



**DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL MANEJO Y CONTROL DE
INVENTARIO PARA EMPRESAS RECUPERADORAS DE HIERRO**

**JONATHAN BAQUERO RODRÍGUEZ
JUAN SEBASTIÁN ARGÜELLO LOZANO**



**UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
FLORIDABLANCA
2023**



DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL MANEJO Y CONTROL DE INVENTARIO PARA EMPRESAS RECUPERADORAS DE HIERRO

**JONATHAN BAQUERO RODRÍGUEZ (324390)
JUAN SEBASTIÁN ARGÜELLO LOZANO (345476)**

PROYECTO DE GRADO

**DIRECTOR:
ING. DANITH PATRICIA SOLÓRZANO ESCOBAR**



**UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
FLORIDABLANCA
2023**



AGRADECIMIENTOS

Nos gustaría expresar nuestro más sincero agradecimiento a nuestra apreciada familia y nuestros queridos padres. Su inquebrantable apoyo y presencia constante en cada etapa de este proceso profesional han sido de un valor incalculable. Su amor, dedicación y aliento incondicional han sido el pilar sobre el cual hemos construido nuestra formación y crecimiento además de superar desafíos y alcanzar nuestros objetivos. Estamos profundamente agradecidos por su inmenso sacrificio y por ser nuestra fuente inagotable de inspiración y fortaleza. Este logro es también suyo, y estamos eternamente agradecidos por su presencia y amor incondicional en este viaje.

También deseamos expresar nuestro más profundo agradecimiento a nuestro mentor, cuya colaboración y dedicación se convirtieron en una parte fundamental de nuestro proyecto. Su experiencia y conocimientos en el campo fueron fundamentales para guiar nuestro proceso de trabajo, permitiéndonos abordar los desafíos con confianza y eficacia. Estamos verdaderamente agradecidos por su apoyo incondicional y su compromiso continuo durante todo el desarrollo de este proyecto.

No podemos dejar de mencionar a las empresas y organizaciones que nos brindaron su colaboración y facilitaron el acceso a los datos y recursos necesarios. Su apertura y disposición fueron cruciales para la realización de este proyecto y nos permitieron obtener resultados sólidos y relevantes.

Por último nos gustaría resaltar la importancia de cada uno de los integrantes del equipo en el desarrollo del proyecto. Trabajar juntos fue clave para enfrentar los desafíos, y lograr los objetivos establecidos. La colaboración y el espíritu de equipo demostrados a lo largo de este proceso fueron pilares para nuestro progreso y crecimiento como grupo. Estamos orgullosos de lo que hemos logrado como equipo y agradecidos por la oportunidad de trabajar junto a personas tan talentosas y comprometidas.

Llegamos a un punto en el que nuestros caminos se separan, pero llevamos con nosotros los aprendizajes y las experiencias. Nos esperan nuevos desafíos y proyectos de vida que abrazaremos con entusiasmo y determinación. El futuro nos depara emocionantes oportunidades, y estamos seguros de que cada uno de nosotros brillará en nuestros respectivos caminos.



DEDICATORIA

Dedicamos este proyecto a nuestros amados padres Eder Giovani Baquero Romero, Margarita Esther Rodríguez Pinzón y Liliana Lozano peña, Luis Fernando Ortiz, quienes han sido fuentes inagotables de amor, apoyo y guía a lo largo de nuestras vidas. Su dedicación y sacrificio nos han brindado las herramientas necesarias para enfrentar cada desafío y alcanzar nuestras metas. Agradecemos su confianza en nosotros y por ser nuestros pilares inquebrantables.

También dedicamos este proyecto a nuestro mentor Danith Patricia Solórzano Escobar quien ha sido una figura invaluable en nuestro crecimiento profesional. Gracias por su orientación, sabiduría y por compartir con nosotros su vasto conocimiento. Su apoyo constante y estímulo nos ha impulsado a superar nuestros límites y a alcanzar un nivel superior de excelencia.

A nuestros padres y mentor, les expresamos nuestro más profundo agradecimiento por creer en nosotros, por ser nuestros guías y por brindarnos su incondicional apoyo a lo largo de este proyecto. Sin su presencia y aliento, este logro no hubiera sido posible. Les dedicamos este proyecto como muestra de nuestro eterno agradecimiento y amor.



TABLA DE CONTENIDO

1.	RESUMEN DEL PROYECTO	15
1.1	INTRODUCCIÓN.....	15
1.2	SITUACIÓN PROBLEMA	16
1.3	JUSTIFICACIÓN.....	18
2.	OBJETIVOS	19
2.1	OBJETIVO GENERAL.....	19
2.2	OBJETIVOS ESPECÍFICOS.....	19
3.	MARCO REFERENCIAL	19
3.1	ANTECEDENTES	20
3.2	MARCO CONCEPTUAL.....	22
3.2.1	GLOBALIZACIÓN.....	22
3.2.2	EQUIPO DE CÓMPUTO	23
3.2.3	SERVICIOS DE ALMACENAMIENTO/BASES DE DATOS.....	24
3.2.4	COPIA DE SEGURIDAD/BACK-UP.....	26
3.2.5	APLICATIVO DESKTOP	27
3.2.6	FRAMEWORK DE DESARROLLO.....	28
3.2.7	MARCO DE TRABAJO	29
3.3	MARCO TECNOLÓGICO	31
3.3.1	NETBEANS.....	31
3.3.2	JAVA	32
3.3.3	MYSQL.....	33
3.3.4	DB4FREE.....	35
3.3.5	STAR UML	36
3.3.6	TRELLO	36

3.3.7 GITHUB.....	37
4. METODOLOGÍA	39
4.1 REVISIÓN BIBLIOGRÁFICA.....	39
4.2 DEFINICIÓN DE REQUERIMIENTOS.....	39
4.3 DISEÑO Y DESARROLLO DE LOS COMPONENTES DEL SISTEMA DE INFORMACIÓN... 39	
4.4 EJECUCIÓN PRUEBAS DE CAMPO	42
4.5 ELABORACIÓN INFORME FINAL	42
5. CRONOGRAMA.....	43
6. PRESUPUESTO	43
7. RESULTADOS.....	45
7.1 ANÁLISIS Y ESPECIFICACIÓN DE REQUERIMIENTOS	45
7.1.1 ESTUDIO DE CAMPO	45
7.1.1.1 “METALES DE SANTANDER”	47
7.1.1.1.1 AMBIENTE “METALES DE SANTANDER”	48
7.1.1.1.2 ENTREVISTA “METALES DE SANTANDER”	49
7.1.1.2 “EKOL-RECICLABLE”	50
7.1.1.2.1 AMBIENTE “EKOL-RECICLABLE”	51
7.1.1.2.2 ENTREVISTA “EKOL-RECICLABLE”	52
7.1.2 REQUERIMIENTOS DEL SISTEMA	53
7.1.2.1 REQUERIMIENTOS FUNCIONALES	54
7.1.2.1.1 LISTA CLIENTE PROVEEDOR	54
7.1.2.1.2 VISTA DE LA LISTA DE MATERIALES POR PERSONA	55
7.1.2.1.3 ACTUALIZACIÓN DE LA LISTA DE MATERIALES..... 56	
7.1.2.1.4 VALOR CUENTA..... 57	
7.1.2.1.5 IMPRESIÓN DE CUENTAS Y DATOS DEL USUARIO..... 58	
7.1.2.1.6 ACTUALIZACIÓN DE LA LISTA DE USUARIOS	59
7.1.2.1.7 DATOS POR DEFECTO..... 60	

7.1.2.1.8	MODIFICACIÓN DE DATOS DE USUARIOS.....	61
7.1.2.1.9	VISUALIZACIÓN DE CUENTAS Y SU HISTORIAL.....	62
7.1.2.1.10	MODIFICACIÓN CUENTAS A TRAVÉS DE VALOR	63
7.1.2.1.11	IDENTIFICACIÓN DE CUENTAS EN DEUDA.....	64
7.1.2.1.12	DESCONTAR CUENTAS	65
7.1.2.1.13	CÁLCULOS DE DEUDAS TOTALES	66
7.1.2.1.14	PRÉSTAMO A CLIENTES Y PROVEEDORES	67
7.1.2.1.15	CAJA MENOR	68
7.1.2.1.16	REGISTROS DE CAJA MENOR.....	69
7.1.2.1.17	AFFECTACIONES REGISTRO DE INVENTARIO.....	70
7.1.2.1.18	FILTRACIÓN DE MATERIALES	71
7.1.2.1.19	INGRESOS Y EGRESOS	72
7.1.2.2	REQUERIMIENTOS NO FUNCIONALES	73
7.1.2.2.1	CAPACIDAD DE ALMACENAMIENTO	73
7.1.2.2.2	DISPONIBILIDAD	74
7.1.2.2.3	EFICIENCIA	74
7.1.2.2.4	MANTENIMIENTO	75
7.1.2.2.5	HERRAMIENTAS DE CÓDIGO LIBRE.....	75
7.1.2.2.6	CONFIDENCIALIDAD DE LA INFORMACIÓN	76
7.1.2.2.7	ESCALABILIDAD	76
7.1.3	PRODUCT BACKLOG	77
7.1.4	GESTIÓN Y PRIORIZACIÓN DE TAREAS Y REQUERIMIENTOS	79
7.1.4.1	REVISIÓN BIBLIOGRÁFICA.....	80
7.1.4.2	RECOLECCIÓN Y REFINAMIENTO DE REQUISITOS	81
7.1.4.3	EVALUACIÓN DEL PROTOTIPO POR EL CLIENTE Y REFINAMIENTO	84
7.1.4.4	PRODUCTO DE INGENIERÍA.....	85
7.2	DISEÑO Y MODELADO DEL APLICATIVO	87

7.2.1	DIAGRAMAS	87
7.2.1.1	DIAGRAMA ENTIDAD RELACIÓN.....	87
7.2.1.2	DIAGRAMA DE CLASES.....	89
7.2.1.3	DIAGRAMA DE CASOS DE USO.....	91
7.2.1.4	DIAGRAMA DE ARQUITECTURA DESKTOP	93
7.2.1.5	DIAGRAMA DE ARQUITECTURA WEB	94
7.2.2	VISTAS DE USUARIO.....	95
7.2.2.1	LOGIN DEL SISTEMA.....	96
7.2.2.2	REGISTRO DE USUARIO.....	97
7.2.2.3	MENÚ-CUENTAS	98
7.2.2.4	NUEVA CUENTA.....	98
7.2.2.5	MENÚ-PERSONAS	99
7.2.2.6	AGREGAR PROVEEDOR.....	99
7.2.2.7	EDITAR PERSONA	100
7.2.2.8	MENÚ-INGRESOS Y EGRESOS	100
7.2.2.9	AGREGAR TIPO MOVIMIENTO	101
7.2.2.10	MENÚ-PRÉSTAMOS.....	101
7.2.2.11	REGISTRAR PRÉSTAMO.....	102
7.2.2.12	MENÚ-INVENTARIO	102
7.2.3	RECIBOS Y FACTURAS.....	103
7.2.3.1	INFORME GENERAL DE LOS MOVIMIENTOS.....	103
7.2.3.2	INFORME GENERAL DEL INVENTARIO.....	104
7.2.3.3	IMPRESIÓN DE FACTURA NÚMERO 1.....	106
7.2.3.4	IMPRESIÓN DE FACTURA NÚMERO 2.....	107
7.3	IMPLEMENTACIÓN DEL SISTEMA DE INFORMACIÓN	108
7.3.1	BUENAS PRÁCTICAS DE DESARROLLO-MODELO-VISTA-CONTROLADOR	109
7.3.1.1	MODELO.....	109



7.3.1.2 VISTA	110
7.3.1.3 CONTROLADOR	111
7.3.2 BUENAS PRÁCTICAS DE DESARROLLO-BASE DE DATOS.....	113
7.3.2.1 BUENAS PRÁCTICAS DE DESARROLLO-BASE DE DATOS MONTADA EN “DB4FREE”	114
7.3.3 COPIA DE SEGURIDAD	114
7.4 VALIDACIÓN DEL FUNCIONAMIENTO DEL SISTEMA DE INFORMACIÓN	116
7.4.1 PRUEBAS UNITARIAS	117
7.4.1.1 PRUEBA UNITARIA MODELO	117
7.4.1.2 PRUEBA UNITARIA VISTA	117
7.4.1.3 PRUEBA UNITARIA CONTROLADOR	118
7.4.1.4 PRUEBA UNITARIA CONTABILIDADMETALES	118
7.4.1.5 PRUEBA UNITARIA CLASES_FRONTEND	118
7.4.2 PRUEBAS DE ACEPTACIÓN DE USUARIO	119
7.4.2.1 PRUEBAS DE ACEPTACIÓN “METALES DE SANTANDER”	120
7.4.2.1.1 REGISTRO DE CARACTERÍSTICAS DE PRUEBAS	121
7.4.2.1.2 CUESTIONARIO.....	122
7.4.2.1.3 GRAFICO.....	123
7.4.2.1.4 CONCLUSIONES	123
7.4.2.1.5 RECOMENDACIONES	124
7.4.2.2 PRUEBAS DE ACEPTACIÓN “EKOL-RECICLABLES”	125
7.4.2.2.1 REGISTRO DE CARACTERÍSTICAS DE PRUEBAS	126
7.4.2.2.2 CUESTIONARIO.....	127
7.4.2.2.3 GRAFICO.....	128
7.4.2.2.4 CONCLUSIONES	128
7.4.2.2.5 RECOMENDACIONES	129
8. CONCLUSIONES FINALES	130
9. RECOMENDACIONES FINALES.....	131

TABLA DE FIGURAS

Fig. 1 Ejemplo de compañía recuperadora de hierro “Don Juan”	17
Fig. 2 Ejemplo de compañía recuperadora de hierro “Donde la Mona”	17
Fig. 3 Datos básicos, características equipo de cómputo	24
Fig. 4 Datos básicos, características equipo de cómputo, tarjeta grafica	24
Fig. 5 Servicio de almacenamiento [17]	25
Fig. 6 Marco de trabajo Modelado por Prototipos [21]	30
Fig. 7 NetBeans[23].....	32
Fig. 8 Java [24].....	32
Fig. 9 Diagrama cliente-servidor “DB4free” [27].....	35
Fig. 10 Logo Trello [30]	37
Fig. 11 Diagrama Modelo-Vista-Controlador.....	38
Fig. 12 Marco de trabajo Modelado por Prototipos [21]	40
Fig. 13 Entrevistas presenciales “Metales de Santander”	47
Fig. 14 Metales de Santander” (1).....	48
Fig. 15 Metales de Santander” (2).....	48
Fig. 16 Entrevista “Metales de Santander”	49
Fig. 17 Entrevistas presenciales “Ekol-Reciclable”	50
Fig. 18 Ambiente “Ekol-Reciclable” (1).....	51
Fig. 19 Ambiente “Ekol-Reciclable” (2).....	51
Fig. 20 Entrevista “Ekol-Reciclable”	52
Fig. 21 Product backlog.....	78
Fig. 22 Tablero de gestión de tareas, semana 1-4.....	80
Fig. 23 Tablero de gestión de tareas, semana 4-9.....	81
Fig. 24 Priorización de requerimientos según el Product backlog	82
Fig. 25 Numero de requerimientos.....	83
Fig. 26 Tablero de gestión de tareas, semana 9-20.....	83
Fig. 27 Tablero de gestión de tareas, semana 20-24.....	84
Fig. 28 Tablero de gestión de tareas, tareas concluidas.....	85
Fig. 29 Proyecto en GitHub	86
Fig. 30 Diagrama entidad relación	88
Fig. 31 Diagrama de clases.....	90
Fig. 32 Diagrama de casos de uso.....	92
Fig. 33 Diagrama de arquitectura Desktop.....	93
Fig. 34 Diagrama de arquitectura Web	94
Fig. 35 Software antes de aplicar experiencia de usuario (1)	95
Fig. 36 Software antes de aplicar experiencia de usuario (2)	96
Fig. 37 Login del sistema	96
Fig. 38 Registro de usuario	97
Fig. 39 Menú-Cuentas.....	98
Fig. 40 Nueva cuenta	98
Fig. 41 Menú-Personas	99

Fig. 42 Agregar proveedor	99
Fig. 43 Editar persona	100
Fig. 44 Menú-Ingresos y egresos.....	100
Fig. 45 Agregar tipo movimiento	101
Fig. 46 Menú-Préstamos	101
Fig. 47 Registrar préstamo.....	102
Fig. 48 Menú-Inventario	102
Fig. 49 Informe general de los movimientos	103
Fig. 50 Informe general del inventario.....	105
Fig. 51 Impresión de factura número 1	106
Fig. 52 Impresión de factura número 2	107
Fig. 53 Historial de reuniones por Microsoft Teams	108
Fig. 54 Buenas prácticas de desarrollo, Modelo-vista-Controlador.....	109
Fig. 55 Modelo.....	109
Fig. 56 Buenas prácticas de desarrollo, Modelo	110
Fig. 57 Vista	110
Fig. 58 Buenas prácticas de desarrollo, Vista	111
Fig. 59 Controlador.....	111
Fig. 60 Buenas prácticas de desarrollo, Controlador	112
Fig. 61 Buenas prácticas de desarrollo, Base de datos (1).....	113
Fig. 62 Buenas prácticas de desarrollo, Base de datos (2).....	113
Fig. 63 Base de datos montada en "DB4free".....	114
Fig. 64 Programador de tareas "Quartz"	115
Fig. 65 Copia de seguridad generada	115
Fig. 66 Software implementado.....	116
Fig. 67 Prueba Unitaria Modelo.....	117
Fig. 68 Prueba Unitaria Vista	117
Fig. 69 Prueba Unitaria Controlador.....	118
Fig. 70 Prueba Unitaria ContabilidadMetales	118
Fig. 71 Prueba Unitaria Clases_FrontEnd.....	118
Fig. 72 Pruebas de aceptación "Metales de Santander" (1).....	120
Fig. 73 Pruebas de aceptación "Metales de Santander" (2).....	120
Fig. 74 Grafico cuestionario "Metales de Santander"	123
Fig. 75 Pruebas de aceptación "Ekol-Reciclables" (1)	125
Fig. 76 Pruebas de aceptación "Ekol-Reciclables" (2)	125
Fig. 77 Grafico cuestionario "Ekol-Reciclables"	128



LISTA DE TABLAS

Tabla. 1 Cronograma de Ejecución de las Actividades Propuestas	43
Tabla. 2 Presupuesto Estimado	44
Tabla. 3 Lista cliente proveedor	54
Tabla. 4 Vista de la lista de materiales por persona	55
Tabla. 5 Actualización de la lista de materiales	56
Tabla. 6 Valor cuenta	57
Tabla. 7 Impresión de cuentas y datos del usuario	58
Tabla. 8 Actualización de la lista de usuarios	59
Tabla. 9 Datos por defecto	60
Tabla. 10 Modificación de datos de usuarios	61
Tabla. 11 Visualización de cuentas y su historial	62
Tabla. 12 Modificación cuentas a través de valor	63
Tabla. 13 Identificación de cuentas en deuda	64
Tabla. 14 Descontar cuentas	65
Tabla. 15 Cálculos de deudas totales	66
Tabla. 16 Préstamo a clientes y proveedores	67
Tabla. 17 Caja menor	68
Tabla. 18 Registros de caja menor	69
Tabla. 19 Afectaciones registro de inventario	70
Tabla. 20 Filtración de materiales	71
Tabla. 21 Ingresos y egresos	72
Tabla. 22 Capacidad de almacenamiento	73
Tabla. 23 Disponibilidad	74
Tabla. 24 Eficiencia	74
Tabla. 25 Mantenimiento	75
Tabla. 26 Herramientas de código libre	75
Tabla. 27 Confidencialidad de la información	76
Tabla. 28 Escalabilidad	76
Tabla. 29 Prueba de aceptación de usuario("Metales de Santander").....	121
Tabla. 30 Cuestionario "Metales de Santander"	122
Tabla. 31 Prueba de aceptación de usuario("Ekol-Reciclables")	126
Tabla. 32 Cuestionario "Ekol-Reciclables"	127



RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: Desarrollo de un sistema de información para el manejo y control de inventario para empresas recuperadoras de hierro

AUTOR(ES): Jonathan Baquero Rodríguez - Juan Sebastián Argüello lozano

FACULTAD: Ingeniería de Sistemas e Informática

DIRECTOR(A): Danith Patricia Solórzano Escobar

RESUMEN

El presente informe refleja los resultados obtenidos en un proyecto de investigación centrado en el desarrollo de un software destinado a la gestión y control del inventario en empresas dedicadas a la recuperación de hierro. Para llevar a cabo este proyecto, se recopilaron fuentes primarias, como un estudio realizado por la unidad de educación de la Universidad de Uppsala, que analizó las necesidades tecnológicas en este ámbito. También se realizaron entrevistas a empresas recuperadoras de hierro de bajos recursos, en donde se conocieron las necesidades de estas compañías y sus requerimientos más importantes para el desarrollo del software. Basándose en los hallazgos de estas investigaciones, se procedió a diseñar e implementar el sistema, utilizando como base el framework Netbeans IDE. Además, se llevaron a cabo pruebas de validación en entornos reales de trabajo, en colaboración con empresas recuperadoras de hierro. Durante el desarrollo del aplicativo, se optó por emplear un enfoque de modelado por prototipos, ya que, tras un análisis exhaustivo del estudio y del tiempo disponible, se determinó que era el marco de trabajo más apropiado para este proyecto en particular. El resultado final del trabajo de investigación es un aplicativo que ofrece una solución integral para la gestión y control del inventario en empresas recuperadoras de hierro. El uso de tecnologías y herramientas adecuadas, así como la implementación de buenas prácticas, fueron factores determinantes para garantizar un desarrollo eficiente. La amplia documentación, y recursos disponibles en el framework utilizado, contribuyeron significativamente al aprendizaje y dominio de las tecnologías utilizadas. Asimismo, el uso de herramientas como Trello y la creación de diagramas en Star UML desempeñaron un papel crucial para mantener el orden del proyecto.

PALABRAS CLAVE: Software, Framework, Recuperación, UML, Gestión, Aplicativo, Marco.

Vº Bº DIRECTOR DE TRABAJO DE GRADO



GENERAL SUMMARY OF WORK OF GRADE

TITLE: Development of an information system for inventory management and control for iron recovery companies

AUTHOR(S): Jonathan Baquero Rodríguez - Juan Sebastián Argüello lozano

FACULTY: Faculty of Systems Engineering and Informatics

DIRECTOR: Danith Patricia Solórzano Escobar

ABSTRACT

This report reflects the results obtained in a research project focused on the development of software for inventory management and control in companies dedicated to iron recovery. To carry out this project, primary sources were collected, such as a study carried out by the education unit of Uppsala University, which analyzed the technological needs in this field. Interviews were also conducted with low-income iron recovery companies, where the needs of these companies and their most important requirements for software development were known. Based on the findings of these investigations, we proceeded to design and implement the system, using the Netbeans IDE framework as a base. In addition, validation tests were carried out in real work environments, in collaboration with iron recovery companies. During the development of the application, it was decided to use a prototype modeling approach, since, after an exhaustive analysis of the study and the time available, it was determined that it was the most appropriate framework for this particular project. The final result of the research work is an application that offers a comprehensive solution for inventory management and control in iron recovery companies. The use of appropriate technologies and tools, as well as the implementation of good practices, were determining factors to guarantee efficient development. The extensive documentation, and resources available in the framework used, contributed significantly to the learning and mastery of the technologies used. Also, the use of tools like Trello and diagramming in Star UML played a crucial role in keeping the project orderly.

KEYWORDS: Software, Framework, Recovery, UML, Management, Application, Framework.

Vº Bº DIRECTOR OF GRADUATE WORK



1. RESUMEN DEL PROYECTO

1.1 Introducción

La propuesta de proyecto de grado que se presenta a continuación tiene como principal objetivo la creación de una solución de software que permita a las empresas recuperadoras de hierro tener control de su inventario y organizar los productos que manejan, a fin de sostener la obtención de los objetivos del negocio y mejorar su reconocimiento en el mercado.

La propuesta planteada para alcanzar este objetivo se basa en el diseño y desarrollo de un software que permita llevar un seguimiento minucioso de cada artículo ingresado al inventario, así como del estado en el que se encuentra en todo momento. Con esta información, se pretende lograr una gestión más eficiente del inventario, lo que se traducirá en un progreso en la toma de decisiones en cuanto a la productividad, venta y distribución de los productos.

La identificación de la situación problema y pregunta de investigación es la base para la construcción del presente sistema de información. Un idóneo planteamiento de la pregunta de investigación permite establecer con precisión el problema se va a solucionar, lo que resulta crucial para garantizar la relevancia y la viabilidad del presente estudio. Se desarrolla dicha propuesta en la sección dos de este documento.

La sección objetivo general y específicos es de gran importancia al presente documento, en esta sección se establecen las finalidades que se aspiran alcanzar con el estudio, así como los procedimientos concretos que deben ser llevados a cabo para alcanzar dicho objetivo; acto seguido vienen los antecedentes, en donde se presenta el resultado de una búsqueda sistemática en la literatura, con el fin de identificar proyectos que hayan tratado el contexto del problema. En la Justificación que como su nombre lo indica, se presenta la razón de ser del proyecto y su pertinencia, desde la respuesta a estas preguntas: ¿Por qué se hace?, ¿Con que fin se va a realizar este proyecto?

A continuación, se pasa a las secciones de Marco referencial y tecnológico, en donde Marco referencial, es quien define la base teórica necesaria para abordar el problema y en marco tecnológico, es el cual se expone la tecnología, herramientas e infraestructura aplicable durante el desarrollo del proyecto, tan para su construcción como para la gestión y documentación de este.

Para el desarrollo de la solución se plantea el marco de trabajo “Modelado por Prototipos” puesto que permite llevar el ciclo de vida del software como un proceso crucial en el desarrollo de un producto exitoso. Para describir este proceso, se construye un prototipo que va refinando y mejorando la funcionalidad del producto final. Este proceso se repite hasta que se alcanza un nivel satisfactorio de calidad y funcionalidad en el producto.

Con el marco de trabajo claro, se debe ajustar un cronograma que se acople a las actividades propuestas, para el caso del presente proyecto, se implica la realización de



varias actividades, lo que hace aún más crucial la elaboración de un cronograma detallado y preciso.

Ya para finalizar, es importante la estimación de recursos, es decir, el uso de un presupuesto, ya que permite planificar y controlar el capital necesario para llevar a cabo las actividades del proyecto de manera efectiva y eficiente, asegurando así su viabilidad financiera; Es importante cumplir con la filosofía de utilizar herramientas de uso libre para así cuidar el presupuesto y tener un bajo costo en el desarrollo de la solución.

1.2 Situación problema

La globalización ha influenciado de manera significativa a la población y a las empresas en la actualidad [1], el mundo aspira a la modernización a partir del uso de dispositivos electrónicos o digitales, sobre todo dispositivos móviles como, por ejemplo, los Smartphones (teléfonos inteligentes)[2]; también, de equipos de cómputo, los cuales, son uno de los más importantes intensificadores del cambio de la forma de interacción entre los individuos y las formas de gestionar empresas y los artículos/inventarios de estas [3].

De esta manera, los servicios de almacenamiento y digitalización de inventario surgen a raíz de la necesidad de dar respaldo y control a la información para mantener una protección de los datos y llevar a cabo un seguimiento para cada artículo; lo cual, facilita los proyectos y tareas cotidianas que realizan las empresas.

La unidad de educación de la Universidad de Uppsala, ha realizado un estudio sobre las necesidades tecnológicas para la gestión y control del inventario en empresas dedicadas a la recuperación de hierro [4]. El estudio ha identificado una necesidad que deriva una oportunidad de negocio para mejorar los indicadores de digitalización de inventario para las empresas mediante un software.

Es importante destacar que las empresas recuperadoras de hierro suelen ser de bajos recursos económicos y no cuentan con acceso a internet debido a su incapacidad financiera para contratar este tipo de servicios. [5] Al encontrarse en una situación económica limitada, estas empresas a menudo tienen que limitar su inversión en tecnología y equipamiento, y priorizar el uso de sus recursos para cubrir otros gastos operativos esenciales. Por lo tanto, la falta de acceso a internet es un factor que puede limitar significativamente su capacidad para mejorar su gestión y control de inventario, así como su facultad para rivalizar en un mercado cada vez más digitalizado.

Dicho lo anterior esta limitante tecnológica afecta su capacidad para perfeccionar la gestión y control de su inventario y, por ende, su rentabilidad y competitividad en el mercado. Por lo tanto, la solución tecnológica propuesta, busca ofrecer un software local que sea accesible y asequible para estas empresas, permitiéndoles mejorar sus procesos de gestión de inventario y operaciones diarias sin incurrir en costos excesivos de inversión o en una mayor dependencia de recursos tecnológicos externos. De esta manera, se espera que la implementación de esta solución contribuya a mejorar la eficiencia y sostenibilidad económica de estas empresas.

Para ello, y aunado a la demanda previamente citada que también concierne a sociedades y personas jurídicas, se resuelve optar por la administración asistida de recursos compartidos; persiguiendo la optimización de operaciones paralelamente a una inversión sin cuantioso valor en lo que al músculo financiero aportado por la entidad en cuestión para afrontar el proyecto respecta.



Fig. 1 Ejemplo de compañía recuperadora de hierro “Don Juan”



Fig. 2 Ejemplo de compañía recuperadora de hierro “Donde la Mona”

Dibujado tal escenario, se aborda la siguiente pregunta de investigación:

¿Cómo se puede administrar y digitalizar el inventario de las empresas recuperadores de hierro, las cuales no cuentan con un presupuesto para el contrato de recursos externos?

1.3 Justificación

Las diferentes empresas que gestionan un control de inventario, debido a varias áreas o temas que manejan, acumulan mucha información y productos en inventario como documentos, artículos o materiales, no solo en una sino en varias bodegas o espacios de almacenamiento de materiales creando desorden, sin saber el estado de cada material y pérdida de datos esenciales [15].

Poder manejar estos recursos y gestionarlos, no siempre se consigue realizar debido a varios aspectos, entre ellos el coste que significa pagar una suscripción a los distintos sitios o plataformas que existan para este fin, por tal razón para evitar estos inconvenientes se propone la creación del software presente.

Este sistema de información bajo la arquitectura cliente servidor, busca dar solución a esta problemática evitando la pérdida de información y desorganización sobre los artículos de la empresa, permitiendo la sincronización de datos en su servicio cliente servidor, autorizando el ingreso de información sobre el estado de cada artículo desde cualquier sitio y generando una copia de seguridad con el fin de mantener un respaldo de la información ya sea por fallos eléctricos, virus, desastres naturales, robos, entre otros, y que las diferentes empresas desocupen memoria en sus ordenadores de una manera sencilla y agradable.

