definidos, responde correctamente ante distintos escenarios y está listo para su implementación real, contribuyendo a mejorar la eficiencia

administrativa, reducir errores manuales y garantizar un mayor control financiero.

CONCLUSIONES

Como resultado del desarrollo de este proyecto, se ha logrado cumplir con los objetivos planteados, permitiendo implementar una solución tecnológica que responde de forma eficaz a las necesidades administrativas de la Unidad Educativa Genoveva Germán. A continuación, se detallan las conclusiones alcanzadas en función de cada objetivo específico:

En primer lugar, el análisis de los requerimientos permitió identificar los procesos críticos que dificultaban la gestión de cobros en la institución, evidenciando problemas como la duplicación de registros, pérdida de información y retrasos en la atención a los padres de familia. Esta fase fue fundamental para comprender el funcionamiento actual y establecer las bases sobre las cuales se construiría la solución.

Posteriormente, se llevó a cabo un estudio detallado sobre las metodologías de desarrollo de software y las herramientas tecnológicas más adecuadas para este tipo de sistema. La selección del enfoque RUP permitió organizar las fases del proyecto de forma estructurada, mientras que el uso de tecnologías como PHP, MySQL, Laravel y JavaScript facilitó la implementación de una plataforma robusta, segura y adaptable al entorno institucional.

Con base en el levantamiento de requisitos, se diseñó una base de datos acorde a las necesidades del sistema, tomando en cuenta los actores, los casos de uso y los requisitos tanto funcionales como no funcionales. Esta etapa permitió establecer una estructura clara de relaciones entre las entidades, garantizando la integridad y coherencia de los datos.

A continuación, se desarrolló el sistema web, incorporando módulos específicos para la gestión académica y financiera, como el registro de estudiantes, control de pagos, generación de reportes y asignación de roles de usuario. El diseño centrado en la funcionalidad y la facilidad de uso permitió que el sistema se integrara de forma efectiva en las tareas cotidianas del personal administrativo.

Finalmente, se realizaron pruebas funcionales y de validación utilizando técnicas como las pruebas de caja negra, a fin de garantizar el correcto funcionamiento del sistema en condiciones reales. Los resultados obtenidos confirmaron que la plataforma cumple con los requisitos establecidos, mejorando

notablemente la eficiencia en la gestión de cobros, reduciendo errores y optimizando los tiempos de atención.

RECOMENDACIONES

Para asegurar una utilización adecuada los secretarios, recaudadores y otro personal administrativo que estará interfiriendo con el sistema deben ser capacitados con cursos especializados adaptados a sus roles. Una capacitación apropiada ayudará a minimizar errores promoviendo la independencia del personal en las operaciones diarias.

Se aconseja establecer sistemas de respaldo automatizados para la base de datos para protegerse contra la pérdida de información crítica debido a fallos técnicos u otros incidentes inesperados. Esto mejorará la protección y la confianza en la administración institucional al garantizar la preservación de los datos académicos y financieros dentro del sistema.

Además, establecer otras funcionalidades del sistema como mensajería interna alertas de pagos atrasados o integración con sistemas bancarios requiere designar cronogramas periódicos de actualización y mantenimiento del sistema que no se han mencionado previamente.

Otra recomendación importante es estudiar la posibilidad de proporcionar acceso remoto controlado de forma segura a representantes legales para visualizar cuentas y generar documentos requeridos externamente esto mejorará la apertura institucional y el compromiso familiar.

Por último, la acción más importante que la institución debe tomar es realizar auditorías internas periódicas de los datos registrados para verificar la información y mantener la integridad de la institución.

BIBLIOGRAFÍA

- Aleman, Y., & Thomas Sosa, Y. (2024). *Módulo de configuración para el mecanismo de rastreo del buscador Orión*. Recuperado el 7 de febrero de 2025, de Universidad de las Ciencias Informáticas: https://www.researchgate.net/publication/264002736_Modulo_de_configuracion_para_el_mecanismo_de_rastreo_del_buscador_Orion
- Annala, M. (2022). *Exploring the Power of MySQL: A Comprehensive Guide*. Recuperado el 25 de enero de 2025, de https://www.researchgate.net/publication/379693876_Exploring_the_Power_of_MySQL_A_Comprehensive_Guide
- Apiag, P. C., Cádiz, E. B., & Lincopinis, D. (2023). *A Review on PHP Programming Language*. Recuperado el 17 de enero de 2025, de https://www.researchgate.net/publication/371166635_A_Review_on_PHP_Programming_Language
- Camayo Ávila, M., & Maita Diaz, D. (2025). Herramientas tecnológicas en educación: Revisión sistemática. *Horizontes*, 548–560. doi:<https://doi.org/10.33996/revistahorizontes.v9i36.937>
- Durvas Jayaraman, K. (2025). *Mastering net core and entity framework: building secure and scalable web applications mastering net core and entity framework: building secure and scalable web applications*. Recuperado el 3 de diciembre de 2024, de https://www.researchgate.net/publication/390335591_MASTERING_NET_CORE_AND_ENTITY_FRAMEWORK_BUILDING_SECURE_AND_SCALABLE_WEB_APPLICATIONS_MASTERING_NET_CORE_AND_ENTITY_FRAMEWORK_BUILDING_SECURE_AND_SCALABLE_WEB_APPLICATIONS
- Espinoza, D., & Mendez, F. (2022). Viabilidad comercial y técnica del procesamiento de pulpa de naranjilla: Limón Indanza, Morona Santiago- Ecuador. *MQRInvestigar*, 1143–1164. doi:<https://doi.org/10.56048/MQR20225.6.3.2022.1143-1164>
- Esquirol-Caussa, J., & Sánchez-Aldeguer, J. (2020). La revisión bibliográfica base fundamental de la investigación. *Actualizaciones en Fisioterapia*. Recuperado

