

UNIVERSIDAD METROPOLITANA DEL ECUADOR



FACULTAD DE INGENIERÍAS

CARRERA DE SISTEMAS DE INFORMACIÓN

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

TEMA:

**DESARROLLO DE UN SISTEMA INFORMÁTICO QUE PERMITA AGILIZAR EL
PROCESO DE REGISTRO DE SOLICITUDES DE PAGO A PROVEEDORES, EL
SEGUIMIENTO Y CONTROL DE LOS PROYECTOS DE CONSTRUCCIÓN PARA LA
EMPRESA NYLIC.**

AUTOR:

JIMMY JESÚS SALAZAR GÓMEZ

ASESOR:

ING. JORGE SOLER MCCOOK MSC.

GUAYAQUIL – 2023

CERTIFICADO DEL ASESOR

ING. JORGE MIGUEL SOLER MCCOOK MSc., 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 Salazar Gómez Jimmy Jesús, portador de la cédula de identidad No. 092127999-8 con el Tema: “Desarrollo de un sistema informático que permita agilizar el proceso de registro de solicitudes de pago a proveedores, el seguimiento y control de los proyectos de construcción para la empresa NYLIC.”. El señor mencionado 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. Jorge Miguel Soler McCook MSc.

TUTOR DE TRABAJO DE TITULACIÓN

CERTIFICADO DE AUTORÍA DE TRABAJO DE TITULACIÓN.

Yo, Salazar Gómez Jimmy Jesús, estudiante de la Universidad Metropolitana del Ecuador "UMET", Ingeniería de Sistemas de Información, declaro en forma libre y voluntaria que el presente trabajo de titulación que versa sobre: DESARROLLO DE UN SISTEMA INFORMÁTICO QUE PERMITA AGILIZAR EL PROCESO DE REGISTRO DE SOLICITUDES DE PAGO A PROVEEDORES, EL SEGUIMIENTO Y CONTROL DE LOS PROYECTOS DE CONSTRUCCIÓN PARA LA EMPRESA NYLIC y las expresiones vertidas en la misma, son autoría de la compareciente, las cuales se han realizado en base a recopilación bibliográfica, consultas de internet y consultas de campo.

Por lo tanto, me hago responsable de asegurar la originalidad de este trabajo y de citar adecuadamente las fuentes bibliográficas correspondientes para respaldar el contenido presentado.

Atentamente,

Jimmy Jesús Salazar Gómez

C.C.: 0921279998

AUTOR

CESIÓN DE DERECHOS DE AUTOR

Yo, Salazar Gómez Jimmy Jesús, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación, Desarrollo de un sistema informático que permita agilizar el proceso de registro de solicitudes de pago a proveedores, el seguimiento y control de los proyectos de construcción para la empresa NYLIC, modalidad Proyecto de Investigación de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, 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 digitalización y publicación de este trabajo de titulación en el repositorio virtual, de conformidad a lo 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,

Jimmy Jesús Salazar Gómez

C.C.: 0921279998

DEDICATORIA

Deseo expresar mi más profundo agradecimiento a Dios, quien ha sido mi guía constante a lo largo de este emocionante viaje académico.

A mis queridos padres, les estoy eternamente agradecido por su amor incondicional, apoyo inquebrantable y sacrificios incansables. Su constante aliento y dedicación me han inspirado a esforzarme y perseguir mis metas.

A mi hermana, le agradezco por su compañía y por ser siempre un apoyo en cada paso del camino. Su presencia ha sido un recordatorio constante de la importancia de la unidad y el apoyo familiar.

A mis amigos, les agradezco por su amistad sincera, consejos y momentos compartidos. Su compañía y aliento han iluminado mis días y han enriquecido mi experiencia.

Este logro es el resultado del amor, apoyo y contribuciones de cada uno de ustedes.

ÍNDICE

CERTIFICADO DE AUTORÍA DE TRABAJO DE TITULACIÓN.....	II
CESIÓN DE DERECHOS DE AUTOR.	III
DEDICATORIA	IV
RESUMEN.....	XII
ABSTRACT.....	XIII
INTRODUCCIÓN.....	1
Situación Problemática	3
Formulación del Problema.....	3
Objetivos.....	4
Objetivo General.....	4
Objetivos específicos.....	4
Justificación del Problema	4
Viabilidad Técnica.....	5
Hipótesis.....	6
Estructura del documento.....	6
CAPITULO I.....	8
1. Fundamentación Teórica	8
1.1. Conceptos asociados a la investigación.....	8
1.2. Objetivos estratégicos de la organización objeto de la investigación.	9
1.2.1. Visión:.....	9
1.3. Objeto de Estudio.....	9
1.3.1. Campo de Acción:.....	9
1.3.2. Flujo actual del Proceso.....	10
1.3.2.1. Análisis de la ejecución de los procesos.....	10
1.3.2.2. Procesos Objeto de automatización.....	11
1.4. Sistemas automatizados existentes relacionados con el campo de acción...	11
1.4.1. Herramientas para la Gestión de Proyectos de Consultoría.....	11

1.4.1.1. Alfresco.....	12
1.4.1.2. Jira.....	12
1.4.1.3. Airtable.....	12
1.4.2. Comparación con algunos Sistemas Informáticos existentes.....	12
1.5. Tendencias y Tecnologías.....	14
1.6. Fundamentación de la Metodología Utilizada.....	14
1.6.1. Metodología RUP.....	14
1.6.2. Lenguaje Unificado de Modelado UML.....	17
1.7. Herramientas y Tecnologías.....	17
1.7.1. Visual Paradigm.....	17
1.7.2. Lenguaje de Programación Visual Basic.Net.....	18
1.7.3. Sistema de Base de Datos Relacional SQL Server.....	19
1.8. Conclusión del Capítulo.....	20
CAPITULO II.....	21
2. Modelo del Negocio.....	21
2.1. Breve descripción del Capítulo.....	21
2.2. Descripción del entorno.....	21
2.2.1. Modelo de casos de uso del negocio.....	22
2.2.2. Diagrama de Casos de uso del negocio.....	23
2.2.3. Actores y Trabajadores del Negocio.....	24
2.2.3.1. Actores del negocio.....	24
2.2.3.2. Trabajadores del Negocio.....	24
2.2.4. Descripción de los Casos de usos del negocio.....	25
2.2.5. Diagrama de Actividades.....	25
2.2.6. Reglas del Negocio para considerar.....	28
2.3. Especificación de los requisitos de Software.....	29
2.3.1. Requisitos funcionales (RF).....	30
2.3.2. Requisitos No Funcionales (RNF).....	32

2.4.	Diagrama de casos de uso del sistema.....	33
2.4.1.	Listado de Casos de uso.	35
2.5.	Descripción de los casos de usos principales del sistema.....	36
2.5.1.	Descripción del Caso de uso “Crear Proyecto”.....	37
2.5.2.	Descripción del Caso de Uso “Crear Contratos”.....	38
2.5.3.	Descripción del Caso de Uso “Crear Pagos a Proveedores”.....	39
2.6.	Conclusión del Capítulo.....	40
CAPITULO III.....		41
3.	Descripción de la Propuesta de Solución.	41
3.1.	Breve descripción del Capítulo.....	41
3.2.	Arquitectura Cliente – Servidor.....	41
3.3.	Arquitectura para Diseño del Sistema (Modelo – Vista – Controlador).	42
3.4.	Diagrama de Clases de Diseño.....	43
3.5.	Diagrama de Secuencia.	45
3.6.	Tratamiento de Errores.....	46
3.7.	Modelo Físico de Datos.....	48
3.8.	Tabla de Base de Datos.....	48
3.9.	Diagrama de Componentes.....	49
3.10.	Diagrama de Despliegue.....	50
3.11.	Conclusión del Capítulo.	51
CAPITULO IV		52
4.	Pruebas y Análisis de Factibilidad.	52
4.1.	Breve descripción del Capítulo.....	52
4.2.	Planificación basada en uno de los métodos de estimación.	52
4.2.1.	Caracterización de los requisitos según su complejidad.....	53
4.2.2.	Clasificación de los requisitos funcionales.....	54
4.2.3.	Definición de factores de complejidad.	55
4.2.4.	Factores Técnicos.....	56

4.2.5. Factores Ambientales.....	58
4.2.6. Métricas para medir para realizar la estimación.....	60
4.2.7. Porcentaje de actividades por ciclo de vida.....	61
4.3. Pruebas de caja negra y su validación.....	65
4.3.1. RF3.1: Crear Registro de Proyecto.....	66
4.4. Verificación del Sistema.....	68
4.5. Errores encontrados durante la Instalación de la Aplicación.....	70
4.6. Conclusión del Capítulo.....	71
CONCLUSIONES.....	71
RECOMENDACIONES.....	72
REFERENCIAS.....	73

ÍNDICE DE TABLAS

Tabla 1. Comparación de Sistemas.....	13
Tabla 2. Detalle de Costo	13
Tabla 3. Comparación de Metodologías.....	14
Tabla 4. Metodología de Trabajo con RUP.....	16
Tabla 5. Componentes de caso de uso.	23
Tabla 6. Descripción de los actores del negocio.....	24
Tabla 7. Descripción de los Trabajadores del Negocio.....	24
Tabla 8. Requerimientos No Funcionales	32
Tabla 9. Priorización de los Casos de Uso.	35
Tabla 10. Descripción de Caso de Uso - Crear Proyecto.	37
Tabla 11. Descripción de Caso de Uso - Crear Contrato.....	38
Tabla 12. Descripción de Caso de Uso - Crear Pago a Proveedores.....	39
Tabla 13. Caracterización de RF según su Complejidad.....	53
Tabla 14. Clasificación de RF	54
Tabla 15. Definición de factores de complejidad.	56
Tabla 16. Definición de factores	56
Tabla 17. Descripción de los Factores técnicos.....	57
Tabla 18. Descripción de los valores de ponderación de factores técnicos.....	58
Tabla 19. Descripción de los factores ambientales.....	59
Tabla 20. Ponderación de factores Ambientales	60
Tabla 21. Porcentajes de Actividades del ciclo de vida	61
Tabla 22. Porcentaje de actividades de ciclo de vida.	64
Tabla 23. Identificación Clases Equivalentes - Crear Proyecto -	66

Tabla 24. Casos de Pruebas Clases válidas - Crear Proyecto -	67
Tabla 25. Casos de Pruebas Clases inválidas - Crear Proyecto -	67
Tabla 26. Matriz caso de prueba - Crear Proyecto -	68

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Diagrama de casos de uso del negocio.....	23
Ilustración 2. Diagrama de Actividades - Registro de Proyectos.	26
Ilustración 3. Diagrama de Actividades - Control de Contratos.....	27
Ilustración 4. Diagrama de Actividades - Pago a Proveedores.....	28
Ilustración 5. Diagrama de Caso de Uso del Sistema - Gestionar Usuario.....	33
Ilustración 6. Diagrama de Caso de Uso del Sistema - Gestionar Proyectos.	34
Ilustración 7. Diagrama de Casos de Uso del Sistema - Gestionar Contratos.....	34
Ilustración 8. Diagrama de Caso de Uso del Sistema - Gestionar Aprobación de Pago.	35
Ilustración 9. Arquitectura del Sistema	42
Ilustración 10. Modelo - Vista - Controlador.....	43
Ilustración 11. Diagrama de Clases de Diseño (Crear Registro de Proyectos).....	44
Ilustración 12. Diagrama de secuencia caso de uso - Crear Proyecto, referencia al CU- 3.1.....	46
Ilustración 13. Tratamiento de Errores.....	47
Ilustración 14. Modelación lógico de datos	48
Ilustración 15. Diagrama de Componentes.....	50
Ilustración 16. Diagrama de Despliegue	51
Ilustración 17. Método de Estimación.....	63
Ilustración 18. Gráfico Porcentajes de Actividades.....	64
Ilustración 19. Gráfico Comparativo de Costos y Requerimientos.....	65
Ilustración 20. Verificación del sistema.....	70

RESUMEN

El objetivo del trabajo actual es lograr agilizar el proceso administrativo del registro de solicitudes de pago en la empresa NYLIC, que en la actualidad se realiza con Excel. Con el fin de garantizar la integridad de los datos, Esto es un aspecto crucial porque se da una mayor eficiencia y optimización del tiempo.

Después de estudiar el negocio, se identifican deficiencias y fallas organizativas en los procesos actuales con lo cual se determinó realizar distintos módulos para el ingreso y generación de Solicitudes y Aprobaciones, además del módulo de administración.

Este aplicativo se desarrolló bajo ambiente de escritorio en Lenguaje Visual Basic.Net, además como almacenamiento de datos se utilizará SQL Server.

En este trabajo de titulación se implementa la metodología UML (Unified Modeling Language) que es un proceso de modelado del sistema con la finalidad de describir los componentes y funciones claves del proyecto desarrollado en este estudio.

Como resultado, se obtiene un sistema de registros que gestiona la información relativa a las solicitudes de aprobación, la supervisión de diversos proyectos, la vigilancia de contratos y los informes mensuales.

Este sistema brindará ayuda al personal de la Consultora NYLIC encargado de este proceso.

Palabras clave: Optimización, Sistema de Registros, Supervisión de Proyectos, Vigilancia de Contratos, Informes Mensuales.

ABSTRACT

The objective of the current work is to streamline the administrative process of registering payment requests in the NYLIC company, which is currently done with Excel. To guarantee the integrity of the data, this is a crucial aspect because it gives greater efficiency and optimization of time.

After studying the business, deficiencies and organizational failures in the current processes were identified, with which it was determined to carry out different modules for the entry and generation of Requests and Approvals, in addition to the administration module.

This application was developed under a desktop environment in Visual Basic.Net Language, and SQL Server will also be used as data storage.

In this degree work, the UML (Unified Modeling Language) methodology is implemented, which is a system modeling process to describe the key components and functions of the project developed in this study.

As a result, a records system is obtained that manages the information related to the approval requests, the supervision of various projects, the surveillance of contracts and the monthly reports.

This system will help the staff of the NYLIC Consultant in charge of this process.

Keywords: Optimization, Record System, Project Supervision, Contract Monitoring, Monthly Reports

INTRODUCCIÓN

Según lo que indica (Daim, Dabic, & Bayraktaroglu, 2016). La innovación juega un papel crucial en el crecimiento, evolución y sostenibilidad económica de las organizaciones. Proporciona una diferenciación competitiva al permitir el desarrollo de productos, servicios o procesos únicos que destacan frente a la competencia. Además, impulsa la mejora continua dentro de la empresa, alentando soluciones más eficientes y creativas para los desafíos. Esto conlleva a la optimización de procesos, reducción de costos y mejora de la calidad.

Otro aspecto importante es su capacidad para adaptarse al cambio en un entorno empresarial en constante evolución. La innovación facilita la agilidad y flexibilidad de la organización, lo que asegura su resiliencia. Asimismo, abre puertas a nuevas oportunidades de negocio, permitiendo a la empresa explorar mercados no atendidos, atraer clientes y expandir su alcance, generando crecimiento y mayores ingresos.

La sostenibilidad económica es un resultado directo de la innovación. Al mantenerse relevante y competitiva, la empresa asegura su futuro a largo plazo. La capacidad de evolucionar y adaptarse a los cambios garantiza su supervivencia en un mercado en constante transformación. (Daim, Dabic, & Bayraktaroglu, 2016, pág. 4556)

En los últimos años, el Ecuador ha experimentado un crecimiento significativo en cuanto a la innovación tecnológica. Se han llevado a cabo diversas iniciativas para fomentar y promover la innovación en el país.

En un artículo titulado "Cambio de la Matriz Productiva vía innovación: el caso de Ecuador", los autores señalan que la inversión en innovación tecnológica es fundamental para el crecimiento económico y la generación de empleo en el país. Además, destacan la importancia de fortalecer los vínculos entre la academia, la industria y el gobierno para impulsar la innovación (Macias, Cadena, Maldonado, & Alcivar, 2017, págs. 84-103).

Un estudio realizado por (Carvache, Gutierrez, & Frias, 2022): Analizó la incidencia de la innovación y la tecnología en el desarrollo competitivo de las pequeñas y medianas empresas (Pymes) de Ecuador. Los resultados mostraron que las Pymes

que adoptaron tecnologías innovadoras experimentaron un aumento significativo en su competitividad y rentabilidad.

La empresa Nylic, que presta servicios de consultoría y fiscalización para proyectos de ingeniería civil desde hace más de 20 años, ha visto un aumento significativo en el número de clientes y variedad de proyectos en los últimos años. Sin embargo, este crecimiento ha revelado algunas deficiencias con respecto a los procedimientos administrativos que aún no han sido actualizados. Como resultado, han surgido desafíos que amenazan con causar retrasos en los procesos de construcción de los proyectos. Por ejemplo, las solicitudes de pago pueden tardar más tiempo de lo esperado debido a la necesidad de buscar entre carpetas y registros los proyectos respectivos. Esto retarda su aprobación y, en última instancia, el desarrollo del proyecto. También ha habido problemas con la pérdida de datos y su duplicación, así como información falsa de contratos, lo que dificulta la verificación de las solicitudes de pago procesadas. En resumen, existe un control irregular de la información de los proyectos en Nylic.

El crecimiento de los proyectos de ingeniería civil ha planteado desafíos en el control administrativo manual asociado a ellos, lo que ha llevado a una serie de inconvenientes que obstaculizan el progreso de los registros y la optimización de los recursos. Estos problemas han surgido debido a los retrasos en los procesos, especialmente cuando el uso de tablas de Excel, que alguna vez fue efectivo, ya no puede manejar el aumento del volumen de solicitudes. Como resultado, Estos procesos presentan una marcada dependencia de la información y del conocimiento. (Rodríguez & Galán, 2019).

Para superar estos inconvenientes, la consultora ha optado por implementar un sistema informático de escritorio para la gestión y control de la información. Esta solución busca mejorar la eficiencia y la precisión en la gestión administrativa, así como proporcionar una base sólida para el manejo adecuado de los proyectos de ingeniería civil (Björk, 2021).

Situación Problemática

A continuación, se describen los procesos internos para gestionar solicitudes y mantener el flujo de continuidad en proyectos. Cuando se firma un nuevo proyecto, se registra en un libro Excel con detalles sobre el presupuesto de inversión y los rubros a ejecutarse. También se registran los proveedores y sus contratos, lo que proporciona información relevante como montos, plazos, garantías y anticipos.

Se utiliza un formulario de aprobación específico para cada proyecto, el cual contiene información detallada sobre el proveedor, los contratos, pagos realizados, descuentos y valores del pago actual, así como datos del ítem y el rubro asociado. Con esta información almacenada en libros de Excel, se generan informes mensuales que reflejan los gastos por rubros y controlan los contratos para evitar sobregiros.

Estos procesos son gestionados principalmente por el departamento de Fiscalización, Programación y Control, generando una amplia documentación que se actualiza hasta el cierre de cada mes, lo que implica un consumo significativo de recursos, especialmente tiempo.

La documentación generada durante todo el proceso muestra múltiples errores en los registros de solicitudes, control de contratos y secuencia de ingreso de datos, ya que cada proceso depende de otro y puede propagar errores. Al finalizar los registros, el supervisor encargado revisa y aprueba la documentación.

Esta revisión a menudo revela errores que podrían afectar el flujo del proyecto y la empresa. Por lo tanto, se recomienda que el departamento de Fiscalización, Programación y Control realice correcciones necesarias para mejorar la eficiencia y evitar futuros problemas.

Formulación del Problema

¿Cómo se puede mejorar la gestión y control de la información para garantizar la eficiencia en el flujo de proyectos y el uso efectivo de recursos en la empresa?

Objetivos.

Objetivo General.

Desarrollar un sistema informático basado en aplicaciones de escritorio utilizando Visual Basic.net y SQL Server para la Empresa Nylic, con el propósito de agilizar y optimizar la gestión de información y procesos administrativos relacionados con las solicitudes de pagos, aprobaciones y contratos en la consultora.

Objetivos específicos.

- Sustentar las metodologías y herramientas tecnológicas propuestas, mediante un análisis crítico y técnico, de acuerdo con el contexto en el que se desempeña la empresa.
- Analizar los problemas en los procesos actuales, la definición de requisitos, investigación de soluciones existentes, la selección de tecnologías.
- Diseñar el Sistema informático propuesto usando las herramientas ya sustentadas.
- Validar mediante pruebas de Caja Negra, y así, garantizar su correcto funcionamiento y cumplimiento de los objetivos establecidos.
- Implementar el software informático a partir del diseño propuesto.