La implementación de este sistema de información cliente servidor, implicaría un impacto positivo en el manejo y seguridad de los distintos artículos de inventario por parte de las diversas empresas, permitiendo distribuir los materiales y saber el estado en que se encuentra cada uno entre los diferentes administradores, manteniendo su respectiva copia de seguridad en el servidor del sistema de información.

Es importante aclarar que en el presente proyecto requirió y se hizo uso de los aprendizajes y conocimientos adquiridos durante la formación como ingeniero de sistemas en la Universidad Pontificia Bolivariana (UPB). En particular, se usó la experiencia obtenida en el diseño y desarrollo de software basados en la arquitectura cliente-servidor, así como en la gestión de datos complejos y en la implementación de estrategias de seguridad de la información; además, se aprovecharon las habilidades en la identificación de requerimientos de los usuarios y en la producción de soluciones personalizadas que satisfagan los requerimientos del presente proyecto; todo ello, con el fin de desarrollar un sistema de información efectivo y rentable que permita a las empresas recuperadoras de hierro optimizar su gestión de inventario y mejorar su eficiencia operativa.



2. OBJETIVOS

2.1 Objetivo general

Desarrollar un software que permita la digitalización y administración eficiente del inventario en las empresas que se dediquen a la recuperación de hierro, siguiendo el marco de trabajo de modelado por prototipos y criterios de seguridad en el manejo de la información, considerando procesos como la recepción de materiales, clasificación, y venta.

2.2 Objetivos específicos

-Identificar los requerimientos funcionales y no funcionales necesarios para el desarrollo del aplicativo a través de la utilización de técnicas de especificación como lo son historias de usuario, siguiendo buenas prácticas de diseño de software.

-Diseñar los componentes del aplicativo incluyendo el modelo de bases de datos, la arquitectura y el diseño de interfaz de usuario, mediante la elaboración de diagramas de clases y casos de uso, siguiendo criterios de calidad en ingeniería del software.

-Implementar el aplicativo propuesto siguiendo buenas prácticas de desarrollo de software y despliegue de aplicaciones, así como de documentación de aplicaciones y criterios de seguridad en el manejo de los datos de la organización.

-Realizar pruebas de campo del aplicativo en dos empresas recuperadoras de hierro para validar la funcionalidad y pertinencia de la aplicación, documentando los resultados obtenidos.

3. MARCO REFERENCIAL

Para investigar los recursos necesitados para la creación del proyecto, se realizó una investigación bibliográfica sobre los antecedentes del proyecto y los instrumentos necesarios para la elaboración del sistema de información. La base del análisis en general se basó en la situación problema la cual ya fue descrita. Las herramientas y técnicas para utilizar son: Equipo de cómputo que se encarga del alojamiento del software, como framework de desarrollo se optó por Netbeans 18, como lenguaje de programación se eligió java 18, como base de datos para el alojamiento de información MySQL 8, DB4free como servidor en la nube para alojar la base de datos en caso de que la compañía tenga acceso a internet, Star UML para la creación de diagramas, Trello para administrar la gestión del proyecto, y GitHub para el control de versiones.

3.1 Antecedentes

Si se contempla la situación problema, se podrá notar que el manejo de recursos en digitalización de inventarios en pequeñas empresas o incluso en hogares es una situación que se presenta a menudo, principalmente en compañías de escasos recursos o que recién están empezando [6], las cuales muchas veces no tienen acceso a las licencias pagas de plataformas que permitan el manejo de recursos. Normalmente estas permiten un almacenamiento gratis y gestión de inventarios, el cual muchas veces se queda algo corto para todos los artículos que una empresa debe manejar.

En el mundo hay muchas plataformas que permiten una solución a este problema, la mayoría desarrollada por grandes empresas como Odoo Inventory, Holded y Factusol, pero como mencionamos, la mayoría con licencias pagas y una versión gratuita que limita las funcionalidades ofrecidas. El uso de estas herramientas ha aumentado de forma notoria [7], una de las que más ha aumentado el servicio de estos servidores es Odoo [8], con su servicio de nube y open source. Según bistaSolutions, “Odoo ayuda a las empresas a administrar su inventario de productos o activos sin obstáculos.” [9] Ya que el software de la empresa cuenta con órdenes de entrega, portal de cliente, alertas personalizadas, o incluso un planificador inteligente.

Como se mencionó anteriormente, Odoo Inventory es uno de los principales servicios de manejo de inventario en el mundo, el cual es definido por Odoo como: “una única plataforma necesaria para manejar el negocio, la cual cuenta con aplicaciones integradas y sencillas, usadas por millones de usuarios.” [10]

A diferencia del presente software, Odoo Inventory es pago, este solo cuenta con una versión de prueba, que ya finalizado, empieza a facturar, mientras que el software del presente documento tiene una licencia totalmente gratuita ya que está diseñado para pequeñas y medianas empresas con bajo presupuesto.

Javi Fondevila y Bernat Ripoll describen a Holded como un software moderno de gestión empresarial, el cual reúne todo lo que una empresa necesita en un solo lugar: contabilidad, facturación, gestión de equipos y proyectos e inventario [11]. La parte que más resalta de esta es probablemente su apartado de facturación e inventario, ya que como su nombre lo indica, ayuda a dar un seguimiento de cada artículo/producto en el inventario, si este fue facturado, si hay stock o su estado actual, además de que cuenta con informes en tiempo real, esto permitiendo consultar cualquier artículo a las 24 horas del día. [11]

El presente proyecto dispone de un servicio similar para el caso de facturación y gestión de inventario, pudiendo así consultar un artículo o ver el stock y estado de cada uno en tiempo real. Sumado a esto, el uso gratuito del aplicativo cliente servidor ya que, a diferencia de Holded, este dispone de una prueba gratuita. Una vez pasada la prueba, el software pide una licencia de pago, cosa que las pequeñas y medianas empresas no pueden adquirir tan fácilmente.



Por último y evitando mencionar la gran cantidad de plataformas que existen, para mencionar las tres que se consideraron más importantes o por lo menos más reconocidas, hablemos un poco de Factusol, desarrollada por Delsol. Ellos la definen como: "Todos los controles de la gestión comercial de la empresa, incluyendo compras y ventas, control de inventarios y control de existencias." [12]

Básicamente, Factusol es un software de facturación para la administración de las cuentas de su compañía y el seguimiento del inventario, así como la gestión de los servicios de facturación. Esta se acopla de manera adecuada a las necesidades de los clientes y además proporciona datos de forma visual. Es decir, cuenta con un entorno gráfico especializado en mostrar esquemas o gráficas que muestren el rendimiento de la compañía o la venta de los productos. [12]

El presente proyecto, de igual forma, puede mostrar información gráfica para interpretar mejor los datos. Pero hay algo más importante que diferencia el aplicativo de los tres softwares principales mencionados, y es que todos ellos requieren una licencia de pago o tienen limitantes en su versión gratuita, mientras que el aplicativo para las empresas recuperadoras de hierro no requiere ningún costo ni licencias de pago para poder usarse. Se han implementado otros estudios, como es el caso de "International Research Journal of Modernization in Engineering Technology and Science", donde se buscó la implementación de un software hecho en inteligencia artificial(IA) para las empresas acumuladoras de hierro [13].

Este estudio dice que la cantidad de residuos es grande y no se ha hecho ningún intento por fortalecer la condición y hacer mejoras a largo plazo. Por lo cual, es importante contar con un sistema inteligente de eliminación de residuos, dado que los residuos generados superan los 2500 millones de toneladas. Por lo tanto, este estudio propone que los desechos deben clasificarse de manera básica para que en los vertederos sea posible garantizar que los residuos se eliminan correctamente. [13]

La evaluación de residuos mediante procesamiento de imágenes puede ser una herramienta muy eficiente en el proceso de gestión de residuos; sin embargo, a diferencia del presente proyecto, este tema es más abarcado a ciudades inteligentes con un presupuesto mucho mayor y no a empresas específicas recuperadoras de hierro que solo quiere digitalizar el inventario teniendo un mejor orden. Además, estas pequeñas empresas no tienen un presupuesto tan elevado como para abarcar temas de inteligencia artificial en recolección de residuos en ciudades.

Otro estudio realizado es el de "Industry 4.0 sustainable supply chains: An application of an IoT enabled scrap metal management solution". Este ya abarca el tema de automatización de los procedimientos de seguimiento y negociación de la chatarra producida. Aunque se considera que los metales se reciclan infinitamente, la gestión de la chatarra suele ser ineficiente debido a varios factores, incluidos los procesos de recolección y las limitaciones del mercado que tienen impactos ambientales y económicos significativos.

El propósito de ese estudio es proporcionar evidencia del impacto de una solución IoT en el desempeño de la gestión sustentable de la cadena de suministro. Se presentó un estudio sobre un productor de chatarra que opera en la industria de ascensores y una empresa de gestión de residuos para ilustrar cómo el despliegue de una solución de industria 4.0 de última generación tiene el potencial de mejorar la sostenibilidad tanto en



la parte de la compañía como en la cadena de suministro. La solución propuesta es beneficiosa en términos de minimización de residuos de CO₂, disponibilidad de recursos y optimización del tiempo de respuesta. [14]

Si bien este estudio también propone agilizar los procesos mediante la digitalización y mejorar los tiempos de respuesta mediante el uso de software, está más enfocado en disminuir la cantidad de CO₂ producida por estas industrias y, a su vez, mejorar el proceso de reciclaje. Según el artículo citado, la gestión de la chatarra suele ser inefficiente debido a varios factores. Además, este estudio no está enfocado en empresas de bajo presupuesto ya que, al ser un proyecto mucho más amplio abarcando más temas, incluso con el uso de dispositivos IoT, hace que el presupuesto aumente de manera considerable, lo que imposibilita la participación de empresas de escasos recursos. El presente proyecto se enfoca exclusivamente a empresas recuperadoras de hierro de bajos recursos, únicamente para digitalizar su inventario sin la necesidad de invertir muchos recursos usando herramientas de uso libre.

Si bien hay varios artículos relacionados con el tema del reciclaje o incluso el reciclado en ciudades inteligentes, hay muy pocos estudios enfocados en la digitalización abarcada por una empresa recuperadora de hierro. Básicamente, esto es un tema nuevo que muy pocas empresas o personas enfocadas en este sector han tocado. Por lo cual, con el uso de estas nuevas tecnologías en estas industrias, puede haber un mejoramiento del orden de los artefactos presentes y mejor gestión empresarial. También se pueden tener digitalizados los datos de las finanzas, usuarios y metales.

3.2 Marco conceptual

El presente apartado muestra una consulta sobre aquellos parámetros y datos a abordar en la elaboración del sistema de información. Se evidenciarán fragmentos de significado de parámetro antes nombrado, los cuales cubren las áreas a trabajar como lo son Bases de Datos (BD), desarrollo de software cliente-servidor e ingeniería de software

3.2.1 Globalización

Para comenzar el marco conceptual, es importante considerar la globalización, que implica el proceso de adaptar las aplicaciones de software para que sean funcionalmente adecuadas para su uso en múltiples países [16].

Esta globalización requiere una combinación de conocimientos de informática que no son fáciles de encontrar. Se cree que después del proceso de desarrollo es más fácil obtener software globalizado simplemente traduciendo el texto en la interfaz del cliente. Pero esta creencia está lejos de la verdad [16]. El proceso de globalización se ve obstaculizado por muchos factores complejos. Si un desarrollador cree que, durante todo el proceso de construcción, que su producto tiene que ser traducido y usado en todas partes, el producto será un resultado global.



Es importante que el software tenga cambios, no solo de idioma, si no de interfaz, imágenes, manual, entre otros, para que si sea un software globalizado y se pueda usar internacionalmente.

Basando en lo anterior y como se expuso en la situación problema, gracias a que la globalización influyó de manera significativa y se están usando cada vez más dispositivos electrónicos como computadores, haciendo el uso de inventarios digitales más importante, de ahí surge la idea y necesidad de crear este sistema de información.

3.2.2 Equipo de cómputo

Teniendo en cuenta el concepto y el desarrollo de la globalización, es importante un dispositivo de cómputo, un dispositivo electrónico que almacena y procesa información y luego se la muestra al usuario en una interfaz usable, permitiendo que el hardware (la parte táctil) interactúe con el software (la parte invisible). Suelen formar parte de estos equipos, que se denominan periféricos: teclados, ratones, monitores, adaptadores de vídeo, etc. En cuanto al proyecto actual, lo más importante es la existencia de un equipo de cómputo, ya que a través de estos dispositivos se puede representar el sistema de información.

En conclusión, el equipo de cómputo es una parte fundamental en el desarrollo de una aplicación de escritorio como la que se está desarrollando para las empresas recuperadoras de hierro. Se requiere de un equipo con ciertas especificaciones técnicas que permitan el correcto funcionamiento del software, por lo que es importante considerar la capacidad de procesamiento, memoria RAM, capacidad de almacenamiento de información y tarjeta gráfica necesaria para asegurar una excelente experiencia de usuario y un rendimiento adecuado del sistema. Además, se debe considerar la compatibilidad del equipo con el sistema operativo y otros programas que se requieran para el desarrollo y la ejecución del software.

Cabe aclarar que, para el desarrollo del software para las empresas recuperadoras de hierro, también se hace uso de un equipo de cómputo.

A continuación, se muestra las características básicas del equipo de cómputo donde se desarrolló el software:

Procesador: Intel Core i7-3770 3.40GHz (séptima generación)

RAM: 10GB

Tarjeta gráfica: NVIDIA GeForce GTX 1650

Disco duro: HDD DE 500GB

Id del dispositivo: 1A7F8B05-5984-4B5A-BF82-1E96F1CEF9F3

Nombre del dispositivo	DESKTOP-PJMDCKJ
Procesador	Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz 3.40 GHz
RAM instalada	10,0 GB
Id. del dispositivo	1A7F8B05-5984-4B5A-BF82-1E96F1CEF9F3
Id. del producto	00330-80000-00000-AA147
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada manuscrita

Fig. 3 Datos básicos, características equipo de cómputo

```

Nombre: NVIDIA GeForce GTX 1650
Fabricante: NVIDIA
Tipo de chip: NVIDIA GeForce GTX 1650
Tipo de DAC: Integrated RAMDAC
Tipo de dispositivo: Dispositivo de pantalla completa
Memoria total aprox.: 9043 MB
Memoria de pantalla
(VRAM): 3949 MB
Memoria compartida: 5094 MB

```

Fig. 4 Datos básicos, características equipo de cómputo, tarjeta grafica

3.2.3 Servicios de almacenamiento/Bases de datos

Ya teniendo claro la globalización y equipos de cómputo, es importante abarcar las bases de datos, ya que el almacenamiento, es un servicio gestionado en el que un proveedor proporciona a los usuarios un acceso a una plataforma de almacenamiento de información, dicho servicio se puede entregar en las instalaciones de una infraestructura dedicada a usuarios individuales o como un servicio compartido en una nube pública, adquirido como una suscripción. Los usuarios usan el protocolo de interfaz del sistema estándar o la interfaz de programación de aplicaciones, para tener acceso al servicio de almacenamiento de datos. Los productos típicos incluyen capacidad de almacenamiento parcial o completa; cartera de materia prima; Sistema de archivos de red; Mas opciones de almacenamiento.

Algunas de sus ventajas son:

- ✓ Espacio con mayor seguridad.
- ✓ Costos operativos con menos recursos.
- ✓ Mayor cantidad de almacenamiento.
- ✓ Entrada a los datos desde la mayoría de dispositivos electrónicos.
- ✓ Compartir archivos con facilidad.

En el caso de un software para administrar los recursos y gestionar el inventario de materiales para empresas recuperadoras de hierro, las bases de datos son esenciales para guardar y organizar la información sobre los materiales y los recursos de la empresa. Se puede utilizar bases de datos relacionales tales como MySQL para almacenar información estructurada como inventarios, compras, ventas y transacciones de materiales. Además, dado que el software es gratuito y dirigido a empresas de escasos recursos, se puede considerar el uso de una base de datos de uso libre como alternativa a las opciones comerciales; como el ejemplo antes mencionado MySQL, que es una base de datos de uso libre ampliamente utilizada y gratuita que puede ser adecuada para su caso de uso.

Es importante tener en cuenta que, sin importar la base de datos que se halla elegido, se debe tener un diseño bien estructurado y optimizado para el rendimiento y la escalabilidad, a fin de garantizar que el software funcione correctamente y sea capaz de manejar grandes volúmenes de datos.

Este servicio es el principal que proporciona el sistema de información ya que como se menciona en la situación problema, se requiere optar por la administración de inventario de varios materiales, es decir, poder gestionar el estado de cada artículo, para que su información pueda ser reflejada en pantalla, manteniendo su debida copia de seguridad.

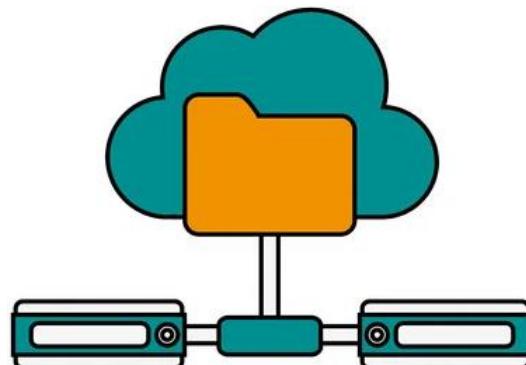


Fig. 5 Servicio de almacenamiento [17]

3.2.4 Copia de seguridad/Back-up

Abarcada la base de datos, es fundamental para la información del sistema tener su propio back-up, lo cual consiste en copias de los datos y archivos originales para evitar la pérdida de parte o toda la información del disco duro u otro dispositivo.

Las copias de seguridad generalmente se realizan en medios que no son los medios originales, como dispositivos de almacenamiento externo. De esta manera, los archivos almacenados en el equipo de cómputo, no se pueden perder o corromper en el peor de los casos. En este caso, los usuarios están más satisfechos sabiendo que sus datos e información están respaldados, previniendo fallas informáticas inesperadas o mayores por mal funcionamiento, errores técnicos o ciberataques. Es importante ejecutar un Back-up de los datos guardados cada cierto tiempo, especialmente de aquellos que se consideran muy valiosos, como textos, imágenes o videos, porque en el peor de los casos, si se pierde toda la información, puede llegar incluso a haber un punto de estrés, pánico o incluso demandas.

Para el caso del software para empresas recuperadoras de hierro, es importante tener una estrategia de copia de seguridad adecuada para garantizar que los datos importantes estén seguros y sean recuperables en caso de desastres naturales, errores humanos o fallas técnicas. La copia de seguridad es una herramienta de gran consideración en la construcción de software, especialmente para una aplicación como la de administración de recursos y gestión de inventario para empresas recuperadoras de hierro. La pérdida de datos en una empresa de este tipo podría tener un gran impacto en sus finanzas y operaciones diarias, por lo que el accionamiento de una estrategia de back-up adecuada es crucial para minimizar los riesgos.

En definitiva, la implementación de una estrategia de back-up efectiva es esencial para garantizar la seguridad y recuperación de la información importante en caso de cualquier eventualidad. Así, se puede asegurar que el software cumpla con su propósito de administrar los recursos y la gestión del inventario de las empresas recuperadoras de hierro de manera efectiva y sin interrupciones.

3.2.5 Aplicativo Desktop

Mencionado lo anterior, un pilar importante para destacar son las aplicaciones Desktop cliente servidor, ya que el software se hará en este tipo de arquitectura. A diferencia de las aplicaciones en la nube, que se ubican en otra computadora (un servidor) a la que nos conectamos a través de una red o de internet, las aplicaciones de escritorio son aquellas que se instalan en la computadora o sistema de almacenamiento (USB) y se pueden usar sin Internet en nuestro sistema operativo.

Este tipo de arquitectura Desktop cliente servidor nos ofrece algunas ventajas como, por ejemplo:

- ✓ El sistema de información puede ser más robusto
- ✓ El tiempo de espera para cada proceso es mucho más corto
- ✓ Facilita el uso de atajos por medio del teclado
- ✓ No requiere una conexión a internet para acceder a ella, como sería el caso del presente software.

La construcción del software en un aplicativo de escritorio se ha examinado como la elección más viable para el caso de las empresas recuperadoras de hierro, ya que como se dijo en el apartado de ventajas, este tipo de software tiene la ventaja de no depender de la conexión a internet para su funcionamiento; además, permite a los usuarios tener un mayor control sobre los recursos del sistema y la administración de sus datos. Otro factor relevante es la capacidad de ofrecer una interfaz de usuario intuitiva y personalizable, lo que puede mejorar la eficacia y rendimiento de los clientes en su trabajo diario. Por último, el aplicativo de escritorio se ajusta perfectamente a las necesidades de las empresas recuperadoras de hierro, que suelen tener equipos informáticos con especificaciones modestas.

Sin embargo, también es importante tener en cuenta que el software debe ser actualizado y mantenerse al día con las versiones más actuales de los sistemas operativos y así garantizar su funcionamiento óptimo. Además, el transcurso de instalación y actualizaciones del sistema de información debe ser sencillo y accesible para los usuarios.

Es importante que la compañía tenga un equipo principalmente asignado para el sistema de información, ya que como se mencionó, por medio de este es que el software se aloja y se ejecuta.

3.2.6 Framework de desarrollo

A medida que se analizan las aplicaciones cliente servidor, es necesario aclarar cómo se desarrollaron estas aplicaciones o qué se utilizó para crearlas.

En resumen, el Framework de desarrollo, es una estructura o software, que se puede ser usado para diseñar un proyecto planeado, en este caso el sistema de información actual. Estos son Frameworks diseñados para desarrollar proyectos en línea (web) o de escritorio(Desktop). Esto incluye desde el diseño de páginas web y aplicaciones de escritorio simples, hasta aplicativos webs más robustos y específicos tales como Facebook o Instagram.

Dependiendo del lenguaje de programación usado, hay varios tipos de este sistema. Sin embargo, no hay nada que impida que un Framework diseñado originalmente para cierto lenguaje de programación, se use en otro. Para un buen desarrollador de software, en muchos casos es mucho más sencillo adaptar un entorno de programación a otro, que cambiar completamente el proyecto para otros fines.

Otros usos que un Framework puede tener son:

1. Para el desarrollo aplicativos generales

Permite que el sistema operativo agregue la estructura del aplicativo.

Por ejemplo, la compañía Microsoft desarrolló “NET Framework”, que ya está instalado en los sistemas Windows, para ayudar a los programadores a reutilizar un sistema ya bien desarrollado. [18]

2. Para un estándar de programación organizado

Los frameworks de desarrollo también permiten llevar a cabo un esquema de programación ordenado y estructurado, como el caso de java, que se maneja en programación orientada a objetos, además de tener cada una de las clases ordenadas, incluso también ayudan a depurar y detectar errores en el código, cosa que con un bloc de notas o editor de texto común no se podría [19].

El uso de un framework de desarrollo en el proyecto de software para empresas recuperadoras de hierro permite aprovechar una base sólida y probada en lugar de comenzar desde cero. Esto no solo puede acelerar el proceso de desarrollo, sino que también puede garantizar que el software sea más robusto, escalable y fácil de mantener. Los frameworks de desarrollo también proporcionan una estructura organizada y coherente para el presente proyecto, lo que facilita la colaboración entre desarrolladores



y la identificación de errores. En general, el uso de un framework puede ayudar a crear un software de mayor calidad en menos tiempo.

Aclarado lo anterior, el framework para el presente proyecto, se uso como herramienta de desarrollo para construir el aplicativo cliente servidor.

3.2.7 Marco de trabajo

El sistema de información se desarrollará utilizando el marco de trabajo “Modelado por Prototipos”, pero primero se debe comprender cómo funciona dicho método.

El modelo de prototipos se basa en el diseño rápido, el cual está centrado en una imagen de las propiedades del software que serán visibles para el usuario. El prototipo debe construirse rápidamente usando los programas apropiados con un uso mínimo de recursos. Dicho diseño lleva a la creación de un prototipo temprano, o innovación, que es valorado por el usuario final para un feedback; como resultado, se refinan las necesidades del sistema de información que se desarrollará. La interacción tiene lugar cuando el prototipo se acopla para complacer las necesidades del usuario. Gracias a esto, el desarrollador puede comprender lo que se debe hacer y, al mismo tiempo, permite que el cliente vea los resultados rápidamente. [20]

El modelo de prototipos sigue una serie de pasos:

- 1) *Recolección y refinamiento de requisitos:* En esta sección es donde se realizan y afinan los requisitos del software según las necesidades del cliente o empresa
- 2) *Diseño rápido:* En cuanto al diseño rápido, se refiere a crear y diseñar una interfaz de usuario que sea eficaz y de óptimo entendimiento para el cliente final; Una buena interfaz de usuario debe ser fácil de usar, intuitiva y visualmente atractiva, para aumentar la experiencia del usuario y lograr que el producto sea mucho más accesible y efectivo.
- 3) *Construcción del prototipo:* Durante esta fase, se utilizan las especificaciones y requisitos recopilados de las fases anteriores para crear un prototipo funcional y tangible que pueda ser evaluado y refinado antes de la producción final. Se utilizan herramientas de desarrollo rápidas y flexibles para construir el prototipo.
- 4) *Evaluación del prototipo por el cliente:* Es cuando se presenta el prototipo construido al usuario final para que lo pruebe y brinde retroalimentación. Durante esta fase, el cliente evalúa el prototipo en función de los requisitos y especificaciones establecidos en las fases anteriores y proporciona comentarios para mejorar el diseño o la funcionalidad. La retroalimentación proporcionada en esta fase es esencial para refining el diseño y garantizar que el software cubra las expectativas del usuario.

- 5) *Refinamiento del prototipo*: Se realiza una revisión detallada del prototipo construido y se hacen mejoras o cambios necesarios. Durante esta fase, se utilizan los comentarios y sugerencias obtenidos de la evaluación del cliente para realizar mejoras en el diseño o en la funcionalidad del prototipo.
- 6) *Producto de ingeniería*: En esta fase ya es cuando se crea el producto final a partir del prototipo refinado. Durante esta fase, se utiliza el prototipo refinado como modelo para crear el producto final. Una vez que se ha completado el desarrollo del software, se puede lanzar al mercado y se puede continuar el monitoreo y evaluación del rendimiento del producto en el mercado. [20]

El marco de trabajo de "Modelado por Prototipos" es una excelente elección para el desarrollo de un software para las empresas recuperadoras de hierro, ya que permite una iteración continua del diseño y la funcionalidad, lo que asegura que el sistema de información satisfaga los requerimientos de los clientes. Al proporcionar el software de forma gratuita a empresas con recursos limitados, el marco de trabajo de "Modelado por Prototipos" puede ser una ayuda para acortar el valor de desarrollo y evitar problemas de implementación. En resumen, es una opción muy adecuada para la construcción de este tipo de aplicaciones, ya que garantiza que el sistema de información sea fácil de manipular, mantener en el futuro, y que cumpla con las necesidades específicas de las empresas recuperadoras de hierro.

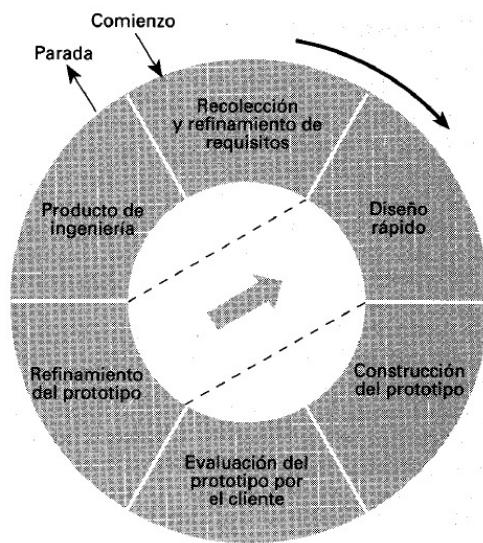


Fig. 6 Marco de trabajo Modelado por Prototipos [21]

Ya aclarados los conceptos, se brinda una base firme para la construcción de un software que ayude a la administración de los procesos de recuperación de hierro en las empresas del sector. Los conceptos y teorías expuestos en este marco permiten una comprensión profunda de los desafíos y oportunidades que enfrenta este sector y, por tanto, una mejor identificación de las necesidades específicas que el software debe abordar. Además, la elección del marco de trabajo de "Modelado por Prototipos" para el desarrollo del



software, garantiza una iteración continua del diseño y la funcionalidad, lo que permitiría una mejor adaptabilidad en las necesidades del usuario final.

3.3 Marco Tecnológico

Tras haber analizado los diferentes conceptos que se manejaran al cabo del desarrollo del proyecto, es importante aclarar las tecnologías se implementaran y usaran para este sistema distribuido. En el marco conceptual se abarcaron conceptos tales como Framework, Bases de datos, desarrollo de aplicaciones, etc. A continuación, se presentan y explican las tecnologías que se ocuparan para el desarrollo del sistema de información.

3.3.1 Netbeans

En el presente proyecto se utilizará NetBeans como un entorno de desarrollo integrado para la construcción del presente software de gestión de inventario para empresas recuperadoras de hierro.

Para empezar, Netbeans será el framework principal de desarrollo, lo que quiere decir que es la herramienta principal de este proyecto, además Netbeans permite el desarrollo de aplicaciones basadas en una colección de partes de software conocidos como módulos [22]. Un módulo es un fichero de Java que abarca archivos especialmente designados (archivos de manifiesto) que lo identifican como un módulo y clases de Java descritas a comunicarse con la API de NetBeans[22].

Hay que recalcar que Netbeans es completamente gratuito, por lo cual no será un costo adicional para el proyecto, además para el desarrollo del software, se tendrán en cuenta métodos, clases y una sintaxis adecuada para evitar tener el sistema desorganizado en cuanto a código.

Otra de las razones fundamentales por las que se optó por usar Netbeans es que encaja en línea con la filosofía del proyecto de utilizar herramientas de uso libre. Los miembros del proyecto están comprometidos a desarrollar un software de calidad que sea accesible y económico para las empresas recuperadoras de hierro, y utilizar herramientas de uso libre es una forma de cumplir con esta misión.

Hay que aclarar que, en términos del proyecto, el uso de Netbeans no está directamente relacionado con una parte específica del patrón Modelo-Vista-Controlador (MVC). En cambio, se utiliza como herramienta de desarrollo en general para implementar y trabajar en el código de todo el proyecto, incluyendo las tres partes del patrón MVC. Por lo tanto, Netbeans se podría considerar como una herramienta que ayuda en la implementación del patrón MVC en el proyecto de software.