el 18 de diciembre de 2024, de <https://www.researchgate.net/publication/319260924>

Gómez Peñuela, J., Silva Labarca, G., & Ovalles Pabón, F. (2019). Análisis de la Propuesta de un Sistema de Información para la Gestión de los Ambientes de Formación del SENA CIES Cúcuta. *Mundo Fesc*, 31-35. doi:<https://doi.org/10.61799/2216-0388.445>

Góngora Duarte, A., Morales Espitia, F., & Trujillo-González, J. (2022). *Tecnologías tradicionales para el beneficio de producciones a pequeña escala*. Recuperado el 3 de diciembre de 2024, de https://www.researchgate.net/publication/362932020_TECNOLOGIAS_TRADICIONALES_PARA_EL_BENEFICIO_DEL_CACAO_EN_PRODUCCIONES_A_PEQUENA_ESCALA

Hoefling, S. (2022). *Microsoft Graph, Web APIs, and MyFilesPage*. Recuperado el 17 de enero de 2025, de https://www.researchgate.net/publication/364075248_Microsoft_Graph_Web_APIs_and_MyFilesPage

Honest, N., Patel, A., & Patel, B. (2013). Sessionization Process for the Pages Designed with the Concept of CMS. *International Journal of Advanced Research in Computer Science and Software Engineering*, 362-368. Recuperado el 27 de enero de 2025, de https://www.researchgate.net/publication/275713255_Sessionization_Process_for_the_Pages_Designed_with_the_Concept_of_CMS

Iqbal, S., Al Azzoni, I., Allen, G., & Ullah Khan, H. (2020). Extending UML Use Case Diagrams to Represent. *e-Informatica Software Engineering Journal*. Recuperado el 3 de diciembre de 2024, de <https://www.e-informatyka.pl/index.php/einformatica/volumes/volume-2020/issue-1/article-4/>

Juárez López, H. A., Ramírez Jiménez, M., & Rivera Orozco, C. E. (2020). UML: una manera de representar, interpretar, analizar y desarrollar el pensamiento computacional, caso tecnologías de la información e ingeniería en videojuegos en Culagos - UDG. *Tecnologías educativas y estrategias didácticas*.

- Recuperado el 13 de enero de 2025, de <https://dialnet.unirioja.es/servlet/articulo?codigo=7787803>
- Kimmel, P. (18 de Marzo de 2019). *Manual de UML*. México: McGraw Hill. Recuperado el 3 de diciembre de 2024, de https://www.academia.edu/29446479/Manual_de_UML_-_Paul_Kimmel.pdf
- Koc, H., & Mert Erdoğan, A. (2021). UML Diagrams in Software Engineering Research: A Systematic Literature Review. *Proceedings*. Recuperado el 6 de enero de 2025, de https://www.researchgate.net/publication/349973644_UML_Diagrams_in_Software_Engineering_Research_A_Systematic_Literature_Review
- Kruchten, P. (2022). *The Rational Unified Process--An Introduction*. Recuperado el 3 de enero de 2025, de https://www.researchgate.net/publication/220018149_The_Rational_Unified_Process--An_Introduction
- Leuthold, S. (2021). *User Interface, Navigation Design and Content Representation: Three Perspectives on World Wide Web Navigation*. Recuperado el 3 de diciembre de 2024, de University of Basel: https://www.researchgate.net/publication/228584191_User_Interface_Navigation_Design_and_Content_Representation_Three_Perspectives_on_World_Wide_Web_Navigation
- Madan, C., & Teitge, B. (2021). The Benefits of Undergraduate Research: The Student's Perspective. *The Mentor*. Recuperado el 19 de diciembre de 2024, de <https://journals.psu.edu/mentor/article/view/61274/60907>
- Marzuki Shofi, I., Damar Nugraha, R., Anggraini, N., & Hakiem, N. (2023). *Implementation of Organization Goal-Oriented Requirements Engineering (OGORE) Method in Designing a Muhammadiyah High School Library Information System*. doi:10.1007/978-1-4842-2499-1_4
- Narváez, D., & Triviño, L. (2011). *Eco parque de Guayaquil, una propuesta de recolección de residuos en Guayaquil*. Recuperado el 3 de diciembre de 2024, de Escuela Superior Politécnica del Litoral: <http://www.dspace.espol.edu.ec/handle/123456789/19512>