Justificación del Problema

En el entorno empresarial actual, el éxito en la ejecución de proyectos es esencial para el crecimiento y la competitividad. La empresa enfrenta desafíos significativos en la gestión y control de información relacionada con los proyectos. Los procesos basados en libros Excel han demostrado ser propensos a errores y consumen recursos innecesarios.

La dependencia de múltiples procesos y la gestión manual conlleva a errores en los registros, control de contratos y secuencia de datos, afectando la integridad de la información y la correcta ejecución de los proyectos.

Además, el trabajo manual requiere un alto consumo de tiempo y esfuerzo para actualizar registros y generar informes mensuales, afectando la eficiencia general.

El desarrollo de un software de gestión y control de proyectos es fundamental para garantizar un flujo más fluido, minimizar errores y aumentar la productividad. Una solución personalizada con Visual Basic.Net y SQL Server ofrece una interfaz amigable y funcional, garantizando un acceso seguro a la información y simplificando la generación de informes y análisis.

El objetivo del proyecto es desarrollar una solución que se ajuste a las necesidades específicas de la empresa, optimizando la asignación de recursos, el seguimiento en tiempo real de avances y el control detallado de contratos y pagos a proveedores.

Con la implementación exitosa, se espera mejorar significativamente el desempeño general de la empresa. La optimización de los procesos impulsará una mayor productividad y un uso más efectivo de recursos, lo que se traducirá en proyectos exitosos y una ventaja competitiva en el mercado actual.

Viabilidad Técnica.

La viabilidad del desarrollo del software utilizando Visual Basic.net y SQL Server para optimizar la gestión de información y procesos administrativos en la consultora Nyllic es fundamentada por varias razones.

En primer lugar, Visual Basic.NET es un lenguaje de programación altamente versátil y potente que permite la creación de aplicaciones personalizadas y adaptadas a las necesidades específicas de la empresa (Utech Academy, 2023). Los productos de software se construyen con un nivel de urgencia que no se veía en años anteriores. La prioridad más alta de las compañías es reducir el tiempo de salida al mercado, que es la base del desarrollo rápido. (Maida & Pacienza, 2015, pág. 22).

Por otro lado, SQL Server es un sistema de gestión de bases de datos robusto y escalable, que permitirá almacenar y gestionar eficientemente la gran cantidad de información relacionada con las solicitudes de pagos, aprobaciones y contratos en la consultora. Además, SQL Server ofrece herramientas avanzadas para la manipulación

de datos y la generación de informes, lo que facilitará el análisis y seguimiento de los procesos administrativos (Larson, 2016).

Por último, la elección de Visual Basic.net y SQL Server también asegura un bajo costo de desarrollo y mantenimiento, ya que ambos son tecnologías ampliamente disponibles y cuentan con una amplia comunidad de desarrolladores que brindan soporte y recursos de aprendizaje (Voon Kiong, 2017).

En conclusión, el desarrollo del software utilizando Visual Basic.net y SQL Server es una opción viable y adecuada para mejorar la gestión de información y procesos administrativos en la consultora Nylis, proporcionando una solución eficiente y de bajo costo para agilizar los flujos de trabajo y garantizar un uso efectivo de los recursos en la empresa.

Hipótesis

La implementación del software de gestión y control de proyectos basado en Visual Basic.Net y SQL Server mejorará la eficiencia en la ejecución de proyectos, minimizando errores y optimizando la asignación de recursos, lo que conducirá a un mayor desempeño general y una ventaja competitiva en el mercado empresarial.

Estructura del documento.

Este desarrollo inicia con la problemática de la situación a mejorar. Esta parte detalla antecedentes de la investigación, Posterior a esto se definen los Objetivos, los mismos que permiten delimitar y orientar el trabajo dándole un camino para direccionar el trabajo.

El Capítulo I, se detalla el marco teórico, que es fundamental para el soporte del trabajo con temas relacionados al tema principal del trabajo. Se muestra el proceso actual de los procedimientos, se establecen los términos y conceptos fundamentales relacionados con la investigación. Además, se examinarán los principales sistemas vinculados al ámbito de acción y se describirán las tecnologías, herramientas y metodologías que se utilizarán para desarrollar un sistema de entorno de escritorio.

El capítulo II, Se expone las normas de negocio, y se describirán los requerimientos funcionales y no funcionales que se tomarán en cuenta para abordar el problema. Esta evaluación permitirá evaluar la viabilidad de una propuesta.

En el capítulo III, se lleva a cabo la implementación del sistema propuesto, enfocándose en la planificación de la realización del software, tales como son los diagramas de clases, diseño físico, lógico y visualización de la base de datos.

En el capítulo IV, se establece la validación del software, aplicando pruebas de caja negra, técnicas, resultados de las pruebas ya aceptación de los usuarios, finalizando con el análisis de factibilidad técnica.

CAPITULO I

1. FUNDAMENTACIÓN TEÓRICA

Este capítulo brindará una visión comprehensiva de los procesos actuales de NYLIC, contextualizará la investigación en relación con sistemas análogos y proporcionará una justificación respaldada para la metodología y las tecnologías adoptadas, estableciendo así las bases sólidas que sustentan la evolución y mejora de los procesos empresariales en la consultora.

1.1. Conceptos asociados a la investigación

- **Registro de Solicitudes de Pago a Proveedores:** Este concepto abarca la acción de recibir y documentar de manera precisa las facturas y comprobantes de gastos presentados por los proveedores con los que la empresa interactúa.
- **Control de Contratos:** Este término engloba el seguimiento y administración integral de los contratos y acuerdos establecidos con los proveedores. Incluye la gestión de fechas, términos, condiciones, asegurando la conformidad y cumplimiento de ambas partes involucradas en la relación comercial.
- **Control de Presupuesto:** Este concepto refiere a la asignación y supervisión de recursos financieros en función de los objetivos y necesidades de la empresa. Implica el establecimiento de un presupuesto planificado y su monitoreo continuo para garantizar un uso eficiente y efectivo de los fondos disponibles.
- **Informes Mensuales:** Este término hace referencia a la generación de informes detallados que resumen y analizan el desempeño y avance del proyecto en un periodo de tiempo específico, generalmente mensual.

Estos conceptos son esenciales para comprender y abordar los procesos que serán optimizados a través del desarrollo del software propuesto. Su comprensión y análisis detenido permiten delinear las funcionalidades necesarias para garantizar una solución integral y efectiva en la gestión y control de la información en la consultora.

1.2. Objetivos estratégicos de la organización objeto de la investigación

1.2.1. Visión

Seremos una firma líder en el ámbito de la consultoría de fiscalización en proyectos de ingeniería civil, destacándonos por nuestra excelencia en la prestación de servicios de máxima calidad y por nuestro arraigado compromiso con la plena satisfacción de nuestros clientes. Buscamos ser reconocidos como un referente en el mercado, impulsando la innovación y la eficiencia en cada proyecto que abordemos. (Nylic, 2023)

1.2.2. Misión:

En NYLIC, nuestra misión es proporcionar servicios de consultoría de fiscalización en ingeniería civil de excelencia, cumpliendo rigurosamente con las exigencias técnicas y legales de nuestros clientes. Trabajamos incansablemente para contribuir al éxito de sus proyectos, ofreciendo soluciones adaptadas a sus necesidades específicas. Mediante enfoques personalizados y orientados a la optimización de recursos, garantizamos la eficiencia en la gestión y el cumplimiento puntual de plazos y presupuestos. A través de nuestra dedicación al cliente, establecemos relaciones de confianza y duraderas, colaborando en su crecimiento a largo plazo y siendo parte fundamental de su éxito (Nylic, 2023).

1.3. Objeto de Estudio.

El objeto de estudio abarca los sistemas informáticos diseñados para optimizar y agilizar los procesos clave de la gestión empresarial, enfocados en las Solicitudes de Pagos a Proveedores, el Control de Contratos y la Administración de Presupuestos en proyectos de ingeniería civil. Estos sistemas se conciben como herramientas tecnológicas que buscan mejorar la eficiencia, la precisión y la integridad de los datos en cada etapa de los proyectos, desde la solicitud de pagos hasta la presentación de informes mensuales.

1.3.1. Campo de Acción:

El ámbito de acción se centra en los procesos vitales que prevalecen el flujo ejecutivo de los proyectos de ingeniería civil en las firmas consultoras. Estas operaciones

engloban la evaluación y autorización de pagos a proveedores según el estado de los proyectos, la supervisión de los acuerdos y contratos pactados con los proveedores, y la asignación estratégica y seguimiento de los presupuestos destinados a cada proyecto específico.

1.3.2. Flujo actual del Proceso.

1.3.2.1. Análisis de la ejecución de los procesos.

La Consultora sigue una secuencia bien definida en la ejecución de sus operaciones, con pasos que abarcan el registro de solicitudes de pago a proveedores, la tramitación de aprobaciones, el control de contratos y la creación de informes. No obstante, se ha evidenciado un ineficiente manejo de la información, lo que ha desencadenado una serie de inconvenientes. Específicamente, el uso extensivo de hojas de cálculo Excel ha resultado en la proliferación de archivos y la generación de dificultades. Este tratamiento inadecuado de la información ha llevado a jornadas laborales prolongadas en el departamento correspondiente y a la imposibilidad de generar aprobaciones de manera ágil, lo cual ha derivado en considerables retrasos en la gestión de pagos y en el descontento de los involucrados.

Un ejemplo es el proceso de generación de aprobaciones, el cual produce una gran cantidad de datos que no son almacenados adecuadamente. Esto provoca un retraso significativo en los procedimientos de aprobación y suscita descontento por parte de los proveedores.

La gestión manual de la información en la inscripción de contratos también ha demostrado ser ineficiente. La revisión de los procesos anteriores ha revelado que el almacenamiento de los datos generados no es apropiado, lo que genera demoras significativas para la consultora al momento de cerrar los contratos.

Los desafíos mencionados tienen un impacto significativo en la eficiencia operativa y la calidad de la gestión en el departamento correspondiente. La asignación ineficiente de recursos y el derroche de tiempo repercuten en la capacidad de la organización para lograr una utilización óptima de sus activos y talentos. Por otro lado, la generación de información poco fiable debido a errores introduce incertidumbre en el

proceso de aprobación, lo que a su vez origina retrasos y problemas en la ejecución de proyectos y en las relaciones con los proveedores (Brandon-Jones & Slack, 2018)

1.3.2.2. Procesos Objeto de automatización.

La automatización implica el desarrollo e implementación de un sistema informático que permita registrar y gestionar las solicitudes de pago de manera más eficiente, automatizar los flujos de aprobación, almacenar de manera adecuada la información de los contratos y generar informes de manera automatizada y precisa. Al emplear tecnologías como Visual Basic.Net y SQL Server, se podría crear un software personalizado que automatice estos procesos, reduciendo la carga de trabajo manual, mejorando la precisión de los datos y agilizando la toma de decisiones.

La automatización puede ofrecer una mayor trazabilidad y visibilidad de los procesos, facilitando la toma de decisiones basada en datos y mejorando la colaboración entre departamentos y partes interesadas (Davenport, 2018).

1.4. Sistemas automatizados existentes relacionados con el campo de acción.

Mejorar y agilizar el registro de solicitudes generaría más tiempo para realizar otras actividades, el tiempo de espera para el cobro del valor a pagar se reduciría, el control de contrato se realizaría de forma organizada. Todo esto permitirá agilizar los procesos en la Consultora.

En el transcurso del desarrollo de este trabajo de titulación, se identificó la carencia de un software completamente adaptado para la gestión administrativa de proyectos de ingeniería civil en la Consultora. Aunque existen aplicaciones que abordan el control administrativo, algunas se centran en el seguimiento de avances y ejecuciones de tareas, mientras que otras se enfocan exclusivamente en aspectos contables (Sutherland & Beedle, 2020). Este vacío en el mercado resalta la necesidad de desarrollar una solución específica que integre las diversas facetas de gestión que requiere la Consultora para sus proyectos de ingeniería civil.

1.4.1. Herramientas para la Gestión de Proyectos de Consultoría.

La gestión efectiva de proyectos de consultoría se beneficia de herramientas especializadas que facilitan la planificación, ejecución y control. En este sentido,

soluciones como Alfresco, Jira y Airtable ofrecen capacidades esenciales, pero no las necesarias para optimizar la gestión de proyectos.

1.4.1.1. Alfresco.

Alfresco es una plataforma de gestión de contenido empresarial que permite el almacenamiento, organización y colaboración en documentos relacionados con proyectos. Proporciona flujos de trabajo personalizables para la revisión y aprobación de documentos, lo que mejora la eficiencia en la gestión de información crítica (Alfresco Software, Inc., 2023).

1.4.1.2. Jira.

Jira, por otro lado, es una herramienta de seguimiento de problemas y gestión de proyectos ampliamente utilizada en entornos de desarrollo ágil. Ofrece tableros visuales, asignación de tareas, seguimiento de avances y permite una colaboración efectiva entre los miembros del equipo. Jira se adapta especialmente bien a equipos que siguen metodologías ágiles como Scrum o Kanban (Atlassian, 2023).

1.4.1.3. Airtable.

Airtable es una plataforma de gestión de bases de datos en línea que combina la flexibilidad de las hojas de cálculo con las capacidades de una base de datos. Puede utilizarse para la planificación de proyectos, seguimiento de tareas, gestión de recursos y generación de informes personalizados. Airtable es altamente personalizable y se adapta a diferentes flujos de trabajo (Airtable Inc, 2023).

1.4.2. Comparación con algunos Sistemas Informáticos existentes.

A continuación, se presentan los indicadores en la Tabla 1, que son los requisitos necesarios para la Consultora.

Tabla 1. Comparación de Sistemas.

No.	Características.	Alfresco		Jira		Airtable	
		Cumple	No Cumple	Cumple	No Cumple	Cumple	No Cumple
1	Registro de Proyectos y sus Presupuestos	x		x		x	
2	Registro de Proveedores	x			x	x	
3	Registro de Contratos	x			x		x
4	Ingreso de Solicitudes		x		x		x
5	Generar Aprobación		x		x		x
6	Generar Informe		x		x		x
7	Control de Contratos	x			x	x	
TOTAL		4	3	1	6	3	4

Elaborado: Jimmy Jesús Salazar Gómez.

En la Tabla 2, se detallan los costos por adquisición de estos productos.

Tabla 2. Detalle de Costo

No.	Programas	Costo de la Licencia Por Usuario / año	Entorno
1	Alfresco	5.400,00	Web
2	Jira Software	1.525,00	Web
3	Airtable	720,00	Web

Elaborado: Jimmy Jesús Salazar Gómez.

Después del análisis llevado a cabo, se puede afirmar que el costo de los diferentes sistemas es excesivo. Por consiguiente, se llega a la conclusión de que resulta imprescindible implementar un software personalizado, teniendo en cuenta que no se realizan todos los procesos que se desean implementar, y se considera un costo que se ajuste a la realidad económica del país. Además, los sistemas mencionados en la tabla

1 no satisfacen los requisitos necesarios para garantizar el correcto funcionamiento de las actividades de la Consultora.

1.5. Tendencias y Tecnologías.

La elección de la metodología adecuada desempeña un papel crucial en el éxito de cualquier proyecto. Proporciona un enfoque estructurado para la planificación, ejecución y seguimiento, lo que resulta fundamental para optimizar los recursos y lograr los objetivos establecidos (Kerzner, 2017).

Tabla 3. Comparación de Metodologías.

Característica	Metodologías ágiles	Metodologías tradicionales
Enfoque	Iterativo e incremental	Secuencial
Planificación	Flexible y adaptable a cambios	Rigurosa y detallada
Entregables	Funcionalidades operativas	Solución completa y validada
Comunicación	Comunicación frecuente y directa con el cliente	Comunicación formal y limitada con el cliente
Documentación	Prioridad en el software y poca documentación formal	Prioridad en la documentación formal y exhaustiva
Control de calidad	Pruebas continuas y frecuentes	Pruebas finales y exhaustivas
Equipo	Autogestionado y multifuncional	Roles y jerarquías bien definidos

Fuente: Agile vs Waterfall (Gil López, 2018).

Elaborado: Jimmy Jesús Salazar Gómez.

1.6. Fundamentación de la Metodología Utilizada.

1.6.1. Metodología RUP.

La elección de una metodología adecuada es un paso crucial en la planificación de proyectos, ya que influye en la forma en que se abordan los procesos y se logran los objetivos establecidos. Tras una evaluación exhaustiva, se ha decidido adoptar la metodología Rational Unified Process (RUP) para el proyecto en cuestión. RUP es una

metodología tradicional que ofrece un enfoque estructurado para el desarrollo de software, centrándose en procesos, productos y la definición clara de actividades y responsabilidades.

Los conceptos esenciales de RUP son fundamentales para comprender su aplicación efectiva en el proceso de desarrollo:

- a) **Ciclos de Vida en Fases:** RUP se organiza en fases de ciclo de vida, incluyendo Inicio, Elaboración, Construcción y Transición. Cada fase tiene objetivos específicos y produce entregables que guían el progreso del proyecto (Ambler, 2010).
- b) **Enfoque Iterativo e Incremental:** RUP promueve el desarrollo iterativo e incremental, en el cual el proyecto se divide en iteraciones cortas y enfocadas en objetivos específicos. Cada iteración produce un incremento funcional del producto y permite ajustar los requisitos y mejoras a lo largo del tiempo (Ambler, 2010).
- c) **Roles y Responsabilidades Definidos:** RUP asigna roles claros y definidos a los miembros del equipo, como arquitectos, analistas y desarrolladores. Cada rol tiene responsabilidades específicas para garantizar una colaboración efectiva y una distribución adecuada del trabajo (Ambler, 2010)
- d) **Artefactos y Documentación:** RUP enfatiza la creación de artefactos, como modelos, diagramas y documentación, para capturar requisitos, especificaciones y decisiones de diseño. Estos artefactos fomentan la comunicación y el entendimiento entre el equipo y las partes interesadas (Ambler, 2010)

La metodología RUP se adapta particularmente a proyectos que requieren una planificación rigurosa y una estructura clara para garantizar la calidad y el control a lo largo del proceso de desarrollo. Su enfoque en ciclos de vida en fases, iteraciones y roles definidos proporciona una base sólida para la gestión exitosa de proyectos de software.

1.6.1.1. Metodología de trabajo basada en RUP.

A continuación, se presenta la Metodología de Trabajo basada en Rational Unified Process (RUP). Esta metodología proporciona un marco sólido para el desarrollo de

proyectos, asegurando la planificación efectiva, la colaboración y la entrega de productos de alta calidad.

Tabla 4. Metodología de Trabajo con RUP.

Proceso	Fases							
	Inicio		Elaboración		Construcción		Transición	
	Actividades	Artefacto	Actividades	Artefacto	Actividades	Artefacto	Actividades	Artefacto
Modelado del Negocio	Elaborar casos de uso del negocio	Casos de uso del negocio			Emplear casos de uso del Negocio	Casos de uso del negocio		
	Realizar diagrama de actividades							
Requerimientos	Establecer requerimientos funcionales		Desarrollar casos de uso del sistema		Emplear casos de uso del sistema	Casos de uso del sistema		
	Establecer requerimientos no funcionales							
Análisis y Diseño			Elaborar la arquitectura del sistema	Diagrama MVC				
			Realizar diagrama de Clases	Diagrama de Clases				
Implementación			Crear diagrama de secuencias		Emplear diagrama de secuencias	Diagrama de secuencias		
			Crear diagrama de componentes		Emplear Diagrama de componentes	Diagrama de componentes		
Pruebas					Preparar pruebas y análisis de la factibilidad		Mostrar formularios de prueba	Casos de Prueba

Elaborado: Jimmy Jesús Salazar Gómez.

1.6.2. Lenguaje Unificado de Modelado UML.

El Lenguaje Unificado de Modelado (UML) es una herramienta gráfica fundamental para la visualización, especificación, construcción y documentación de sistemas de software (Pressman & Maxim, 2015). UML está compuesto por una serie de diagramas que representan diversos aspectos del sistema, incluyendo diagramas de casos de uso, clases, objetos, secuencias, comunicaciones, actividades, componentes y despliegue. Cada uno de estos diagramas ofrece una perspectiva única del sistema, permitiendo a los desarrolladores y diseñadores comprender mejor la interrelación entre las partes del sistema. Además, UML proporciona una notación que permite la representación de restricciones, comentarios y otros aspectos del diseño del sistema.