En conclusión, la elección de Netbeans como herramienta de desarrollo de software tiene múltiples beneficios para el presente proyecto de tesis. Es una herramienta potente, de



fácil uso, comprensión, y más importante, de uso libre, lo que permite cumplir con el objetivo de desarrollar un software de calidad a un costo accesible para las empresas recuperadoras de hierro.



Fig. 7 NetBeans[23]

3.3.2 Java

Teniendo claro que se maneja NetBeans, hay que basarse en el lenguaje de programación que este maneja, el cual es Java. Este se basa en la programación orientada a objetos (POO), la cual permite una ejecución de un único programa en varios sistemas operativos que además permite ejecutar código de forma segura en sistemas remotos [24]. Su rango de uso es tan amplio que se utiliza tanto en dispositivos móviles como en electrodomésticos, además de que el framework antes mencionado (NetBeans) trabaja sobre este lenguaje, algunas características de Java son:

4. De alto nivel
5. Portable
6. Robusto
7. Orientado a objetos
8. Distribuido

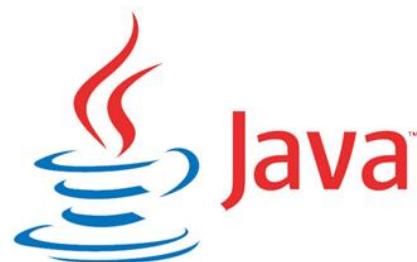


Fig. 8 Java [24]

Una de las primordiales razones por las que opto Java para el desarrollo del software, es que Java es un lenguaje de programación orientado a objetos muy flexible y poderoso que brinda una inmensa gama de propiedades y funcionalidades [24]. Además, al utilizar



Java, se puede garantizar la compatibilidad con diferentes sistemas operativos y se puede tener una mayor facilidad para mantener el software a largo plazo, lo que lo hace una elección sólida para el desarrollo de esta aplicación. [24]. Otra ventaja importante de Java es su amplia gama de bibliotecas y frameworks disponibles que pueden ayudar a simplificar y acelerar el transcurso de desarrollo del sistema de información para las compañías recuperadoras de hierro.

Además, Java es un lenguaje de código libre, lo que se ajusta a la filosofía del proyecto de utilizar herramientas de código libre y evitar altos costos de licencia. La integración con Netbeans, también hace que Java sea una opción lógica para el proyecto.

En cuanto el modelo-vista controlador, Java se ubica en la capa del Controlador, ya que se encargará de administrar las comunicaciones entre la Vista y el Modelo, permitiendo el intercambio de datos entre dichas capas. La elección de Java como el lenguaje de programación para el Controlador se debe gracias su capacidad para manejar la lógica de la aplicación, procesar los datos recibidos desde la Vista y actualizar el Modelo en consecuencia; ya en la parte de vista, se usará Java Swing ya que esta herramienta es una biblioteca de clases que hace posible crear gráficos de interfaz de usuario en Java; dicha herramienta se utilizó para diseñar la interfaz de usuario para la aplicación de gestión de inventario. Esto incluye la creación de formularios de entrada de información, y para la visualización de información. Al separar la capa de vista de la lógica de negocio, se logra un mayor modularidad en el diseño de la aplicación y una mejor mantenibilidad, lo que es esencial para cumplir con los objetivos del proyecto.

En resumen, el uso de Java es una decisión estratégica que permite a los desarrolladores del presente proyecto de tesis, crear un software robusto y escalable para cubrir las necesidades de las compañías recuperadoras de hierro, al tiempo que mantiene la filosofía del proyecto de utilizar herramientas de código libre y mantener un bajo presupuesto.

3.3.3 MySql

Es importante utilizar una base de datos para guardar las credenciales de los administradores e información de los materiales en el inventario. En este caso, se ofrece Mysql de Oracle, que se considera como uno de los mejores sistemas administradores de bases de datos [25]. Hoy en día, es importante optimizar y administrar de manera inteligente las bases de datos MySQL que se integran en aplicaciones web o Desktop como el presente proyecto. El estado de estos factores dependerá de desastres como ataques de hackers y la necesidad de proteger datos, aplicaciones o fallas en los servidores que albergan los proyectos [25]. Entender los diferentes gestores de información es muy importante en cualquier sistema de bases de datos. Por lo tanto, es de suma importancia elegir el producto adecuado.



Se decide utilizar MySQL como administrador de base de datos para el software, debido a su capacidad para manejar grandes cantidades de información, su escalabilidad y su amplia disponibilidad en el mercado. Además, MySQL es una opción de código abierto y gratuita, lo que se alinea con la filosofía del presente proyecto de proporcionar una herramienta de gestión de recursos accesible y gratuita para las empresas de escasos recursos [25].

Como se ha mencionado anteriormente, muchas de estas compañías no tienen acceso a internet o tienen una conexión inestable. Para solucionar esto, la base de datos se instala de manera local para facilitar la conexión del software, a continuación, se proporcionan los pasos a seguir para la montar la base de datos de manera local:

1. Se instala el servidor de base de datos MySQL en cada máquina donde se desea implementar el software contable. Esto permitirá crear una base de datos local para el almacenamiento de la información.
2. Una vez instalado MySQL, se importa la base de datos específica para el software contable.
3. Se debe descargar el controlador JDBC de MySQL. Este controlador es esencial para establecer la conexión entre el software contable y la base de datos MySQL.
4. Se deben conectar los equipos al servidor donde está alojada la base de datos
5. Ejecutar consultas y procesar resultados, una vez establecida la conexión, se pueden ejecutar consultas SQL de prueba.

Ya instalado, para conectar el software contable a otra máquina dentro de la misma red, solo se necesita modificar el valor de "url" en el código Java, reemplazando el nombre o la dirección IP del servidor MySQL por el de la máquina en la que se ha instalado MySQL y la base de datos de la empresa recuperadora correspondiente.

Estos pasos permiten una instalación local y una conexión efectiva del software contable a la base de datos MySQL, facilitando así la gestión y control del inventario en las empresas recuperadoras de hierro sin acceso a internet.

En cuanto al modelo-vista-controlador Mysql se encuentra en el modelo. Esto se debe a que esta parte se encarga de manejar los datos y la lógica del negocio, y la base de datos es una parte esencial en la gestión de información del sistema. Mysql se utiliza como el gestor de base de datos del proyecto debido a que es una herramienta open-source, lo que se alinea con la filosofía del proyecto de utilizar herramientas de software libre para evitar un alto presupuesto. Además, Mysql es una opción popular y confiable para manejar grandes cantidades de datos y se integra fácilmente con el lenguaje de programación Java, el cual se utiliza para el desarrollo del software en Netbeans.

Por eso mismo el equipo del sistema de información optó por el uso de Mysql como gestor de base de datos ya que se consideró la más adecuada para el proyecto por su seguridad, eficacia, automatización de back-up para el respaldo de información subida por el usuario y además que todo el equipo conoce el uso de esta herramienta.

3.3.4 DB4free

Como se ha mencionado anteriormente, el presente proyecto está enfocado en empresas recuperadoras de hierro de bajos recursos, las cuales en su mayoría no tienen acceso a internet de manera constante. Sin embargo, también se reconoce que existen compañías con acceso a Internet. Para estas empresas, se ha implementado una solución alternativa. Se sube la base de datos a la plataforma de hosting en la nube “DB4free”, la cual es un espacio virtual que permite a los desarrolladores implementar, administrar y escalar la base de datos de manera sencilla [26], lo que permitirá que estas empresas que cuentan con un servicio de internet estable accedan a la base de datos de forma remota sin necesidad de almacenarla localmente. Esta solución proporcionará flexibilidad y conveniencia para las empresas que pueden aprovechar los beneficios de la infraestructura en la nube, además su uso es completamente gratuito cumpliendo con el objetivo de usar tecnologías de uso libre para tener un presupuesto mínimo.

La elección de “DB4free” para las compañías con un servicio de internet estable, brinda una serie de beneficios clave. En primer lugar, proporciona una solución de implementación en la nube altamente eficiente y confiable [26]. Al aprovechar su infraestructura escalable, se puede desplegar y gestionar la base de datos del presente software de manera sencilla, sin preocupaciones por la configuración de servidores o la administración de recursos. Otro aspecto destacado de “DB4free” es su enfoque en la seguridad y la gestión de la infraestructura. La plataforma brinda medidas de seguridad avanzadas, como la protección de datos [26], lo que permite salvaguardar la información sensible de los clientes y garantizar la confidencialidad de la información.

En resumen, al elegir “DB4free” para el presente proyecto, se obtiene una solución integral que simplifica y agiliza el proceso de implementación, permite escalar de manera eficiente y brinda la tranquilidad de contar con una infraestructura segura y confiable.

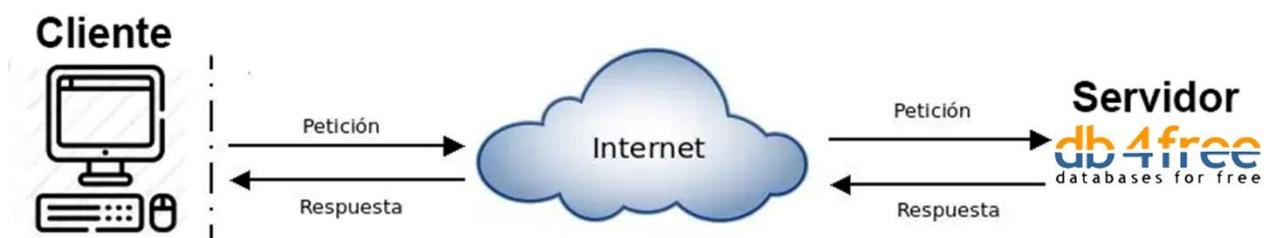


Fig. 9 Diagrama cliente-servidor “DB4free” [27]

3.3.5 Star UML

StarUML es una herramienta de modelado para software que se basa en los estándares UML (Unified Modeling Language). Originalmente era un producto comercial que evolucionó de un proyecto comercial a una licencia de código abierto bajo GNU/GPL [28].

El programa ha conservado todas las funciones de la versión comercial y ha mejorado gradualmente estas funciones, que incluyen:

- ✓ Diagrama de estados
- ✓ Diagrama de colaboración.
- ✓ Diagrama de clase
- ✓ Diagrama de actividades.
- ✓ Diagrama de secuencia
- ✓ Diagrama de casos de uso. [28]

Se optó Star UML como herramienta de modelado del presente proyecto debido a que permite representar gráficamente la estructura y el comportamiento del software de una manera clara y sencilla, lo que puede facilitar la comprensión y comunicaciones entre los participantes del equipo de desarrollo y los clientes. Además, Star UML ofrece una gran gama de propiedades e instrumentos para el modelado de software, lo que permite crear modelos detallados que ayudan en el proceso de desarrollo [28]. En general, el uso de Star UML mejora significativamente la eficiencia y la precisión en el desarrollo del software.

En resumen, este es uno de los mejores softwares de modelado en Internet [28], y en el caso del presente sistema de información, es una columna importante para los diversos diseños de gráficos mencionados anteriormente.

3.3.6 Trello

Ya con las herramientas anteriores definidas, se puede hacer la base de datos, diagramas y programación; pero queda faltando una herramienta para poder llevar un control y priorización de las tareas y/o requerimientos que se estén llevando a cabo.

En este sentido, se seleccionó la herramienta Trello como herramienta para poder tener un control sobre las tareas y requerimientos a programar del software. Trello proporciona un sistema de gestión de proyectos visual y colaborativo que permite organizar y monitorear las actividades relacionadas con el desarrollo del software para las empresas recuperadoras de hierro. Con Trello, se crearon tableros que representan diferentes etapas del proyecto, como "por hacer", "en progreso" y "completado", y asignar tareas a los miembros del equipo, establecer fechas límite y establecer la prioridad de los requerimientos [29]. Esta herramienta facilita la comunicación y el seguimiento de las

actividades, asegurando que todas las tareas se realizaran de manera oportuna y eficiente, permitiendo una gestión efectiva de los requerimientos del proyecto.

En resumen, la adopción de Trello en el presente proyecto se justifica por su capacidad para facilitar la gestión y organización de tareas, así como la priorización de los requerimientos [29]. Con Trello, se cuenta con una herramienta sólida y accesible para gestionar de manera efectiva el flujo de trabajo y lograr el éxito en la implementación del software para las empresas recuperadoras de hierro.



Fig. 10 Logo Trello [30]

3.3.7 GitHub

Aunque Trello ayuda a planificar las tareas del proyecto y organizar la ejecución de cada uno de los requerimientos, es necesaria una herramienta para ayudar a gestionar el versionado de software, es aquí donde entra GitHub ya que proporciona un control de versiones efectivo [31], lo que permite un seguimiento preciso de todas las modificaciones realizadas en el software. Esto facilita la colaboración entre los miembros del equipo, ya que se puede trabajar en ramas separadas y fusionar los cambios de manera eficiente.

El uso de GitHub proporciona numerosos beneficios en el presente proyecto. Se puede crear un repositorio centralizado donde almacenar y organizar todo el código del proyecto [31], lo que facilita el trabajo en equipo y la colaboración entre los desarrolladores del software para las empresas recuperadoras de hierro. Otra ventaja de GitHub es su capacidad para manejar problemas y solicitudes de cambios de manera efectiva, lo que permite mantener un seguimiento de las tareas pendientes, discutir ideas y realizar contribuciones al proyecto de manera ordenada [31], además se puede tener un link disponible para que se pueda descargar el proyecto en cualquier momento. En resumen, utilizar GitHub ayuda a mantener una gestión eficiente del código, promover la colaboración entre el equipo de desarrollo y garantizar la calidad y el avance del proyecto de manera estructurada.

Ya teniendo el concepto del ¿por qué?, ¿cómo? y ¿para qué? se usarán las herramientas propiamente dichas. Se puede aclarar que el marco tecnológico de este proyecto de tesis se enfoca en el uso de herramientas de código libre y de fácil acceso, con la meta de reducir los precios y reforzar la eficacia del desarrollo del software para empresas recuperadoras de hierro. Se seleccionó Netbeans como el ambiente de desarrollo, Java como lenguaje de programación, MySQL como el administrador de base de datos, star UML como editor para los diagramas del proyecto, Trello como herramienta para llevar un control y priorización de las tareas y/o requerimientos y GitHub para mantener un control de versiones y tener el proyecto disponible para su descarga

La arquitectura del software sigue el patrón Modelo-Vista-Controlador, donde Java Swing se encarga de la implementación de la Vista, Java se utiliza como el Controlador y MySQL es el encargado de manejar el Modelo. La implementación de esta arquitectura permite una separación clara de responsabilidades y un mayor modularidad del software, lo que facilita su mantenimiento y escalabilidad en el futuro. A continuación, se presentará el diagrama del Modelo-Vista-Controlador que se utilizará en la construcción del software.

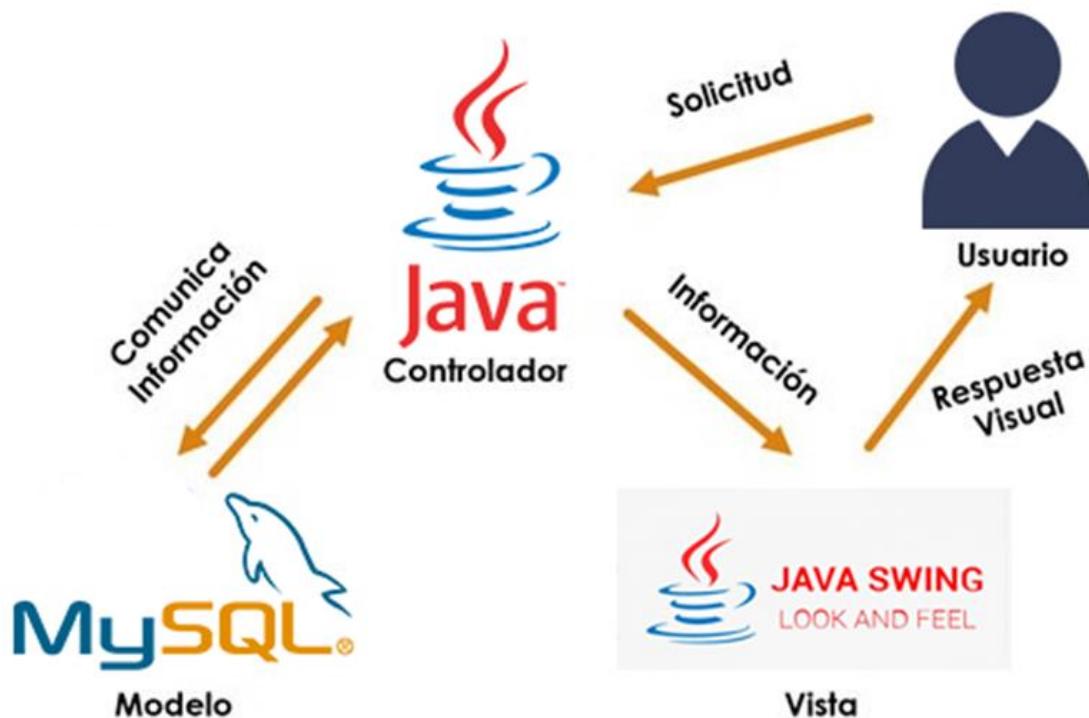


Fig. 11 Diagrama Modelo-Vista-Controlador



4. METODOLOGÍA

4.1 Revisión bibliográfica

Se ejecuta una inspección bibliográfica de artículos científicos, páginas web oficiales y otras fuentes regionales afines al tema de interés que se desarrolla en el respectivo documento.

4.2 Definición de requerimientos

Al empezar con el desarrollo del software, es importante la definición de cada uno de los requerimientos de este, ya que estos son una descripción del servicio que ofrece el sistema de información.

Cada uno de los requisitos ayudan a abarcar desde la declaración abstracta de alto nivel de los requerimientos del cliente final, hasta las especificaciones más detalladas para lograr comprender el comportamiento previsto del sistema de información.

Para la definición de los requisitos se incluye la siguiente información:

- Particularidades de las operaciones realizadas en cada pantalla
- La metodología de gestión de datos que se introduce en el sistema.
- Debe quedar claro quién puede agregar, modificar y eliminar datos del sistema de información

En cuanto a actualizaciones o mantenimiento que se le tenga que hacer al sistema de información, en caso de añadir una nueva funcionalidad se crea otro requerimiento, describiendo la actividad que se va a realizar, con el fin de tener todo el software con una documentación completa.

4.3 Diseño y desarrollo de los componentes del sistema de información

Ya definidos todos los requerimientos es cuando se comienzan a diseñar todos los componentes del sistema, desde el aplicativo cliente servidor, hasta el sistema administrador de base de datos.

Ante el presente proyecto, tal como se mencionó en el marco conceptual, el marco de trabajo que se llevará a efecto es “Modelado por Prototipos”, la cual requiere que los participantes del proyecto se adhieran a diferentes métodos de trabajo para lograr una mejora progresiva en el desarrollo del software. Cabe aclarar que este es un marco de trabajo para gestionar proyectos, la cual se divide en varias fases. Cada uno de estos ciclos comienzan poco después de que la fase anterior haya terminado. En este artículo se explican los pasos de la administración de proyectos y lo útil que puede ser para que el equipo del presente proyecto alcance sus objetivos.



Este marco de trabajo ayuda a cumplir los objetivos de diseño de los elementos del software, accionamiento de servicios y validación del sistema de información.

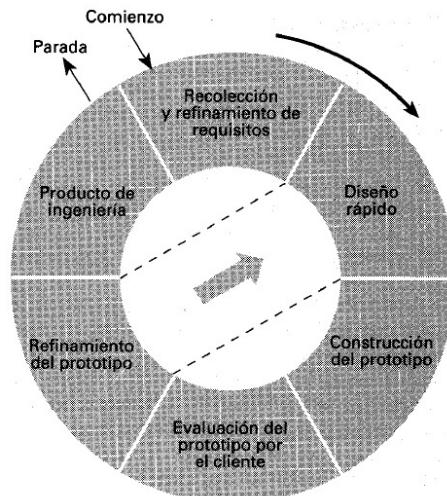


Fig. 12 Marco de trabajo Modelado por Prototipos [21]

Cada ciclo termina con la creación de una parte del producto para los clientes.

Fases:

1. *Recolección y refinamiento de requisitos*: La primera fase es importante porque sienta las bases para el sistema que se desarrolla. Sin entrar en los aspectos técnicos, en la documentación se describen detalladamente los requisitos marcados por el cliente que son el resultado final del software para las empresas recuperadoras de hierro. En general, la fase de Recolección y Refinamiento de requisitos ayuda a asegurar que el software se acople a las expectativas de las compañías recuperadoras de hierro, lo que aumenta las posibilidades de éxito del proyecto.
2. *Diseño rápido*: En la fase de Diseño Rápido, se realiza una aproximación rápida a la definición de la interfaz y funcionalidad del software. Esta fase es importante en el presente proyecto porque permite obtener rápidamente una idea visual de cómo es la interfaz del software, y permite identificar de manera temprana problemas o limitaciones en el diseño. De esta manera, se pueden realizar mejoras en el diseño antes de continuar con el desarrollo del proyecto, lo que ahorra tiempo y recursos. Además, esta fase puede servir como un instrumento de comunicación con los clientes o usuarios finales del software, que vendrían siendo las empresas recuperadoras de hierro, quienes pueden proporcionar comentarios y sugerencias sobre el diseño del software antes de su implementación.

- 
3. *Construcción del prototipo*: La fase de Construcción del prototipo es crucial en el marco de trabajo de “Modelado por Prototipos”, porque es aquí donde se construye y se desarrolla el software propiamente dicho, utilizando los requisitos y el diseño que se establecieron en las fases anteriores. Esta fase implica la creación de un prototipo funcional que pueda ser probado y evaluado por las empresas recuperadoras de hierro en la siguiente fase de “evaluación del prototipo por el cliente”.
 4. *Evaluación del prototipo por el cliente*: Ya con el software desarrollado, esta fase permite obtener retroalimentación sobre el software en desarrollo, directamente de las empresas recuperadoras de hierro. Es importante involucrar a los clientes en esta fase para que puedan proporcionar comentarios valiosos sobre el prototipo y sugerir mejoras que se deben implementar en el siguiente ciclo de desarrollo. De esta manera, se puede mejorar el software para cumplir los requerimientos de las compañías recuperadoras de hierro y aumentar la probabilidad de que el producto final sea aceptado y utilizado de manera efectiva. La evaluación del prototipo por los clientes también permite identificar problemas tempranos y corregirlos antes de la finalización del software, lo que puede reducir costos y retrasos en el desarrollo del proyecto.
 5. *Refinamiento del prototipo*: Ya en esta fase, se permite mejorar y optimizar el software a partir de los comentarios y sugerencias por los clientes. En este apartado, se determinan los inconvenientes que puedan surgir en la interfaz y se realizan ajustes para mejorar la usabilidad y la funcionalidad del software. Además, se lleva a cabo una revisión exhaustiva del código para identificar y corregir errores o fallas. Esto asegura que el sistema de información final satisfaga los requisitos y expectativas de las empresas recuperadoras de hierro, lo que aumenta las posibilidades de éxito del proyecto. En resumen, la fase de Refinamiento del prototipo es fundamental para la creación de un software eficiente, confiable y a la par con los requerimientos del usuario.
 6. *Producto de ingeniería*: Ya en la última etapa se realiza la construcción de la versión final del software, basándose en el prototipo seleccionado previamente. Durante esta fase ya es cuando se implementa el software en un ambiente real de trabajo. También se lleva a cabo la documentación del software, lo que es esencial para futuras modificaciones y mantenimiento.

En el caso específico del presente proyecto de software para empresas recuperadoras de hierro, la fase de Producto de ingeniería es crucial para crear una herramienta que sea fácil de usar, eficiente en el manejo de inventarios y que se ajuste a las necesidades de las empresas. La documentación y el mantenimiento del software también son importantes para asegurar su uso continuo y efectivo. En resumen, la fase de Producto de ingeniería es esencial para transformar el prototipo en un producto final de alta calidad y funcionalidad.



Ya explicadas las fases del marco de trabajo hay que tener en cuenta que al momento de desarrollar cada uno de los requerimientos del sistema, se hacen una serie de iteraciones como se mostraron en la figura 11; ya que los requisitos se van desarrollado según como son priorizados en el product backlog donde cada vez que se termine un requerimiento, el equipo de desarrollo se devuelve a la fase de diseño rápido hasta llegar nuevamente al apartado de refinamiento del prototipo, y así para desarrollar cada requerimiento planteado.

Una vez finalizado el marco de trabajo del Modelado por Prototipos, se cuenta con un software funcional que cumple con los requisitos y expectativas del cliente. Además, se logra una mejor comprensión de los requisitos y necesidades del cliente a lo largo del proceso de desarrollo, lo que permite una mayor eficiencia en el desarrollo de futuros proyectos similares. También se identifican y resuelven posibles problemas y errores durante el proceso de desarrollo, lo que aumentará la calidad y fiabilidad del software. En resumen, se logra un software eficiente y de calidad que contribuye a la gestión de inventarios de las empresas recuperadoras de hierro y mejorar su rendimiento.

4.4 Ejecución pruebas de campo

Ya con el software desarrollado, se procede a validar la funcionalidad y pertinencia del aplicativo desarrollado mediante la realización de pruebas de campo en dos empresas recuperadoras de hierro.

Para lograrlo, se planea una serie de pasos, que incluyen la instalación y configuración del software en los sistemas de las empresas y la extracción de datos sobre el desempeño de la herramienta mediante entrevistas al personal de la empresa.

4.5 Elaboración informe final

Una vez desarrolladas las actividades planteadas, se elabora el informe final con los resultados obtenidos, el análisis de estos, y conclusiones.

5. CRONOGRAMA

El tiempo estimado para ejecutar a cabalidad el marco de trabajo propuesto es de 6 meses.

Actividades	Mes 1				Mes 2			
	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8
Revisión Bibliográfica								
Definición de requerimientos								

Actividades	Mes 3				Mes 4				Mes 5			
	Semana 9	Semana 10	Semana 11	Semana 12	Semana 13	Semana 14	Semana 15	Semana 16	Semana 17	Semana 18	Semana 19	Semana 20
Diseño y desarrollo de los componentes del sistema de información												
Revisión Bibliográfica												

Actividades	Mes 6			
	Semana 21	Semana 22	Semana 23	Semana 24
Ejecución pruebas de campo				
Revisión Bibliográfica				
Elaboración informe final				

Tabla. 1 Cronograma de Ejecución de las Actividades Propuestas

6. PRESUPUESTO

Se estableció un presupuesto bajo que se alinea con la idea de maximizar el valor obtenido de cada inversión realizada. Para lograr esto, se ha priorizado el uso de herramientas y tecnologías de código abierto, que no requieren costosas licencias o



adquisiciones de software propietario. Asimismo, se ha fomentado la reutilización de recursos y la búsqueda de alternativas de bajo costo, evitando gastos innecesarios y optimizando el rendimiento de cada inversión.

PRESUPUESTO PROYECTO DE GRADO				
UNIVERSIDAD PONTIFICIA BOLIVARIANA - SECCIONAL BUCARAMANGA				
TÍTULO DEL PROYECTO DE GRADO	DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL MANEJO Y CONTROL DE INVENTARIO PARA LAS EMPRESAS RECUPERADORAS DE HIERRO			
	NOMBRE	HORAS/ SEMANA	HORAS/MES	MESES
DIRECTOR DEL PROYECTO DE GRADO	Danith Patricia Solórzano Escobar	1	4	6
DURACIÓN DEL PROYECTO DE GRADO	6 meses			
INVERSIONES				
DESCRIPCIÓN	COSTO TOTAL (COP)			
EQUIPO DE COMPUTO	\$ 2.500.000			
MATERIALES	\$ 0			
LIBROS (Base de datos x año)	\$ 50.000			
TOTAL INVERSIONES	\$ 2.550.000			
GASTOS DE PERSONAL				
SUELDO (Por mes)	\$ 1.000.000			
SUELDO (Docente)	\$ 2.500.000			
TOTAL GASTOS DE PERSONAL	\$ 21.000.000			
OTROS GASTOS				
TRANSPORTE	\$ 40.000			
FOTOCOPIAS	\$ 15.000			
TOTAL OTROS GASTOS	\$ 55.000			
COSTO TOTAL DEL PROYECTO	\$ 23.605.000			

Tabla. 2 Presupuesto Estimado



7. RESULTADOS

7.1 Análisis y Especificación de Requerimientos

En la presente sección, se presenta el análisis y la especificación de los requerimientos del aplicativo, los cuales han sido obtenidos a partir de un proceso de entrevistas realizadas a las empresas recuperadoras de hierro. Estos diálogos han permitido recopilar información valiosa sobre las necesidades y desafíos específicos que enfrentan estas empresas en su día a día. Con base en estas entrevistas, se han identificado y documentado de manera precisa los requerimientos funcionales y no funcionales necesarios para el desarrollo del aplicativo. A lo largo de esta sección, se detallarán de forma clara y concisa los resultados obtenidos y se presentará una lista exhaustiva de los requerimientos identificados, brindando así una base sólida y fundamentada para el desarrollo del aplicativo orientado a las necesidades de las empresas recuperadoras de hierro.

7.1.1 Estudio de campo

En este tipo de estudio, se adentra en un campo de investigación bastante particular y poco común en comparación con áreas más convencionales como el desarrollo de software sencillo de contabilidad o la creación de sitios web. En lugar de abordar temas amplios y generalizados, el foco se centra en el conocimiento específico y especializado del gremio de la recuperación de hierro.

En el estudio de campo realizado, se entrevistaron a diversas empresas recuperadoras de hierro como lo son “Metales de Santander” y “Ekol-Reciclables” ubicadas en Piedecuesta-Santander. El objetivo fue obtener información directa sobre las necesidades y requerimientos que enfrentan estas compañías en su gestión y control de inventario. Al tener acceso a la palabra de las compañías, se obtuvo una fuente confiable de información que permitió establecer una base sólida para la definición de los requerimientos del software. Estas entrevistas proporcionaron una comprensión más profunda de las operaciones y procesos relacionados con la recuperación de hierro, así como de las funcionalidades y características clave que el software debía abordar para satisfacer las necesidades del sector.

Dichas entrevistas fueron preparadas y se hizo una lista de preguntas para poder obtener la mayor información posible y así poder saber cuáles son los requerimientos del software:

1. ¿Cuál es el proceso actual que siguen para gestionar y controlar el inventario de hierro en su empresa? ¿Considera que la ayuda de un software sería más eficiente?
2. ¿Cuáles son los principales desafíos o dificultades que enfrentan en la gestión de inventario?