- Necula, S. (2024). *Exploring The Model-View-Controller (MVC) Architecture: A Broad Analysis of Market and Technological Applications*. Recuperado el 3 de diciembre de 2024, de https://www.researchgate.net/publication/380197155_Exploring_The_Model-View-Controller_MVC_Architecture_A_Broad_Analysis_of_Market_and_Technological_Applications
- Páez Cárdenas, P., Arias Peralta, C., & Wanumen Silva, L. (2018). Metodología para elaboración de requerimientos en aplicaciones con servicios web. *Revista Vínculos*, 160-174. Recuperado el 3 de diciembre de 2024, de https://www.researchgate.net/publication/336624121_Metodologia_para_elaboracion_de_requerimientos_en_aplicaciones_con_servicios_web
- Pérez A., O. (2023). Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM. *Inventum*. Recuperado el 5 de enero de 2025, de https://www.researchgate.net/publication/320984310_Cuatro_enfoques_metodologicos_para_el_desarrollo_de_Software_RUP_-_MSF_-_XP_-_SCRUM
- Puerto Moreno , S. (2020). *Esencialización de la práctica gestión de requisitos de rup*. doi:<https://doi.org/10.31637/epsir-2024-1070>
- Ramírez Jiménez, M., Pulido-Hernández, K., Gómez Torres, N., & Rivera-Orozco, C. (2024). UML: Una manera de representar, interpretar, analizar y desarrollar el pensamiento computacional. *Ride: Revista Iberoamericana para la Investigación y el Desarrollo Educativo*. Recuperado el 8 de enero de 2025, de https://www.researchgate.net/publication/386396915_UML_Una_manera_de_representar_interpretar_analizar_y_desarrollar_el_pensamiento_computacional
- Rodríguez Rodríguez, C. R., & Peña Abreu, M. (2023). *Herramienta para aplicar métricas al Diagrama de Clases del Diseño Orientado a Objetos*. Recuperado el 4 de febrero de 2025, de https://www.researchgate.net/publication/313794326_Herramienta_para_aplicar_mtricas_al_Diagrama_de_Clases_del_Disenio_Orientado_a_Objetos

- Roman Honores, C., & Ormaza Andrade, J. (2022). Estudio de viabilidad comercial, técnica y económica en la línea de cosméticos, Ecuador. *Visionario Digital*, 29-51. doi:<https://doi.org/10.33262/visionariodigital.v6i4.2292>
- Sánchez Pacheco, C. A. (2020). Vulnerabilidad del protocolo Mysql en redes LAN Bajo plataforma LINUX. *Telematique*. Recuperado el 16 de febrero de 2025, de https://www.researchgate.net/publication/28290829_Vulnerabilidad_del_protocolo_Mysql_en_redes_LAN_Bajo_plataforma_LINUX
- Sánchez Tudela, M., Molina Real, M., & Tudela-Rodríguez, F. (2025). Diálogo entre las nuevas tecnologías y las técnicas tradicionales para redefinir la educación patrimonial. *Revista PH*. Recuperado el 3 de enero de 2025, de https://www.researchgate.net/publication/386266729_Dialogo_entre_las_nuevas_tecnologias_y_las_tecnicas_tradicionales_para_redefinir_la_educacion_patrimonial
- Smith, J. (2023). *Programming Entity Framework Core*. Apress. Recuperado el 3 de diciembre de 2024, de <https://github.com/Apress/beg-entity-framework-core-5>
- Tan, J., Chen, Y., & Jiao, S. (2023). Visual Studio Code in Introductory Computer Science Course: An Experience Report. *ArVix*. Recuperado el 20 de enero de 2025, de <https://arxiv.org/abs/2303.10174>
- Viladas Jené, X. (2023). *El diseño a su servicio : cómo mejorar una idea de negocio con la ayuda de un diseñador*. Obtenido de file:///C:/Users/carrielp/Downloads/Eldiseodeservicioscomonuevocampodeespecializacinprofesional.pdf
- Villa Quishpe, M., Centeno Atavallo, C., Gavilánez Guanoluisa, Á., & Cadena Moreano, J. (2023). Herramientas para el desarrollo de software. Una revisión. *Revista Científica de Informática Encriptar*. Recuperado el 11 de enero de 2025, de https://www.researchgate.net/publication/379562767_Herramientas_para_el_desarrollo_de_software_Una_revision
- Wildermuth, S. (2022). *ASP.NET Core Dependency Injection*. Recuperado el 3 de diciembre de 2024, de <https://blog.wildermuth.com/2016/08/07/ASP-NET-Core-Dependency-Injection/>

Yusup Ali, E., Permana, J., & Achmad Kurniady, D. (2020). Improving the Quality of Academic Services in Higher Education by the Development of SIMANTAP. *Universal Journal of Educational Research*. Recuperado el 25 de enero de 2025, de https://www.researchgate.net/publication/339455060_Improving_the_Quality_of_Academic_Services_in_Higher_Education_by_the_Development_of_SIMANTAP