La estandarización y el reconocimiento internacional hacen que UML sea ampliamente utilizado por empresas de todo el mundo para el diseño y construcción de sistemas de software complejos. En palabras de (Pressman & Maxim, 2015), "UML es un lenguaje gráfico que permite a los desarrolladores de software representar visualmente los componentes y procesos de un sistema, desde la perspectiva del diseño hasta la implementación."

1.7. Herramientas y Tecnologías.

1.7.1. Visual Paradigm.

Visual Paradigm es una herramienta de modelado y diseño ampliamente utilizada en la industria del desarrollo de software. Proporciona una plataforma integral para crear diversos tipos de diagramas y modelos que ayudan en la visualización, especificación y documentación de sistemas de software y procesos empresariales.

Ofrece una amplia gama de características, incluyendo la capacidad de crear diagramas UML (Lenguaje Unificado de Modelado), diagramas de flujo, diagramas de casos de uso, diagramas de clases, diagramas de secuencia, entre otros. Además, la herramienta permite la colaboración en equipo, la generación automática de código a partir de los modelos y la integración con herramientas de desarrollo y gestión de proyectos.

Esta herramienta se ha convertido en una elección popular entre los profesionales del desarrollo de software debido a su versatilidad, facilidad de uso y capacidad para adaptarse a diversas metodologías de desarrollo. Visual Paradigm permite a los equipos de desarrollo crear modelos visuales que ayudan a comunicar de manera efectiva las ideas y el diseño del software, lo que facilita la comprensión y colaboración entre los miembros del equipo (Visual Paradigm, 2023).

1.7.2. Lenguaje de Programación Visual Basic.Net.

El desarrollo de software utilizando Visual Basic.NET se presenta como una opción altamente viable y recomendable para la empresa, especialmente considerando que ya posee una licencia de esta plataforma debido a la implementación de su sistema contable bajo la misma tecnología. Esta elección se basa en una serie de ventajas que hacen que aprovechar la licencia existente sea una estrategia inteligente y económicamente eficiente.

- a) **Aprovechamiento de recursos:** Al tener ya una licencia de Visual Basic.NET debido al sistema contable, la empresa puede aprovechar esta inversión y utilizarla para desarrollar nuevas soluciones. Esto permite maximizar el valor de los recursos ya adquiridos y reducir los costos asociados con la adopción de una nueva tecnología (Larson, 2016, págs. 67-69).
- b) **Coherencia y compatibilidad:** Desarrollar el nuevo software utilizando la misma plataforma en la que se basa el sistema contable garantiza la coherencia y compatibilidad entre ambas soluciones. Esto facilita la integración de los sistemas y la gestión de datos, optimizando la eficiencia operativa y minimizando la complejidad (Larson, 2016, págs. 67-69).
- c) **Aprovechamiento del conocimiento existente:** Si el equipo ya está familiarizado con Visual Basic.NET debido a la implementación del sistema contable, esta experiencia previa puede ser transferida al desarrollo del nuevo software. Esto acelera el proceso de desarrollo y reduce la curva de aprendizaje, permitiendo una implementación más rápida y efectiva (Voon Kiong, 2017).

- d) **Reducción de costos:** Al no necesitar adquirir una nueva licencia ni invertir en una plataforma diferente, la empresa ahorra costos directos e indirectos asociados con la adopción de una nueva tecnología. Esto contribuye a mantener el presupuesto bajo control y a maximizar el retorno de inversión (Voon Kiong, 2017).
- e) **Soporte y comunidad:** Dado que ya trabajan con Visual Basic.NET, la empresa tiene acceso a la comunidad de desarrolladores y al soporte proporcionado por Microsoft. Esto asegura que puedan resolver desafíos y obtener ayuda de manera efectiva durante el proceso de desarrollo y más allá (Voon Kiong, 2017).

La elección de Visual Basic.NET no solo se basa en la licencia existente, sino también en la robustez de la plataforma en sí. Visual Basic.NET es un lenguaje de programación poderoso y versátil que puede adaptarse a diversos tipos de proyectos, incluyendo aplicaciones de escritorio, web y móviles (Microsoft, 2023)

1.7.3. Sistema de Base de Datos Relacional SQL Server.

SQL Server es un sistema de gestión de bases de datos relacional desarrollado por Microsoft que proporciona una plataforma de software diseñada para almacenar, administrar y recuperar datos de forma segura y eficiente. Utiliza el lenguaje de consulta estructurado (SQL) para interactuar con los datos almacenados en la base de datos y es compatible con diversas aplicaciones de software, incluyendo aplicaciones web, empresariales y de análisis de datos. Ofrece una variedad de herramientas de gestión y monitoreo, como herramientas de respaldo y recuperación, supervisión de rendimiento, seguridad y administración de usuarios y permisos, para garantizar la disponibilidad, el rendimiento y la seguridad de la base de datos. (SQL Server - Base de datos relacional de Microsoft, 2023)

En opinión de (Petkovic, 2019), "SQL Server es un producto maduro y bien establecido que se ha convertido en una opción popular para la gestión de bases de datos en la empresa".

En resumen, SQL Server es un sistema de gestión de bases de datos relacional ampliamente utilizado en la industria que ofrece funcionalidades de alta disponibilidad, escalabilidad, seguridad y una amplia variedad de herramientas de administración.

1.8. Conclusión del Capítulo.

Este capítulo presenta los conceptos básicos necesarios para entender el problema a resolver y las soluciones propuestas. Se describen las características del proceso actual de la Consultora, el cual no se ajusta a las necesidades actuales. También se describe un nuevo sistema que automatiza los procesos básicos y reduce los problemas de manipulación y almacenamiento de datos. En cuanto a la gestión de documentos ya generados, se realiza un estudio comparativo de sistemas y se establece la necesidad de desarrollar un nuevo sistema que permita una mejor todo este proceso. Se analizan las herramientas disponibles para el desarrollo del nuevo sistema y se eligen las más adecuadas. Con todo ello, se pretende solucionar la problemática actual y mejorar la eficiencia de la Consultora.

CAPITULO II

2. MODELO DEL NEGOCIO

2.1. Breve descripción del Capítulo.

En el presente capítulo, se expone una descripción del entorno que está siendo considerado para la automatización. Mediante diagramas y diversas técnicas de modelado, se logrará una representación detallada del entorno. Estos recursos gráficos y modelos tienen la capacidad de esclarecer en detalle la dinámica empresarial y los procesos destinados a la automatización, así como destacar los requisitos funcionales esenciales que necesitan ser abordados. Este enfoque estratégico en la visualización y el análisis tiene como propósito primordial brindar una comprensión completa de la estructura y las conexiones al sistema, sentando una sólida base para la planificación y el desarrollo óptimos del nuevo sistema automatizado.

2.2. Descripción del entorno.

En este entorno operativo, el ciclo se inicia cuando el Departamento de Fiscalización recibe los archivos correspondientes al proyecto en cuestión. El proceso comprende la apertura de un archivo Excel nuevo destinado a la captura de datos iniciales esenciales, tales como el nombre del fideicomiso, el proyecto en sí, los presupuestos con sus respectivos rubros y los costos asociados.

A medida que el proyecto avanza, el departamento de fiscalización se convierte en el receptor de las Solicitudes de Pagos a Proveedores. Este sector es responsable de registrar las solicitudes y generar las correspondientes aprobaciones, ajustadas según las particularidades de cada proyecto o proveedor involucrado. Este proceso de registro empieza con la generación de un número de aprobación único, seguido por la incorporación de detalles cruciales como el RUC del proveedor, nombre, contrato y descripción del pago. Además, se efectúa un análisis de los pagos previos realizados por el proveedor, así como las garantías establecidas en el contrato. Los datos recolectados se asientan en una tabla Excel para un control eficiente y, tras las revisiones pertinentes, se procede a la creación de la aprobación final, lista para su firma y envío.

Cada solicitud y proyecto registrados en el departamento de fiscalización dan lugar a la generación de documentos relevantes, lo que facilita un seguimiento riguroso de los procedimientos de fiscalización. Esta información adquiere importancia adicional al ser utilizada en la creación de informes mensuales y en la consolidación de registros respaldatorios para los gastos efectuados.

2.2.1. Modelo de casos de uso del negocio.

Los Casos de Uso de Negocio constituyen una herramienta esencial en la ingeniería de requisitos, proporcionando una descripción detallada de las interacciones y eventos que ocurren en un sistema para cumplir los objetivos y necesidades de un negocio. Estos casos de uso se centran en los aspectos más amplios y estratégicos de la organización, permitiendo una comprensión clara de cómo el sistema interactúa con los usuarios y los procesos empresariales.

Scott W. Ambler (2010) enfatiza que:

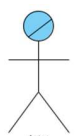




Los casos de uso de negocio son vitales para capturar los requisitos esenciales y definir los flujos de trabajo cruciales. Proporcionan un enfoque claro y centrado en el usuario, promoviendo la alineación entre los objetivos del negocio y la funcionalidad del sistema.

Según (Ambler, 2010).

Los casos de uso de negocio también son valiosos para la identificación temprana de oportunidades de mejora y la resolución proactiva de desafíos operativos. Al involucrar a los usuarios y las partes interesadas en su desarrollo, se logra un entendimiento profundo de los procesos y se establecen bases sólidas para el éxito del proyecto.

La tabla que presenta a continuación detalla los estereotipos que se emplean en el proceso de modelación del negocio.

Tabla 5. Componentes de caso de uso.

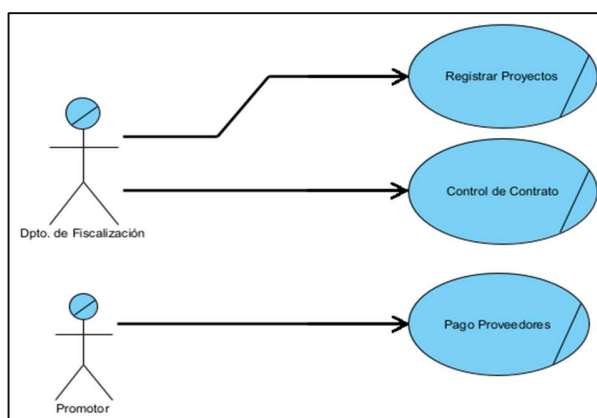
Componentes de Casos de uso.		
Figura	Nombre	Definición.
	Autor	El actor es un rol que representa a un usuario o un conjunto de usuarios que interactúan con el sistema. El actor es cualquier persona, organización o sistema externo que interactúa con el sistema en cuestión, para lograr un objetivo específico.
	Caso de Uso	Dentro del óvalo, se escribe el nombre del caso de uso. Además, es común agregar una descripción corta del caso de uso debajo del nombre. La descripción puede incluir información adicional sobre el propósito y las características del caso de uso.
	Conector	Estos conectores son líneas que se utilizan para indicar las relaciones entre los casos de uso y los actores, así como entre los casos de uso.
	Extensión	El estereotipo "extend" se utiliza en UML para indicar que un caso de uso opcional (el caso de uso "extensor") puede ser ejecutado en un momento determinado del caso de uso base (el caso de uso "extendido") para agregarle más funcionalidades.
	Inclusión	El estereotipo "include" en UML representa una relación obligatoria entre dos casos de uso en la que uno es parte del otro y se integra en su flujo de eventos, permitiendo la reutilización y evitando la duplicación de código.

Elaborado: Jimmy Jesús Salazar Gómez.

2.2.2. Diagrama de Casos de uso del negocio.

A Continuación, se presentan los diagramas de casos de uso del negocio.

Ilustración 1. Diagrama de casos de uso del negocio.



Elaborado: Jimmy Jesús Salazar Gómez.

2.2.3. Actores y Trabajadores del Negocio.

2.2.3.1. Actores del negocio.

Según (Montes de Oca, 2015, pág. 78). “Los actores del negocio se refieren a todas las entidades externas con las que la organización interactúa, ya sean individuos, grupos, entidades, organizaciones, máquinas o sistemas de información”. Esta definición abarca la diversidad de interacciones que pueden surgir en un entorno empresarial y destaca la importancia de comprender y considerar todos los roles involucrados en el sistema.

Tabla 6. Descripción de los actores del negocio

Descripción de los actores del Negocio	
Actores del Negocio	Función
Dpto. de Fiscalización	Departamento encargado dentro de la Empresa Consultora
Promotor	Representante legal Encargada de Proyectos

Elaborado: Jimmy Jesús Salazar Gómez.

2.2.3.2. Trabajadores del Negocio.

La definición de los actores implica la especificación de comportamientos y responsabilidades (roles) que recaen en individuos, grupos, sistemas automatizados o máquinas que colaboran como un equipo. Estos actores realizan una serie de actividades y asumen la propiedad de elementos particulares dentro del contexto del sistema (Montes de Oca, 2015)

Tabla 7. Descripción de los Trabajadores del Negocio.

Descripción de los actores del Negocio	
Actores del Negocio	Función
Jefe Departamento	Encargado de gestionar la información relacionada con los Registros de Proyectos, Control de Contratos, e informes
Asistente	Encargado de gestionar la información relacionada con los Registros de Proyectos, Generar las Aprobaciones de Pagos a Proveedores, e informes

Elaborado: Jimmy Jesús Salazar Gómez.

2.2.4. Descripción de los Casos de usos del negocio.

El proceso de pago a proveedores comienza cuando el departamento de fiscalización y control recibe la solicitud generada por la entidad contratante. El asistente encargado de la verificación y análisis de la solicitud se asegura de que se cumplan los requisitos necesarios para su aprobación. Para ello, se verifica la información de la solicitud, incluyendo el código, datos del proveedor, si existe contrato con el proveedor, descripción de la solicitud, rubro al que se carga el pago, posibles descuentos y saldos de contratos y rubros de presupuesto.

Una vez que se ha analizado y verificado la información, se procede a registrar la nueva solicitud en un archivo de Excel que está dividido en varias hojas, cada una de las cuales contiene información sobre los contratos, proveedores, rubros del presupuesto y un historial de las solicitudes anteriores. Si el proveedor o el contrato no se encuentran registrados, se procede a su registro, así mismo como el nombre de proyecto y su respectivo presupuesto.

A continuación, el asistente busca el archivo de Excel que contiene la información necesaria para la aprobación y modifica los datos con la nueva solicitud. Una vez que se ha completado este proceso, la aprobación se registra en las hojas de Excel correspondientes y se genera un archivo en formato PDF para su firma y envío posterior.

El proceso de registro de Proyecto con su presupuesto empieza recibiendo la información, y jefe del departamento, luego de verificar la información, procede a crear el archivo de Excel para ser usado en la generación de Aprobaciones de pago.

2.2.5. Diagrama de Actividades.

El Diagrama de Actividades en UML (Lenguaje Unificado de Modelado) es una herramienta fundamental para representar la secuencia y el flujo de actividades dentro de un proceso o sistema. Este diagrama ofrece una visión visual que abarca desde el punto de inicio hasta el punto final, permitiendo identificar las decisiones y bifurcaciones que pueden ocurrir en el desarrollo de las actividades (Fowler, 2019, pág. 45)

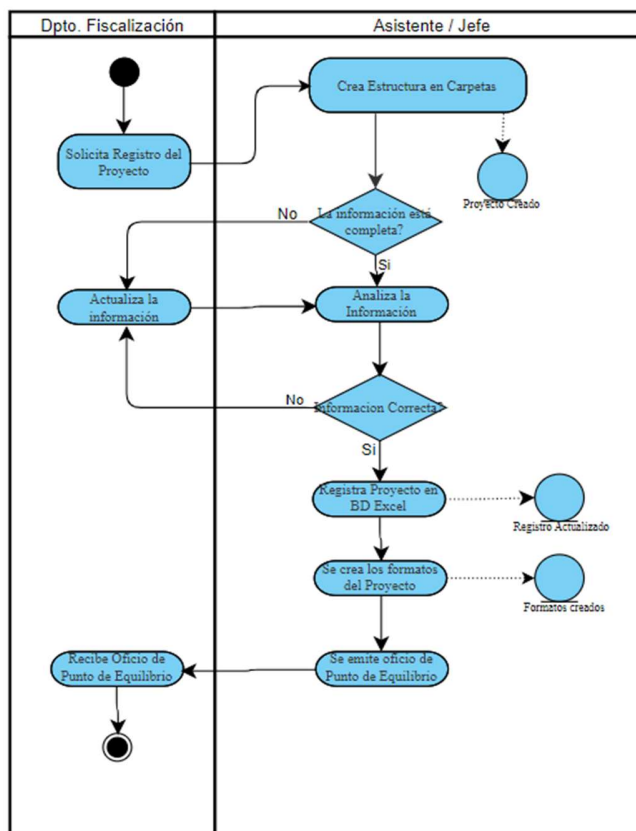
Una de las principales ventajas del Diagrama de Actividades radica en su capacidad para mejorar la visualización y comprensión de la secuencia de acciones, lo

que ayuda a los involucrados a tener una visión clara del proceso. Además, este tipo de diagrama es especialmente útil para identificar puntos críticos y cuellos de botella en el flujo de trabajo, lo que facilita la identificación de áreas que requieren optimización (Fowler, 2019, pág. 45).

La comunicación dentro del equipo y con otros interesados se ve favorecida gracias a la representación gráfica y concisa que ofrece el Diagrama de Actividades. Por su valor comunicativo, estos diagramas permiten documentar el proceso y rastrear su evolución, lo que resulta crucial para mantener un registro claro de las acciones realizadas (Fowler, 2019, pág. 45)

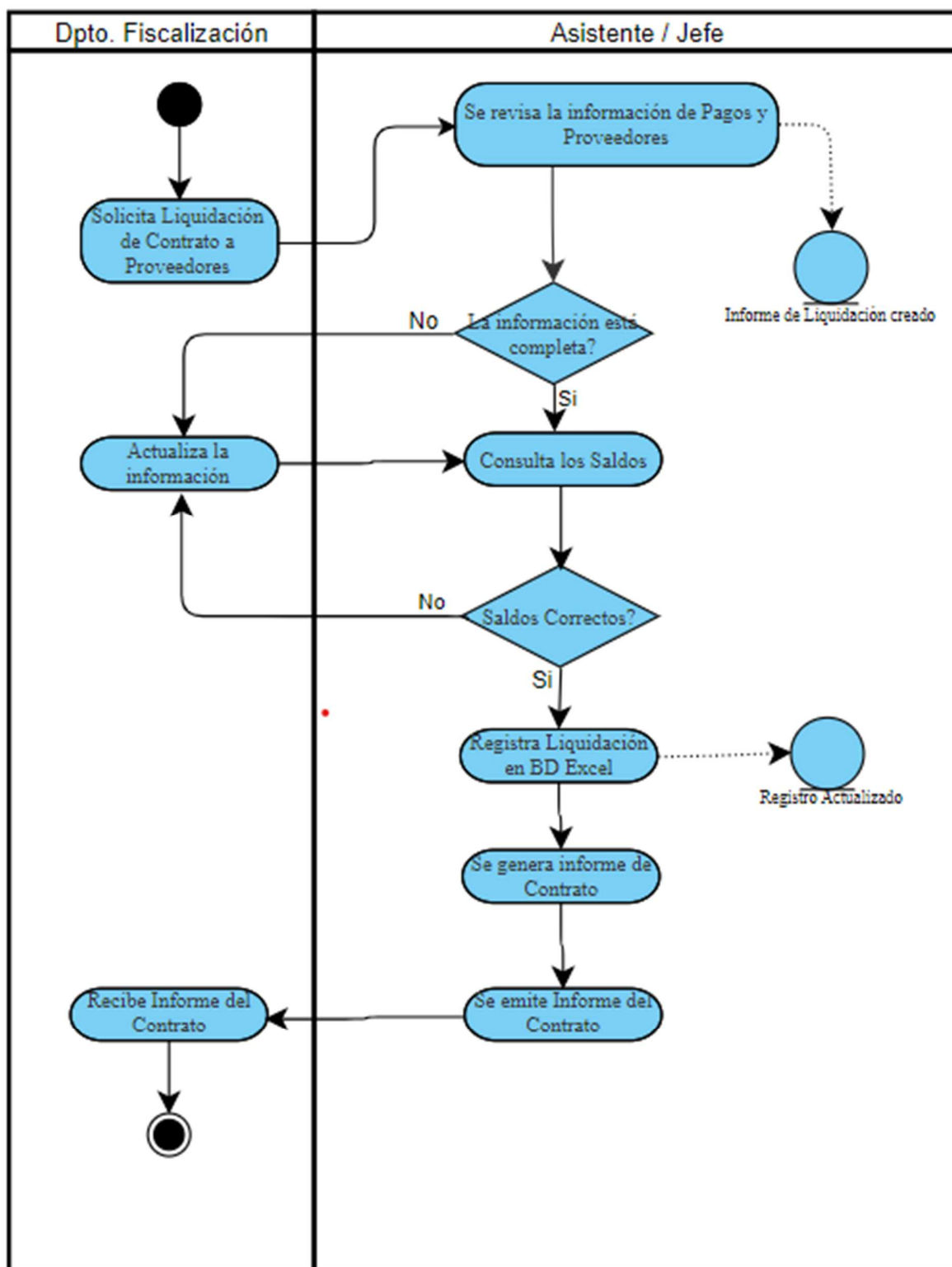
El análisis y mejora de procesos también se benefician de los Diagramas de Actividades. Al analizar el flujo de trabajo y las decisiones tomadas, se pueden identificar áreas propicias para la automatización, lo que contribuye a aumentar la eficiencia y reducir el tiempo necesario para completar el proceso (Fowler, 2019, pág. 45).