- 
- 3.** ¿Qué herramientas o métodos utilizan actualmente para llevar el registro de inventario? ¿Cree que con un software evitaría la perdida de la información?
 - 4.** ¿Cómo gestionan las compras y ventas de hierro en su empresa? ¿Le gustaría tener las compras y ventas registradas en una base de datos?
 - 5.** ¿Qué funcionalidades o características consideran imprescindibles en un software de gestión de inventario para su empresa?
 - 6.** ¿Existe alguna necesidad específica relacionada con la gestión de inventario que no esté siendo satisfecha actualmente?
 - 7.** ¿Cómo les gustaría que un software de gestión de inventario optimizara sus procesos y brindara soluciones a sus desafíos?
 - 8.** ¿Qué funcionalidades o características consideran imprescindibles en un software de gestión de inventario para su empresa?
 - 9.** ¿Cómo obtienen actualmente informes detallados de ventas, compras, caja menor y préstamos realizados? ¿Qué datos desearían filtrar en estos informes?
 - 10.** ¿Cómo concilian actualmente las cuentas entre clientes o proveedores? ¿Qué información desearían tener disponible para realizar esta tarea de manera más eficiente?
 - 11.** ¿Qué acciones específicas realizan cuando una cuenta está pendiente de pago o cobro? ¿Cómo les gustaría poder abonar o cancelar saldos correspondientes?
 - 12.** ¿Qué información desean poder modificar en los datos de las personas registradas en su sistema?
 - 13.** ¿Qué información les gustaría tener disponible en el apartado de visualización general de la cuenta?
 - 14.** ¿Cómo gestionan actualmente los valores correspondientes a los materiales seleccionados? ¿Tienen un sistema establecido?
 - 15.** ¿Cómo manejan actualmente el registro y seguimiento de los materiales y su respectiva cantidad en su empresa?

Se fue personalmente a dichas compañías para hacer las respectivas entrevistas y tomar evidencias del ambiente y estado de la empresa.



A continuación, se mostrarán fotografías tomadas en el proceso de las entrevistas y el ambiente de las empresas “Ekol-Reciclables” y “Metales de Santander” ubicadas en Piedecuesta-Santander.

7.1.1.1 “Metales de Santander”

Se tomaron evidencias del ambiente y estado de “Metales de Santander” siendo esta una de las compañías recuperadoras de hierro de bajos recursos, a continuación, una fotografía evidenciando que se fue personalmente a la empresa.

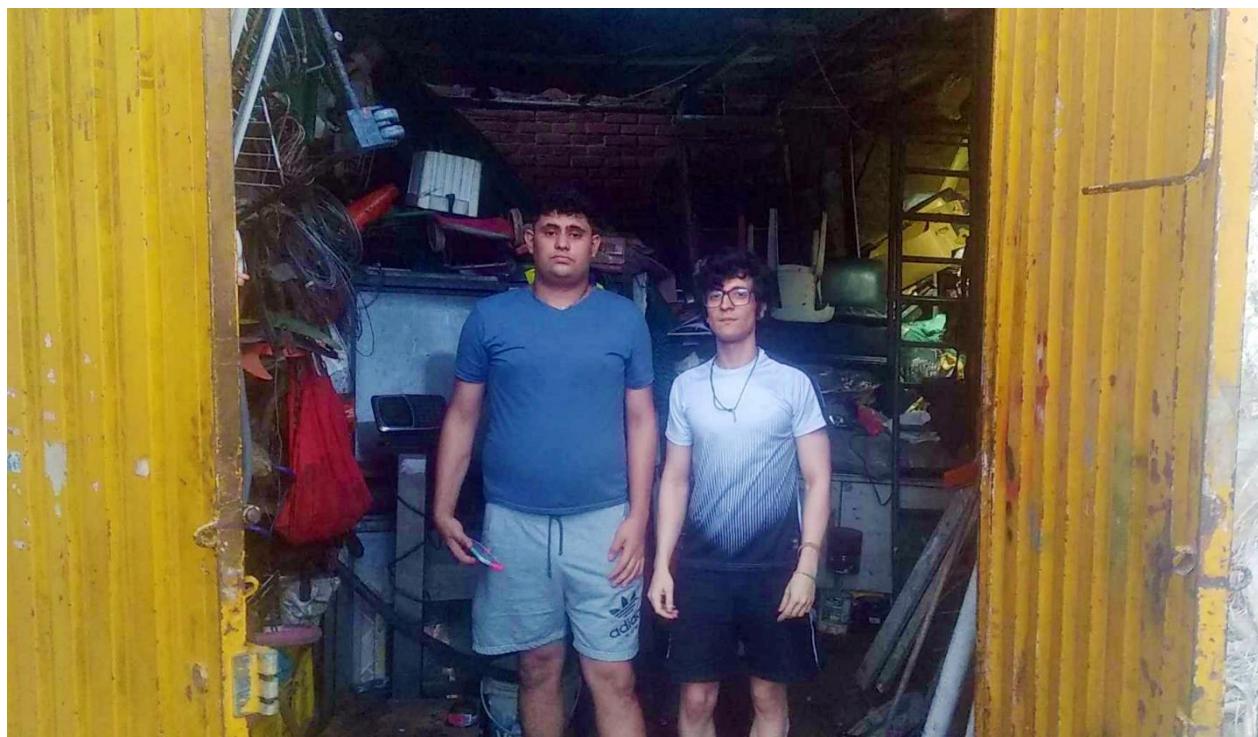


Fig. 13 Entrevistas presenciales “Metales de Santander”

7.1.1.1.1 Ambiente “Metales de Santander”

Se evidencia que “Metales de Santander” es una de las compañías recuperadoras de hierro de bajos recursos, como se ha comentado en el transcurso de este documento.

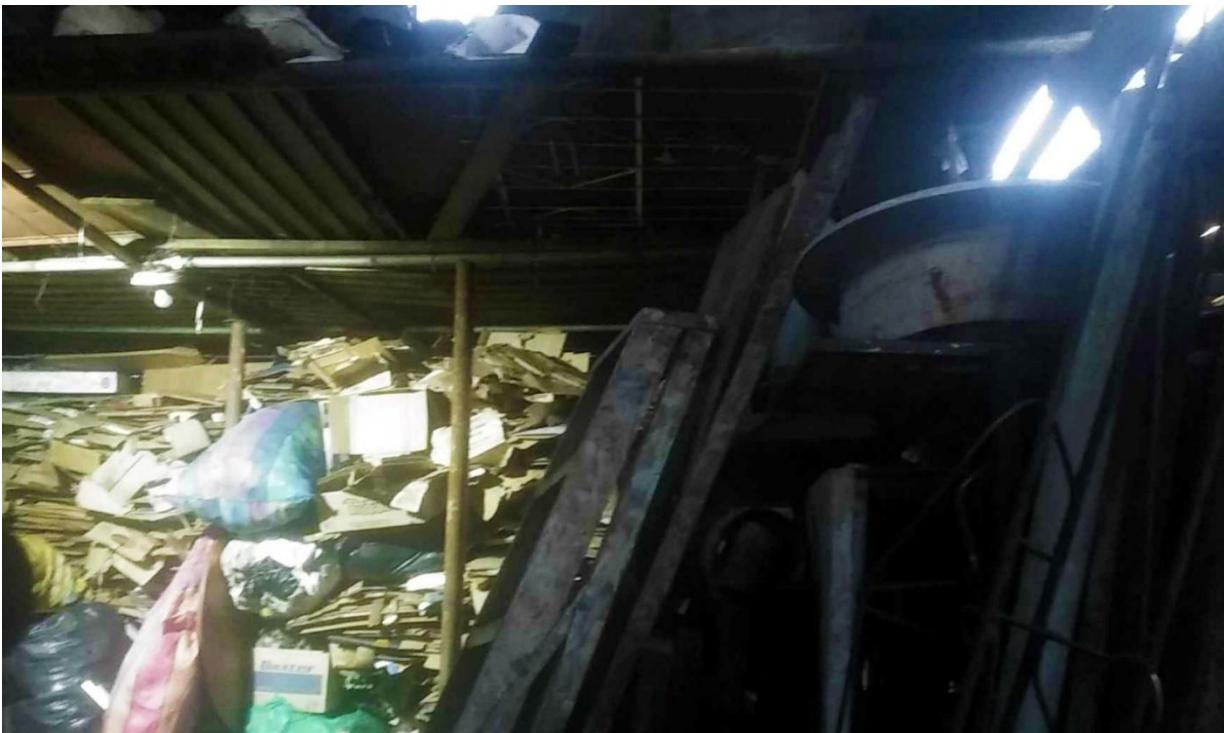


Fig. 14 Metales de Santander” (1)



Fig. 15 Metales de Santander” (2)

7.1.1.2 Entrevista “Metales de Santander”

Se entrevistó al personal presente en la compañía en ese momento, quienes solicitaron que se protegiera su identidad manteniendo el anonimato por su seguridad.

A continuación, se deja la URL de la entrevista realizada: <https://youtu.be/rYNhjd5cOd0>

URL entrevistas en GitHub:

<https://github.com/JonathanBaquero01/ContabilidadMetales/tree/main/Recursos%20tesis/Entrevistas>

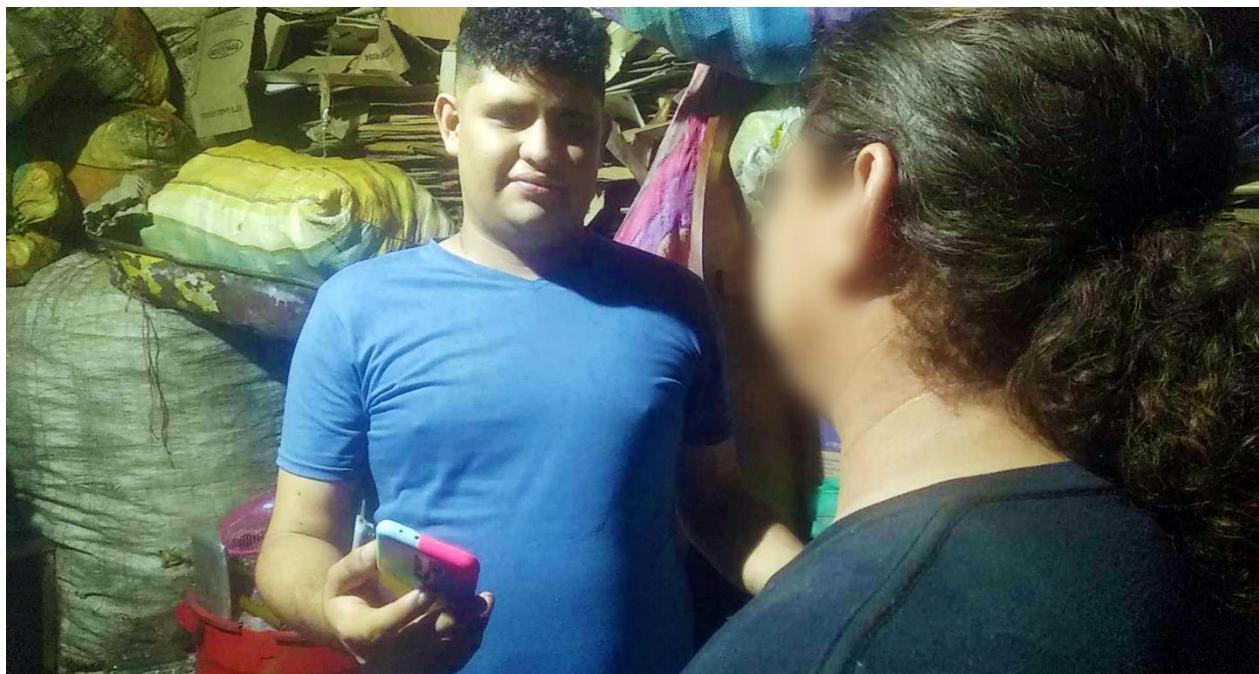


Fig. 16 Entrevista “Metales de Santander”



7.1.1.2 “Ekol-Reciclable”

Al igual que “Metales de Santander”, se tomaron evidencias del ambiente y estado de “Ekol-Reciclable”, siendo esta una de las compañías recuperadoras de hierro de bajos recursos, a continuación, una fotografía evidenciando que se fue personalmente a la empresa.



Fig. 17 Entrevistas presenciales “Ekol-Reciclable”

7.1.1.2.1 Ambiente “Ekol-Reciclabl”

Se evidencia igualmente que “Ekol-Reciclabl” es una de las compañías recuperadoras de hierro de bajos recursos, como se ha ido comentado en el transcurso de este documento.



Fig. 18 Ambiente “Ekol-Reciclabl” (1)



Fig. 19 Ambiente “Ekol-Reciclabl” (2)

7.1.1.2.2 Entrevista “Ekol-Reciclable”

Se entrevistó al personal presente en la compañía en ese momento, quienes de igual manera solicitaron que se protegiera su identidad manteniendo el anonimato por su seguridad.

A continuación, se deja la URL de la entrevista realizada: <https://youtu.be/OV51wGyvoAM>

URL entrevistas en GitHub:

<https://github.com/JonathanBaquero01/ContabilidadMetales/tree/main/Recursos%20tesis/Entrevistas>



Fig. 20 Entrevista “Ekol-Reciclable”



7.1.2 Requerimientos del sistema

Gracias a las entrevistas realizadas a las compañías que brindaron su colaboración y facilitaron el acceso a los datos y recursos necesarios, se pudo obtener una fuente sólida y confiable para la creación de los requerimientos del software.

Los requerimientos del sistema son esenciales en cualquier proyecto. En el caso del presente software, es importante contar con un conjunto claro de especificaciones que guíen todo el proceso de desarrollo. Estos requerimientos deben ser cuidadosamente diseñados para satisfacer las necesidades de las organizaciones del sector y garantizar que el software sea de fácil uso, seguro y cumpla con los estándares de calidad necesarios.

En esta sección, se presenta los requerimientos que se han identificado mediante entrevistas a empresas dedicadas a la recuperación de hierro para el sistema que se desarrolla, esto con el objetivo de establecer una base sólida para el éxito del proyecto.

7.1.2.1 Requerimientos funcionales

7.1.2.1.1 Lista cliente proveedor

Requerimientos funcionales						
Prioridad: Alta						
CÓDIGO DEL REQUERIMIENTO:	RF01	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Lista cliente proveedor					
Descripción						
El administrador puede elegir el tipo de persona entre cliente o proveedor, y al momento de elegirlo aparece todos los proveedores y clientes registrados en el software con su respectiva lista de materiales.						
Funcionalidad						
Permitir al administrador elegir el tipo de persona, ya sea cliente o proveedor, dentro del software. Una vez seleccionado el tipo de persona, se muestra en pantalla todos los proveedores o clientes registrados en el sistema, junto con su respectiva lista de materiales. Esto brinda al administrador la capacidad de visualizar de manera rápida y conveniente la información relevante de los proveedores o clientes, como los materiales que ofrecen o han adquirido, facilitando así la gestión y toma de decisiones relacionadas con la relación comercial con ellos.						
Criterios de aceptación	1. El administrador debe tener la capacidad de seleccionar el tipo de persona como "cliente" o "proveedor" desde una opción o menú desplegable. 2. Se deben mostrar todos los proveedores y clientes registrados en el software. 3. Los datos de proveedores, clientes y materiales deben estar actualizados y reflejar la información correcta.					
Restricciones						
El usuario no puede eliminar a otros usuarios de su mismo nivel, no puede acceder o tener conocimientos de la base de datos.						
El administrador solo puede elegir entre los dos tipos de personas disponibles: cliente o proveedor. No se permite la creación de nuevos tipos de personas.						
Solo se muestra en la lista los proveedores y clientes que estén registrados en el software. No se incluyen personas que no estén debidamente registradas.						

Tabla. 3 Lista cliente proveedor

7.1.2.1.2 Vista de la lista de materiales por persona

Requerimientos funcionales						
Prioridad: Muy alta						
CÓDIGO DEL REQUERIMIENTO:	RF02	Actor	Usuario y administrador			
NOMBRE DEL REQUERIMIENTO	Vista de la lista de materiales por persona					
Descripción						
El usuario puede seleccionar un material de una lista según si es cliente o proveedor, y se muestra el valor correspondiente por material. Luego, podrá registrar la cantidad del material seleccionado en kilogramos (kg) en una cuenta previamente vinculada a la persona propietaria. Toda esta información es reflejada en un apartado dedicado para permitir la visualización general de la cuenta.						
Funcionalidad						
Permitir a los usuarios seleccionar un material de una lista según si es cliente o proveedor, y mostrar el valor correspondiente por material. Los usuarios también pueden registrar la cantidad del material seleccionado en kilogramos (kg) en una cuenta previamente vinculada a la persona propietaria. Esta información se refleja en un apartado dedicado para permitir la visualización general de la cuenta.						
Criterios de aceptación	1. El sistema debe permitir al usuario seleccionar un material de una lista de materiales dependiendo de si es un cliente o un proveedor. 2. El sistema debe permitir al usuario registrar los kilogramos del material seleccionado en una cuenta. 3. El sistema debe mostrar un apartado para permitir la visualización general de la cuenta.					
Restricciones						
El usuario debe tener una cuenta previamente vinculada para poder realizar la selección de materiales y registrar la cantidad de kilogramos.						
La cuenta vinculada a la persona propietaria debe estar previamente registrada en el sistema. No se permite asociar la cantidad del material a una cuenta inexistente.						

Tabla. 4 Vista de la lista de materiales por persona

7.1.2.1.3 Actualización de la lista de materiales

Requerimientos funcionales						
Prioridad: Alta						
CÓDIGO DEL REQUERIMIENTO:	RF03	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Actualización de la lista de materiales					
Descripción						
El administrador puede modificar la lista de materiales por persona después de haber seleccionado el tipo de persona.						
Funcionalidad						
Permitir al administrador modificar la lista de materiales asociada a una persona una vez que haya seleccionado el tipo de persona. Esto significa que el administrador tiene la capacidad de agregar, editar o eliminar los materiales disponibles para cada persona en particular. Estas modificaciones bridan flexibilidad al sistema, permitiendo ajustar y personalizar la lista de materiales según las necesidades y características específicas de cada persona registrada.						
Criterios de aceptación	1. El sistema debe permitir que el administrador modifique la lista de materiales después de haber seleccionado el tipo de persona. 2. El sistema debe permitir que el administrador guarde los cambios realizados en la lista de materiales. 3. El sistema debe mostrar los cambios realizados en la lista de materiales al administrador.					
Restricciones						
El administrador solo puede modificar la lista de materiales de una persona si el tipo de persona ya ha sido seleccionado previamente.						

Tabla. 5 Actualización de la lista de materiales

7.1.2.1.4 Valor cuenta

Requerimientos funcionales						
Prioridad: Alta						
CÓDIGO DEL REQUERIMIENTO:	RF04	Actor	Usuario			
NOMBRE DEL REQUERIMIENTO	Valor cuenta					
Descripción						
El usuario puede ver todo el tiempo el valor a pagar a lo largo de la cuenta.						
Funcionalidad						
Permitir al usuario ver el valor total a pagar en cualquier momento durante el proceso de pago, lo que le permite mantener un control sobre el monto total de la cuenta. Esto significa que en todo momento, el usuario puede acceder y ver claramente el monto total que se debe pagar en relación a la cuenta en cuestión.						
Criterios de aceptación	1. El usuario debe poder ver el valor a pagar antes de realizar el pago. 2. El valor a pagar debe ser actualizado de manera inmediata cuando se realicen cambios en la cuenta. 3. El valor a pagar debe ser preciso y reflejar correctamente la suma total de los montos a pagar a lo largo de la cuenta					
Restricciones						
El sistema debe proporcionar una interfaz clara y legible para mostrar el valor a pagar de la cuenta de manera continua. El valor a pagar debe ser calculado de acuerdo con las reglas y políticas establecidas por la empresa o el sistema, asegurando su precisión y consistencia.						

Tabla. 6 Valor cuenta

7.1.2.1.5 Impresión de cuentas y datos del usuario

Requerimientos funcionales						
Prioridad: Alta						
CÓDIGO DEL REQUERIMIENTO:	RF05	Actor	Usuario y administrador			
NOMBRE DEL REQUERIMIENTO	Impresión de cuentas y datos del usuario					
Descripción						
Al finalizar el registro de todos los materiales en la cuenta, se podrá imprimir las cuentas y todos los datos del cliente o el proveedor generando así una factura en un formato PDF.						
Funcionalidad						
Permitir al usuario finalizar el registro de todos los materiales en la cuenta y generar una factura en formato PDF que contenga todos los datos del cliente o proveedor. Una vez que se hayan seleccionado y registrado los materiales, el sistema proporcionará la opción de generar la factura, la cual contendrá la información detallada de los materiales, cantidades, precios y cualquier otro dato relevante. Esta funcionalidad permite al usuario obtener un registro oficial y tangible de la transacción realizada, facilitando la gestión contable y proporcionando un comprobante para el cliente o proveedor involucrado.						
Criterios de aceptación	1. La factura generada debe tener el formato PDF. 2. La factura debe contener los datos del cliente o proveedor. 3. La generación de la factura en formato PDF se realiza de manera rápida y eficiente, sin retrasos significativos en el proceso.					
Restricciones						
El software debe tener los permisos y accesos necesarios para interactuar con los recursos de manipulación de ficheros y generar el archivo PDF de manera adecuada.						

Tabla. 7 Impresión de cuentas y datos del usuario

7.1.2.1.6 Actualización de la lista de usuarios

Requerimientos funcionales						
Prioridad: Muy alta						
CÓDIGO DEL REQUERIMIENTO:	RF06	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Actualización de la lista de usuarios					
Descripción						
El administrador puede registrar editar y eliminar tanto persona como proveedor al igual que usuario en un apartado propio de personas						
Funcionalidad						
Permitir al administrador registrar, editar y eliminar tanto personas como proveedores, así como usuarios en un apartado específico de personas. El administrador también puede gestionar la información de los usuarios, así como los datos de los proveedores y personas, permitiendo así un control completo sobre el manejo de información de cada uno de los roles.						
Criterios de aceptación	<ol style="list-style-type: none">1. El administrador debe poder registrar, editar y eliminar proveedores y usuarios.2. El administrador debe poder ver una lista de personas, proveedores y usuarios registrados.3. Las modificaciones realizadas en los registros de personas, proveedores y usuarios se reflejan de manera correcta y actualizada en el sistema.					
Restricciones						
El usuario debe tener una identificación única para poder ser registrado. Solo el administrador tiene los permisos necesarios para registrar, editar y eliminar personas, proveedores y usuarios en el apartado de personas.						

Tabla. 8 Actualización de la lista de usuarios



7.1.2.1.7 Datos por defecto

Requerimientos funcionales						
Prioridad: Baja						
CÓDIGO DEL REQUERIMIENTO:	RF07	Actor	Software			
NOMBRE DEL REQUERIMIENTO	Datos por defecto					
Descripción						
El software tiene el nombre de usuario y una contraseña registrada administradora que viene por defecto.						
Funcionalidad						
El software tiene un sistema de autenticación para los usuarios, que incluye un nombre de usuario y una contraseña registrada administradora predeterminada. Esto permite a los usuarios iniciar sesión y acceder a los recursos del software por primera vez.						
Criterios de aceptación	1. El software debe tener un nombre de usuario y una contraseña administradora predeterminada. 2. El acceso con la cuenta de administrador se limita a las funcionalidades y privilegios correspondientes a dicho rol. 3. El software permite cambiar la contraseña administradora por defecto a una personalizada, asegurando la confidencialidad y la seguridad de la cuenta.					
Restricciones						
El nombre de usuario y la contraseña administradora por defecto deben ser seguros. El software debe asegurarse de que el nombre de usuario y la contraseña administradora por defecto no sean accesibles para usuarios no autorizados.						

Tabla. 9 Datos por defecto



7.1.2.1.8 Modificación de datos de usuarios

Requerimientos funcionales						
Prioridad: Media						
CÓDIGO DEL REQUERIMIENTO:	RF08	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Modificación de datos de usuarios					
Descripción						
El administrador puede modificar los datos de cada persona registrada en el sistema						
Funcionalidad						
Permitir al administrador modificar los datos de cada persona registrada en el sistema. Esto implica proporcionar una interfaz que permita al administrador acceder y editar la información de las personas, como nombres, números de contacto, entre otros datos relevantes. El software debe garantizar la seguridad y la integridad de los datos.						
Criterios de aceptación	<ol style="list-style-type: none">1. El administrador debe poder acceder a los datos de cada persona registrada en el sistema.2. El administrador debe poder modificar los datos de cada persona registrada en el sistema.3. Los cambios realizados por el administrador deben ser guardados.4. El administrador debe poder ver los cambios realizados.					
Restricciones						
Solo el administrador o usuarios autorizados deben tener permisos para modificar los datos de las personas registradas en el sistema. Se deben implementar medidas de seguridad para evitar cambios no autorizados.						

Tabla. 10 Modificación de datos de usuarios

7.1.2.1.9 Visualización de cuentas y su historial

Requerimientos funcionales						
Prioridad: Alta						
CÓDIGO DEL REQUERIMIENTO:	RF09	Actor	Usuario y administrador			
NOMBRE DEL REQUERIMIENTO	Visualización de cuentas y su historial					
Descripción						
El usuario puede ver las cuentas y el historial de cuentas						
Funcionalidad						
Permitir a los usuarios acceder y visualizar sus cuentas personales, así como el historial de cuentas asociado. Esto brinda a los usuarios la capacidad de revisar y monitorear sus transacciones pasadas, saldos y registros financieros. La funcionalidad permite una mayor transparencia y control sobre las actividades financieras, facilitando la toma de decisiones y el seguimiento de la información relacionada con las cuentas.						
Criterios de aceptación	1. El usuario debe poder ver sus cuentas y el historial de estas 2. La información de las cuentas y el historial es clara y legible para el usuario. 3. El sistema proporciona una interfaz clara y accesible para que el usuario pueda ver las cuentas existentes.					
Restricciones						
El usuario debe tener una cuenta válida para acceder a la información de la cuenta y el historial. El software puede requerir autorización para garantizar que solo los usuarios permitidos puedan acceder a las cuentas y el historial asociado.						

Tabla. 11 Visualización de cuentas y su historial

7.1.2.1.10 Modificación cuentas a través de valor

Requerimientos funcionales						
Prioridad: Media						
CÓDIGO DEL REQUERIMIENTO:	RF10	Actor	Usuario y administrador			
NOMBRE DEL REQUERIMIENTO	Modificación cuentas a través de valor					
Descripción						
El usuario puede modificar las cuentas a través de su valor, es decir lo único modificable sería el valor total a pagar o pagado de la cuenta						
Funcionalidad						
Permitir al usuario modificar las cuentas, centrándose en la modificación del valor total a pagar o pagado de la cuenta. Esta funcionalidad brinda al usuario la capacidad de ajustar y actualizar los montos correspondientes a las cuentas. El enfoque principal está en la modificación del valor total, proporcionando flexibilidad para adaptar y mantener actualizada la información financiera en el sistema.						
Criterios de aceptación	1. El usuario debe ser capaz de modificar las cuentas a través de su valor 2. El sistema debe guardar los cambios realizados por el usuario. 3. El sistema valida que el nuevo valor total a pagar o el nuevo valor pagado sean numéricos y cumplan con las reglas establecidas, como no ser negativos o superar el valor total a pagar.					
Restricciones						
Solo se puede editar el valor total a pagar o pagado de la cuenta. El usuario debe tener acceso a la información de la cuenta antes de modificar el valor total.						

Tabla. 12 Modificación cuentas a través de valor

7.1.2.1.11 Identificación de cuentas en deuda

Requerimientos funcionales						
Prioridad: Alta						
CÓDIGO DEL REQUERIMIENTO:	RF11	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Identificación de cuentas en deuda					
Descripción						
El administrador puede identificar cuentas para proveedores o para clientes, y determinar si una cuenta de un cliente aún se debe o una cuenta de un proveedor aún se debe. Esto se refleja en una tabla dentro del módulo de cuentas, donde se puede seleccionar la cuenta por pagar y abonar o cancelar el saldo correspondiente, positivo o negativo, según sea el caso.						
Funcionalidad						
Permitir al administrador identificar y gestionar las cuentas de proveedores y clientes. Este debe poder determinar si una cuenta de un cliente aún está pendiente de pago o una cuenta de un proveedor aún está pendiente de cobro. Además, se proporciona la capacidad de seleccionar una cuenta por pagar y abonar o cancelar el saldo correspondiente, ya sea positivo o negativo. Toda esta información se refleja de manera clara y organizada en una tabla dentro del módulo de cuentas, facilitando el seguimiento y la administración eficiente de las transacciones y saldos pendientes.						
Criterios de aceptación	<ol style="list-style-type: none"> 1. El administrador debe poder identificar cuentas para proveedores y clientes. 2. El administrador debe poder determinar si una cuenta de un cliente aún se debe o una cuenta de un proveedor aún se debe. 3. Debe haber una tabla dentro del módulo de cuentas donde se pueda seleccionar la cuenta por pagar y abonar o cancelar el saldo correspondiente. 4. El saldo debe ser positivo o negativo según sea el caso. 					
Restricciones						
El sistema debe tener la capacidad de almacenar información sobre cuentas de proveedores y clientes. El sistema debe tener la capacidad de mostrar una tabla con la información de las cuentas por pagar.						

Tabla. 13 Identificación de cuentas en deuda

7.1.2.1.12 Descontar cuentas

Requerimientos funcionales						
Prioridad: Baja						
CÓDIGO DEL REQUERIMIENTO:	RF12	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Descontar cuentas					
Descripción						
El software permite realizar cruces de cuentas, es decir, si un cliente o un proveedor tienen una cuenta por cobrar, podrán descontar del valor de la cuenta por cobrar en la cuenta correspondiente.						
Funcionalidad						
Permitir realizar cruces de cuentas entre clientes o proveedores. Esto significa que, si un cliente o proveedor tiene una cuenta por cobrar, podrán descontar del valor de esa cuenta por cobrar en la cuenta correspondiente. Esta funcionalidad facilita la conciliación y compensación de cuentas entre las partes involucradas, optimizando la gestión y el seguimiento de los pagos y saldos pendientes.						
Criterios de aceptación	<ol style="list-style-type: none"> 1. El software debe permitir realizar cruces de cuentas entre clientes o proveedores. 2. El software debe permitir descontar el valor de la cuenta por cobrar en la cuenta correspondiente. 3. El software valida que el valor a descontar no supere el saldo de la cuenta por cobrar 					
Restricciones						
<p>Se debe contar con cuentas válidas para poder realizar cruces de cuentas.</p> <p>Antes de permitir el cruce de cuentas, el software debe verificar que existan saldos suficientes tanto en la cuenta por cobrar como en la cuenta correspondiente.</p> <p>El sistema debe realizar los cálculos de descuentos de manera precisa y confiable, asegurando que los valores se deduzcan correctamente de las cuentas involucradas.</p>						

Tabla. 14 Descontar cuentas

7.1.2.1.13 Cálculos de deudas totales

Requerimientos funcionales						
Prioridad: Media						
CÓDIGO DEL REQUERIMIENTO:	RF13	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Cálculos de deudas totales					
Descripción						
El administrador puede ver los cálculos de las deudas totales para cada persona los cuales serán positivos o negativos dependiendo del tipo de persona						
Funcionalidad						
Permitir al administrador visualizar los cálculos de las deudas totales para cada persona, presentando los resultados como valores positivos o negativos según el tipo de persona. Esta funcionalidad brinda al administrador una visión clara y completa de las deudas acumuladas por cada individuo, facilitando el seguimiento y la gestión de las obligaciones financieras en el sistema.						
Criterios de aceptación	1. El administrador debe ser capaz de ver los cálculos de las deudas totales para cada persona. 2. Los cálculos de las deudas totales deben ser positivos o negativos dependiendo del tipo de persona. 3. El sistema debe mostrar los resultados de los cálculos de manera clara y precisa.					
Restricciones						
El administrador debe tener acceso a la información de cada persona para poder ver los cálculos de las deudas.						
El sistema debe garantizar la privacidad y confidencialidad de los datos relacionados con las deudas de las personas, asegurando que solo el administrador tenga acceso a ellos.						

Tabla. 15 Cálculos de deudas totales

7.1.2.1.14 Préstamo a clientes y proveedores

Requerimientos funcionales						
Prioridad: Baja						
CÓDIGO DEL REQUERIMIENTO:	RF14	Actor	Software			
NOMBRE DEL REQUERIMIENTO	Préstamo a clientes y proveedores					
Descripción						
El sistema permite el registro de préstamos tanto a clientes como a proveedores a través de un apartado propio en donde se puede también actualizarlo o dar por pagado el préstamo.						
Funcionalidad						
El software registra préstamos tanto a clientes como a proveedores, proporcionando un sistema para ingresar los detalles del préstamo, como el monto. Además, el sistema permite actualizar el estado del préstamo, indicando si está pendiente, pagado o vencido. El software proporciona una interfaz dedicada para administrar y realizar un seguimiento de los préstamos, brindando a los usuarios un control eficiente y una gestión adecuada de los préstamos en el contexto del negocio.						
Criterios de aceptación	1. El sistema debe permitir el registro de préstamos tanto a clientes como a proveedores. 2. El sistema debe tener un apartado dedicado para el registro y actualización de los préstamos. 3. El sistema debe permitir marcar un préstamo como pagado. 4. El sistema debe guardar los datos de los préstamos registrados.					
Restricciones						
Tanto el cliente como el proveedor deben tener cuentas válidas para poder realizar el préstamo. El sistema debe validar y verificar la información ingresada para garantizar la integridad de los datos, como la verificación de la existencia del cliente o proveedor, y la coherencia de los valores de préstamo.						

Tabla. 16 Préstamo a clientes y proveedores

7.1.2.1.15 Caja menor

Requerimientos funcionales						
Prioridad: Muy alta						
CÓDIGO DEL REQUERIMIENTO:	RF15	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Caja menor					
Descripción						
El administrador tiene un apartado de caja menor en dónde estará el valor reflejado de capital y podrán tener registrados los gastos propios de la empresa diferentes a las cuentas o préstamos, como lo pueden ser gastos en materiales de oficina, comestibles para invitados, entre otros.						
Funcionalidad						
Permitir al administrador llevar un registro de los gastos propios de la empresa, como los gastos en materiales de oficina, comestibles para invitados, etc., y mantener una caja menor con el valor reflejado de capital. Esta funcionalidad brindará al administrador un control preciso sobre los gastos de la empresa y facilitará la gestión de la caja menor de manera eficiente.						
Criterios de aceptación	1. El apartado de caja menor debe mostrar el valor reflejado de capital. 2. El administrador debe tener la capacidad de registrar gastos propios de la empresa diferentes a las cuentas o préstamos. 3. El administrador debe tener la capacidad de registrar gastos en materiales de oficina, comestibles para invitados, entre otros 4. El administrador deberá tener acceso a un sistema de contabilidad para llevar un registro de los gastos.					
Restricciones						
Solo el administrador o usuarios autorizados deberían tener acceso al apartado de caja menor para registrar y visualizar los gastos propios de la empresa.						

Tabla. 17 Caja menor



7.1.2.1.16 Registros de caja menor

Requerimientos funcionales						
Prioridad: Alta						
CÓDIGO DEL REQUERIMIENTO:	RF16	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Registros de caja menor					
Descripción						
Los registros en caja menor se harán a través del valor de si es un gasto o un ingreso menor y estarán acompañados de una descripción.						
Funcionalidad						
Permitir al usuario realizar registros en caja menor, diferenciando entre gastos e ingresos menores. Cada registro estará acompañado de una descripción para proporcionar información adicional. Esta funcionalidad permitirá un seguimiento preciso de los movimientos de caja menor, facilitando el registro y control de los gastos e ingresos de menor cuantía en la empresa.						
Criterios de aceptación	1. Los registros en caja menor deben ser realizados con el valor de si es un gasto o un ingreso menor. 2. Los registros deben incluir una descripción. 3. Interfaz clara y fácil de usar para ingresar los registros en caja menor.					
Restricciones						
La cantidad de dinero que se puede registrar en caja menor debe estar dentro de un límite predeterminado.						
El software puede requerir que el valor ingresado para el gasto o ingreso sea numérico y válido, evitando caracteres no permitidos.						
Solo los usuarios autorizados deben tener acceso y permisos adecuados para realizar registros en la caja menor, evitando modificaciones no autorizadas o manipulaciones indebidas.						

Tabla. 18 Registros de caja menor



7.1.2.1.17 Afectaciones registro de inventario

Requerimientos funcionales						
Prioridad: Muy alta						
CÓDIGO DEL REQUERIMIENTO:	RF17	Actor	Software			
NOMBRE DEL REQUERIMIENTO	Afectaciones registro de inventario					
Descripción						
El registro de inventarios será afectado dependiendo de los gastos de caja menor y de los gastos de las cuentas modificando en el apartado de inventario un valor llamado capital y todo el peso por material.						
Funcionalidad						
Permitir afectar el registro de inventarios en función de los gastos realizados tanto en caja menor como en las cuentas. Estos gastos modificarán un valor denominado "capital" en el apartado de inventario, reflejando el impacto financiero de los gastos en el inventario de la empresa. Además, el software deberá considerar el peso por material en los cálculos relacionados con el inventario. Estas funcionalidades permitirán mantener un registro actualizado y preciso del valor del inventario, teniendo en cuenta los gastos realizados.						
Criterios de aceptación	1. El registro de inventarios debe reflejar los cambios en los gastos de caja menor y de las cuentas. 2. El registro de inventarios debe mostrar un valor llamado capital. 3. El registro de inventarios debe mostrar el peso por material. 4. El registro de inventarios debe actualizarse de forma correcta cuando se realicen cambios en los gastos de caja menor y de las cuentas.					
Restricciones						
El software puede requerir que los gastos ingresados sean numéricos y válidos, evitando caracteres no permitidos o valores negativos incorrectos.						
Solo los usuarios autorizados deben tener acceso y permisos adecuados para registrar los gastos y modificar el inventario, evitando modificaciones no autorizadas o manipulaciones indebidas.						

Tabla. 19 Afectaciones registro de inventario



7.1.2.1.18 Filtración de materiales

Requerimientos funcionales						
Prioridad: Media						
CÓDIGO DEL REQUERIMIENTO:	RF18	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Filtración de materiales					
Descripción						
El inventario podrá ser filtrado por materiales por valor o por kilogramo permitiendo así que el administrador pueda saber la cantidad de material que hay en la empresa para una venta o si existen la cantidad suficiente para una venta a través de una cuenta de un cliente.						
Funcionalidad						
Permitir al administrador filtrar el inventario por materiales, ya sea por valor o por kilogramo. Esto proporcionará la capacidad de conocer la cantidad de material disponible en la empresa para su venta, así como verificar si existen suficientes cantidades para satisfacer una venta específica a través de una cuenta de un cliente. Estas funcionalidades facilitarán la gestión y planificación de inventario						
Criterios de aceptación	1. El software debe permitir al administrador filtrar el inventario por materiales, ya sea por valor o por kilogramo. 2. El filtrado del inventario por materiales debe proporcionar resultados precisos y actualizados, mostrando la cantidad de material disponible en la empresa. 3. El sistema debe actualizar automáticamente el inventario después de cada venta, reflejando los cambios en la cantidad de material disponible.					
Restricciones						
Solo los usuarios autorizados, como administradores o personal designado, deben tener acceso y permisos adecuados para filtrar el inventario por materiales y acceder a la información relacionada. El software puede requerir que los valores ingresados para el filtrado, ya sea por valor o por kilogramo, sean numéricos y válidos, evitando caracteres no permitidos o valores incorrectos.						

Tabla. 20 Filtración de materiales

7.1.2.1.19 Ingresos y egresos

Requerimientos funcionales						
Prioridad: Alta						
CÓDIGO DEL REQUERIMIENTO:	RF19	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Ingresos y egresos					
Descripción						
El administrador tendrá "ingresos y egresos", un apartado propio en dónde se podrán obtener los informes obtenidos en ventas, compras, caja menor y préstamos los cuales se podrán filtrar por fechas o por tipo de gastos o ingresos.						
Funcionalidad						
El software proporcionará al administrador un apartado denominado "Ingresos y Egresos" donde podrá obtener informes detallados de ventas, compras, caja menor y préstamos realizados. Estos informes podrán ser filtrados según fechas específicas o por tipo de gastos e ingresos. Esta funcionalidad permitirá al administrador acceder de manera rápida y sencilla a la información relevante, facilitando la toma de decisiones y el análisis financiero.						
Criterios de aceptación	1. El software debe proporcionar un apartado dedicado al administrador llamado "Ingresos y Egresos" donde se puedan obtener informes detallados de ventas, compras, caja menor y préstamos. 2. El apartado de "Ingresos y Egresos" debe permitir al administrador filtrar los informes por fechas específicas, gastos, e ingresos, lo que garantizará la capacidad de obtener datos relevantes para un período determinado. 3. Los informes obtenidos en el apartado de "Ingresos y Egresos" deben ser precisos, completos y actualizados, reflejando de manera confiable las transacciones realizadas en ventas, compras, caja menor y préstamos.					
Restricciones						
El software debe garantizar la privacidad y seguridad de los datos financieros. Solo los usuarios autorizados deben tener acceso a la información de ingresos y egresos. El software debe generar informes de manera rápida y eficiente, especialmente al aplicar filtros por fechas o por tipo de gastos o ingresos.						

Tabla. 21 Ingresos y egresos

7.1.2.2 Requerimientos no funcionales

Los requerimientos no funcionales desempeñan un papel fundamental en el presente proyecto, ya que definen las características y atributos clave que no están directamente relacionados con la funcionalidad del software, pero que son esenciales para su éxito. Estos requerimientos abarcan aspectos como, la seguridad, la disponibilidad, la eficiencia y el mantenimiento, entre otros. Es importante considerarlos porque permiten establecer estándares de calidad y garantizar que el software cumpla con los criterios de desempeño y satisfacción del usuario.

Al igual que en la sección anterior, se presentarán los requerimientos no funcionales que se han identificado para el sistema que se está desarrollando, con el objetivo de establecer una base sólida para el éxito del proyecto.

7.1.2.2.1 Capacidad de almacenamiento

Requerimientos no funcionales						
Prioridad	Alta					
CÓDIGO DEL REQUERIMIENTO:	RNF01	Actor	Base de datos			
NOMBRE DEL REQUERIMIENTO	Capacidad de almacenamiento					
Descripción	Tener una gran capacidad de almacenamiento para poder almacenar varios usuarios y la información que se maneja					
Restricciones						
El sistema debe ser capaz de adaptarse y escalar su capacidad de almacenamiento según sea necesario. Esto implica que debe ser posible aumentar la capacidad de almacenamiento de manera fácil y eficiente a medida que aumente la cantidad de usuarios o la cantidad de datos que se manejen en el sistema. El sistema debe ser capaz de crecer y adaptarse a medida que las necesidades de almacenamiento cambien con el tiempo.						

Tabla. 22 Capacidad de almacenamiento

7.1.2.2.2 Disponibilidad

Requerimientos no funcionales						
Prioridad	Muy alta					
CÓDIGO DEL REQUERIMIENTO:	RNF02	Actor	Servidor			
NOMBRE DEL REQUERIMIENTO	Disponibilidad					
Descripción	Tener el sistema a disponibilidad de los usuarios las veinticuatro(24) horas de la semana gracias al servidor donde se alojara el software					
Restricciones						
Para garantizar la disponibilidad del sistema las veinticuatro (24) horas del día, es necesario contar con un servidor robusto y confiable. Esto implica que la infraestructura de servidor debe ser capaz de manejar cargas de trabajo intensivas y estar diseñada para minimizar los tiempos de inactividad. Se debe asegurar que el servidor cuente con suficiente capacidad de procesamiento, memoria y almacenamiento para soportar las demandas de los usuarios y evitar cuellos de botella.						

Tabla. 23 Disponibilidad

7.1.2.2.3 Eficiencia

Requerimientos no funcionales						
Prioridad	Baja					
CÓDIGO DEL REQUERIMIENTO:	RNF03	Actor	Software			
NOMBRE DEL REQUERIMIENTO	Eficiencia					
Descripción	Se requiere eficiencia en la implementación del aplicativo de tal forma que los tiempos de respuesta para los procesos ejecutados no sean tan grandes al punto de generar insatisfacción al usuario.					
Restricciones						
Para lograr una implementación eficiente del aplicativo, es necesario contar con recursos de hardware adecuados. Esto incluye disponer de suficiente capacidad de procesamiento, memoria y almacenamiento para soportar la carga de trabajo esperada y ejecutar los procesos de manera rápida y eficiente. Es importante considerar requisitos de hardware para el aplicativo y asegurarse de que se cumplan para garantizar un rendimiento óptimo.						

Tabla. 24 Eficiencia

7.1.2.2.4 Mantenimiento

Requerimientos no funcionales						
Prioridad	Media					
CÓDIGO DEL REQUERIMIENTO:	RNF04	Actor	Software			
NOMBRE DEL REQUERIMIENTO	Mantenimiento					
Descripción	Se requiere el mantenimiento del sistema de tal forma que al realizar modificaciones se minimice el impacto por cambiar código, para lo cual es contempla la separación entre el servidor de aplicaciones y el de bases de datos					
Restricciones						
Se debe mantener una documentación actualizada y detallada sobre los cambios realizados en el sistema. Esto incluye registrar los cambios en el código, describir su propósito y documentar las decisiones de diseño y arquitectura.						

Tabla. 25 Mantenimiento

7.1.2.2.5 Herramientas de código libre

Requerimientos no funcionales						
Prioridad	Muy alta					
CÓDIGO DEL REQUERIMIENTO:	RNF05	Actor	Software			
NOMBRE DEL REQUERIMIENTO	Herramientas de código libre					
Descripción	No se requiere de herramientas de pago para el desarrollo y ejecución del software, todo está basado en herramientas de código libre.					
Restricciones						
Se establece la restricción de no utilizar herramientas que impliquen costos adicionales, ya sean herramientas de desarrollo, bases de datos, u otros componentes necesarios para el desarrollo y ejecución del software. Todas las herramientas utilizadas deben ser de libre disponibilidad y no requerir la compra de licencias o el pago de tarifas.						

Tabla. 26 Herramientas de código libre

7.1.2.2.6 Confidencialidad de la información

Requerimientos no funcionales						
Prioridad	Alta					
CÓDIGO DEL REQUERIMIENTO:	RNF06	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Confidencialidad de la información					
Descripción	Es importante garantizar la confidencialidad, integridad y disponibilidad de la información almacenada en el aplicativo. El sistema debe contar con mecanismos de autenticación y autorización para controlar el acceso a los datos de los usuarios.					
Restricciones						
El sistema debe contar con mecanismos de autenticación y autorización para controlar el acceso a los datos de los usuarios. Esto implica que se debe verificar la identidad de los usuarios antes de permitirles acceder al sistema, esto utilizando métodos como contraseñas seguras						

Tabla. 27 Confidencialidad de la información

7.1.2.2.7 Escalabilidad

Requerimientos no funcionales						
Prioridad	Media					
CÓDIGO DEL REQUERIMIENTO:	RNF07	Actor	Administrador			
NOMBRE DEL REQUERIMIENTO	Escalabilidad					
Descripción	Es importante que el aplicativo tenga la capacidad de adaptarse y crecer de manera eficiente para satisfacer las necesidades futuras de las empresas recuperadoras de hierro. El sistema debe ser capaz de manejar un aumento en la cantidad de usuarios, transacciones y datos sin comprometer su rendimiento y funcionalidad.					
Restricciones						
El sistema debe ser fácil de mantener y permitir futuras actualizaciones y mejoras. Esto implica seguir buenas prácticas de desarrollo de software, utilizar estándares y patrones de diseño adecuados. Se deben establecer procesos y herramientas para el mantenimiento del sistema, como la documentación adecuada, para facilitar las futuras modificaciones y evoluciones.						

Tabla. 28 Escalabilidad

7.1.3 Product backlog

Ya con todos los requerimientos definidos se puede hacer y seguir el Product backlog, el cual es importante en el presente proyecto ya que sirve como una lista priorizada de todos los requisitos, funcionalidades y mejoras planificadas para el aplicativo de las empresas recuperadoras de hierro. Actúa como una guía para el equipo de desarrollo, proporcionando una visión clara y organizada de las características y tareas que deben implementarse.

Alta prioridad			
	Nombre del requerimiento	Código	Prioridad 1-10
1	Actualización de la lista de usuarios	RF06	10
2	Vista de la lista de materiales por persona	RF02	9.5
3	Caja menor	RF15	9.2
4	Afectaciones registro de inventario	RF17	9
5	Ingresos y egresos	RF19	8.7
6	Registros de caja menor	RF16	8.5
7	Lista cliente proveedor	RF01	8
8	Visualización de cuentas y su historial	RF09	7.5
9	Identificación de cuentas en deuda	RF11	7
10	Valor cuenta	RF04	7
11	Actualización de la lista de materiales	RF03	6
12	Impresión de cuentas y datos del usuario	RF05	5
13	Modificación cuentas a través de valor	RF10	5



14	Cálculos de deudas totales	RF13	5
15	Modificación de datos de usuarios	RF08	4.5
16	Filtración de materiales	RF18	5
17	Descontar cuentas	RF12	3
18	Préstamo a clientes y proveedores	RF14	2
19	Datos por defecto	RF07	1
<i>Baja prioridad</i>			

Fig. 21 Product backlog

7.1.4 Gestión y priorización de tareas y requerimientos

Aunque para la gestión del proyecto se utilizó principalmente el marco de trabajo de "Modelado por Prototipo", también se incorporaron elementos del enfoque "SCRUM" con el fin de llevar el control de las tareas que se iban realizando y que se consignaron en un tablero Trello.

La gestión de un proyecto es fundamental para garantizar su éxito y el cumplimiento de los objetivos establecidos. En este sentido, los marcos de trabajo como SCRUM ofrecen un enfoque estructurado y colaborativo para la planificación, seguimiento y entrega de los proyectos de software [32]. Aunque en el presente caso se ha modificado el enfoque SCRUM debido a la limitación de personal y recursos, aún se puede aprovechar algunos de sus principios y prácticas para mejorar la gestión. Se adaptó el marco para ajustarlo a nuestras necesidades, omitiendo roles específicos como el Product Owner y el Scrum Master, así como el equipo de desarrollo dedicado. Sin embargo, se sigue utilizando elementos esenciales como la división del trabajo en iteraciones, la realización de reuniones periódicas de seguimiento y la priorización de tareas con ayuda del Product Backlog. Esto permite mantener un enfoque ágil, colaborativo y adaptativo en la gestión del proyecto, maximizando la eficiencia y minimizando los riesgos.

Ya con el Product backlog definido se procede a planear el desarrollo de los componentes del aplicativo. Tal y como se mencionó en el marco tecnológico, se optó por usar Trello para llevar un control sobre la gestión y priorización de tareas y requerimientos del presente proyecto; En el tablero Trello se utilizaron una serie de colores para evaluar la prioridad de cada tarea en el lapso de tiempo en que esta se encuentre, en donde el color verde es una tarea poco prioritaria, amarillo media y rojo alta.

7.1.4.1 Revisión bibliográfica

Se siguió al pie de la letra el marco de trabajo y el cronograma planteado, en donde de la primera a la cuarta semana se empieza con la revisión bibliográfica, en donde se busca y revisa diferentes fuentes de información, como libros, revistas científicas, artículos, entre otros, con el objetivo de obtener una comprensión más profunda y actualizada sobre el tema de investigación, además de ir documentando paso a paso cómo va el desarrollo del proyecto, garantizando así la trazabilidad y transparencia en el desarrollo del software.

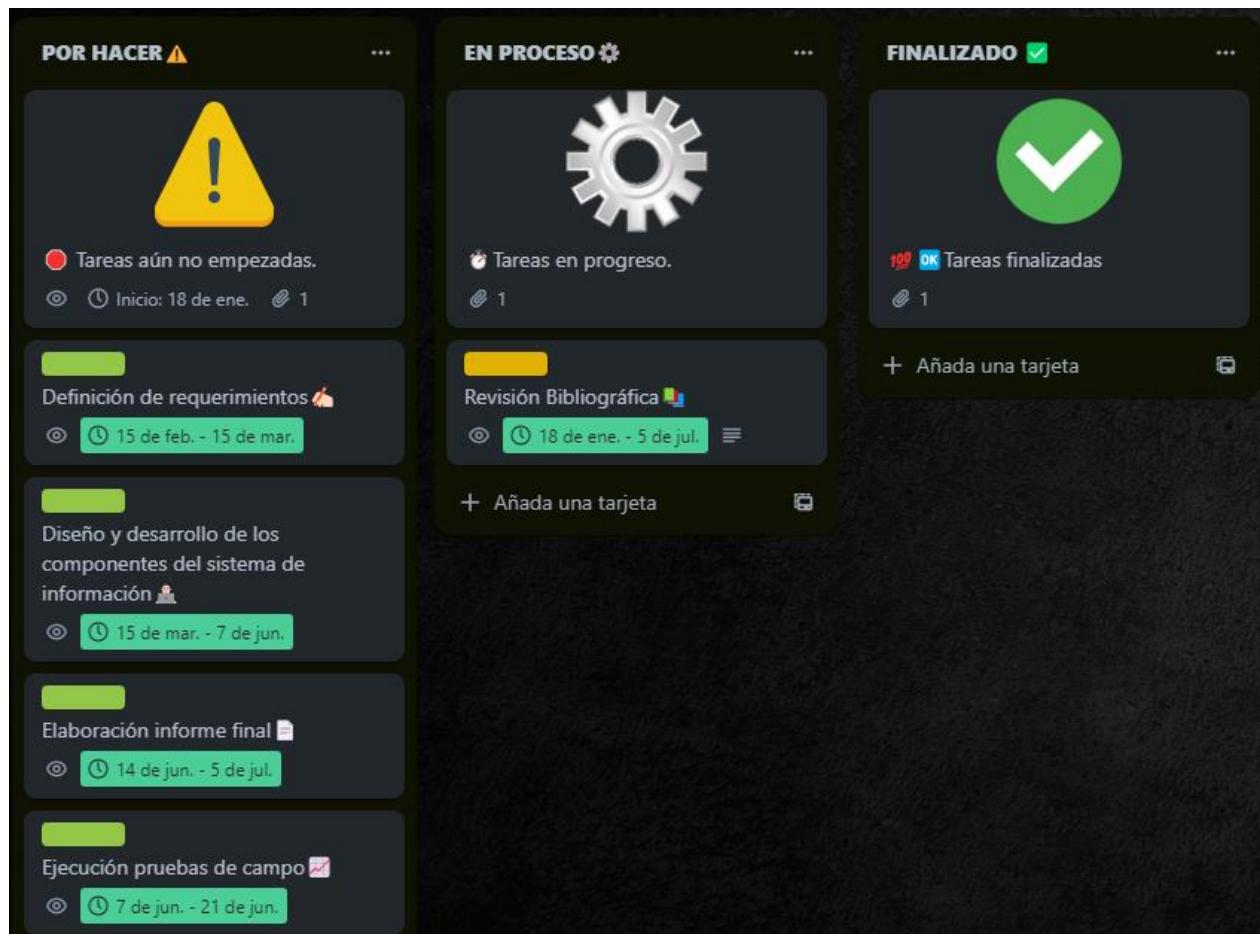


Fig. 22 Tablero de gestión de tareas, semana 1-4

7.1.4.2 Recolección y refinamiento de requisitos

Ya pasadas las primeras cuatro semanas del inicio de la revisión bibliográfica junto con su documentación, se pasa a la tarea de definición de requerimientos, la cual es un pilar importante ya que de esto depende de que funcionalidades tiene el software; dicha tarea va desde la quinta semana a la octava.

Durante este período, se llevó a cabo un exhaustivo análisis de las necesidades y expectativas de las empresas recuperadoras de hierro. Se realizaron búsquedas en la web y entrevistas a personas dedicadas al sector de recuperación de hierro para recopilar información detallada sobre los procesos existentes, los desafíos y las oportunidades de mejora. Con base en esta información, se elaboró un conjunto claro y completo de requerimientos funcionales que sirvieron como guía para el desarrollo del software, siguiendo así la primera fase del marco de trabajo “Modelado por Prototipos”, que como su nombre lo indica es la Recolección y refinamiento de requisitos.



Fig. 23 Tablero de gestión de tareas, semana 4-9

Ya finalizada la tarea de definición de requerimientos, se lleva a cabo la tarea más importante de todo el proyecto, la cual es diseñar y desarrollar el software mediante los requerimientos planteados, este apartado incluye la fase dos y tres del marco de trabajo “Modelado por Prototipos”, las cuales son Diseño rápido y Construcción del prototipo.

Siguiendo la prioridad de los requerimientos que se elaboran en el Product backlog, se ejecutan las funcionalidades según su priorización, a medida que se completa un requerimiento se va marcando como completado para así mantener un orden y respetar el Product backlog.

Teniendo en cuenta lo anterior, y tal como se mencionó en el apartado del marco de trabajo, al momento de desarrollar cada uno de los requerimientos del sistema, se hacen una serie de iteraciones; ya que los requisitos se van desarrollado según como son priorizados en el product backlog, donde cada vez que se termine un requerimiento, el equipo de desarrollo se devuelve a la fase de diseño rápido hasta llegar nuevamente al apartado de refinamiento del prototipo, y así para desarrollar cada requerimiento planteado.

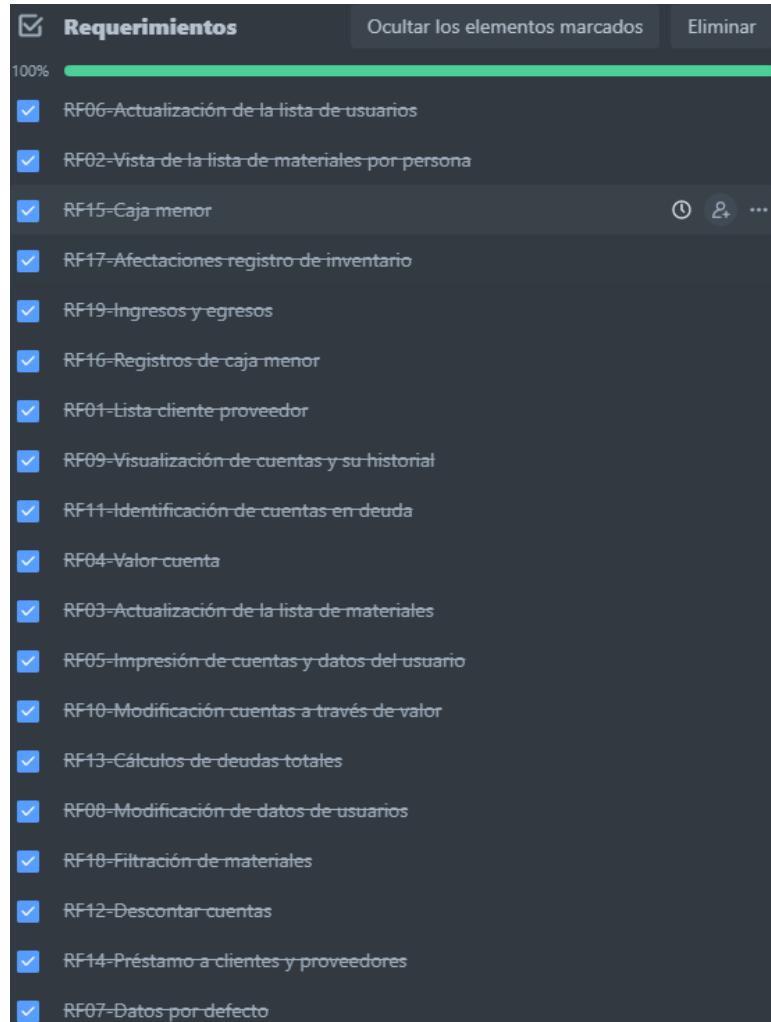


Fig. 24 Priorización de requerimientos según el Product backlog

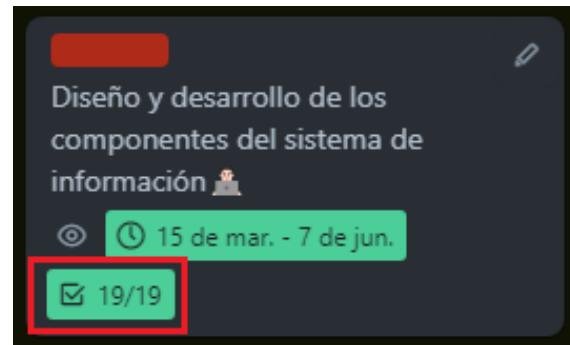


Fig. 25 Número de requerimientos

En esta etapa, se aplican los conocimientos que se adquieren durante la definición de requerimientos para diseñar una solución óptima y eficiente. Se emplean las herramientas tecnológicas mencionadas en el marco tecnológico, como NetBeans, Java, MySQL y StarUML, para implementar el software de manera robusta y funcional. El equipo de desarrollo trabaja en estrecha colaboración, siguiendo los principios del marco de trabajo de “Modelado por Prototipos”, y llevando a cabo iteraciones sucesivas para perfeccionar y ajustar la solución de acuerdo con las necesidades de las empresas recuperadoras de hierro.