Ilustración 2. Diagrama de Actividades - Registro de Proyectos.



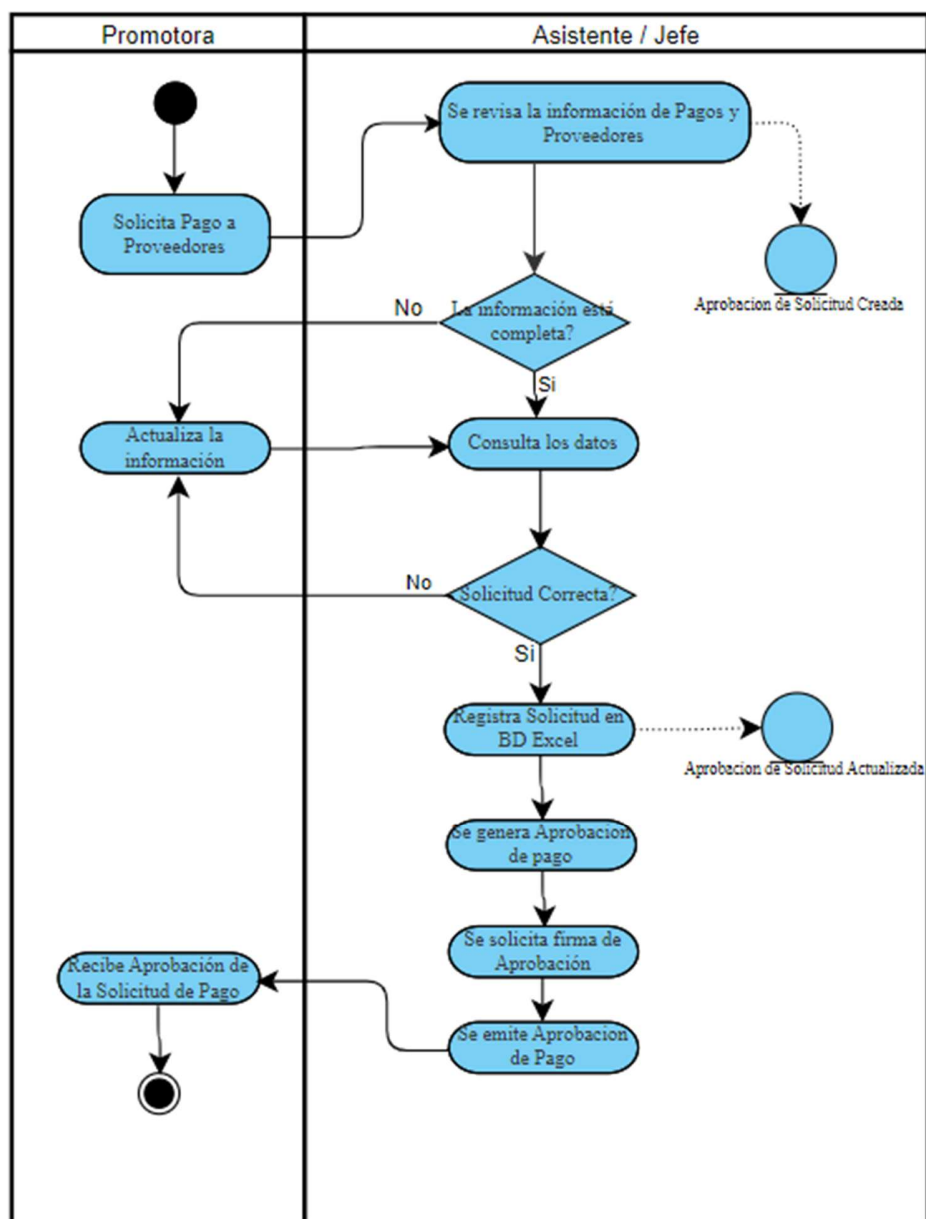
Elaborado: Jimmy Jesús Salazar Gómez.

Ilustración 3. Diagrama de Actividades - Control de Contratos.



Elaborado: Jimmy Jesús Salazar Gómez.

Ilustración 4. Diagrama de Actividades - Pago a Proveedores



Elaborado por: Jimmy Jesús Salazar Gómez.

2.2.6. Reglas del Negocio para considerar.

Una regla de negocio es una declaración que describe una política, restricción o procedimiento empresarial que debe seguirse en una organización para lograr un objetivo específico. Estas reglas se utilizan para establecer normas y directrices que gobiernan el comportamiento de la organización y pueden abarcar varios aspectos del

negocio, como la gestión de inventario, el servicio al cliente, la facturación, la contabilidad, la seguridad de la información y la gestión de proyectos. (Ozkan, 2021, pág. 10).

Para el desarrollo actual del sistema se consideran las siguientes normas empresariales relevantes:

- Toda información respecto a los contratos con los Fideicomisos es confidencial.
- Cada Fideicomiso debe tener su Contrato con la Consultora.
- El jefe de departamento debe emitir una aprobación de Punto de Equilibrio del proyecto
- Cada Proyecto debe tener su Fideicomiso.
- El Proyecto debe tener su Presupuesto Aprobado por la consultora
- El Proveedor debe tener RUC
- El proveedor puede o no tener un contrato.
- El contrato debe ser revisado por el jefe del Departamento.
- Una solicitud es aprobada y enviada siempre que esté revisada y firmada por el jefe del departamento.
- Una aprobación firmada y emitida, no puede ser borrada ni alterada.
- Solo los usuarios con credenciales de Administrador deben tener todas las opciones de Ediciones dentro del sistema.
- Los contratos de los proveedores cambian del estado "En Curso" a "Liquidado", siempre y cuando todos los valores estén pagados.

2.3. Especificación de los requisitos de Software.

Los requerimientos funcionales de un sistema se refieren a los servicios o funciones específicas que el sistema debe proporcionar para satisfacer las necesidades de los usuarios y cumplir con los objetivos del negocio. Estos requerimientos se centran

en lo que el sistema debe hacer y no en cómo debe hacerlo (Pressman & Maxim, 2015, pág. 78).

Según (Pfleeger & Atlee, 2009, pág. 42), "Los requerimientos funcionales describen las funciones específicas del sistema, cómo se espera que se comporten en respuesta a ciertas entradas y cómo se espera que el sistema interactúe con los usuarios y otros sistemas."

2.3.1. Requisitos funcionales (RF).

1. RF1: Autenticar Usuario: Esta función se encarga de verificar la autenticidad de las credenciales proporcionadas por el usuario que intenta acceder al sistema.
2. RF2: Gestión de Fideicomiso: Esta función se encarga de gestionar la información correspondiente a los datos solicitados por el sistema al usuario.
 - RF 2.1 Crear Fideicomiso.
 - RF 2.2 Eliminar Fideicomiso.
 - RF 2.3 Modificar Fideicomiso.
 - RF 2.4 Buscar Fideicomiso
3. RF3: Gestionar registro de Proyectos: Se ocupará de introducir y procesar los datos según los requisitos o las exigencias que tenga.
 - RF 3.1 Crear Registro de Proyectos.
 - RF 3.2 Eliminar Registro de Proyectos.
 - RF 3.3 Modificar Registro de Proyectos.
 - RF 3.4 Buscar Registro de Proyectos.
4. RF4: Gestionar Registro de Presupuesto: Se ocupará de introducir y procesar los datos según los requisitos o las exigencias que tenga.
 - RF 4.1 Crear Registro de Presupuesto.
 - RF 4.2 Eliminar Registro de Presupuesto.
 - RF 4.3 Modificar Registro de Presupuesto.

- RF 4.4 Buscar Registro de Presupuesto.
5. RF5: Gestionar Registro de Proveedores: Se ocupará de introducir y procesar los datos según los requisitos o las exigencias que tenga.
 - RF 5.1 Crear Registro de Proveedores.
 - RF 5.2 Eliminar Registro de Proveedores.
 - RF 5.3 Modificar Registro de Proveedores.
 - RF 5.4 Buscar Registro de Proveedores.
 6. RF6: Gestionar Registro de Contratos: Se ocupará de introducir y procesar los datos según los requisitos o las exigencias que tenga.
 - RF 6.1 Crear Registro de contratos.
 - RF 6.2 Eliminar Registro de contratos.
 - RF 6.3 Modificar Registro de contratos.
 - RF 6.4 Buscar Registro de contratos.
 7. RF7: Entrega de Pago a Proveedores: Su responsabilidad es producir las autorizaciones utilizando la información requerida en el formato preestablecido por la empresa. Posteriormente, esta información se exporta en formato PDF.
 - RF 7.1 Crear Pago de Proveedores.
 - RF 7.2 Eliminar Pago de Proveedores.
 - RF 7.3 Modificar Pago de Proveedores.
 - RF 7.4 Buscar Pago de Proveedores.
 8. RF8: Reporte informe mensual: Se encarga de generar un informe con todos los datos requeridos en ese momento.

2.3.2. Requisitos No Funcionales (RNF).

“Los requisitos no funcionales se refieren a características del sistema que no están directamente relacionadas con su funcionalidad, pero que son importantes para el éxito del sistema” (Sommerville, 2011).

Tabla 8. Requerimientos No Funcionales

Id	Requisito no Funcionales (RNF)	Prioridad del requerimiento
Interfaz de Usuario (1)		
RNF 1.1	La aplicación debe ser fácil de usar y tener una interfaz de usuario intuitiva.	Alta
Usabilidad (2)		
RNF 2.1	Las funciones del Sistema no deben estar a más de 4 clics de la pantalla inicial o selector de módulos.	Alta
RNF 2.2	No más de 5 segundos debe demorar la respuesta del sistema	Alta
Seguridad (3)		
RNF 3.1	La aplicación debe estar protegida contra amenazas externas y evitar cualquier tipo de vulnerabilidad.	Alta
RNF 3.2	Cada Usuario debe tener su login de ingreso al sistema	Alta
Fiabilidad (4)		
RNF 4.1	Los errores deben ser manejados de manera adecuada y los usuarios deben ser informados de manera clara sobre las posibles fallas	Alta
Eficiencia (5)		
RNF 5.1	La aplicación debe ser capaz de manejar un aumento en el número de usuarios y el volumen de datos sin disminuir el rendimiento.	Alta
Arquitectura (6)		
RNF 6.1	Compatible con Windows 7 en adelante.	Alta
RNF 6.2	Memoria Ram mínima de 4Gb en adelante.	Alta
RNF 6.3	Procesador mínimo Intel I3 en adelante.	Alta

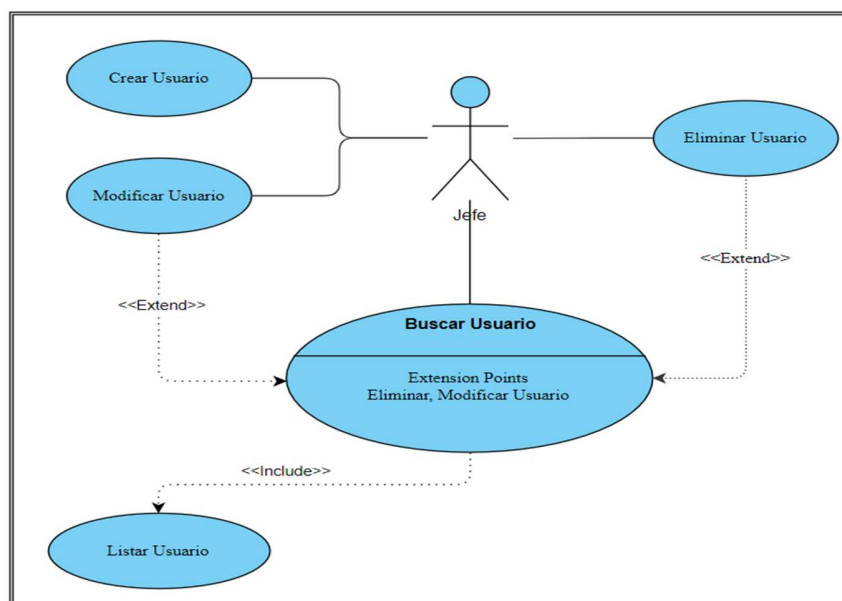
Elaborado: Jimmy Jesús Salazar Gómez.

2.4. Diagrama de casos de uso del sistema.

El Diagrama de Casos de Uso es una herramienta de modelado de requisitos de software que representa la interacción entre los actores (usuarios, sistemas externos, etc.) y el sistema en cuestión, mediante la identificación de los diferentes casos de uso y sus relaciones. Según (Sommerville, 2011, pág. 143), un caso de uso es "un conjunto de actividades que el sistema realiza en colaboración con uno o más usuarios externos para producir un resultado observable".

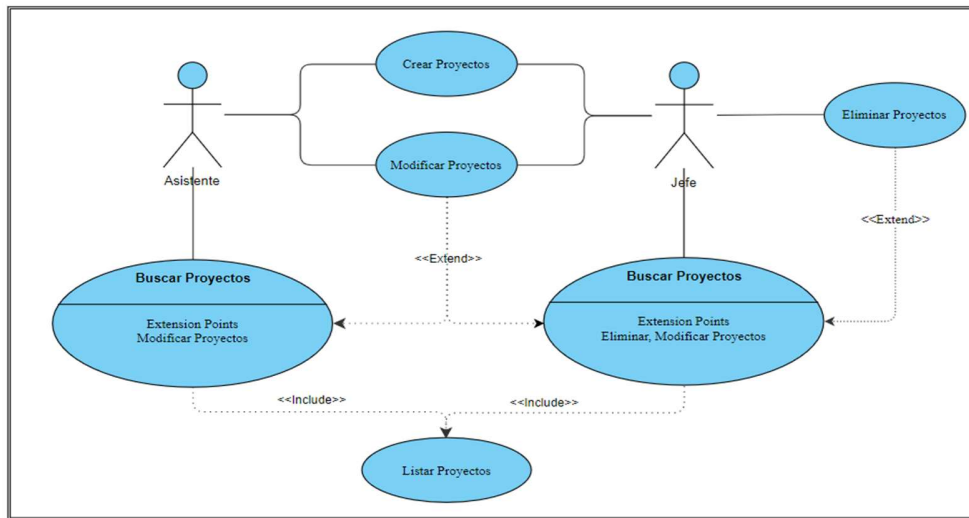
El Diagrama de Casos de Uso se utiliza para definir los requisitos funcionales del sistema, es decir, las funcionalidades y comportamientos que el sistema debe ser capaz de realizar para satisfacer las necesidades de los usuarios y alcanzar los objetivos del negocio. Según (Larman, 2004, pág. 93), el Diagrama de Casos de Uso "es un modelo de comportamiento que describe la funcionalidad propuesta del sistema desde la perspectiva del usuario final o de un actor"

Ilustración 5. Diagrama de Caso de Uso del Sistema - Gestionar Usuario



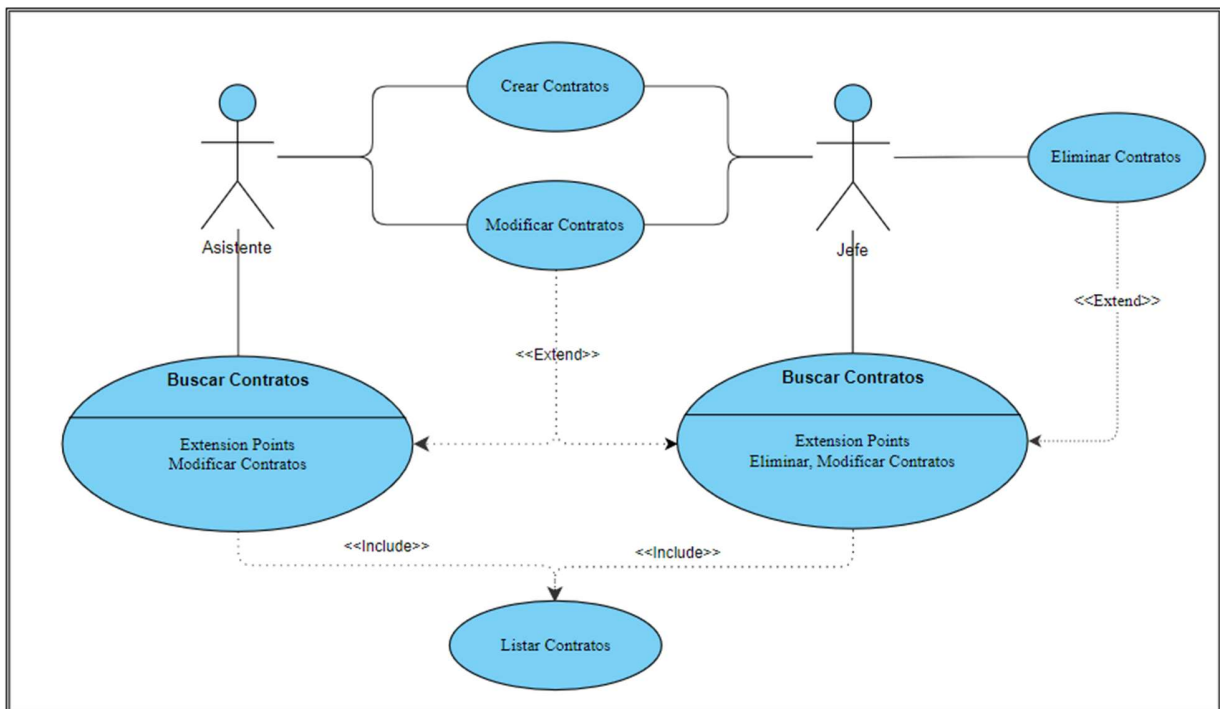
Elaborado: Jimmy Jesús Salazar Gómez.

Ilustración 6. Diagrama de Caso de Uso del Sistema - Gestionar Proyectos.