POR HACER	EN PROCESO	FINALIZADO
<p>⚠ Tareas aún no empezadas.</p> <p>⌚ Inicio: 18 de ene. ⌚ 1</p>	<p>⚙️ Tareas en progreso.</p> <p>⌚ 1</p>	<p>✅ OK Tareas finalizadas</p> <p>⌚ 1</p>
<p>Elaboración informe final 📄</p> <p>⌚ 14 de jun. - 5 de jul.</p>	<p>Diseño y desarrollo de los componentes del sistema de información 🏢</p> <p>⌚ 15 de mar. - 7 de jun.</p> <p>19/19</p>	<p>Definición de requerimientos 🛡️</p> <p>⌚ 15 de feb. - 15 de mar.</p>
<p>Ejecución pruebas de campo ✅</p> <p>⌚ 7 de jun. - 21 de jun.</p>	<p>Revisión Bibliográfica 📚</p> <p>⌚ 18 de ene. - 5 de jul.</p>	<p>+ Añada una tarjeta</p>

Fig. 26 Tablero de gestión de tareas, semana 9-20

7.1.4.3 Evaluación del prototipo por el cliente y Refinamiento

Una vez finalizada la fase más importante la cual fue "Diseño y desarrollo de los componentes del sistema de información", se avanzó hacia la etapa final del proyecto. En esta nueva etapa, con el software completamente funcional, se realizaron pruebas exhaustivas en empresas especializadas en el sector de recuperación de hierro. El objetivo principal fue observar el comportamiento del software y evaluar su eficiencia, así como ver que eficiencia y mejoras tienen estas compañías al utilizar el aplicativo. Durante este proceso, se documentaron detalladamente todas las observaciones relevantes en el informe final del proyecto; cabe aclarar que este apartado incluye la fase cuatro y cinco del marco de trabajo "Modelado por Prototipos", siendo estas Evaluación del prototipo por el cliente y Refinamiento del prototipo, en donde además de ver la mejora de las empresas al utilizar el software, estas ayudaban con comentarios y sugerencias para posibles mejoras al aplicativo

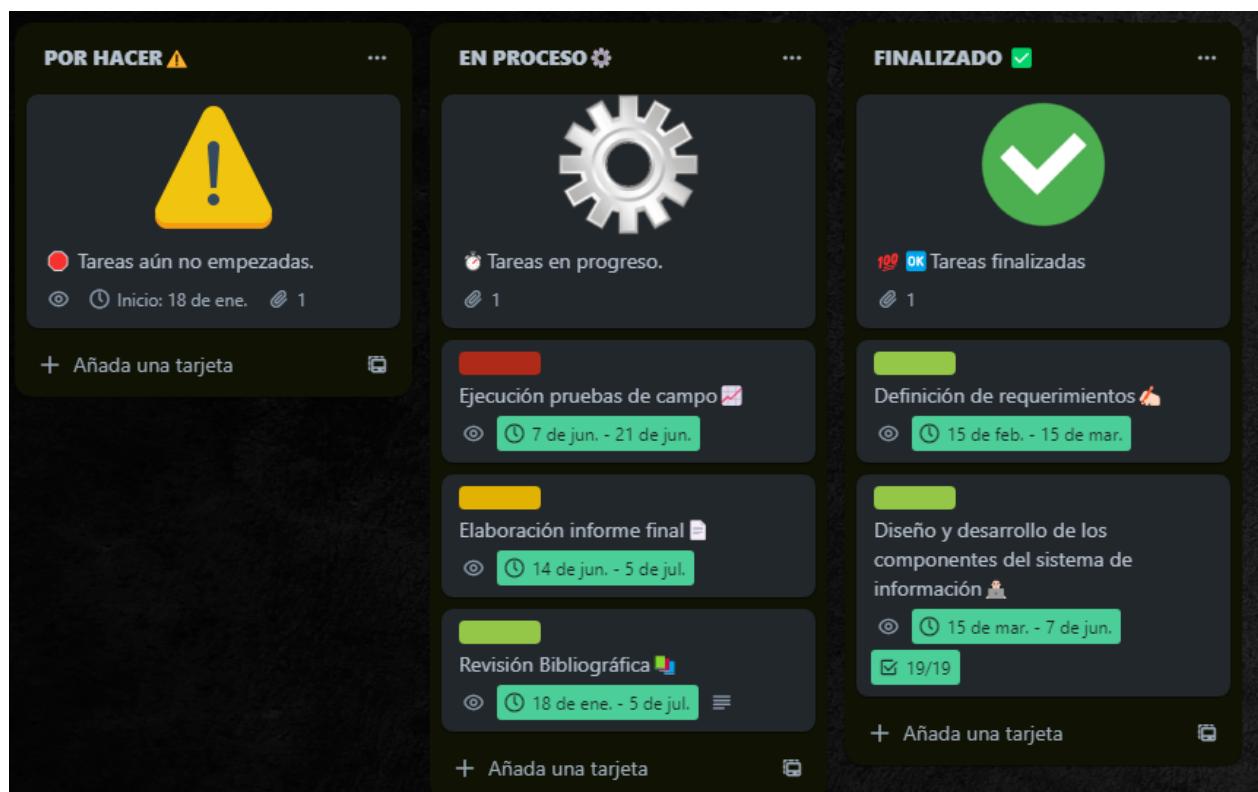


Fig. 27 Tablero de gestión de tareas, semana 20-24

7.1.4.4 Producto de ingeniería

Con todas las fases y tareas del proyecto concluidas, se puede afirmar con certeza que se logró una gestión eficiente del tiempo, cumpliendo rigurosamente con el cronograma y marco de trabajo establecido llegando a su última fase donde el software ya es un producto de ingeniería. Cada etapa del proyecto se desarrolló de manera oportuna y se asignaron los recursos necesarios para su ejecución exitosa. Esta gestión efectiva del tiempo permitió optimizar los recursos disponibles y garantizar la finalización del proyecto dentro de los plazos previstos.

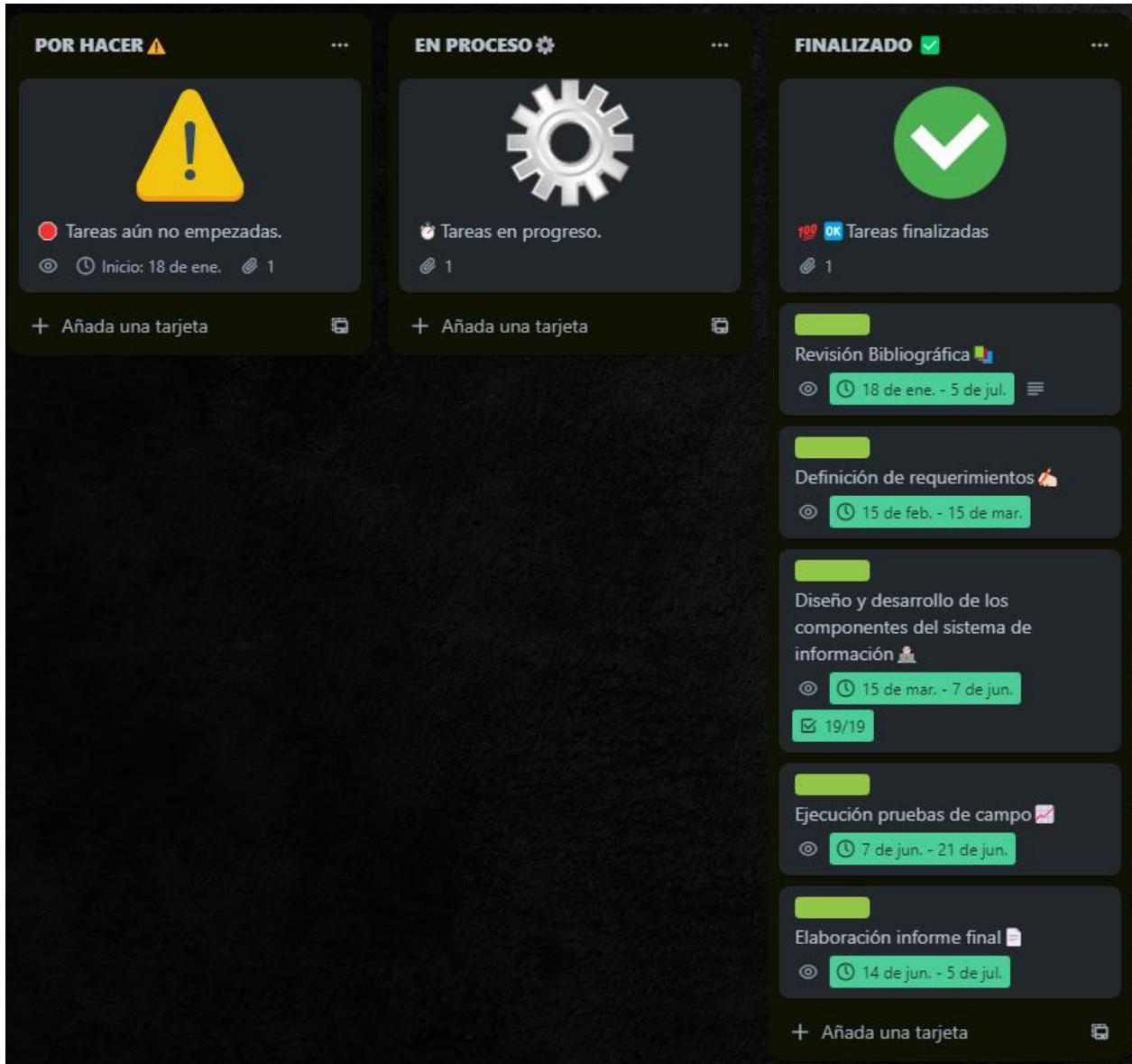


Fig. 28 Tablero de gestión de tareas, tareas concluidas

Ya con el producto en fase de ingeniería se procedió a subirlo a GitHub para ir controlando su versión y poder descargar el software.

Dicho repositorio está disponible en:

<https://github.com/JonathanBaquero01/ContabilidadMetales>

The screenshot shows the GitHub repository page for 'ContabilidadMetales'. At the top, it displays the repository name 'ContabilidadMetales' (Public), a 'Pin' button, and an 'Unwatch' button with a count of 1. Below this, there are buttons for 'main' (branch), '1 branch' (tag), and '0 tags'. On the right, there are buttons for 'Go to file', 'Add file', and 'Code'. The main area shows a single commit from 'JonathanBaquero01' titled 'first commit' at commit hash 'b5c25d1' 4 minutes ago, which includes 1 commit. The commit details show 21 files added, all of which are first commits, updated 4 minutes ago. The files listed include various Java classes, XML files, and supporting documents like DataBase.sql and Diagrama_Entity_relacion.png.

File	Commit	Last Updated
1.0-ContabilidadMetales	first commit	4 minutes ago
Imagenes	first commit	4 minutes ago
Recursos tesis	first commit	4 minutes ago
lib	first commit	4 minutes ago
src/main/java	first commit	4 minutes ago
target	first commit	4 minutes ago
1.0-ContabilidadMetales.rar	first commit	4 minutes ago
Cuenta 122.txt	first commit	4 minutes ago
DataBase.sql	first commit	4 minutes ago
Diagrama_Entity_relacion.png	first commit	4 minutes ago
ReporteGeneral	first commit	4 minutes ago
ReporteGeneral.pdf	first commit	4 minutes ago
factura de Persona Anonima.pdf	first commit	4 minutes ago
factura de Persona Anonima147.pdf	first commit	4 minutes ago
imagen.png	first commit	4 minutes ago
informe general inventario.pdf	first commit	4 minutes ago
modeloSql.mwb	first commit	4 minutes ago
nbactions.xml	first commit	4 minutes ago
pom.xml	first commit	4 minutes ago

Fig. 29 Proyecto en GitHub

7.2 Diseño y modelado del aplicativo

En la presente sección, se presentan los diagramas, modelos y esquemas que han sido desarrollados para representar de manera clara y precisa la arquitectura, funcionalidades y componentes del aplicativo. A través del diseño y modelado, se busca garantizar la coherencia, eficiencia y usabilidad del sistema de información, permitiendo una implementación efectiva y una experiencia satisfactoria para los usuarios.

7.2.1 Diagramas

7.2.1.1 Diagrama entidad relación

Siguiendo el contexto del presente proyecto de las empresas recuperadoras de hierro, se ha utilizado el diagrama entidad-relación como una herramienta fundamental en el proceso de diseño y modelado del sistema de información. Este diagrama permite representar de manera visual y clara las entidades o elementos relevantes en el dominio de la aplicación, así como las relaciones que existen entre ellas en la base de datos.

A continuación, se procede a explicar cada cuadro junto con sus relaciones:

Caja_menor: Como su nombre lo indica, registra transacciones de caja menor, incluyendo el tipo de movimiento, monto, descripción, y la fecha. Se relaciona con la tabla “Tipo_movimiento” para vincular cada transacción con un tipo específico de movimiento.

Cuenta: Registra información sobre las cuentas asociadas a personas, incluyendo su valor, estado, fecha de creación, actualización, tipo de cuenta y la persona asociada. Se relaciona con las tablas “Tipocuenta” y “Persona” para vincular cada cuenta y persona determinada.

Inventario: Registra información sobre los elementos del inventario, como su peso, descripción, material asociado y valor. Se relaciona con la tabla “Material” para vincular cada elemento del inventario con una materia prima específica.

Material: Registra información sobre diferentes materiales, como su nombre, descripción y valor. Cada elemento tiene un ID único asignado automáticamente.

Persona: Registra información sobre diferentes personas, como su nombre, número de celular, identificación, tipo de documento, archivo relacionado, descripción y contraseña. Cada persona tiene un ID único asignado automáticamente y está asociada a un tipo de individuo específico.

Prestamo: Registra información sobre los préstamos realizados, incluyendo la persona y el tipo de persona asociados, la fecha del préstamo, el valor del préstamo, la descripción y el abono realizado. Cada préstamo tiene un ID único asignado automáticamente y está relacionado con un individuo y un tipo de persona específica.

Seguimiento inventario: Registra información sobre el seguimiento y cambios realizados en el inventario, como el peso nuevo, descripción, material, valor aproximado y fecha del cambio. Cada registro en la tabla tiene un ID único asignado automáticamente.

Tipocuenta, Tipodocumento, Tipopersona y Tipos_movimientos: Almacenan información de configuración y categorización para los diferentes tipos de cuentas, tipos de documento, tipos de persona y tipos de movimientos respectivamente.

En resumen, esta base de datos permite gestionar y mantener un registro de transacciones financieras, cuentas asociadas a personas, elementos de inventario, materiales, información de personas, préstamos y seguimiento del inventario, facilitando el análisis y la gestión eficiente de estos datos.

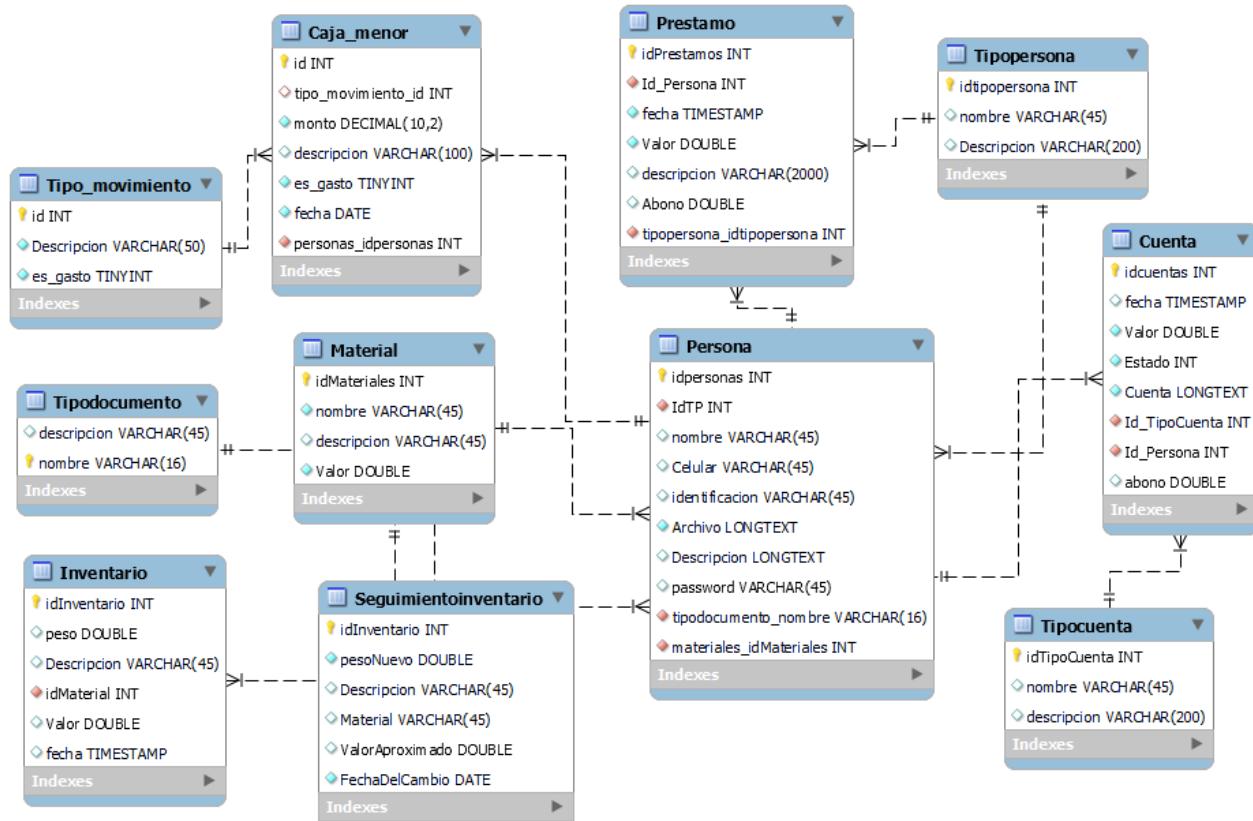


Fig. 30 Diagrama entidad relación

7.2.1.2 Diagrama de clases

Tomando en consideración el marco del proyecto en cuestión, el diagrama de clases desempeña un papel central en el diseño y modelado del sistema de información. Este diagrama permite visualizar de manera precisa las clases que componen el software junto con sus atributos y relaciones. Al utilizar el diagrama de clases en el proyecto, se logra una estructura clara y organizada, lo que facilita la implementación de funcionalidades, el trabajo en equipo y la creación de un sistema eficiente y adaptable a las necesidades de las empresas recuperadoras de hierro.

En el siguiente análisis, se destacan las clases más relevantes e influyentes en dicho diagrama, aquellas que desempeñan un papel fundamental en la arquitectura y el flujo de trabajo del sistema.

Persona: Dicha clase esta resumida con atributos comunes de un individuo para el software, como lo son el nombre, apellidos, celular, descripción, dirección y la más importante, una lista de materia prima, que es un atributo agregado de la clase “Material” qué tiene precio, nombre y unos métodos para su registro y control. Los usuarios y los clientes heredan de la clase persona obteniendo sus atributos y utilizando dichos métodos.

Usuario: Esta clase agrega un atributo nuevo que es contraseña y unos métodos propios como lo son creación de la clave y validación de la misma. Estos procedimientos son utilizados a la hora de crear un atributo en la clase “Login”

Login: Se usa para la validación del cliente a través del método de autenticación de la clase “Usuario”.

Registro: Esta clase por su parte tiene unas propiedades que son solicitadas para la creación de un cliente a través del método registrar usuario que necesita los atributos del registro

CajaMenor: Esta clase utiliza la clase “Usuario” y cuenta con atributos específicos para cada registro de la caja menor. Además, requiere de un atributo cliente para realizar diversas consultas, como la inserción de un nuevo registro, actualizaciones o eliminación. Esta propiedad también es necesaria en su método constructor para crear dichos registros. Cabe destacar que existe una relación de composición entre esta clase y el usuario, ya que siempre están interconectados.

Cliente: La presente agrega un nuevo atributo llamado “tipoCliente”, el cual identifica el tipo de usuario para el cual se crean las cuentas o préstamos. En relación a estas, se utiliza el atributo “cliente” para acceder a su método heredado “listaDeMateriales”, que como su nombre lo indica, consiste en un listado de materia prima con atributos y propiedades provenientes de la clase “Material”, tales como nombre y precio, así como los métodos para registrar y actualizar materiales. De esta manera, se puede generar una

lista de pesadas las cuales se obtienen de la clase “Pesadas”, que tiene atributos como el material obtenido, el peso total, el costo y la fecha; estos datos son necesarios para el registro de la pesada.

Factura: Esta es la clase principal que ejecutara la creación de un recibo. La presente necesita varios atributos de la clase “Cuenta” para la creación de la factura, como “Listapesadas”, Además también es necesaria para el registro de las pesadas en el inventario. La idea detrás de esto es facilitar el seguimiento de las pesadas y su registro correspondiente en el inventario.

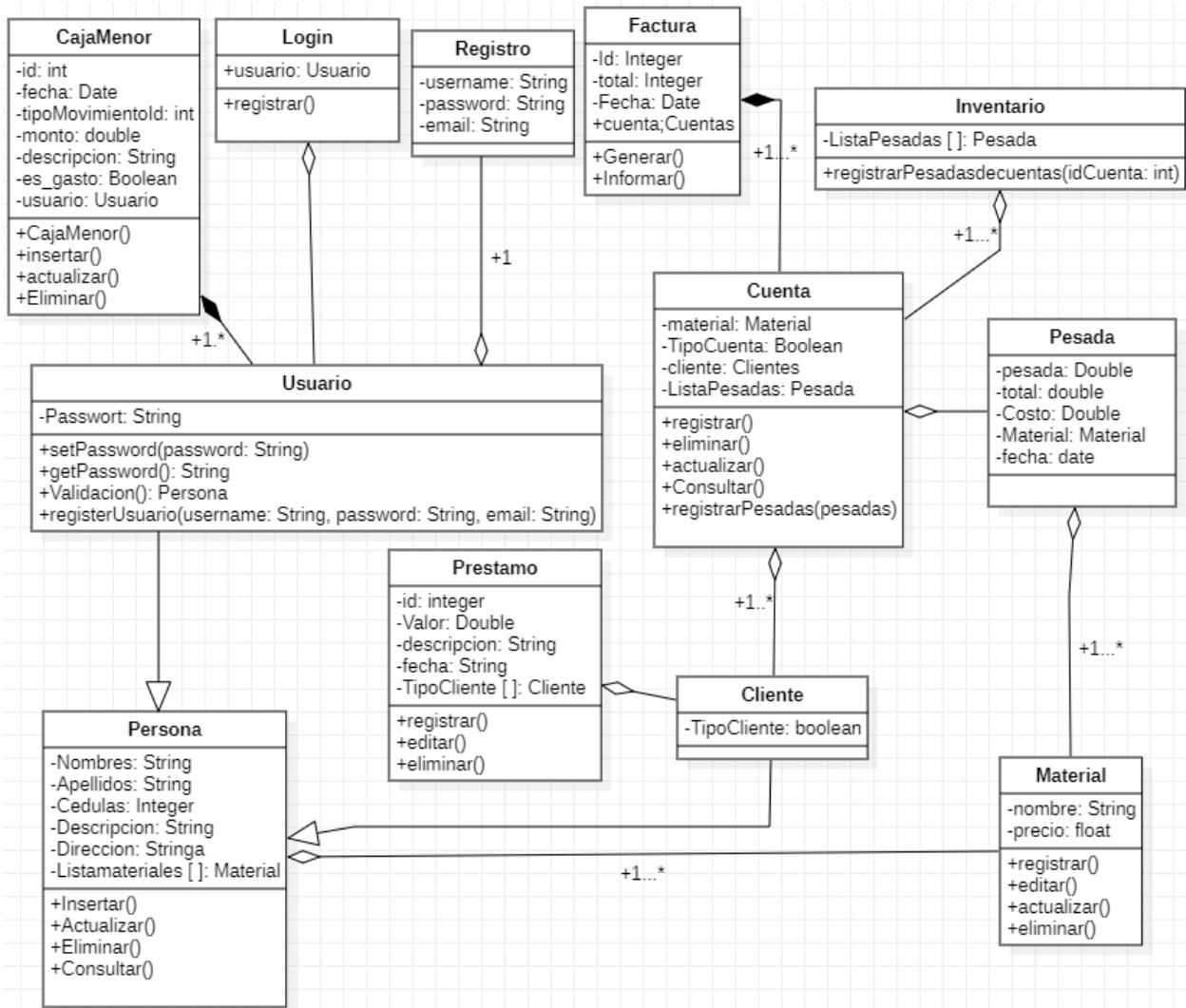


Fig. 31 Diagrama de clases

7.2.1.3 Diagrama de casos de uso

Ahora, considerando el contexto del proyecto actual enfocado en las empresas recuperadoras de hierro, el diagrama de casos de uso juega un papel fundamental en el diseño y modelado del sistema de información. Este esquema permite visualizar de manera clara y concisa las interacciones entre los actores y las funcionalidades del software. Al utilizar el diagrama de casos de uso el presente proyecto, se logra una comprensión detallada de los requisitos y escenarios de uso del sistema, lo que facilita la identificación de las funcionalidades clave y la alineación de las necesidades de los usuarios con el diseño del software.

En el siguiente análisis, se destacan los casos de uso más relevantes e influyentes en dicho diagrama, aquellos que desempeñan un papel fundamental en la interacción entre los actores y el sistema, además de que permiten cumplir con los objetivos y requerimientos establecidos para las empresas recuperadoras de hierro.

Existen dos tipos de usuario en el aplicativo

Admin

Encargado de usar todos los módulos como lo son:

- El módulo de ingresos y egresos del sistema el cual gestiona y almacena cada transacción administrada por el sistema.
- El inventario que contiene cada uno de los materiales con su kilo y registros realizados en el software
- Los préstamos que son otorgados a cada uno de los usuarios del sistema y el control de los registros de los usuarios que usan la caja
- La administración de las cuentas del registro de los clientes, proveedores y la facturación de los mismos, pudiendo además modificar las cuentas, publicarlas e insertarlas.

Cajero

Encargado de usar el modulo:

- El proceso de administración de las cuentas del registro de los clientes y proveedores es fundamental para garantizar un control integral de la facturación asociada y mantener un flujo de información preciso y actualizado. Además, se cuenta con la flexibilidad necesaria para realizar modificaciones pertinentes en dichas cuentas, adaptándose a los cambios y necesidades del negocio. Asimismo, se destaca la capacidad de publicar e insertar información relevante en las cuentas de manera oportuna.

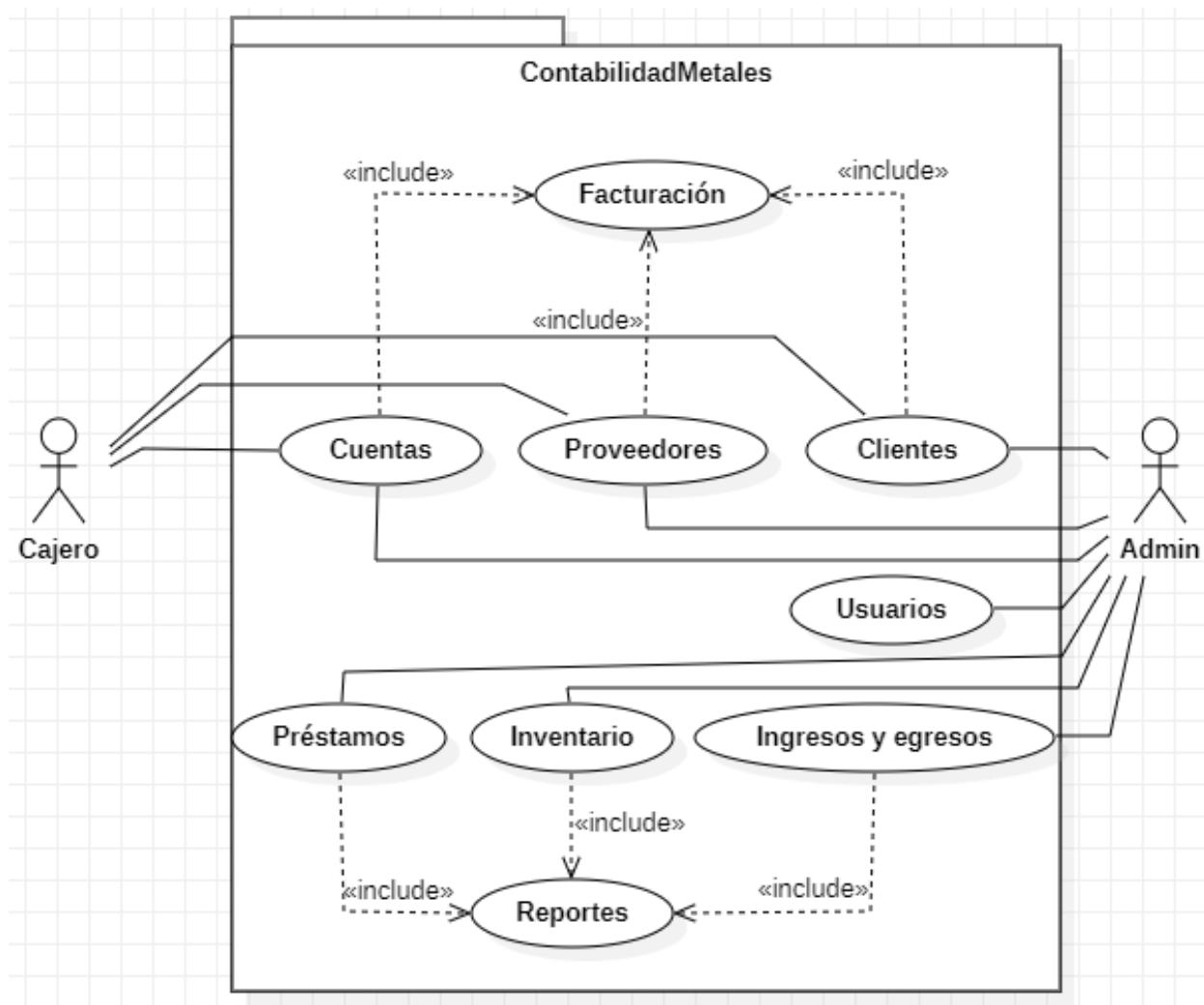


Fig. 32 Diagrama de casos de uso

7.2.1.4 Diagrama de arquitectura Desktop

Con todas las tecnologías que se han ido mencionando y la explicación de cómo es el aplicativo, es importante diseñar el diagrama de arquitectura el cual es una representación visual de la estructura del software que se desarrolla. Es importante porque permite comprender y comunicar cómo se organizan los diferentes componentes del sistema, cómo interactúan entre sí y cómo se distribuyen en el entorno tecnológico. Esto ayuda a tener una visión clara de la arquitectura del software, y sienta las bases para el mantenimiento y evolución del proyecto a largo plazo.

Se mostrará el diagrama de arquitectura para las compañías que, no tengan acceso a internet, ya que en este caso la base de datos debe ser local. Para este caso se usan tecnologías para la conexión a la base de datos:

JDBC API: La API proporciona los métodos para establecer y administrar las conexiones con la base de dato. Desde la aplicación Java se especifica la URL de conexión, el nombre de usuario y la contraseña para acceder a la base de datos. La “JDBC API” se conecta a la “JDBC driver” para establecer una comunicación efectiva entre la aplicación Java y el sistema de gestión de bases de datos.

JDBC Driver: Es un componente que actúa como puente entre la API y el sistema de gestión de bases de datos (SGBD), permitiendo la interacción y el intercambio de datos. La razón principal por la que la “JDBC API” necesita una “JDBC driver” es que cada SGBD tiene su propio protocolo de comunicación y formato de datos. La JDBC driver traduce las llamadas de la API a instrucciones y comandos específicos del SGBD correspondiente.

2 Niveles (BD en 1 Server)

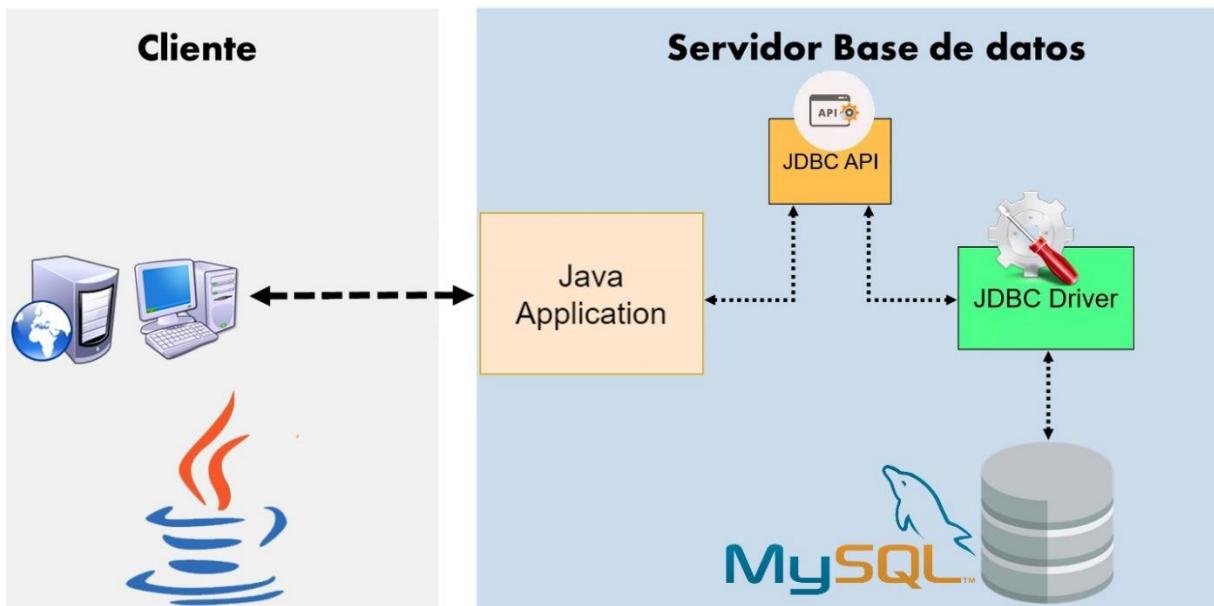


Fig. 33 Diagrama de arquitectura Desktop

7.2.1.5 Diagrama de arquitectura Web

Ahora, para este caso se mostrará el diagrama de arquitectura para las compañías que, si tengan acceso a internet, ya que en este caso se podrá montar la base de datos en la nube.

El cliente accede al aplicativo y a través de internet, se conecta a la base de datos alojada en “DB4free”, a su vez usando tecnología Mysql. Además, es importante destacar que el servidor llevará a cabo las actualizaciones de la copia de seguridad a través de internet.

2 Niveles (BD en 1 Server)

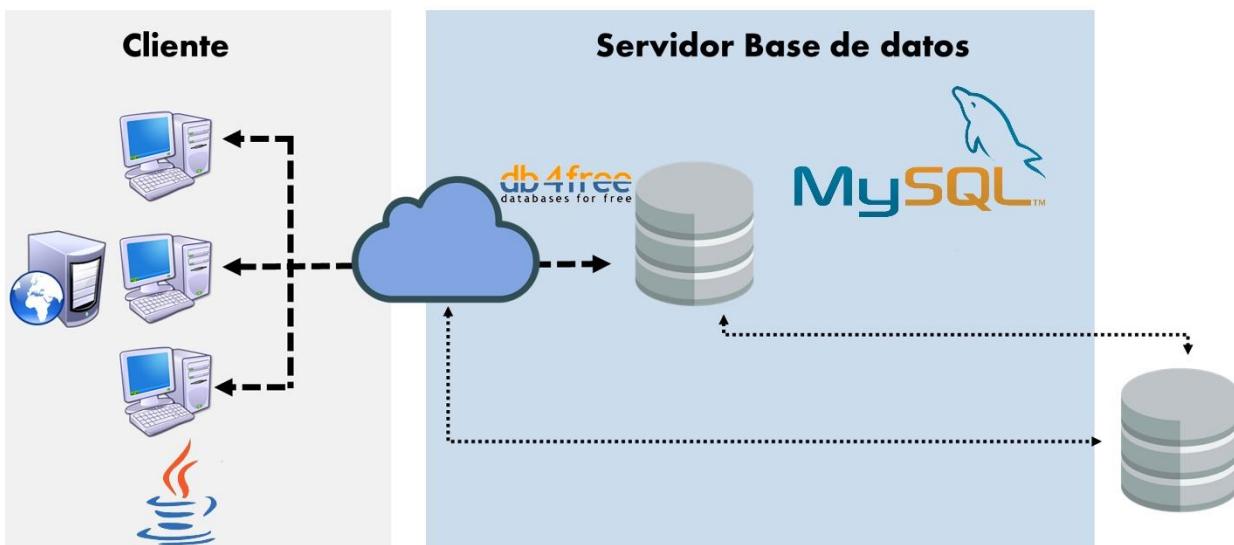


Fig. 34 Diagrama de arquitectura Web

7.2.2 Vistas de usuario

Esta sección exhibe algunas vistas antiguas y todas las interfaces finales de usuario, donde cada imagen presenta el título correspondiente a su actividad o ventana que le corresponde.

Los colores y la distribución de los botones en el software fueron cuidadosamente seleccionados y diseñados después de un proceso de investigación y retroalimentación con los usuarios finales, incluida la tutora Danith Patricia Solórzano Escobar. Durante el proceso de diseño, hubo preocupación por crear una interfaz intuitiva y atractiva que garantice una experiencia de usuario satisfactoria.

La elección de los colores se basó en principios de usabilidad y diseño visual, considerando aspectos como la legibilidad, la coherencia con la identidad de nuestra marca y la capacidad de transmitir emociones adecuadas para nuestro público objetivo. A través de la investigación y las conversaciones con los usuarios finales, incluida la tutora Danith Patricia Solórzano Escobar, se logró comprender sus preferencias y necesidades, lo que nos permitió seleccionar una paleta de colores que se alineara con sus expectativas y brindara una experiencia agradable y armoniosa.

A continuación, algunas capturas de cómo se veía el software antes de aplicar la distribución de botones, colores y experiencia de usuario:

Registro

Tipo de persona	Proveedor	nombre completo	<input type="text"/>
Celular	<input type="text"/>	identificacion	<input type="text"/>
Tipo Documento	cc		
Descripcion	<input type="text"/>		
Lista de precios	carton,150.0 plastico,300.0 cobre,25000.0 vidrio,50.0 pvc,500.0 motor de arranque,150000.0		
Material	<input type="text"/>	Valor	Registrar a la lista
Registrar Cancelar			

Fig. 35 Software antes de aplicar experiencia de usuario (1)



Metales de santander							Capital : 500.000 \$
Cuenta Personas Ingresos y egresos Prestamos							Inventario Ayuda
Registrar prestamo		Actualizar Estado Prestamo			Cancelar Prestamo		
idPrestamos	Id_Persona	Id_Tipo_Persona	fecha	Valor	descripcion	Abono	
31	42	2	2023-05-03 22...	2000000.0	el prestamo c...		
32	42	2	2023-05-09 00...	1000000.0	antisipo		
			2023-05-16 11...	200000.0	prueba		

Fig. 36 Software antes de aplicar experiencia de usuario (2)

A continuación, las ventanas del software ya aplicada la experiencia de usuario:

7.2.2.1 Login del sistema

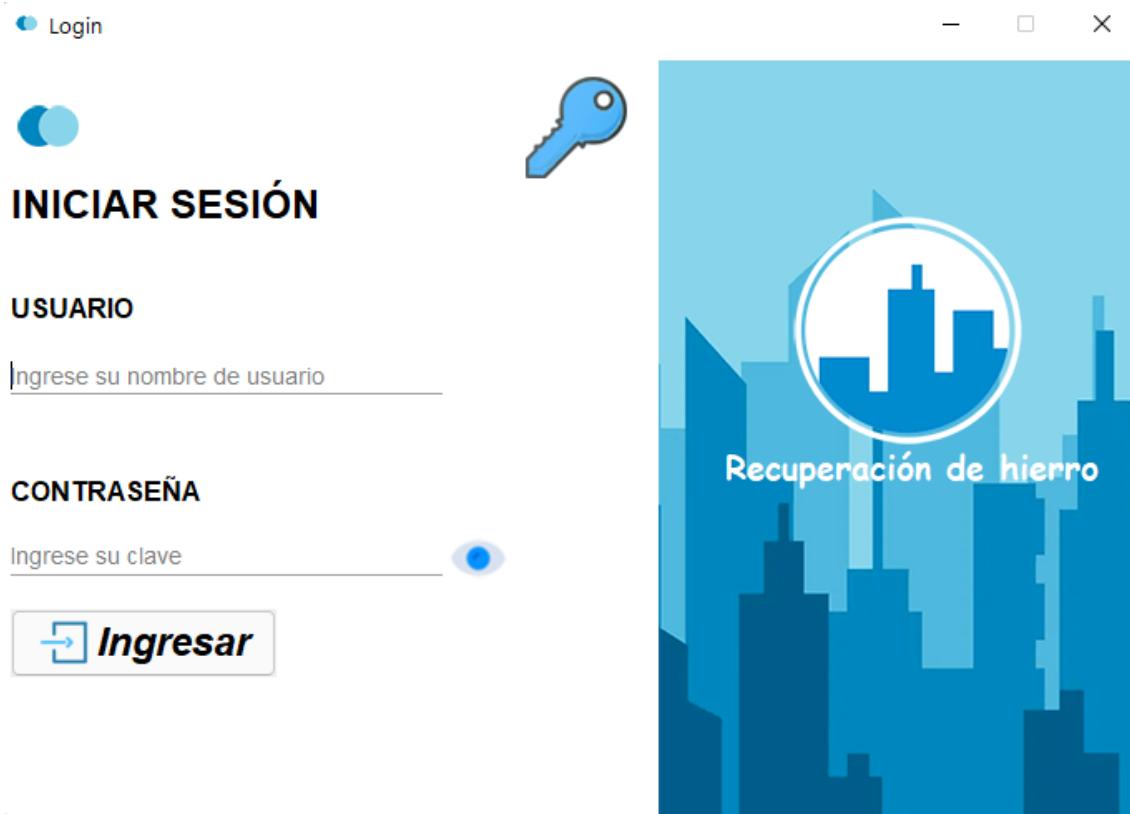


Fig. 37 Login del sistema

7.2.2.2 Registro de usuario

The screenshot shows a user registration form titled "REGISTRO". The form includes fields for "Nombre completo", "Contraseña", "Tipo de documento: cc", "Número de identificación", "Número de celular", "Tipo de persona: Usuario", and a large "Descripción" field containing placeholder text "Por favor, ingrese la descripción". At the bottom are "Registrar" and "Cancelar" buttons.

Registrar usuario

REGISTRO

Nombre completo

Contraseña

Tipo de documento: cc

Número de identificación

Número de celular

Tipo de persona: Usuario

Descripción:

Por favor, ingrese la descripción

Registrar Cancelar

Fig. 38 Registro de usuario

7.2.2.3 Menú-Cuentas

CUENTAS POR COBRAR: 500.000 \$									
Capital: -974000.0									
Id	Fecha	Valor	Cuenta	TipoCuenta	Id	Fecha	Valor	Cuenta	TipoCuenta
117	2023-05-03...	95070.0	...I valor total= 95070.0	compra	137	2023-05-22...	664000.0	... valor total= 664000.0	venta
121	2023-05-03...	11250.0	...I valor total= 11250.0	compra	145	2023-05-23...	310000.0	... 10000 X 1 = -3100.0	venta
122	2023-05-03...	1647450.0	...I valor total= 1647450.0	compra					
123	2023-05-03...	38970.0	...I valor total= 38970.0	compra					
124	2023-05-03...	10314.0	...I valor total= 10314.0	compra					
125	2023-05-03...	1350.0	... el valor total= 1350.0	compra					
126	2023-05-03...	9810.0	... el valor total= 9810.0	compra					
127	2023-05-22...	200.0	... el valor total= 270.0	compra					
128	2023-05-03...	9702.0	... el valor total= 9702.0	compra					
129	2023-05-03...	9180.0	... el valor total= 9180.0	compra					
130	2023-05-03...	10260.0	...I valor total= 10260.0	compra					
131	2023-05-03...	1890.0	... el valor total= 1890.0	compra					
132	2023-05-03...	1440.0	... el valor total= 1440.0	compra					
133	2023-05-03...	1500.0	... el valor total= 1500.0	compra					
134	2023-05-23...	197000.0	... no 740 X 1 = 197000.0	compra					
136	2023-05-08...	120800.0	... valor total= 120800.0	compra					
138	2023-05-16...	11000.0	...I valor total= 11000.0	compra					
139	2023-05-16...	15013.0	...I valor total= 15013.0	compra					
140	2023-05-16...	30500.0	...I valor total= 30500.0	compra					
141	2023-05-16...	665500.0	...I valor total= 665500.0	compra					
143	2023-05-17...	10000.0	...I valor total= 10000.0	compra					

Fig. 39 Menú-Cuentas

7.2.2.4 Nueva cuenta

Fig. 40 Nueva cuenta

7.2.2.5 Menú-Personas



Fig. 41 Menú-Personas

7.2.2.6 Agregar proveedor

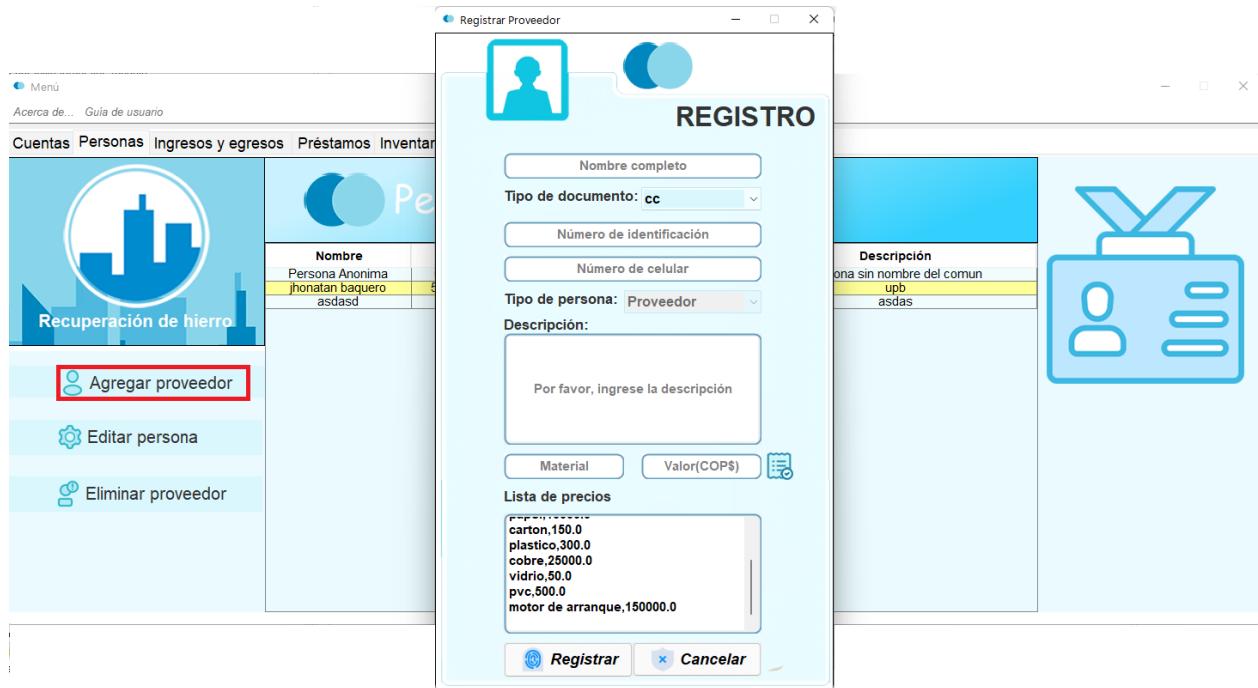


Fig. 42 Agregar proveedor

7.2.2.7 Editar persona

Fig. 43 Editar persona

7.2.2.8 Menú-Ingresos y egresos

id	fecha	tipo_movimiento_id	monto	descripción
1	2023-05-21	1	10000.0	pago pedro
2	2023-05-22	2	10000.0	Donación del gobierno se vendieron 4kg
3	2023-05-22	4	4000.0	
4	2023-05-22	4	2000.0	2k
5	2023-05-22	7	10000.0	rueda de carretilla metal
6	2023-06-28	2	100000.0	metal

Fig. 44 Menú-Ingresos y egresos

7.2.2.9 Agregar tipo movimiento

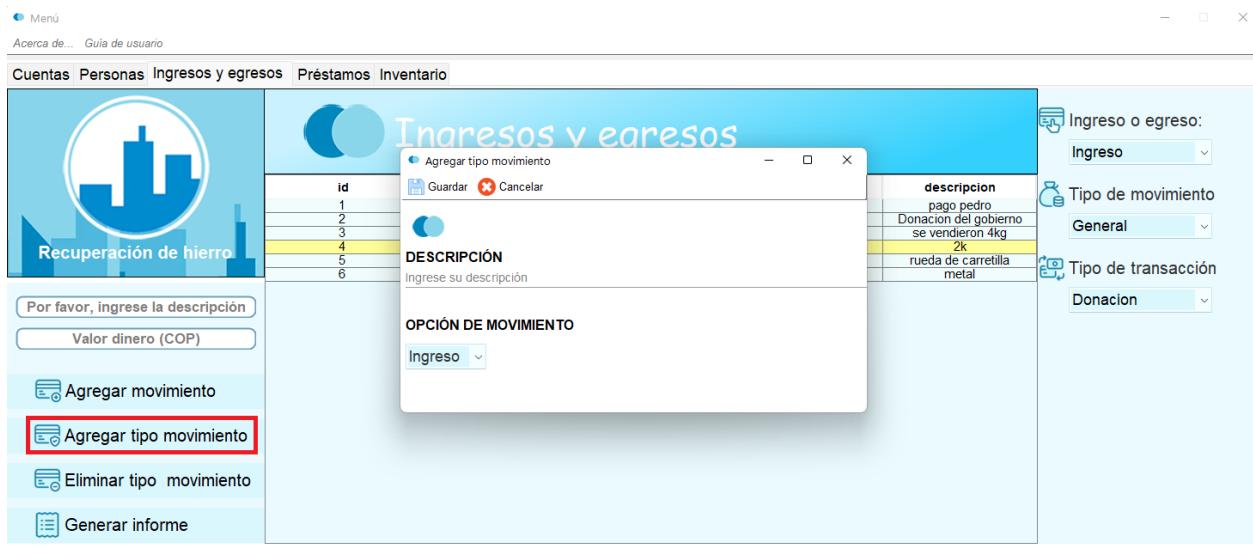


Fig. 45 Agregar tipo movimiento

7.2.2.10 Menú-Préstamos



Fig. 46 Menú-Préstamos

7.2.2.11 Registrar préstamo

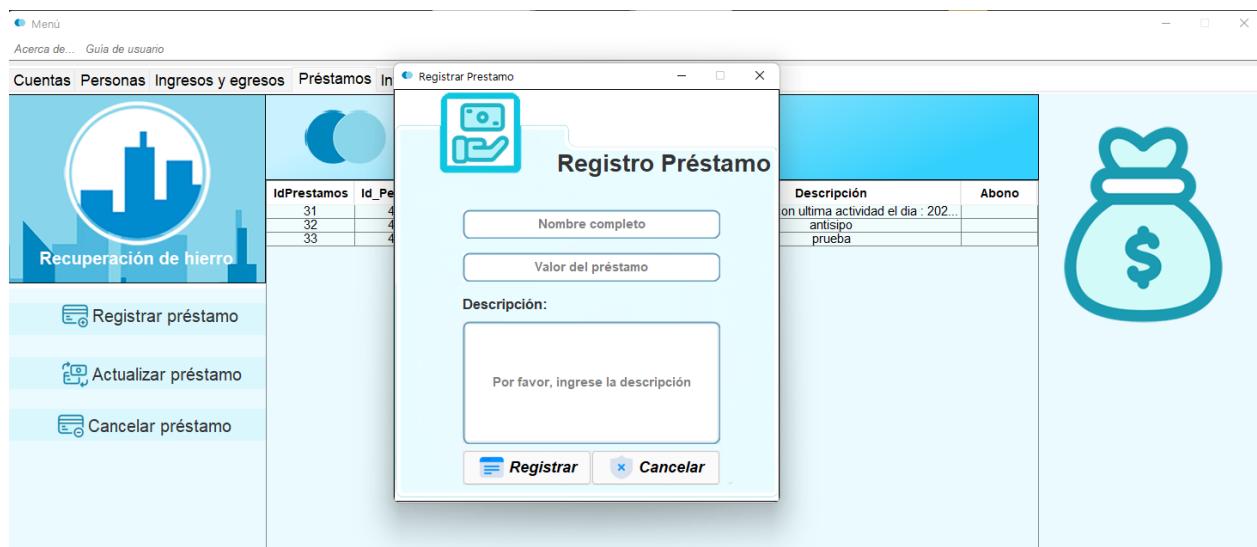


Fig. 47 Registrar préstamo

7.2.2.12 Menú-Inventario



Fig. 48 Menú-Inventario

7.2.3 Recibos y facturas

El aplicativo ha sido diseñado para generar una variedad de recibos y facturas, brindando a los usuarios la capacidad de visualizar y diferentes tipos de documentos, tales como movimientos, comprobantes o datos de inventario. Estas funcionalidades permiten filtrar la información de manera eficiente, proporcionando a los usuarios la flexibilidad de seleccionar y presentar los datos relevantes en los recibos y facturas generadas. Con esto, se busca facilitar la organización y el control de la información financiera y administrativa, agilizando los procesos de registro y seguimiento en el contexto de las empresas recuperadoras de hierro.

7.2.3.1 Informe general de los movimientos

En este informe se hace un análisis de los movimientos realizados por la empresa a lo largo del uso del software, además se muestra el dinero gastado e invertido en diferentes aspectos, pero orientado al gasto de la empresa en algo distinto a las cuentas.

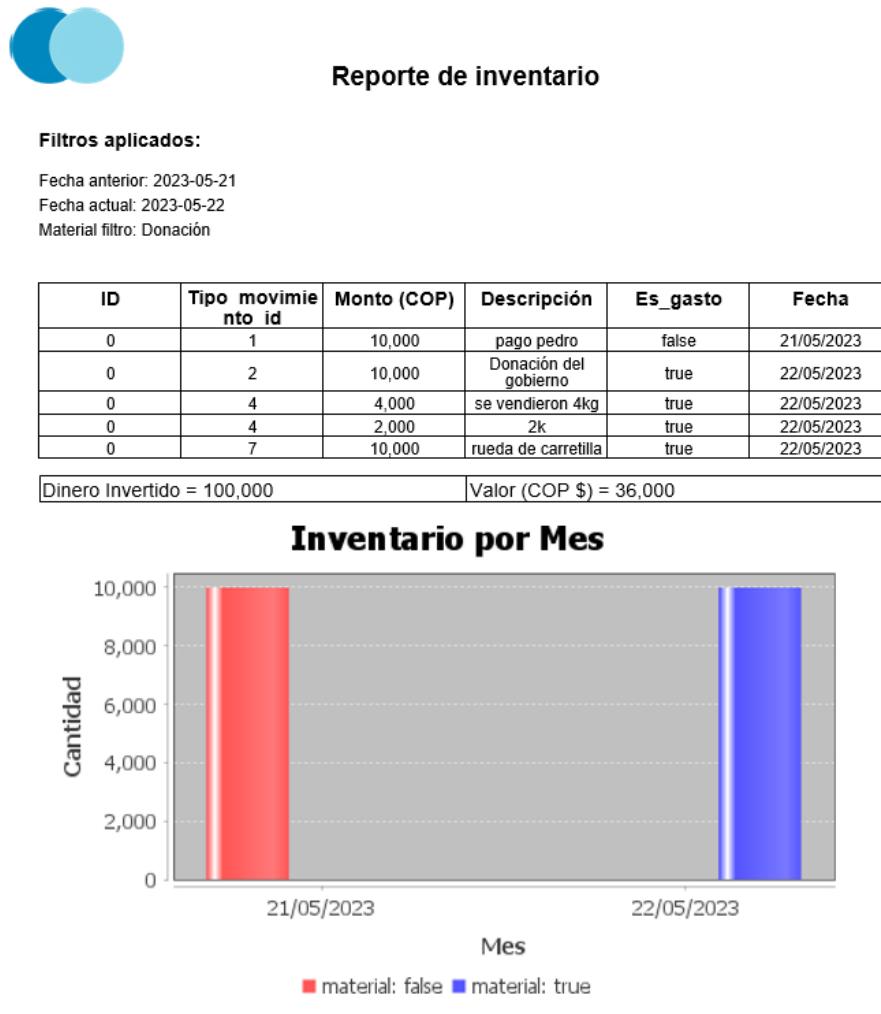


Fig. 49 Informe general de los movimientos



7.2.3.2 Informe general del inventario

Este es un documento con una tabla con todos los datos del inventario registrados desde la fecha inicial del comienzo de los registros, hasta la fecha actual. A través de una tabla y una gráfica esto en grandes rasgos da una perspectiva muy amplia de todo lo que ha hecho la empresa.

Reporte de Inventario

Filtros aplicados:

Fecha anterior: 2023-05-16

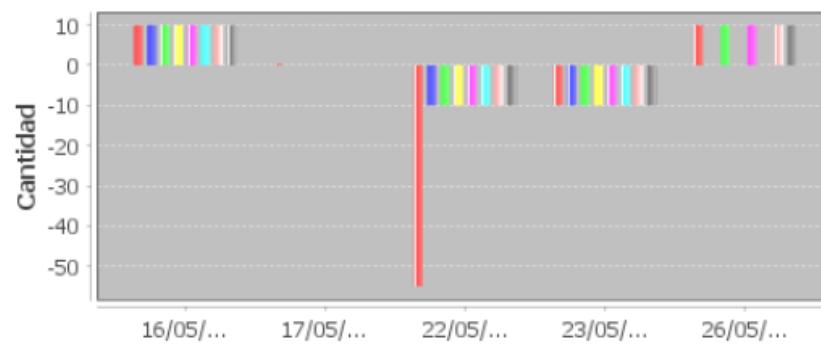
Fecha actual: 2023-05-26

Material filtro: null

ID	Peso (KG)	Número de Cuenta	Material(Cantidad)	Valor (COP)	Fecha
438	-10.0	145	1	\$10,000	23/05/2023
439	-0.5	145	1	\$500	23/05/2023
440	-10.0	145	1	\$10,000	23/05/2023
441	-10.0	145	7	\$300,000	23/05/2023
442	-10.0	145	5	\$1,500	23/05/2023
443	-10.0	145	1	\$10,000	23/05/2023
444	-0.5	145	1	\$500	23/05/2023
445	-10.0	145	1	\$10,000	23/05/2023
446	-10.0	145	7	\$300,000	23/05/2023
447	-10.0	145	5	\$1,500	23/05/2023
448	-10.0	145	1	\$10,000	23/05/2023
449	-0.5	145	1	\$500	23/05/2023
450	-10.0	145	1	\$10,000	23/05/2023
451	-10.0	145	7	\$300,000	23/05/2023
452	-10.0	145	5	\$1,500	23/05/2023
453	-10.0	145	1	\$10,000	23/05/2023
454	-0.5	145	1	\$500	23/05/2023
455	-10.0	145	1	\$10,000	23/05/2023
456	-10.0	145	7	\$300,000	23/05/2023
457	-10.0	145	5	\$1,500	23/05/2023
458	-10.0	145	1	\$10,000	23/05/2023
459	-0.5	145	1	\$500	23/05/2023
460	-10.0	145	1	\$10,000	23/05/2023
461	-10.0	145	7	\$300,000	23/05/2023
462	-10.0	145	5	\$1,500	23/05/2023
464	10.0	146	1	\$-10,000	26/05/2023
465	10.0	146	6	\$-3,000	26/05/2023
466	10.0	146	5	\$-1,500	26/05/2023
467	10.0	146	7	\$-300,000	26/05/2023
468	10.0	146	3	\$-200,000	26/05/2023

Peso (KG) = -1028.7599999999999 Valor (COP) = \$2,986,490

Inventario por Mes



█ material: 1 █ material: 8 █ material: 7 █ material: 2 █ material: 3 █ material: 4
█ material: 5 █ material: 6

Fig. 50 Informe general del inventario

7.2.3.3 Impresión de factura número 1

Esta factura es para clientes de gran envergadura, que tienen que tener mucho más detalle, en donde se informa las pesadas en kilogramos, el nombre del material, el valor del material por kilogramo y el total del costo de la pesada



CONTABILIDAD METALES

Nit : 900405928

FACTURA # 137

Cliente: Persona Anonima

Descripción: persona sin nombre del comun

Fecha: 24 de junio de 2023

Concepto	Cantidad (KG)	Precio (COP)	Costo (COP)
chatarra	10.00	\$1,000	\$10,000
aluminio	10.00	\$5,000	\$50,000
bronce	10.00	\$20,000	\$200,000
papel	10.00	\$10,000	\$100,000
carton	10.00	\$150	\$1,500
plastico	10.00	\$300	\$3,000
cobre	10.00	\$30,000	\$300,000
vidrio	10.00	\$100	\$1,000

Total Anterior: \$665,500

Total: \$664,000

Firma: _____

Fig. 51 Impresión de factura número 1

7.2.3.4 Impresión de factura número 2

Esta factura es para clientes del común que tienen que tener mucho menos detalle, pero aun así está toda la data donde se informa las pesadas en kilogramos, el nombre del material, el valor del material por kilogramo y el total del costo de la pesada. Este tipo de recibo se entrega de manera física.