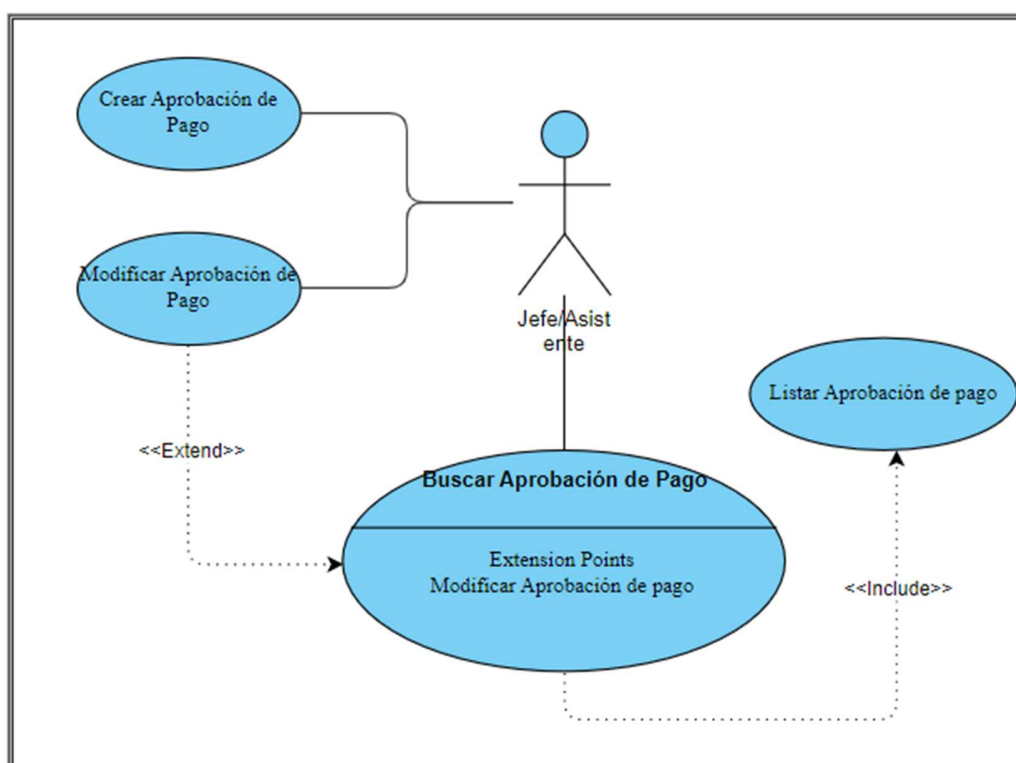
Elaborado: Jimmy Jesús Salazar Gómez.

Ilustración 7. Diagrama de Casos de Uso del Sistema - Gestionar Contratos.



Elaborado: Jimmy Jesús Salazar Gómez.

Ilustración 8. Diagrama de Caso de Uso del Sistema - Gestionar Aprobación de Pago.



Elaborado: Jimmy Jesús Salazar Gómez.

2.4.1. Listado de Casos de uso.

Tabla 9. Priorización de los Casos de Uso.

No.	Nombre del Caso de Uso
1	Autenticar Usuario
2	Gestionar Fideicomiso
	2.1. Crear Fideicomiso
	2.2. Eliminar Fideicomiso
	2.3. Modificar Fideicomiso
	2.4. Buscar Fideicomiso.
3	Gestionar Proyecto
	3.1. Crear Proyecto
	3.2. Eliminar Proyecto
	3.3. Modificar Proyecto
	3.4. Buscar Proyecto.

- 4 Gestionar Presupuesto
 - 4.1. Crear Presupuesto
 - 4.2. Eliminar Presupuesto
 - 4.3. Modificar Presupuesto
 - 4.4. Buscar Presupuesto.
- 5 Gestionar Proveedor
 - 5.1. Crear Proveedor
 - 5.2. Eliminar Proveedor
 - 5.3. Modificar Proveedor
 - 5.4. Buscar Proveedor.
- 6 Gestionar Contrato
 - 6.1. Crear Contrato
 - 6.2. Eliminar Contrato
 - 6.3. Modificar Contrato
 - 6.4. Buscar Contrato.
- 7 Gestionar Pago de Proveedores
 - 7.1. Crear Pago de Proveedores
 - 7.2. Eliminar Pago de Proveedores
 - 7.3. Modificar Pago de Proveedores
 - 7.4. Buscar Pago de Proveedores.
- 8 Crear Informes

Elaborado: Jimmy Jesús Salazar Gómez.

2.5. Descripción de los casos de usos principales del sistema.

En el informe de investigación, se mencionarán elementos claves los casos de uso del sistema 3, 6 y 7, debido a que son los que describen las actividades relevantes del negocio. Solo se incluirá en este trabajo el escenario "crear" de cada uno de estos casos de uso. Para ello, según lo plantea (Pressman & Maxim, 2015), se tendrán en cuenta lo siguiente:

- **Nombre del caso de uso:** El nombre del caso de uso debe ser claro y descriptivo, capturando el objetivo central del caso de uso.

- **Propósito:** Este componente describe el objetivo principal que se persigue con el caso de uso. Es crucial que el propósito esté alineado con los objetivos generales del sistema o software en desarrollo.
- **Actor:** En esta sección se especifican los usuarios o sistemas externos que interactúan con el sistema en cuestión. Los actores son esenciales para comprender quiénes están involucrados en la ejecución del caso de uso y cómo interactúan con el sistema.
- **Precondición:** Esta condición describe el estado en el que debe encontrarse el sistema antes de que el caso de uso pueda ejecutarse. Establece los requisitos previos necesarios para garantizar una ejecución adecuada del caso de uso.

2.5.1. Descripción del Caso de uso “Crear Proyecto”.

Tabla 10. Descripción de Caso de Uso - Crear Proyecto.

CU-3	Gestionar Registro de Proyecto
CU-3.1	Crear Registro de Proyecto
RF: 3.1	
Actores	Asistente, Supervisor
Propósito	Crear un nuevo Proyecto
Resumen	El caso de uso inicia cuando el usuario selecciona la opción de "Registros" y luego selecciona la opción de "Proyectos", se despliega la interfaz pertinente para gestionar la información, la cual se genera la creación de un nuevo registro.
Precondición	El usuario esta autenticado en el sistema
Flujo normal del evento	
Acción del Actor	Respuesta de Sistema
1. Selecciona la opción de "Registro"	2. Muestra las opciones (Fideicomiso, Proyectos, Presupuestos Proveedores, Contratos)

3. Selección de opción deseada	4. Se despliega formulario de registro
5. Digita la información requerida	
6. Selecciona la opción "Guardar"	7. Se realiza la validación de la información ingresada.
	8. Almacena los datos en la base de datos
	9. Muestra mensaje de confirmación de la información guardada.
Flujos alternos	
7.1 En caso de información errónea o campos vacíos, el nuevo registro no se guarda hasta que el actor digite la información faltante/correcta.	

Elaborado: Jimmy Jesús Salazar Gómez.

2.5.2. Descripción del Caso de Uso "Crear Contratos".

Tabla 11. Descripción de Caso de Uso - Crear Contrato.

CU-6	Gestionar Registro de Contrato
CU-6.1	Crear Registro de Contrato
RF: 6.1	
Actores	Asistente, Supervisor
Propósito	Crear un nuevo Contrato
Resumen	El caso de uso inicia cuando el usuario selecciona la opción de "Registros" y luego selecciona la opción de "Contratos", se despliega la interfaz pertinente para gestionar la información, la cual se genera la creación de un nuevo registro.
Precondición	El usuario esta autenticado en el sistema
Flujo normal del evento	
Acción del Actor	Respuesta de Sistema

1. Selecciona la opción de "Registros"	2. Muestra las opciones (Fideicomiso, Proyectos, Presupuestos Proveedores, Contratos)
3. Selección de opción deseada	4. Se despliega formulario de registro
5. Digita la información requerida	
6. Selecciona la opción "Guardar"	7. Se realiza la validación de la información ingresada.
	8. Almacena los datos en la base de datos
	9. Muestra mensaje de confirmación de la información guardada.
Flujos alternos	
7.1 En caso de información errónea o campos vacíos, el nuevo registro no se guarda hasta que el actor digite la información faltante/correcta.	

Elaborado: Jimmy Jesús Salazar Gómez.

2.5.3. Descripción del Caso de Uso "Crear Pagos a Proveedores".

Tabla 12. Descripción de Caso de Uso - Crear Pago a Proveedores.

CU-7	Gestionar Registro de Pago a Proveedores
CU-7.1	Crear Registro de Pago a Proveedores
RF: 7.1	
Actores	Asistente, Supervisor
Propósito	Crear un nuevo Pago a Proveedores
Resumen	El caso de uso inicia cuando el usuario selecciona la opción de "Pagos", se despliega la interfaz pertinente para gestionar la información, la cual se genera la creación de un nuevo registro.
Precondición	El usuario esta autenticado en el sistema.
Flujo normal del evento	
Acción del Actor	Respuesta de Sistema

1. Selecciona la opción de "Pagos"	2. Muestra la ventana de registro
	3. Se despliega formulario de registro
4. Digita la información requerida	
5. Selecciona la opción "Guardar"	6. Se realiza la validación de la información ingresada.
	7. Almacena los datos en la base de datos
	8. Muestra mensaje de confirmación de la información guardada.
Flujos alternos	
<p>4.1 En caso de que no existiera información en la base de datos, se procede a agregar la información faltante yendo a los formularios correspondientes.</p> <p>6.1 En caso de información errónea o campos vacíos, el nuevo registro no se guarda hasta que el actor digite la información faltante/correcta.</p>	

Elaborado: Jimmy Jesús Salazar Gómez.

2.6. Conclusión del Capítulo.

En este capítulo se ha realizado una descripción de los elementos esenciales para el diseño del sistema. Se han presentado los diagramas de casos de uso del negocio, el diagrama de actividades del negocio y los diagramas de casos de uso del sistema, lo que permite una visión detallada y completa de las funcionalidades y procesos que intervienen en el proyecto. Además, se ha proporcionado una lista de los requisitos funcionales, destacando aquellos que resultan críticos para la funcionalidad del sistema, y se han descrito los requisitos no funcionales que se deben cumplir para garantizar la calidad del sistema.

En conjunto, todo lo presentado en este capítulo sienta las bases para un diseño eficiente del sistema.

CAPITULO III

3. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

3.1. Breve descripción del Capítulo.

En este capítulo se ha llevado a cabo una descripción detallada del análisis y diseño del sistema propuesto, destacando la arquitectura utilizada como eje central. Se han utilizado tecnologías, lenguaje y metodología descritos en los diagramas del diseño la arquitectura elegida para el proyecto.

Se ha admitido la implementación del sistema de acuerdo con la arquitectura propuesta, lo que ha incluido la representación de la base de datos a través de un modelo de datos relacional, así como la modelación de los diagramas de componentes. Todo ello ha contribuido a conformar un modelo de implementación sólido y eficiente.

Asimismo, se explica de manera detallada cómo se lleva a cabo el tratamiento de errores, la seguridad, los estándares y estilos utilizados, aspectos fundamentales para garantizar el correcto funcionamiento y la calidad del sistema.

En conjunto, este capítulo ha sido clave en el desarrollo del proyecto, proporcionando una visión clara y precisa de la arquitectura utilizada, la base de datos y los diferentes diagramas de diseño. Todo ello sentará las bases para la implementación del sistema de manera rigurosa y eficiente, garantizando la satisfacción del cliente y usuarios finales.

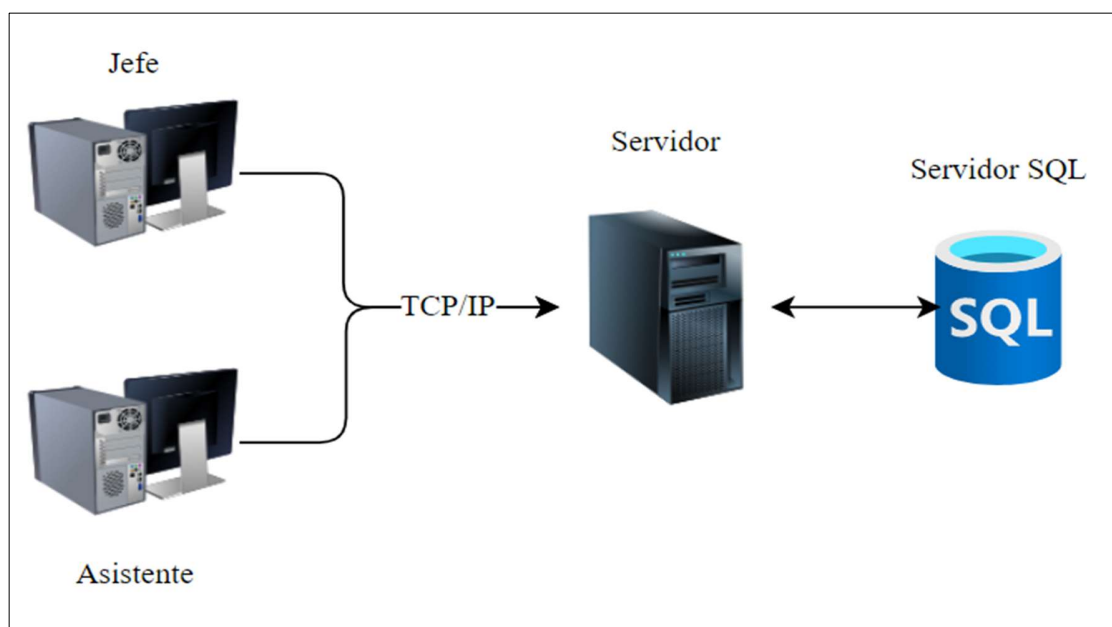
3.2. Arquitectura Cliente – Servidor.

El modelo de arquitectura Cliente-Servidor consiste en un enfoque de diseño para sistemas informáticos, donde el software de aplicación se separa en dos componentes centrales: el cliente y el servidor. El cliente es la parte de la aplicación que interactúa con el usuario y el servidor es la parte que proporciona los servicios que solicita el cliente. Según (Tanenbaum & Van Steen, 2007, pág. 319), "en una arquitectura cliente-servidor, el cliente realiza solicitudes al servidor y el servidor proporciona los servicios necesarios para satisfacer esas solicitudes".

La funcionalidad de este sistema radica en su capacidad para permitir el acceso desde cualquier dispositivo dentro de la red, siempre y cuando se haya instalado la aplicación correspondiente.

En la Ilustración 9, se puede observar cómo los dispositivos se conectan a la red interna de la empresa, la cual es controlada por un servidor de Directorio Activo (AD) que permite la autenticación tanto a nivel de Windows como a nivel de software. Una vez que se han completado las correspondientes autenticaciones, se obtiene acceso a la base de datos.

Ilustración 9. Arquitectura del Sistema



Elaborado: Jimmy Jesús Salazar Gómez.

3.3. Arquitectura para Diseño del Sistema (Modelo – Vista – Controlador).

La arquitectura Modelo-Vista-Controlador (MVC) es un enfoque de diseño de software que segmenta la lógica del negocio (modelo), la interfaz de usuario (vista) y la lógica de control (controlador) en componentes distintos. Ampliamente empleado en aplicaciones web y de escritorio, este enfoque fomenta una mayor modularidad, mantenibilidad y escalabilidad (Pavón, 2019, págs. 2-3).

En este caso, el Modelo está constituido por la capa de acceso a datos, donde se almacenarán los datos del programa y se realizarán las operaciones relacionadas con

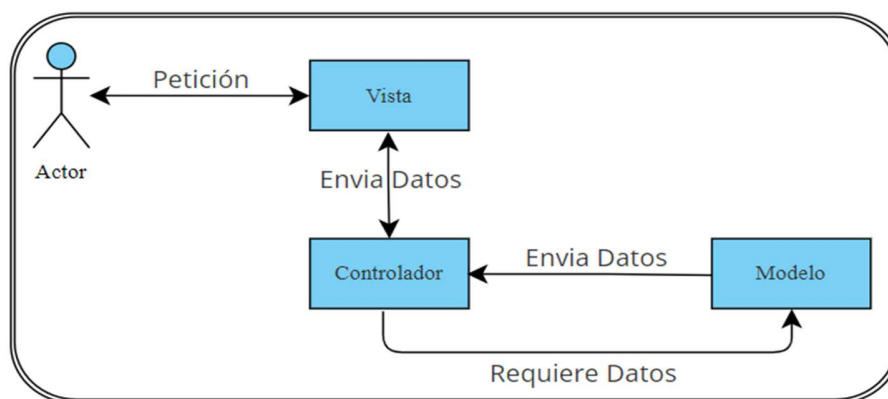
ellos, como leer y escribir en una base de datos. En Visual Basic.Net, esto puede lograrse a través del uso de tecnologías como ADO.NET o LINQ to SQL.

La Vista sería la interfaz gráfica de usuario (GUI) del programa, que se encargará de mostrar los datos al usuario y de capturar las entradas del usuario. En Visual Basic.Net, esto puede lograrse utilizando las herramientas de diseño de formularios integradas en el entorno de desarrollo.

El Controlador sería el código que coordina la interacción entre la Vista y el Modelo. En Visual Basic.Net, esto puede lograrse mediante el uso de eventos y delegados, donde el código en el Controlador responde a los eventos generados por la Vista y actualiza el Modelo en consecuencia.

Según (Aguilar, 2019), el uso del patrón MVC y sus variantes está claro que ha triunfado en todo tipo de desarrollos Web, móvil, de escritorio, y en todo tipo de plataformas, rara es la plataforma actual que no lo implementa para uno o varios tipos de desarrollos.

Ilustración 10. Modelo - Vista - Controlador



Elaborado: Jimmy Jesús Salazar Gómez.

3.4. Diagrama de Clases de Diseño.

Según Booch, el diagrama de clases de diseño es un "diagrama que muestra las clases y sus relaciones, así como sus atributos y operaciones. En un diagrama de clases de diseño, se especifica el nivel de visibilidad de los miembros de la clase, las multiplicidades y los tipos de datos de los atributos, y los parámetros y tipos de datos de las operaciones". (Booch, Maksimchuk, Young, & Engel, 2007, pág. 189).

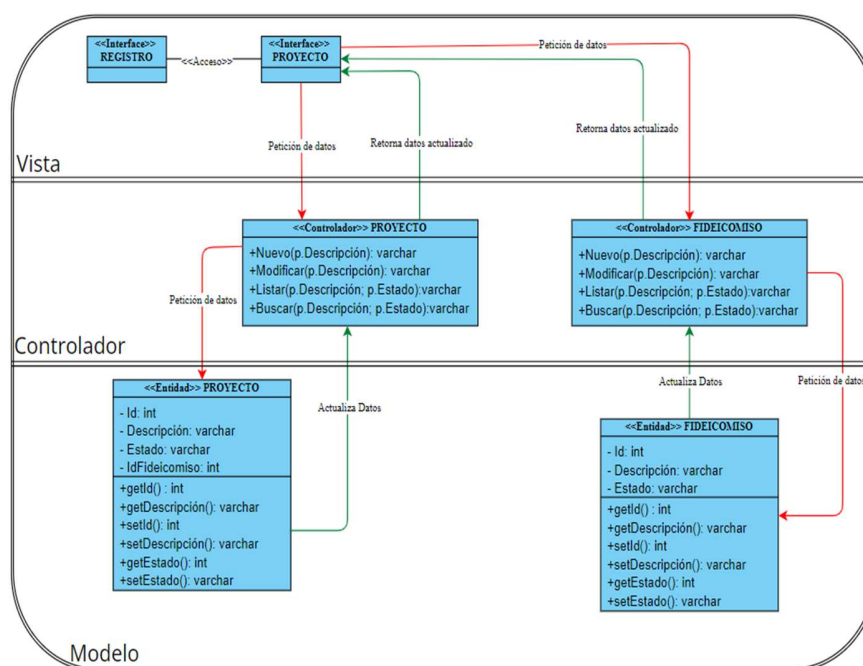
Al describir la lógica operativa del software, es crucial considerar las credenciales del usuario y su relación con las funcionalidades. En la Ilustración 11, se muestra la interfaz principal del programa, donde el usuario puede ejecutar diversas acciones. Por ejemplo, solicitar la creación de un Proyecto y asociarlo a un Fideicomiso. La capa de presentación capta las entradas del usuario y las remite a la siguiente capa.

La capa de control administra la interacción entre la parte visual y la base de datos, resolviendo las peticiones. Esta capa envía solicitudes a la base de datos (Modificar, Crear, Guardar, Eliminar, Buscar) a través de la capa de modelo.

La capa de modelo refleja la base de datos, interactuando para recuperar y almacenar información de manera efectiva.

Este ciclo de interacción entre capas se repite con cada petición desde la interfaz, lo que asegura una interacción fluida. El controlador gestiona múltiples modelos, aumentando la eficiencia y facilitando la gestión de datos.

Ilustración 11. Diagrama de Clases de Diseño (Crear Registro de Proyectos).



Elaborado: Jimmy Jesús Salazar Gómez.

3.5. Diagrama de Secuencia.

Los diagramas de secuencia desempeñan un papel fundamental en la elaboración de sistemas de software. Según (Pons, Giandini, & Perez, 2010, págs. 227-228), los diagramas de secuencia especifican las acciones y evaluaciones necesarias para chequear el objetivo de la prueba, el cual describe lo que se debe probar. El uso de diagramas de secuencia puede ayudar a los desarrolladores a identificar problemas de diseño y a optimizar el rendimiento del sistema.

Además, como menciona (Roques, 2006) en su libro "Modeling Software Systems Using UML2, WILEY-ISTE":

Los diagramas de secuencia son particularmente útiles para representar situaciones en las que la secuencia de eventos es importante, como en la interacción entre un usuario y un sistema. Estos diagramas también pueden utilizarse para describir situaciones en las que hay múltiples objetos que interactúan entre sí, o cuando la secuencia de eventos es crítica para el funcionamiento del sistema.

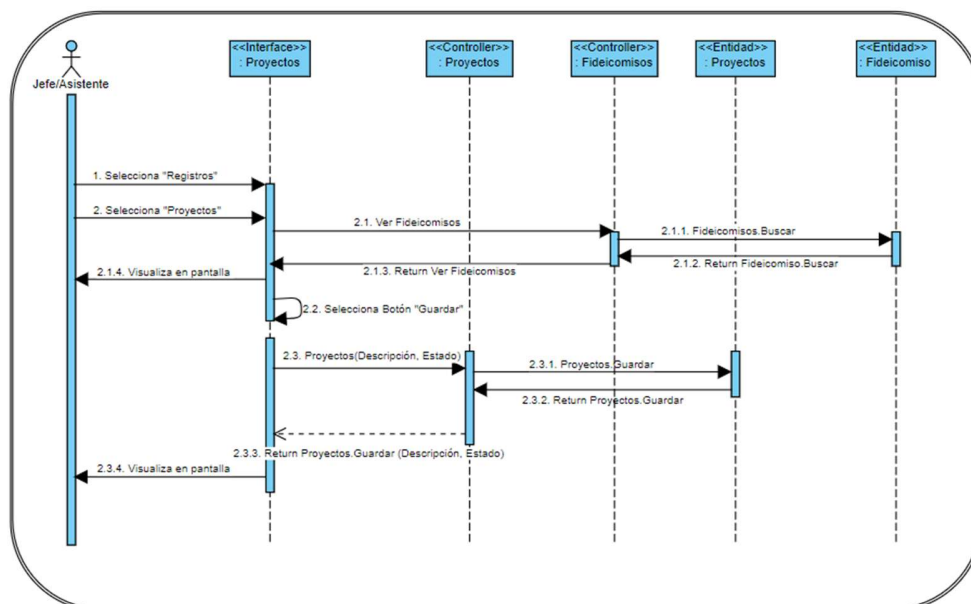
El diagrama de secuencia muestra la interacción cíclica entre las capas de presentación, control y modelo del software. El usuario interactúa con la capa de presentación, generando una acción como la creación de un presupuesto vinculado a un proyecto. Esta información se transmite a la capa de control.

La capa de control, gestionando la relación visual-base de datos, recibe y valida la petición del usuario antes de transmitirla al modelo. La capa de modelo interactúa con la base de datos, ejecutando las operaciones solicitadas por el controlador.

Tras la acción en la capa de modelo, los resultados vuelven a la capa de control y se envían a la de presentación para que el usuario los vea. El ciclo se repite, permitiendo identificar problemas y diseñar soluciones, mejorando la eficiencia. El diagrama también clarifica la interacción entre las capas, esencial para el correcto funcionamiento del software.

A continuación, se muestra el Caso de Uso que hace referencia al Evento de Crear Proyectos, CU-3.1. donde se puede observar la secuencia necesaria para registrar el proceso.

Ilustración 12. Diagrama de secuencia caso de uso - Crear Proyecto, referencia al CU-3.1



Elaborado: Jimmy Jesús Salazar Gómez.

3.6. Tratamiento de Errores.

La gestión adecuada de errores y excepciones es esencial para el correcto funcionamiento del nodo de integración. Según el autor "El manejo correcto de errores y excepciones es importante para que el nodo de integración funcione correctamente. Debe considerar cómo y cuándo la extensión definida por el usuario debe manejar errores y excepciones" (International Business Manager (IBM), 2023).

En el entorno de Visual Studio .NET, las excepciones son situaciones de error que surgen durante la ejecución de una aplicación, interrumpiendo su flujo normal. Pueden ser causadas por diversos motivos, como errores de programación, problemas de hardware o interacciones con el usuario.

Para gestionar eficazmente las excepciones en Visual Studio .NET, se emplean bloques de código "try-catch". Dentro de un bloque "try", se encapsula el código propenso a generar una excepción, mientras que en un bloque "catch", se coloca el código que se ejecuta cuando se produce una excepción.

Además de los bloques "try-catch", Visual Studio .NET ofrece otros mecanismos para el manejo de excepciones, como la instrucción "throw", que posibilita la generación

manual de una excepción, y la instrucción "finally", que garantiza la ejecución de un bloque de código sin importar si ocurre o no una excepción. (Nagel, Evjen, & Glynn, 2008).

A continuación, se presenta un ejemplo del código que ilustra estos conceptos:

Ilustración 13. Tratamiento de Errores.

```
Public Sub EliminarProyecto(ByVal DtgVProyectos As DataGridView, ByVal TxtBuscarProyectos As TextBox)
    Try
        ' Verificar si se ha seleccionado una fila en el DataGridView
        If DtgVProyectos.SelectedRows.Count > 0 Then
            ' Obtener el ID del proyecto seleccionado
            Dim id As Integer = CInt(DtgVProyectos.SelectedRows(0).Cells(0).Value)
            Dim cls As New TesisClassDataContext

            ' Obtener el proyecto a eliminar a través de LINQ
            Dim prov = (From o In cls.Proyectos Where o.id = id Select o).First()

            ' Eliminar el proyecto de la base de datos
            cls.Proyectos.DeleteOnSubmit(prov)
            cls.SubmitChanges()
        End If

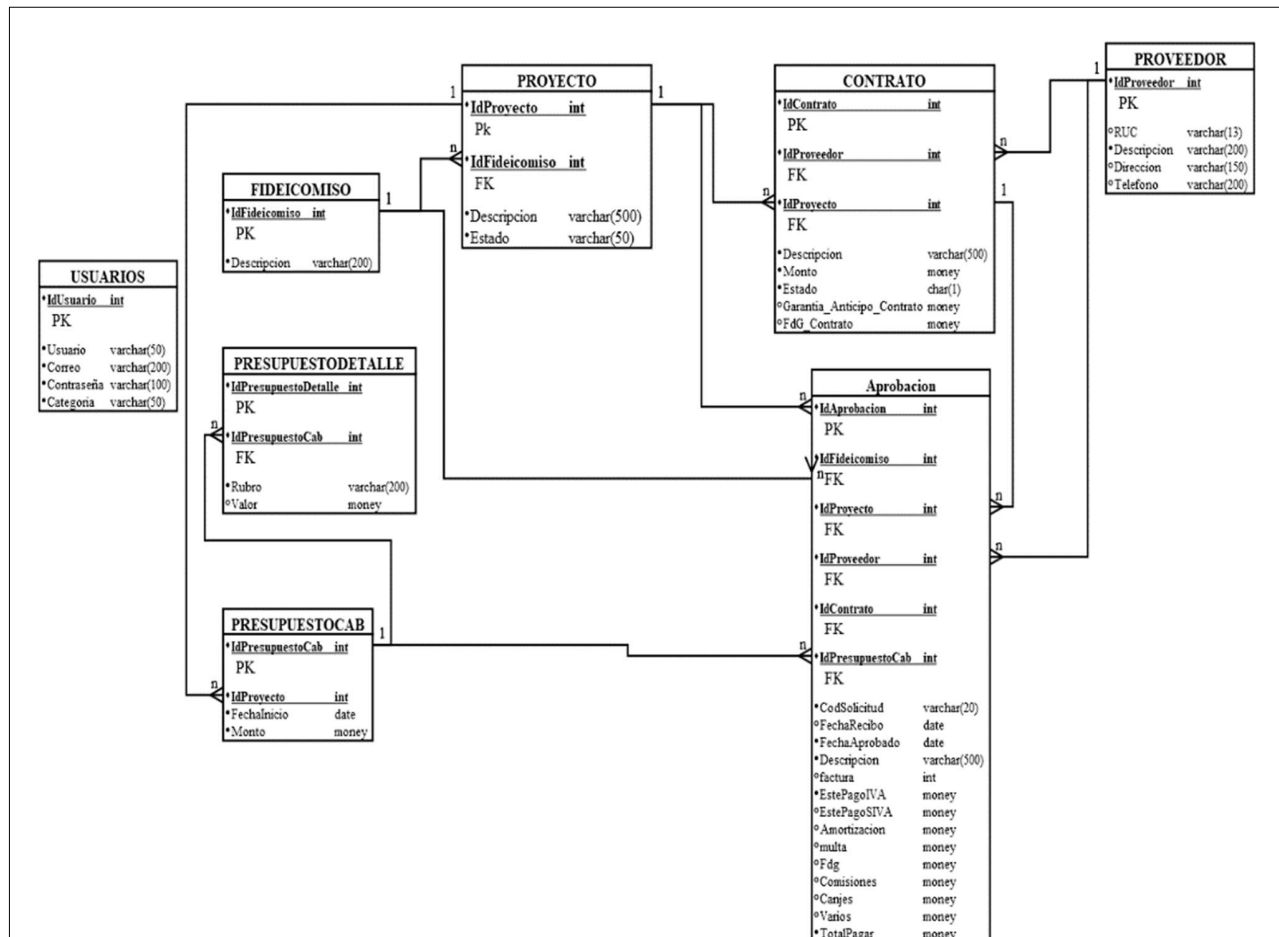
        Catch ex As Exception
            ' Manejar cualquier error que ocurra durante el proceso
            MessageBox.Show("Ocurrió un error al eliminar el proyecto: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

        Finally
            RefreshDtgVProyectos(DtgVProyectos)
        End Try
    End Sub
```

Elaborado: Jimmy Jesús Salazar Gómez.

3.7. Modelo Físico de Datos.

Ilustración 14. Modelación lógica de datos



Elaborado: Jimmy Jesús Salazar Gómez.

3.8. Tabla de Base de Datos.

En las Tablas de base de datos, se especifican el tipo de datos y clave para su verificación, en donde:

- **Primary Key (pk):** En una tabla se encuentra una columna o una combinación de columnas cuyos valores permiten la identificación única de cada fila de la tabla. (Microsoft Corporation, s.f.).

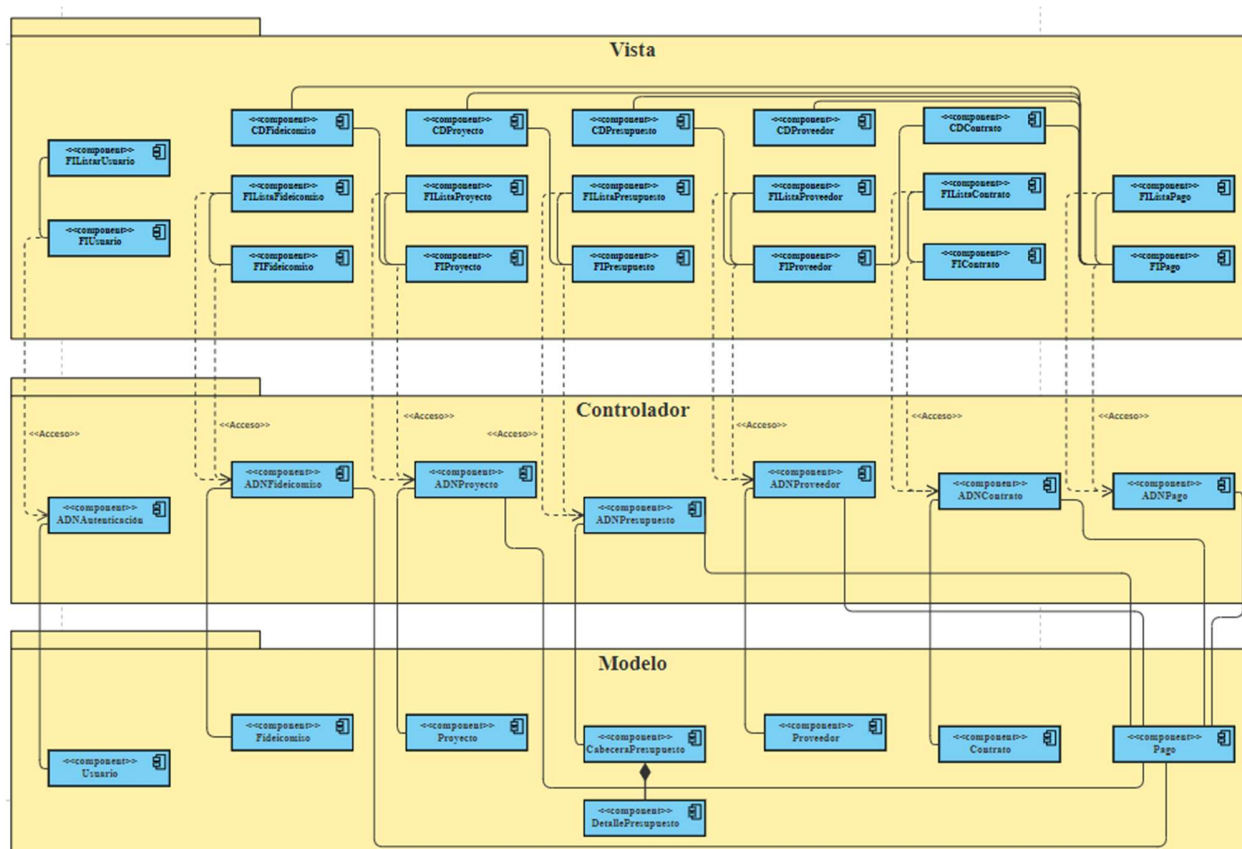
- Foreign Key (fk): Se refiere a una columna o a una combinación de columnas que se emplean para establecer una conexión y asegurar una relación coherente entre los datos que se encuentran en dos tablas diferentes. (Microsoft Corporation, s.f.)
A Continuación, se describe las entidades del diagrama lógico en la Ilustración 14.
- La tabla Aprobación, especifica que entre los datos que serán impresos en las distintas interacciones son: los nombres del Fideicomiso, Proyecto, Proveedor, Contratos, Montos de la Solicitud, los rubros y montos totales.
- La tabla Fideicomiso, especifica la información que será ingresada al momento de llenar los campos.
- En la tabla Proyecto, nos muestra los datos que serán ingresados en el sistema para el registro de los Proveedores y contratos.
- En la tabla PresupuestoCabecera, nos muestra los datos que serán ingresados en el sistema.
- En la tabla PresupuestoDetalle, especifica los datos correspondientes al presupuesto del proyecto.
- En la tabla Proveedor, nos muestra los datos que serán ingresados para el registro en el sistema.

3.9. Diagrama de Componentes.

Según (Pressman & Maxim, 2015). El diagrama de componentes es útil para ilustrar la organización de los componentes y las dependencias entre ellos, lo que proporciona una vista clara de la arquitectura del sistema. Además, permite identificar de manera efectiva cómo los distintos módulos se comunican y cooperan en el intercambio de información.

A continuación, se presenta el diagrama de componentes que establece una conexión tangible entre el análisis realizado y la fase de construcción del software. Este diagrama materializa las relaciones y dependencias identificadas previamente, proporcionando una representación visual de cómo los diversos componentes del sistema interactúan y colaboran para lograr los objetivos establecidos.

Ilustración 15. Diagrama de Componentes.



Elaborado: Jimmy Jesús Salazar Gómez

3.10. Diagrama de Despliegue.

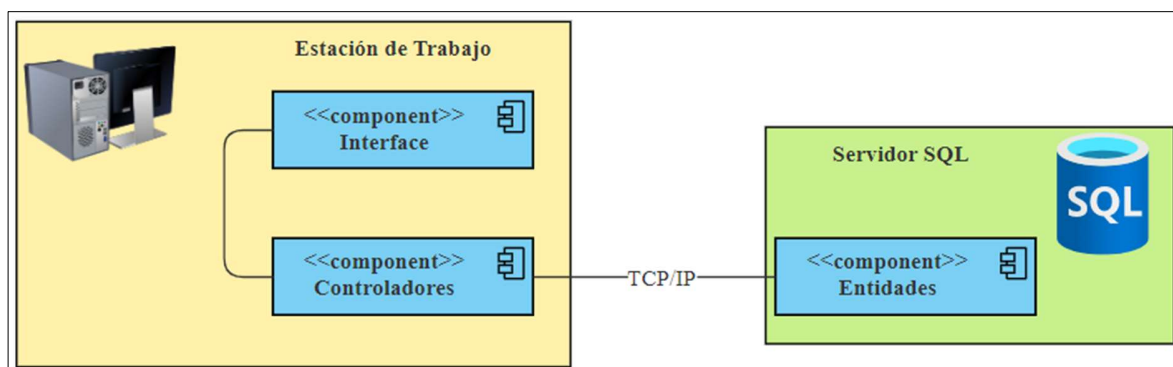
Según (Pressman, 2010), El diagrama de despliegue proporciona una vista clara de cómo los diferentes componentes del sistema se despliegan en la infraestructura física y cómo se comunican entre sí a través de la red. Esto es crucial para comprender cómo los recursos se asignan y cómo se gestionan las interacciones en el entorno de producción.

El diagrama de despliegue también ayuda a identificar posibles cuellos de botella en la red o problemas de escalabilidad en la infraestructura. Además, permite planificar adecuadamente la distribución de los componentes y recursos, asegurando un rendimiento óptimo del sistema en producción (Fowler, 2022).

La Ilustración 16, exhibe nodos físicos que albergan los componentes del sistema, como el servidor de aplicaciones, la base de datos y los dispositivos que ejecutan la

interfaz de usuario. Las conexiones entre estos componentes son visibles, como la comunicación entre el servidor de aplicaciones y la base de datos. El diagrama también refleja cómo los componentes se distribuyen en los nodos. Este esquema es esencial para planificar la implementación y garantizar el rendimiento adecuado del sistema.

Ilustración 16. Diagrama de Despliegue



Elaborado: Jimmy Jesús Salazar Gómez.

3.11. Conclusión del Capítulo.

Se puede concluir que en el capítulo se ha llevado a cabo una descripción detallada de la arquitectura utilizada en el modelamiento del sistema informático que se está desarrollando, así como del contenido de este a través de diferentes tipos de diagramas, como los de clases, secuencia y componentes. Además, se ha realizado una descripción de la arquitectura física y lógica de la base de datos del sistema.

Es importante destacar que se ha utilizado el UML para el diseño del sistema informático, lo que implica que se ha seguido un estándar reconocido en la industria para el modelamiento de sistemas de software. Asimismo, la elección de Visual Basic.Net junto con una base de datos SQL Server responde a la selección de tecnologías ampliamente adoptadas en la industria, lo que garantiza eficiencia en el desarrollo, mantenimiento y escalabilidad del software.

CAPITULO IV

4. PRUEBAS Y ANÁLISIS DE FACTIBILIDAD

4.1. Breve descripción del Capítulo.

Este capítulo se enfoca en la importancia de medir el esfuerzo necesario para el desarrollo del software y en la realización de pruebas de caja negra para garantizar la calidad del sistema.

4.2. Planificación basada en uno de los métodos de estimación.

La estimación del esfuerzo de producción es una necesidad en todas las áreas de la industria y una actividad imprescindible para el estudio de viabilidad de los proyectos. Desde el momento en que las empresas comenzaron a considerar las aplicaciones informáticas como productos industriales, aparecieron algunos problemas fundamentales que resolver: el cumplimiento de los plazos de entrega dentro de costos establecidos manteniendo niveles de calidad; así como poder realizar un seguimiento y control de la evolución de los proyectos. (Ferreira y otros, 2014). Este enfoque emplea la técnica de los puntos de caso de uso para medir el tamaño del proyecto, y posteriormente aplica factores de ajuste que consideran la complejidad del sistema, la experiencia del equipo de desarrollo y otros elementos pertinentes (Perez & Yzquierdo, 2014)

La estimación basada en puntos de caso de uso es una práctica ampliamente aceptada para la planificación de proyectos de software, ya que permite una evaluación más precisa de los recursos requeridos y puede ayudar a mejorar la toma de decisiones en el desarrollo (Wieggers & Beatty, 2013).

Una vez se ha determinado el tamaño del proyecto y se han aplicado los factores de ajuste, es posible estimar el esfuerzo total en horas de trabajo, lo que a su vez permite calcular el número de desarrolladores necesarios y el tiempo requerido para completar el proyecto.

La planificación basada en este método de estimación resulta esencial, ya que proporciona a los desarrolladores y gerentes de proyecto una comprensión precisa de los recursos necesarios para llevar a cabo un proyecto de software. Esto garantiza que

los recursos se asignen adecuadamente y que el proyecto se entregue dentro del presupuesto y en el plazo establecido.

Este método tiene como objetivos:

- Medir la tecnología involucrada en la construcción del software.
- Ofrecer una medida del tamaño del proyecto.
- Proveer una herramienta de estimación del software.

4.2.1. Caracterización de los requisitos según su complejidad.

Los requisitos funcionales fueron caracterizados por su complejidad, tomando en cuenta el esfuerzo para el diseño, implementación e integración de un requisito funcional.

Tabla 13. Caracterización de RF según su Complejidad

Valor	Descripción
ALTA	El requisito funcional incluye 8 o más servicios y requiere la manipulación de 5 o más entidades del sistema. La figura que accede al sistema es una persona y la interfaz de usuario contiene elementos gráficos, de animación y componentes tradicionales de entrada y salida de información. Además, el requisito funcional no utiliza elementos de reutilización y es desarrollado completamente por un único programador.
MEDIA	El requerimiento funcional implica entre 5 y 8 operaciones, se crean de 3 a 5 entidades del sistema para consulta, modificación o creación, y el usuario que accede puede ser una persona o un sistema externo que utiliza un protocolo o una interfaz basada en texto como Servicios Web, XML o correo electrónico. La interfaz diseñada solo cuenta con elementos convencionales para la captura y presentación de información. Además, se utilizan elementos que requieren ajustes en la codificación antes de la generación de la implementación del caso de uso.

BAJA	<p>El requisito funcional contiene de 1 a 4 servicios y se instancian hasta 2 entidades del sistema para consulta, modificación o creación. El usuario que accede puede ser una persona o un sistema externo que aprueba mediante una interfaz de programación como API o DLL. La interfaz diseñada solamente posee mecanismos tradicionales de captura y muestra de información, o incluso puede no tener una interfaz asociada. Además, se utilizan elementos que cumplen con el requisito funcional y sólo se requiere ajustar los atributos de uso.</p>
------	---

Elaborado: Jimmy Jesús Salazar Gómez.

4.2.2. Clasificación de los requisitos funcionales.

Tabla 14. Clasificación de RF

Id	Nombre del Requisito	Complejidad
RF1.1	Crear Usuario	Baja
RF1.2	Eliminar Usuario	Baja
RF1.3	Modificar Usuario	Baja
RF1.4	Buscar Usuario	Baja
RF2.1	Crear Fideicomiso	Baja
RF2.2	Eliminar Fideicomiso	Baja
RF2.3	Modificar Fideicomiso	Baja
RF2.4	Buscar Fideicomiso	Baja
RF3.1	Crear Proyecto	Baja
RF3.2	Eliminar Proyecto	Baja
RF3.3	Modificar Proyecto	Baja

RF3.4	Buscar Proyecto	Baja
RF4.1	Crear Presupuesto	Baja
RF4.2	Eliminar Presupuesto	Baja
RF4.3	Modificar Presupuesto	Baja
RF4.4	Buscar Presupuesto	Baja
RF5.1	Crear Proveedor	Baja
RF5.2	Eliminar Proveedor	Baja
RF5.3	Modificar Proveedor	Baja
RF5.4	Buscar Proveedor	Baja
RF6.1	Crear Contrato	Baja
RF6.2	Eliminar Contrato	Baja
RF6.3	Modificar Contrato	Baja
RF6.4	Buscar Contrato	Baja
RF7.1	Crear Pago	Media
RF7.2	Eliminar Pago	Media
RF7.3	Modificar Pago	Media
RF7.4	Buscar Pago	Media
RF8	Reporte de Proyectos	Media

Elaborado: Jimmy Jesús Salazar Gómez.

4.2.3. Definición de factores de complejidad.

En la Tabla 15, se asignan los grados de complejidad con los cuales se trabajará el análisis.

Tabla 15. Definición de factores de complejidad.

Id	Complejidad del Requisito	Peso
C1	Alta	3
C2	Media	2
C3	Baja	1

Elaborado: Jimmy Jesús Salazar Gómez.

Para la medición de la complejidad de los actores, se procederá a valorar de la siguiente forma.

Tabla 16. Definición de factores

Id	Actores	Peso	Criterio de Clasificación
A1	Alta	0,5	Persona delegada que interactúe con el sistema mediante una interfaz gráfica (GUI).
A2	Media	0,3	Otro sistema interactuando a través de un protocolo o una persona interactuando a través de una interfaz en modo consola.
A3	Baja	0	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, WS).

Elaborado: Jimmy Jesús Salazar Gómez.

4.2.4. Factores Técnicos

Se denominan elementos técnicos a los componentes tecnológicos que tienen un impacto positivo o negativo en los requisitos del software. A diferencia de los factores ambientales, estos elementos pueden ser modificados ya que están relacionados directamente con la complejidad tecnológica y los requisitos no funcionales del sistema.

Estos factores influyen directamente en el esfuerzo requerido, especialmente durante la etapa de implementación.

Tabla 17. Descripción de los Factores técnicos

Id	Factor Técnico	Peso (0,5 - 2)	Valor (0-1)	Descripción
FT1	Sistema distribuido	2	Toma de valor definido	Corresponde con requisitos cuyos componentes hardware y/o software, se encuentran en otros ordenadores conectados en red, se comunican y coordinan sus acciones mediante el paso de información, para el logro de un objetivo.
FT2	Objetivos de rendimiento	1,5	Toma de valor definido	Capacidad del requisito para cumplir con sus funciones, por lo general va muy relacionado con la respuesta que ostenta el requisito ante el acceso de varios usuarios al mismo tiempo y su tiempo de respuesta.
FT3	Eficiencia respecto al usuario final	1	Toma de valor definido	Se refiere a la complejidad visual diseñada para una buena manipulación del sistema por parte del usuario.
FT4	Procesamiento Complejo	1	Toma de valor definido	Complejidad en la definición del requisito y complejidad de los algoritmos a utilizar, ejemplo: Implementación de funciones matemáticas, gráficos, etc.
FT5	Código reutilizable	1	Toma de valor definido	Reutilización de código ya implementado en otro requisito o sistema.
FT6	Instalación sencilla	1	Toma de valor definido	Instalación del programa de manera sencilla y fácil de comprender.
FT7	Validaciones	0,5	Toma de valor definido	Se refiere a la cantidad de validaciones a realizar en cierta acción en el sistema.
FT8	Portabilidad	0	Toma de valor definido	Capacidad de operación del requisito en otras plataformas y dispositivos
FT9	Librerías Externas	1	Toma de valor definido	Necesidad de incorporación de librerías, APIs, componentes desarrollados por terceros para completar el funcionamiento del requisito.
FT10	Uso Concurrente	1	Toma de valor definido	Ejecución de los procesos en un solo sistema o en distintos computadores.

FT11	Características de Seguridad	2	Toma de valor definido	La información esté disponible solo para las entidades permitidas, integridad de la información, disponibilidad de la información por las entidades autorizadas, validar la propiedad de la información en procesos de comunicación.
FT12	Integración con dispositivos externos	0	Toma de valor definido	Integración de la implementación del requisito con hardware externo (lector de huella, lector de código de barra, cámaras, etc.).
FT13	Se requiere formación especial	0	Toma de valor definido	Formación especial para los usuarios finales en la utilización del requisito.

Elaborado: Jimmy Jesús Salazar Gómez.

A Continuación, se presenta la ponderación para los valores por factor

Tabla 18. Descripción de los valores de ponderación de factores técnicos.

Valor	Descripción
0	Falta del factor técnico. (Peor caso)
1	Presencia o influencia parcial del factor técnico (presente en al menos una y no en todas, las transacciones del requisito) o Presencia o influencia total del factor técnico (presente en todas las transacciones del requisito). (Mejo caso).

Elaborado: Jimmy Jesús Salazar Gómez.

4.2.5. Factores Ambientales.

Los factores externos son aquellos elementos fuera del control del equipo del proyecto, pero que pueden influir en las condiciones en las que se definen e implementan los requisitos del proyecto. Estos factores pueden restringir o modificar los requisitos del proyecto y, por lo tanto, afectar su resultado final. La consideración de estos factores externos es esencial en la gestión de proyectos, ya que su influencia dependerá de la cercanía al proyecto. Algunos factores externos pueden ser difíciles de influir, y se deberá actuar en consecuencia. (Meredith & Mantel, 2021).

Tabla 19. Descripción de los factores ambientales

Id	Factor Ambiental	Peso (0,5 - 2)	Valor (1, 2, 3)	Descripción
FA1	Familiarización con el negocio	2	Toma de valor definido	Se refiere al conocimiento del equipo del sistema sobre el negocio a implementar.
FA2	Disponibilidad de equipo	2	Toma de valor definido	La disponibilidad se refiere al tiempo con el que dispone el equipo de sistema para dedicar al sistema.
FA3	Experiencia en el entorno de desarrollo	1,75	Toma de valor definido	Representa las habilidades, conocimientos y experiencia del equipo en las herramientas, tecnologías a utilizar en la implementación del requisito.
FA4	Familiarización con el requisito	2	Toma de valor definido	Se refiere a cuando el responsable de la implementación ha desarrollado requisitos similares.
FA5	Estabilidad del requisito	1,75	Toma de valor definido	El requisito se encuentra correctamente definido, estable, con pocas indefiniciones o poco propenso al cambio.
FA6	Conocimiento en herramienta de control de versiones	2	Toma de valor definido	Habilidades y capacidades del equipo en la utilización de herramientas de control de versiones para código y documentación.
FA7	Motivación	1	Toma de valor definido	Grado de compromiso del equipo con el cumplimiento de los objetivos y/o obtención de recompensas en dependencia del trabajo realizado.
FA8	Equipo Distribuido geográficamente	1	Toma de valor definido	Se refiere si el personal del equipo de proyecto se encuentra ubicado geográficamente disperso.
FA9	Dificultad de lenguaje de programación	1,75	Toma de valor definido	Se refiere a si se considera complejo el uso de las plataformas de desarrollo y el lenguaje de programación a utilizar.

Elaborado: Jimmy Jesús Salazar Gómez.

En la siguiente tabla se muestra la ponderación de los factores ambientales.

Tabla 20. Ponderación de factores Ambientales

Valor	Descripción
3	Ausencia del factor ambiental (Mejor Caso)
2	Presencia o influencia parcial de factor ambiental
1	Presencia o influencia total del factor ambiental. (Peor Caso).

Elaborado: Jimmy Jesús Salazar Gómez.

4.2.6. Métricas para medir para realizar la estimación.

Las métricas para evaluar son:

- Factor de Peso del Actor Sin Ajustar (FPASA). $FPASA = \sum_{i=1}^n \text{Peso}_i * \text{Actor}_i$
- Factor de Peso del Requisito Sin Ajustar (FPRSA): $FPRSA = \text{Peso}$
- Puntos de Requisitos Sin Ajustar (PRSA): $PRSA = FPASA + FPRSA$.
- 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): $PRA = PRSA * FCT * FA$.
- El Esfuerzo estimado en horas Persona: $E = PRA * HH$ « HH= 10Horas »

La estimación de esfuerzo es un elemento crucial en la gestión de proyectos de software. En algunos casos, se utilizan valores estándar para convertir los puntos de función en horas/hombre, lo que permite estimar el esfuerzo requerido para un proyecto. Estos valores estándar pueden variar según la organización o la industria, y en algunos casos, pueden ser basados en datos históricos o en estudios de benchmarking. Sin embargo, es importante recordar que estos valores estándar son solo una guía y que la estimación de esfuerzo debe ser adaptada a las circunstancias específicas de cada proyecto. Además, se recomienda revisar y ajustar estas estimaciones a medida que se obtiene información real sobre la velocidad y la productividad del equipo de desarrollo. (Herron & Garmus, 2017).

4.2.7. Porcentaje de actividades por ciclo de vida.

El objetivo es establecer un tiempo específico para cada fase del ciclo de vida del proyecto utilizando un modelo de estimación adaptado del sistema de la Empresa Nylic. Este sistema está diseñado para gestionar el registro de solicitudes, el pago a proveedores y el control de contratos de proyectos. La implementación de un modelo de estimación adecuado es importante para lograr una planificación eficaz y un control adecuado del tiempo en cada fase del ciclo de vida del proyecto. De esta manera, se pueden establecer plazos realistas y asegurar que el proyecto se complete en el tiempo y presupuesto previstos.

A continuación, se define los porcentajes para el ciclo de vida del proyecto.

Tabla 21. Porcentajes de Actividades del ciclo de vida

Fase	Porcentaje
Actividades de Gestión	15%
Modelado del Negocio	15%
Requerimientos	10%
Análisis y Diseño	10%
Implementación (Codificación)	30%
Pruebas	10%
Despliegue	10%

Elaborado: Jimmy Jesús Salazar Gómez.

Se determinaron los valores de porcentaje tomando en cuenta los porcentajes de distribución para el esfuerzo total en el desarrollo del sistema propuesto por (Herron & Garmus, 2017) en el método inicial ya que en la empresa no se tiene información histórica que sirva para obtener un valor de porcentaje propio. Los porcentajes se irán ajustando

conforme existan datos históricos de la duración real de cada proceso en un periodo de tiempo adecuado por ende los porcentos se irán ajustando con respecto al método descrito anteriormente.

Enseguida se presentan los resultados obtenidos tras aplicar el método de estimación al Sistema de Control para Pagos a Proveedores, el cual será responsable de la gestión del registro de solicitudes, el pago a proveedores y el control de contratos de la empresa Nylic.

Ilustración 17. Método de Estimación.

ID Requisitos	Complejidad	Actores		FPASA	FPRSA	PRSA	Factores Técnicos													Factores Ambientales									PRA	HH	E(Horas)		
		A1	A2				FT1	FT2	FT3	FT4	FT5	FT6	FT7	FT8	FT9	FT10	FT11	FT12	FT13	FCT	FA1	FA2	FA3	FA4	FA5	FA6	FA7	FA8				FA9	FA
RF1.1	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF2.1	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF2.2	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF2.3	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF2.4	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF3.1	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF3.2	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF3.3	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF3.4	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF4.1	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF4.2	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF4.3	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF4.4	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF5.1	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF5.2	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF5.3	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF5.4	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF6.1	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF6.2	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF6.3	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF6.4	Baja	1	0	0,5	2	2,5	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	1,35	10	13,48
RF7.1	Media	1	0	1	3	4	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	2,16	10	21,56
RF7.2	Media	1	0	1	3	4	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	2,16	10	21,56
RF7.3	Media	1	0	1	3	4	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	2,16	10	21,56
RF7.4	Media	1	0	1	3	4	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	2,16	10	21,56
RF8	Media	1	0	1	3	4	1	1	1	1	1	0	1	0	0	1	1	0	1	0,7	1	2	1	2	2	1	1	1	1	0,77	2,16	10	21,56

Total del Esfuerzo del Proyecto 390,78

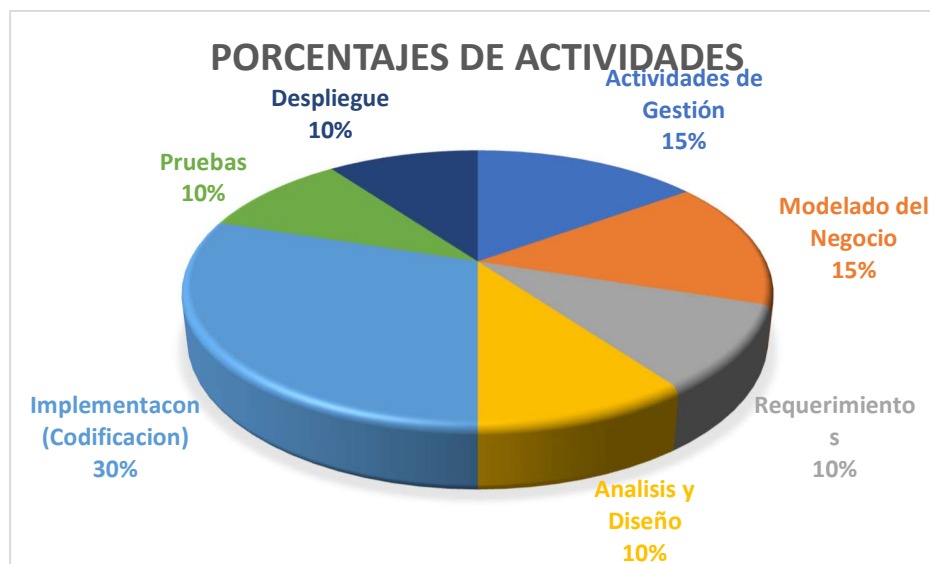
Elaborado: Jimmy Jesús Salazar Gómez.

Tabla 22. Porcentaje de actividades de ciclo de vida.

Fase	Porcentaje	Tiempo/Hombres
Actividades de Gestión	15%	58,62
Modelado del Negocio	15%	58,62
Requerimientos	10%	39,08
Análisis y Diseño	10%	39,08
Implementación (Codificación)	30%	117,23
Pruebas	10%	39,08
Despliegue	10%	39,08
Soporte	NO APLICA	

Elaborado: Jimmy Jesús Salazar Gómez.

Ilustración 18. Gráfico Porcentajes de Actividades



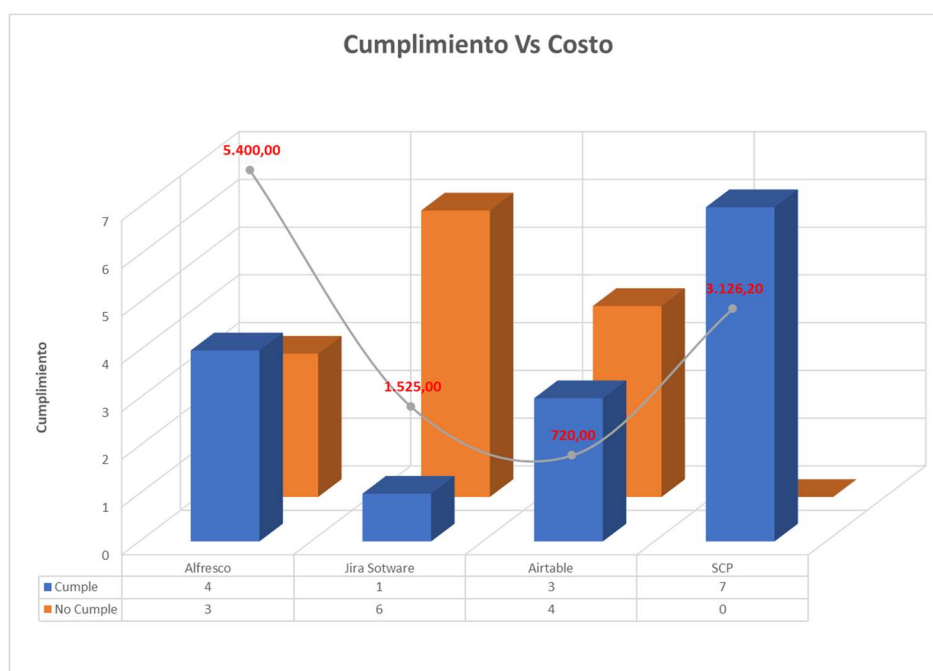
Elaborado: Jimmy Jesús Salazar Gómez.

El esfuerzo requerido para desarrollar un sistema informático es una estimación importante que se utiliza para planificar y asignar recursos adecuados al proyecto. En la Ilustración 17. Método de Estimación. **Ilustración 17**, se indica que el esfuerzo requerido

para desarrollar el sistema es de **390,78 horas/hombre**. Para calcular el costo del desarrollo, se tiene en cuenta el salario mínimo de un ingeniero en sistemas en Ecuador, que se estima en **8 USD/hora**. Por lo tanto, el costo total del desarrollo del sistema por un solo recurso sería de **3,126.20 USD**.

Además, se debe considerar el número de horas laborales en un mes, que es de **160 horas**. En este caso, se estima que el sistema se desarrollará en un tiempo aproximado de **tres meses**, considerando todo el mes de vida tradicional para un proceso de desarrollo de software por un solo recurso. Es importante tener en cuenta que estos cálculos son una estimación y pueden variar según las circunstancias y factores que influyan en el proyecto. Por lo tanto, es necesario revisar y ajustar periódicamente estos cálculos a medida que avanza el proyecto.

Ilustración 19. Gráfico Comparativo de Costos y Requerimientos



Elaborado: Jimmy Jesús Salazar Gómez.

4.3. Pruebas de caja negra y su validación.

A continuación, se llevará a cabo un paso fundamental en el proceso de desarrollo del sistema, que consiste en la identificación de clases equivalentes, el diseño de casos

de prueba válidos e inválidos, y la creación de una matriz de casos de prueba. Para ilustrar este proceso, se tomarán como muestra los siguientes requisitos funcionales:

- Crear Registro de Proyecto
- Crear Proveedor.

Este enfoque detallado, garantizará que tanto el Registro de Proyecto como la creación de Proveedores sean sometidos a pruebas, detectando y solucionando posibles fallos antes de la implementación final del sistema.

Es fundamental llevar a cabo este proceso de pruebas para brindar a los usuarios finales una experiencia fluida y sin contratiempos, así como para garantizar el cumplimiento de los requisitos funcionales establecidos.

4.3.1. RF3.1: Crear Registro de Proyecto

Identificación de las Clases equivalentes:

Tabla 23. Identificación Clases Equivalentes - Crear Proyecto -

Parámetro de entrada	Regla Heurística para aplicar	Clases Validas (CV)	Clases Invalidas (CI)
Nombre del Proyecto	Llenar el campo con Nombre del Proyecto	CV2: Llenar el campo de Nombre	CI2: No llenar el campo.
Nombre del Fideicomiso	Buscar conjunto de valores admitidos	CV3: Seleccionar el Fideicomiso de la Lista	CI3: No seleccionar el Fideicomiso de la Lista
Fecha Inicio	Llenar campo con la fecha de inicio del Proyecto	CV4: Llenar el campo con la fecha de inicio del proyecto, no existen restricciones alfanuméricas ni de longitud	CI4: No llenar el campo
Fecha Finalizado	Llenar campo con la fecha de Final del Proyecto	CV5: Llenar el campo con la fecha de finalización del proyecto, no existen restricciones alfanuméricas ni de longitud	CI5: No llenar el campo cuando el estado sea Finalizado

Estado	Buscar conjunto de valores admitidos	CV6: Seleccionar el Estado de la Lista.	CI6: No seleccionar el Estado de la Lista
--------	--------------------------------------	---	---

Diseño de Casos de prueba para las Clases Válidas.

Tabla 24. Casos de Pruebas Clases válidas - Crear Proyecto -

Nombre del Proyecto	Nombre del Fideicomiso	Fecha Inicio	Fecha Finalización	Estado	Clases válidas cubiertas
Edificio Millenium Towers	Fideicomiso Millenium	1/4/2020	12/12/2022	Finalizado	CV2, CV3, CV4, CV5, CV6.
Millenium Village	Fideicomiso Village	20/6/2022	Vacío	En Ejecución	CV2, CV3, CV4, CV5, CV6.

Elaborado: Jimmy Jesús Salazar Gómez.

Diseño clase de pruebas para las clases inválidas.

Tabla 25. Casos de Pruebas Clases inválidas - Crear Proyecto -

Nombre del Proyecto	Nombre del Fideicomiso	Fecha Inicio	Fecha Finalización	Estado	Clases inválidas cubiertas
Vacío	Fideicomiso Millenium	1/4/2020	12/12/2022	Finalizado	CI3, CI4, CI5, CI6
Edificio Millenium Towers	Vacío	1/4/2020	12/12/2022	Finalizado	CI2, CI4, CI5, CI6
Edificio Millenium Towers	Fideicomiso Millenium	Vacío	12/12/2022	Finalizado	CI2, CI3, CI5, CI6
Edificio Millenium Towers	Fideicomiso Millenium	1/4/2020	Vacío	Finalizado	CI2, CI3, CI4, CI6
Edificio Millenium Towers	Fideicomiso Millenium	1/4/2020	12/12/2022	Vacío	CI2, CI3, CI4, CI5
Millenium Village	Fideicomiso Village	20/6/2022	12/12/2022	En Ejecución	CI2, CI3, CI4, CI6

Elaborado: Jimmy Jesús Salazar Gómez.

Matriz de los casos de prueba.

Tabla 26. Matriz caso de prueba - Crear Proyecto -

Clase Equivalente	Resultado Esperado	Resultado Final
CV2, CV3, CV4, CV5, CV6.	Proyecto Guardado	Proyecto Guardado
CV2, CV3, CV4, CV5, CV6.	Proyecto Guardado	Proyecto Guardado
CI2	Mensaje "Campo de Nombre de Proyecto no puede estar en blanco"	Mensaje "Campo de Nombre de Proyecto no puede estar en blanco"
CI3	Mensaje "Debe seleccionar un Fideicomiso"	Mensaje "Debe seleccionar un Fideicomiso"
CI4	Mensaje "Campo de Fecha de Inicio no puede estar en blanco"	Mensaje "Campo de Fecha de Inicio no puede estar en blanco"
CI5	Mensaje "Campo de Fecha de Finalización no puede estar en blanco"	Mensaje "Campo de Fecha de Finalización no puede estar en blanco"
CI6	Mensaje "Debe seleccionar un Estado"	Mensaje "Debe seleccionar un Estado"

Elaborado: Jimmy Jesús Salazar Gómez.

4.4. Verificación del Sistema.

El software desarrollado para la Empresa NYLIC ha sido implementado y puesto en funcionamiento para la gestión eficiente de los procesos de registro y pago de solicitudes a proveedores. Este software ha sido diseñado y adaptado específicamente para satisfacer las necesidades y requerimientos de la empresa, optimizando sus operaciones y agilizando el flujo de trabajo relacionado con los pagos a proveedores.

Como evidencia de su correcto funcionamiento y validez, se adjunta un certificado emitido por una institución competente que avala la calidad y eficiencia del software implementado. Dicho certificado representa una validación adicional de que el sistema cumple con los estándares y requisitos establecidos, brindando confianza y respaldo a

la Empresa NYLIC en cuanto a la efectividad y fiabilidad de la solución tecnológica implementada.

Este logro en la implementación exitosa del software ha permitido a la Empresa NYLIC mejorar sus procesos de gestión de pagos a proveedores, incrementar la eficiencia operativa y garantizar una mayor precisión en el registro y seguimiento de solicitudes. Además, la implementación del software ha contribuido a la reducción de errores y la optimización de los tiempos de respuesta en el manejo de los pagos, generando un impacto positivo en la productividad y rentabilidad de la empresa.

Ilustración 20. Verificación del sistema



Elaborado: Jimmy Jesús Salazar Gómez.

4.5. Errores encontrados durante la Instalación de la Aplicación.

Durante la implementación de la aplicación en los sistemas de la red, se encontró un problema en la conexión con la base de datos. A pesar de que el Directorio Activo (AD) estaba funcionando correctamente y controlando los nombres de usuario y accesos, la aplicación no lograba establecer una conexión efectiva con el servidor SQL utilizando el nombre de este. Para resolver esta situación, se optó por configurar la aplicación para

que se conectara mediante la dirección IP del servidor SQL en lugar de depender de la resolución de nombres a través del AD. Esta solución permitió evitar las posibles complicaciones asociadas a la resolución de nombres y aseguró una conexión más estable y confiable con la base de datos.

La aplicación no funcionó correctamente en los equipos previstos debido a la falta de permisos de administrador. Para solucionarlo, se otorgaron los permisos necesarios para que la aplicación pudiera ejecutarse sin problemas en esos equipos. Esto permitió que la aplicación funcionara correctamente y se llevara a cabo su implementación de manera exitosa.

4.6. Conclusión del Capítulo.

En este capítulo, se presentó en detalle el proceso de estimación del esfuerzo necesario para el desarrollo del software. Esta metodología permite cuantificar y medir el tamaño y complejidad funcional del sistema, brindando una base sólida para la planificación y asignación de recursos en el desarrollo de software.

Además, se abordó la importancia de las pruebas de caja negra como parte integral del proceso de validación del sistema. Mediante la ejecución de escenarios de prueba exhaustivos, se logró identificar y solucionar diversos problemas, mejorando así el funcionamiento general del sistema.

CONCLUSIONES

Se ha llevado a cabo un análisis crítico y técnico de las metodologías y herramientas tecnológicas propuestas en el contexto de la empresa. Este análisis ha demostrado su viabilidad y relevancia para abordar los desafíos específicos de la organización.

Se ha realizado una evaluación de los problemas existentes en los procesos actuales de la empresa. Además, se ha investigado y analizado soluciones existentes en el mercado. Como resultado de este análisis, se han identificado las tecnologías más adecuadas para resolver los problemas identificados.

Se ha diseñado un sistema informático que aborda los problemas identificados en los procesos actuales. El diseño se ha basado en las metodologías y herramientas tecnológicas, garantizando su coherencia y aplicabilidad en el contexto de la empresa.

Se han realizado pruebas de Caja Negra para validar el funcionamiento del sistema informático propuesto. Estas pruebas han demostrado que el sistema cumple con los objetivos establecidos y funciona correctamente de acuerdo con los requisitos definidos.

Se ha llevado a cabo la implementación del software informático basándose en el diseño propuesto. Esta etapa ha permitido materializar el sistema informático y ponerlo en funcionamiento en el entorno de la empresa.

RECOMENDACIONES.

Se recomienda considerar la posibilidad de mover la aplicación a una plataforma web. Esto tendría varias ventajas, ya que permitiría que la aplicación sea más flexible y esté al alcance de usuarios a través de navegadores web en cualquier tipo de dispositivo.

Sería aconsejable analizar la posibilidad de incorporar módulos que permitan a la Promotora del Proyecto registrar sus solicitudes de pago directamente. Esto agilizaría el intercambio de información y eliminaría la necesidad de depender del correo electrónico para este proceso.

BIBLIOGRAFÍA

- Aguilar, J. (15 de 05 de 2019). *¿Qué es el patrón MVC en programación y por qué es útil?* Recuperado el 24 de 05 de 2023, de Campus MVP.es: <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>
- Airtable Inc. (2023). *The fastest way to build apps*. Recuperado el 24 de 04 de 2023, de <https://airtable.com/>

- Alfresco Software, Inc. (2023). *Productivity Anywhere*. Recuperado el 08 de 08 de 2023, de Hyland: <https://www.alfresco.com/>
- Ambler, S. (2010). *The Object Primer 3rd Edition: Agile Model Driven Development with UML 2*. Cambridge University Press. Recuperado el 25 de 05 de 2023, de https://www.researchgate.net/publication/235616285_The_object_primer_agile_modeling-driven_development_with_UML_20
- Atlassian. (2023). *Jira Software*. Recuperado el 22 de 02 de 2023, de <https://www.atlassian.com/software/jira>
- Björk, B.-C. (2021). Document management – a key IT technology for the construction industry. *ITC - Digital library*, 1000-1009. Recuperado el 15 de 05 de 2023, de <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=543302726e0ca53b66177ce91dea759733133b96>
- Booch, G., Maksimchuk, R., Young, B., & Engel, M. (2007). *Object-Oriented Analysis and Design with Applications*. Addison-Wesley. Recuperado el 19 de febrero de 2023, de <https://zjnu2017.github.io/OOAD/reading/Object.Oriented.Analysis.and.Design.with.Applications.3rd.Edition.by.Booch.pdf>
- Brandon-Jones, A., & Slack, N. (2018). *Operations and Process Management: Principles and Practice for Strategic Impact*. Pearson UK. Recuperado el 20 de 05 de 2023, de https://books.google.com.ec/books?id=vA5MDwAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Carvache, O., Gutierrez, G., & Frias, C. (2022). Impacto de la innovación tecnológica en las Pymes de Ecuador. *Revista Espacios*, 211-230. Recuperado el 02 de 09 de 2023, de <https://www.revistaespacios.com/a18v39n47/a18v39n47p37.pdf>
- Daim, T., Dabic, M., & Bayraktaroglu, E. (2016). El comportamiento emprendedor de los estudiantes: diferencias internacionales y de género. *Journal of Innovation and Entrepreneurship*, 10. Recuperado el 22 de febrero de 2023, de <https://innovation-entrepreneurship.springeropen.com/articles/10.1186/s13731-016-0046-8>
- Davenport, T. (2018). *The AI Advantage: How to Put the Artificial Intelligence Revolution to Work*. MIT Press. doi:<https://doi.org/10.7551/mitpress/11781.001.0001>
- Ferreira, G., Galvez, D., Quintero, L. A., & Anton, J. (2014). Estimación del esfuerzo en proyectos de software utilizando técnicas de inteligencia artificial. La Habana. Recuperado el 25 de 10 de 2023, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992014000400001
- Fowler, M. (2019). *A Brief Guide to the Standard Object Modeling Language*. Boston: Addison-Wesley Professional. Recuperado el 30 de 05 de 2023, de

<https://bedford-computing.co.uk/learning/wp-content/uploads/2015/10/UML-Distrilled-3nd.pdf>

Fowler, M. (2022). *Patterns of Enterprise Application Architecture*. Addison-Wesley.

Gil López, A. (09 de 2018). *Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software*. Recuperado el 25 de 05 de 2023, de Universidad de Valladolid: <https://uvadoc.uva.es/bitstream/handle/10324/32875/TFG-I-1015.pdf?sequence=1&isAllowed=y>

Herron, D., & Garmus, D. (2017). *Function Point Analysis: Measurement Practices for Successful Software Projects*. Boston: Addison-Wesley Professional. Recuperado el 15 de 06 de 2023, de <https://archive.org/details/functionpointana0000garm/page/n5/mode/2up>

International Business Manager (IBM). (2023). *Manejo de errores y excepciones*. Recuperado el 08 de 06 de 2023, de <https://www.ibm.com/docs/es/app-connect/11.0.0?topic=extension-errors-exception-handling>

Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Hoboken: Wiley. Recuperado el 02 de 03 de 2023, de <http://www.mim.ac.mw/books/Kerzner%27s%20Project%20Management%20A%20Systems%20Approach...10thed.pdf>

Larman, C. (2004). *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development*. Prentice Hall. Recuperado el 13 de 06 de 2023, de <https://personal.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf>

Larson, B. (2016). *Microsoft SQL Server 2016 Reporting Services*. México: McGraw-Hill Education.

Macias, G., Cadena, D., Maldonado, J., & Alcivar, M. (2017). Cambio de la Matriz Productiva vía innovación: Caso Ecuador. *Revista de Ciencias Sociales y Económicas - UTEQ*, 1(2), 84-103. Recuperado el 3 de 04 de 2023, de <https://revistas.uteq.edu.ec/index.php/csye/article/view/259>

Maida, E., & Pacienza, J. (2015). *Metodología de Desarrollo de Software*. Recuperado el 20 de 05 de 2023, de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

Meredith, J., & Mantel, S. (2021). *Project Management: A Managerial Approach*. Wiley.

Microsoft. (2023). *Desarrollo de Aplicaciones .Net*. Recuperado el 08 de 08 de 2023, de Visual Studio: <https://visualstudio.microsoft.com/es/vs/features/net-development/>

Microsoft. (2023). *SQL Server - Base de datos relacional de Microsoft*. Recuperado el 08 de 08 de 2023, de <https://www.microsoft.com/es-es/sql-server/>

- Microsoft Corporation. (s.f.). *Restricciones entre claves principales y claves externas*. Recuperado el 29 de 05 de 2023, de <https://learn.microsoft.com/es-es/sql/relational-databases/tables/primary-and-foreign-key-constraints?view=sql-server-ver16>
- Montes de Oca, Y. (2015). *Sistema de Gestión de Información*. Recuperado el 02 de 06 de 2023, de https://www.eumed.net/libros-gratis/2012a/1150/modelo_del_negocio_y_modelo_del_sistema.html#:~:text=Un%20actor%20del%20negocio%20es,de%20sus%20resultados%20%5B20%5D.
- Nagel, C., Evjen, B., & Glynn, J. (2008). *Professional C#*. John Wiley & Sons.
- Nylic. (08 de 08 de 2023). *Empresa Consultora*. Recuperado el 22 de febrero de 2023, de <https://nylic.com.ec/es/>
- Ozkan, S. (2021). *Business Rule Management: A Systematic Review*. IEEE Access.
- Pavón, J. (2019). *Estructura de las Aplicaciones Orientadas a Objetos: El patrón Modelo-Vista-Controlador (MVC)*. Recuperado el 23 de 05 de 2023, de Universidad Complutense Madrid: <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>
- Perez, H., & Yzquierdo, R. (2014). Planificación del Alcance en Proyectos de Software. *Serie Científica de la Universidad de las Ciencias Informáticas*, 1 - 9. Recuperado el 15 de 06 de 2023, de <https://dialnet.unirioja.es/servlet/articulo?codigo=8590133>
- Petkovic, D. (2019). *Microsoft SQL Server: A Beginner's Guide*. McGraw-Hill Education. Recuperado el 08 de 08 de 2023, de <https://dl1.newoutlook.it/book/2020/03/Microsoft-SQL-Server-2019-A-Beginners-Guide.pdf>
- Pfleeger, S., & Atlee, J. (2009). *Software engineering: theory and practice*. Prantice Hall.
- Pons, C., Giandini, R., & Perez, G. (2010). *Desarrollo de Software dirigido por modelos*. Buenos Aires: Editorial de la Universidad Nacional de La Plata (EDULP) / McGraw-Hill Educación. Recuperado el 26 de 05 de 2023, de http://sedici.unlp.edu.ar/bitstream/handle/10915/26667/Documento_completo_.pdf?sequence=1&isAllowed=y
- Pressman, R. (2010). *Ingeniería del Software: Un Enfoque Práctico*. México: McGraw Hill. Recuperado el 3 de diciembre de 2022, de https://www.academia.edu/26345704/Ingenieria_de_software_enfoque_practico_7ed_Pressman_PDF
- Pressman, R., & Maxim, B. (2015). *Software engineering: A practitioner's approach*. New York: McGraw-Hill Education. Recuperado el 01 de 06 de 2023, de https://www.mlsu.ac.in/econtents/16_EBOOK-7th_ed_software_engineering_a_practitioners_approach_by_roger_s._pressman_.pdf

- Rodríguez, Y., & Galán, E. (25 de 09 de 2019). La inteligencia organizacional: necesario enfoque de gestión de información y del conocimiento. *Journal of Engineering Management*, 51-58. Recuperado el 08 de 10 de 2023, de <https://www.scielo.br/j/ci/a/h5tF7FQGcSr38FXfGGsVJmP/?lang=es#>
- Roques, P. (2006). *Modeling Software Systems Using UML2*. Wiley India Pvt.
- Sommerville, I. (2011). *Software Engineering*. Addison-Wesley. Recuperado el 3 de diciembre de 2022, de <https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf>
- Sutherland, J., & Beedle, M. (2020). *Manifesto for agile software development*. Harvard Business Review. Recuperado el 30 de 05 de 2023, de https://ai-learn.it/wp-content/uploads/2019/03/03_ManifestoofAgileSoftwareDevelopment-1.pdf
- Tanenbaum, A., & Van Steen, M. (2007). *Sistemas distribuidos: principios y paradigmas*. México: Pearson Educación.
- Utech Academy. (23 de febrero de 2023). *¿Qué es .NET y por qué deberías aprenderlo? Descubre las ventajas de este lenguaje de programación*. Recuperado el 28 de julio de 2023, de <https://utech.academy/blog/que-es-net-y-por-que-deberias-aprenderlo-descubre-las-ventajas-de-este-lenguaje-de-programacion/>
- Visual Paradigm. (2023). *Visual Paradigm: Leading UML, BPMN, EA, Agile, Project Management Software*. Recuperado el 08 de 08 de 2023, de <https://www.visual-paradigm.com/>
- Voon Kiong, L. (2017). *Visual Basic 2017 Made Easy*. Kindle Edition. Recuperado el 05 de 06 de 2023, de https://www.vbtutor.net/vb2017/vb2017me_preview.pdf
- Wieggers, K., & Beatty, J. (2013). *Software Requirements (Developer Best Practices)*. Microsoft Press. Recuperado el 13 de 05 de 2023, de https://www.booksfree.org/wp-content/uploads/2022/03/Software_Requirements_3rd_Edition_compressed.pdf