Contabilidad Metáles

Número de cuenta: Cuenta 126

chatarra 10.0X900=9.000

chatarra 0.3X900=270

chatarra 0.6X900=540

Total cuenta: 540

Fig. 52 Impresión de factura número 2

7.3 Implementación del Sistema de Información

Como parte de las buenas prácticas de desarrollo del software, en el presente proyecto se implementó la creación de varios prototipos los cuales fueron mostrados y compartidos de manera periódica con la tutora, Danith Patricia Solórzano Esconar, con el objetivo de obtener su retroalimentación y validar la dirección del proyecto. Además, el marco de trabajo de “Modelado por Prototipos” permitió obtener una mejor comprensión de los requisitos y expectativas de los usuarios finales, así como identificar posibles mejoras y ajustes antes de la etapa de desarrollo completa.

La colaboración con la tutora garantizó la alineación del proyecto con las necesidades del cliente y contribuyó a la creación de una aplicación final de alta calidad y satisfacción del usuario.

The screenshot shows a list of meeting participants from Microsoft Teams. The participants listed are Juan Sebastian Arguello Lozano and Danith Patricia Solorzano Escobar. Each participant has two entries: one labeled 'Saliente' (Exiting) and one labeled 'Entrante' (Entering). The entries are repeated multiple times, indicating frequent activity between the two users.

Usuario	Tipo de actividad
Juan Sebastian Arguello Lozano	Saliente
Juan Sebastian Arguello Lozano	Entrante
Danith Patricia Solorzano Escobar	Entrante
Danith Patricia Solorzano Escobar	Saliente
Juan Sebastian Arguello Lozano	Entrante
Danith Patricia Solorzano Escobar	Saliente
Juan Sebastian Arguello Lozano	Saliente
Juan Sebastian Arguello Lozano	Entrante

Fig. 53 Historial de reuniones por Microsoft Teams

7.3.1 Buenas prácticas de desarrollo-Modelo-Vista-Controlador

En el aplicativo, se implementaron buenas prácticas de desarrollo de software, específicamente siguiendo el patrón Modelo-Vista-Controlador (MVC), en el código y clases. El uso de este patrón permitió separar las preocupaciones y responsabilidades en diferentes capas, lo cual facilitó la organización, mantenibilidad y reutilización del código. La adopción del modelo MVC proporcionó una estructura clara y modular, promoviendo la escalabilidad y flexibilidad del sistema.

Las siguientes capturas presentan ejemplos de código visto desde el entorno de programación en el que se desarrolló

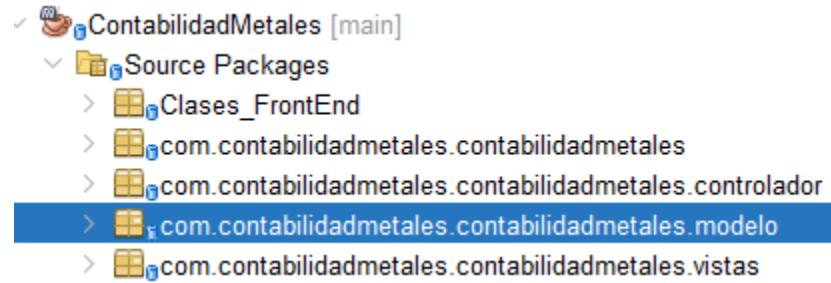


Fig. 54 Buenas prácticas de desarrollo, Modelo-vista-Controlador

7.3.1.1 Modelo

La siguiente captura es un ejemplo de la clase “MovimientoCajaMenor” ubicada en el modelo, se observa como estar ordenados los getters y setters.



Fig. 55 Modelo

```

public class MovimientoCajaMenor {
    private int id;
    private Date fecha;
    private int tipoMovimientoId;
    private double monto;
    private String descripcion;
    private Boolean es_gasto;

    public Boolean getEs_gasto() {
        return es_gasto;
    }

    public void setEs_gasto(Boolean es_gasto) {
        this.es_gasto = es_gasto;
    }
    // Constructor

    public MovimientoCajaMenor(int id, Date fecha, int tipoMovimientoId, double monto, String descripcion, Boolean es_gasto)
    {
        this.fecha = fecha;
        this.tipoMovimientoId = tipoMovimientoId;
        this.monto = monto;
        this.descripcion = descripcion;
        this.es_gasto = es_gasto;
    }
}

```

Fig. 56 Buenas prácticas de desarrollo, Modelo

7.3.1.2 Vista

En el apartado vista, se aprecia cómo están todas las clases que se dedican a mostrar todo el front-end



Fig. 57 Vista



The screenshot shows a software application window. At the top, there are tabs for 'Source', 'Design' (which is selected), and 'History'. Below the tabs is a toolbar with various icons. A message bar at the top right says: 'To select multiple components in an area hold Shift and drag mouse over the components.' There is also a close button 'x'.

In the main area, there is a sidebar on the left containing the following items:

- Agregar a la cuenta
- Editar lista
- Terminar cuenta
- Tipo de usuario (dropdown: Proveedor)
- Usuario (dropdown)
- Material (dropdown)
- Movimiento (dropdown: Compra)
- Acción (dropdown: Cancelado)
- TOTAL: 0.0** (highlighted in red)

The main panel on the right is currently empty.

Fig. 58 Buenas prácticas de desarrollo, Vista

7.3.1.3 Controlador

En el apartado vista, se aprecia cómo están todas las clases que se encargan del controlador



Fig. 59 Controlador

```

public class CLogin {
    private CConexion objetoConexion ;
    public void validaUsuario(String usuario, String contrasenia,JFrame j) {
        try {
            ResultSet rs = null;
            PreparedStatement ps = null;
            objetoConexion= new CConexion();
            String consulta = "select * from personas where personas.nombre =(?) and personas.password=(?) and personas.IdTP=3 ";
            ps = objetoConexion.estableceConexion().prepareStatement(sql: consulta);
            String contra = String.valueOf(obj: contrasenia);
            ps.setString(parameterIndex: 1, x: usuario);
            ps.setString(parameterIndex: 2, x: contra);
            rs = ps.executeQuery();
            if (rs.next()) {
                int i= rs.getInt(columnIndex: 1);
                MetalesDeSantander objetoMenu = new MetalesDeSantander(idUsuario:i);
                objetoMenu.setVisible(b: true);
                // j.setVisible(false);
                j.dispose();
            } else {
                JOptionPane.showMessageDialog(parentComponent: null, message: "El Usuario es INCORRECTO, VUELVA A INTENTAR");
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(parentComponent: null, "ERROR: " + e.toString());
        }
    }
}

```

Fig. 60 Buenas prácticas de desarrollo, Controlador

7.3.2 Buenas prácticas de desarrollo-Base de datos

En la siguiente figura se observa cómo se desarrolló la Gestión de base de datos que fue presentada y en ella se puede apreciar que hay un estándar de programación claro, el código tiene comentarios, los nombres de los métodos coinciden con los mostrados en el diagrama. Cabe aclarar que, aunque en la captura algunos datos como el usuario y clave son de prueba, se censuraron para mayor seguridad.

```
public class CConexion {  
  
    private Connection connection;  
  
    public Connection estableceConexion() {  
  
        try {  
            // Cargar el controlador JDBC  
            Class.forName("com.mysql.jdbc.Driver"); //URL  
  
            // Establecer la conexión con la base de datos  
            String url = "jdbc:mysql://localhost:3306/";  
            String usuario = "root"; //usuario para acceder  
            String contraseña = "1234"; //Clave para acceder  
            + "";//Recuperadoras de hierro1@  
            connection = DriverManager.getConnection(url, user:usuario, password: contraseña); //obtener la conexión  
  
        } catch (ClassNotFoundException | SQLException e) {  
            e.printStackTrace();  
        }  
  
        return connection;  
    }  
}
```

Fig. 61 Buenas prácticas de desarrollo, Base de datos (1)

En la base de datos Mysql, se observa como los datos coinciden con su columna, se puede observar también que hay un estándar de programación claro, los nombres las tablas coinciden con los datos mostrados en la base de datos.

idMateriales	nombre	descripcion	Valor
1	chatarra	NULL	1000
2	aluminio	NULL	5500
3	bronce	NULL	20000
4	papel	NULL	10000
5	carton	NULL	150
6	plastico	NULL	300
7	cobre	NULL	25000
8	vidrio	NULL	50
9	pvc	NULL	500
10	motor de arranque	NULL	150000

Fig. 62 Buenas prácticas de desarrollo, Base de datos (2)

7.3.2.1 Buenas prácticas de desarrollo-Base de datos montada en “DB4free”

Es importante destacar que, como se mencionó en el marco tecnológico, las compañías recuperadoras de hierro que cuentan con acceso a Internet, tienen la posibilidad de utilizar el servicio de base de datos remoto alojado en la nube, el cual está disponible en la plataforma "DB4free", la cual brinda la ventaja de acceder y gestionar la base de datos de forma remota, sin la necesidad de mantener una infraestructura local.

The screenshot shows the phpMyAdmin interface for the 'metales_db' database. The left sidebar lists databases: Nueva, db4free_sys, information_schema, metales_db (selected), and performance_schema. The main area shows a table list with 11 tables: caja_menor, cuentas, inventario, materiales, personas, prestamos, seguimientoinventario, tipocuenta, tipodocumento, tipopersona, and tipos_movimientos. Each table row includes icons for Examinar (View), Estructura (Structure), Buscar (Search), Insertar (Insert), Vaciar (Empty), and Eliminar (Delete). The table has columns for Tabla, Acción, Filas, Tipo, Cotejamiento, Tamaño, and Residuo a depurar. A red box highlights the entire table list area. At the bottom, it says '11 tablas' and 'Número de filas' with values '200' and '320.0 KB'. Below the table, there are buttons for 'Seleccionar todo' (Select all) and 'Para los elementos que están marcados:' (For the selected elements).

Fig. 63 Base de datos montada en "DB4free"

7.3.3 Copia de seguridad

Para la copia de seguridad se utilizaron bibliotecas de programación de tareas como Quartz y "Spring Scheduler". Se implementó una función automatizada que ejecuta regularmente el comando "mysqldump" en intervalos de tiempo predefinidos con el fin de realizar copias de seguridad de la base de datos.

Se configuró el programador de tareas "Quartz" el cual automatiza la tarea de copia de seguridad en un intervalo de tiempo de una vez por semana, específicamente miércoles a las 12 de la noche, el cual es el día menos transcurrido en las empresas recuperadoras de hierro; dicha copia de seguridad es posible guardarla en otra ubicación u incluso en una diferente partición o equipo.

Tanto para la base de datos local como la remota en "DB4free" funcionan las mismas bibliotecas de programación de tareas.

En pocas palabras el software genera la copia de seguridad por medio del Mysql online, el cual se crea desde el código java, ahí mismo en el equipo donde está alojado el software. Es decir, tanto para desktop como para web, el aplicativo hace una consulta de los datos y pone la copia de seguridad en el respectivo equipo donde se aloja el software.

```

--- resources:3.3.0:resources (default-resources) @ ContabilidadMetales ---
skip non existing resourceDirectory C:\Users\narut\Downloads\ContabilidadMetales\src\main\resources

--- compiler:3.10.1:compile (default-compile) @ ContabilidadMetales ---
Changes detected - recompiling the module!
Compiling 37 source files to C:\Users\narut\Downloads\ContabilidadMetales\target\classes
/C:/Users/narut/Downloads/ContabilidadMetales/src/main/java/com/contabilidadmetales/contabilidadmetales/vistas/Cuentas.java: Some input files use or override a deprecated API.
/C:/Users/narut/Downloads/ContabilidadMetales/src/main/java/com/contabilidadmetales/contabilidadmetales/vistas/Cuentas.java: Recompile with -Xlint:deprecation for details.
/C:/Users/narut/Downloads/ContabilidadMetales/src/main/java/com/contabilidadmetales/contabilidadmetales/vistas/Cuentas.java: Some input files use unchecked or unsafe operations.
- /C:/Users/narut/Downloads/ContabilidadMetales/src/main/java/com/contabilidadmetales/contabilidadmetales/vistas/Cuentas.java: Recompile with -Xlint:unchecked for details.

--- exec:3.1.0:exec (default-cli) @ ContabilidadMetales ---
15:08:10.728 [main] INFO org.quartz.impl.StdSchedulerFactory - Using default implementation for ThreadExecutor
15:08:10.732 [main] INFO org.quartz.simpl.SimpleThreadPool - Job execution threads will use class loader of thread: main
15:08:10.743 [main] INFO org.quartz.core.SchedulerSignalerImpl - Initialized Scheduler Signaller of type: class org.quartz.core.SchedulerSignalerImpl
15:08:10.743 [main] INFO org.quartz.core.QuartzScheduler - Quartz Scheduler v.2.3.2 created.
15:08:10.744 [main] INFO org.quartz.simpl.RAMJobStore - RAMJobStore initialized.
15:08:10.745 [main] INFO org.quartz.core.QuartzScheduler - Scheduler meta-data: Quartz Scheduler (v2.3.2) 'DefaultQuartzScheduler' with instanceId 'NON_CLUSTERED'.
Scheduler class: 'org.quartz.core.QuartzScheduler' - running locally.
NOT STARTED.
Currently in standby mode.
Number of jobs executed: 0
Using thread pool 'org.quartz.simpl.SimpleThreadPool' - with 10 threads.
Using job-store 'org.quartz.simpl.RAMJobStore' - which does not support persistence, and is not clustered.

15:08:10.745 [main] INFO org.quartz.impl.StdSchedulerFactory - Quartz scheduler 'DefaultQuartzScheduler' initialized from default resource file in Quartz package: 'quartz.properties'
15:08:10.745 [main] INFO org.quartz.impl.StdSchedulerFactory - Quartz scheduler version: 2.3.2
15:08:10.757 [main] INFO org.quartz.core.QuartzScheduler - Scheduler DefaultQuartzScheduler_6_NON_CLUSTERED started.
15:08:10.758 [DefaultQuartzScheduler_QuartzSchedulerThread] DEBUG org.quartz.core.QuartzSchedulerThread - batch acquisition of 0 triggers

```

Fig. 64 Programador de tareas “Quartz”

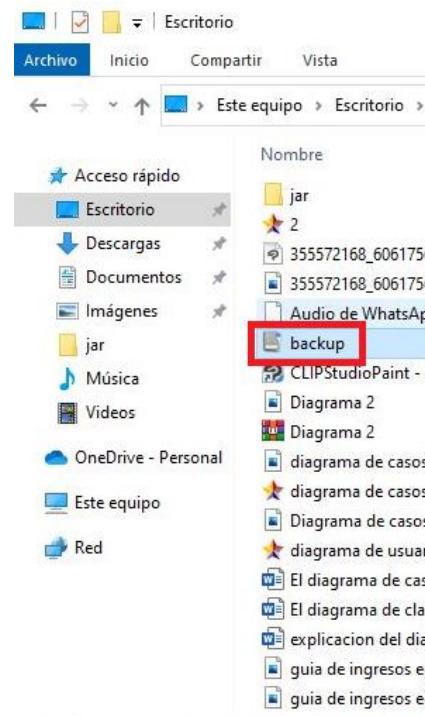


Fig. 65 Copia de seguridad generada

En resumen, se implementó el aplicativo propuesto siguiendo buenas prácticas de desarrollo de software y despliegue de aplicaciones. Se aplicaron metodologías como el modelo-vista-controlador (MVC) para organizar y modularizar el código, lo cual permitió obtener un algoritmo limpio y mantenible. Además, hubo enfoque en la documentación de la aplicación, asegurando que estuvieran disponibles manuales y guías para facilitar su comprensión y uso por parte de los usuarios. También se consideraron criterios de seguridad en el manejo de los datos de la organización, garantizando la privacidad y confidencialidad de la información.



Fig. 66 Software implementado

7.4 Validación del Funcionamiento del Sistema de Información

En este apartado se llevó a cabo la evaluación exhaustiva del sistema desarrollado para asegurar su correcto funcionamiento y cumplimiento de los requisitos establecidos. Para ello, se han realizado pruebas unitarias que examinan y verifican el funcionamiento de cada componente individual del sistema, garantizando su correcta implementación. Además, se han llevado a cabo pruebas de aceptación de usuario, donde los usuarios finales han probado y validado el sistema en situaciones reales, brindando su feedback. Estas pruebas han sido fundamentales para comprobar que el sistema cumple con las expectativas y necesidades de los usuarios, asegurando así su calidad y efectividad en la entrega de los resultados esperados.

7.4.1 Pruebas Unitarias

Antes de realizar las pruebas de campo, se hicieron test unitarios utilizando NetBeans 18 para asegurarse de que cada componente o unidad de código funcione correctamente de forma individual. Estas pruebas permiten detectar y corregir errores y fallos en el código de manera temprana, antes de integrar los diferentes componentes y módulos del software [33].

En el presente proyecto, se consideró las pruebas unitarias como una práctica adecuada por varias razones. En primer lugar, permiten verificar el funcionamiento correcto de cada componente de forma aislada, detectando y corrigiendo errores tempranamente. Además, mejoran la mantenibilidad del software al actuar como salvaguardas para realizar cambios sin romper funcionalidades existentes. Asimismo, fomentan la colaboración y la calidad del código al facilitar la comunicación y promover buenas prácticas de codificación [33].

Se realizaron pruebas unitarias a cada uno de los paquetes del software los cuales contienen todas las clases, dando como resultado un test de 100% pasado.

7.4.1.1 Prueba Unitaria Modelo

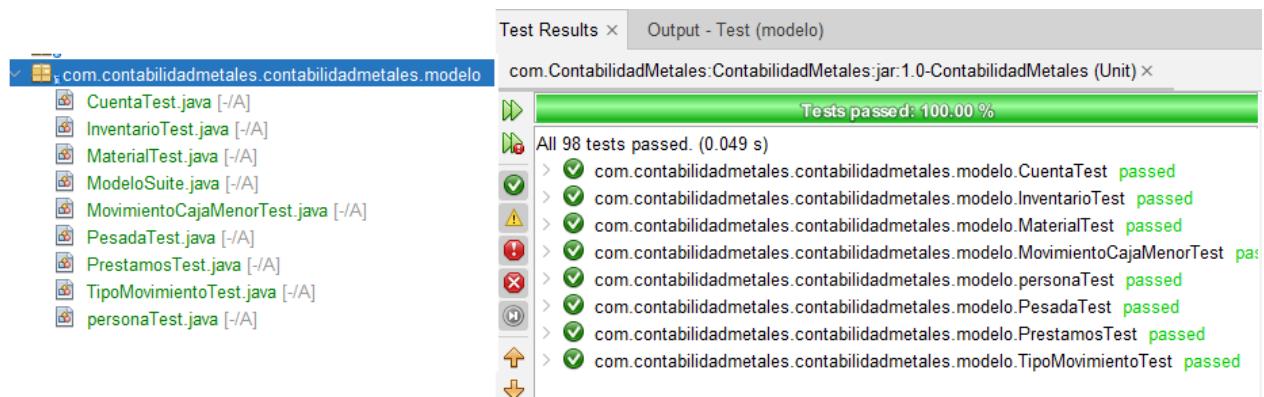


Fig. 67 Prueba Unitaria Modelo

7.4.1.2 Prueba Unitaria Vista

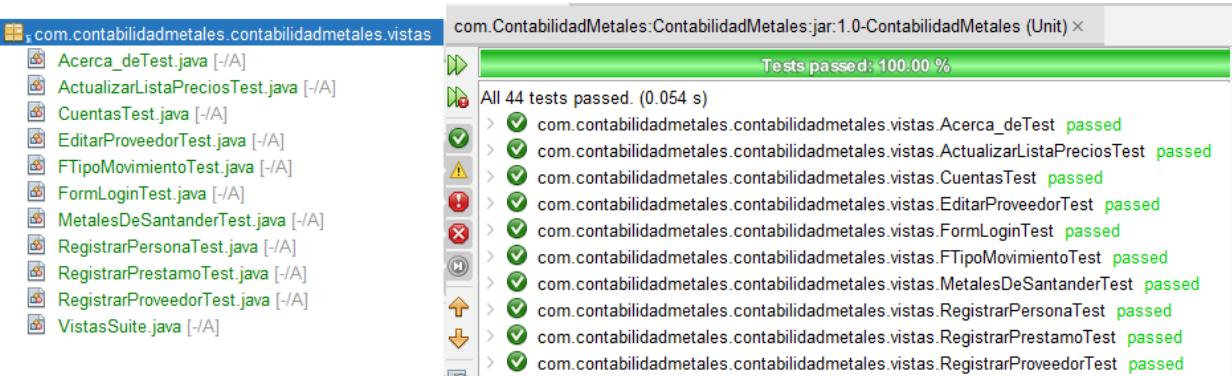
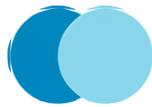


Fig. 68 Prueba Unitaria Vista



7.4.1.3 Prueba Unitaria Controlador

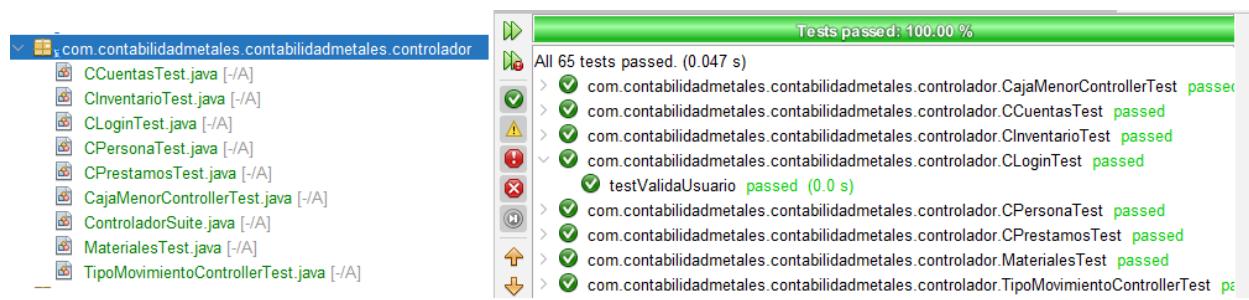


Fig. 69 Prueba Unitaria Controlador

7.4.1.4 Prueba Unitaria ContabilidadMetales

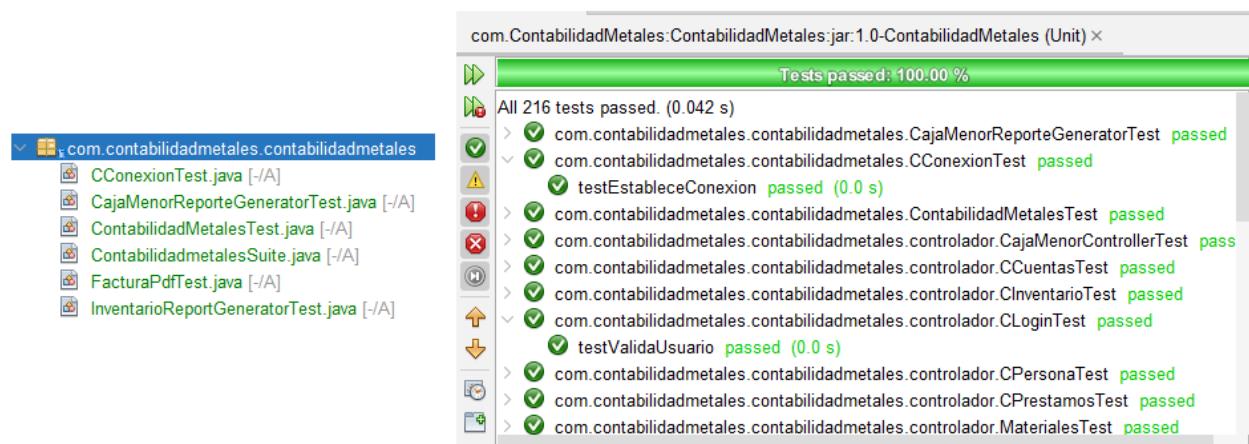


Fig. 70 Prueba Unitaria ContabilidadMetales

7.4.1.5 Prueba Unitaria Clases_FrontEnd

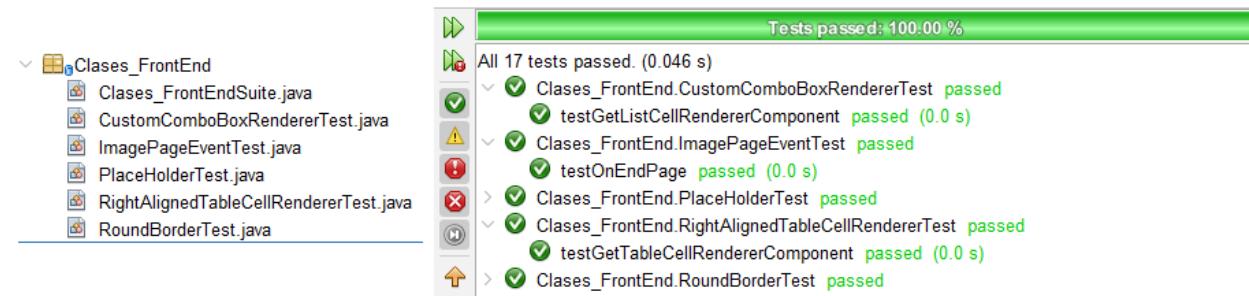


Fig. 71 Prueba Unitaria Clases_FrontEnd

7.4.2 Pruebas de aceptación de usuario

Ya con las pruebas unitarias, se procede a realizar las pruebas de campo del aplicativo en dos empresas recuperadoras de hierro, se llevó a cabo visitas a las mismas dos compañías que se realizaron entrevistas, para así permitir a los usuarios probar personalmente el software y obtener su criterio de aceptación. Durante estas pruebas, se realizaron diversas actividades como la ejecución de escenarios de uso y la validación de funcionalidades clave. Estas pruebas permitieron evaluar la funcionalidad y pertinencia de la aplicación en un entorno real, así como identificar posibles mejoras y ajustes necesarios.

Se consideró hacer test de aceptación, ya que este es una técnica utilizada en el desarrollo de software para evaluar si un sistema cumple con los requisitos y expectativas establecidos por los usuarios finales. Consiste en simular situaciones reales de uso y verificar si el software se comporta de manera adecuada y brinda los resultados esperados.

Los resultados obtenidos se documentaron cuidadosamente para analizar y tener en cuenta en el proceso de mejora continua del aplicativo. Esta validación en campo fue fundamental para garantizar que el software cumpliera con las necesidades y expectativas de los usuarios finales en el contexto específico de las empresas recuperadoras de hierro.

Es de aclarar que los usuarios que probaron el aplicativo, quisieron mantener su anonimato como en las entrevistas, así ocultando su nombre, cedula, firma, siendo muy reservados en sus comentarios

De igual forma se diseñó un manual de usuario para ayudar al cliente final sobre la usabilidad del software:

<https://github.com/JonathanBaquero01/ContabilidadMetales/blob/main/Recursos%20tesis/Manual%20de%20usuario/Manual%20de%20usuario.pdf>



7.4.2.1 Pruebas de aceptación “Metales de Santander”

La compañía “Metales de Santander” al ser de bajos recursos no cuenta con una buena iluminación en su sitio de escrito por lo que las imágenes se ven oscuras, sin el usuario probar el software satisfactoriamente



Fig. 72 Pruebas de aceptación “Metales de Santander” (1)

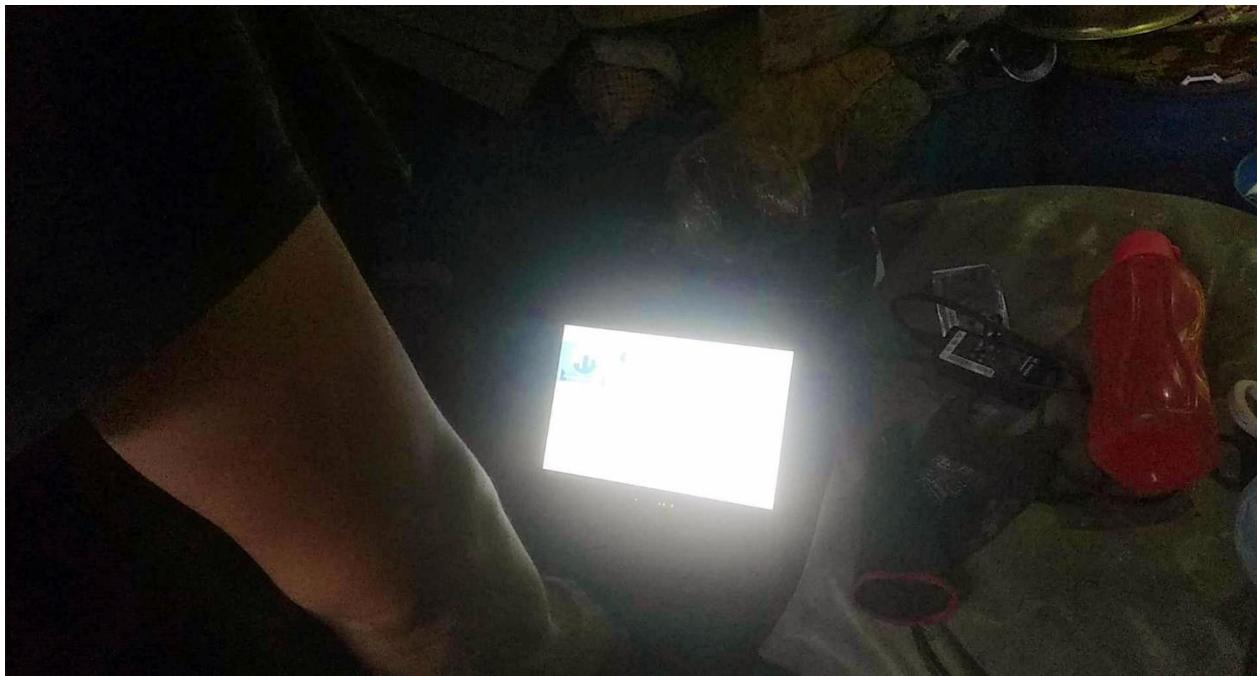


Fig. 73 Pruebas de aceptación “Metales de Santander” (2)

7.4.2.1.1 Registro de características de pruebas

Como las pruebas se basan principalmente en el correcto funcionamiento, se registraron las principales características que se consideraron importantes.

Prueba de aceptación de usuario("Metales de Santander")			
Funcionalidad	Entorno de prueba	Resultado Prueba	Observaciones
Registro de usuario	Empresa "Metales de Santander"	Aprobado	Ninguna
Login	Empresa "Metales de Santander"	Aprobado	Ninguna
Abonar	Empresa "Metales de Santander"	Aprobado	Ninguna
Agregar Proveedor	Empresa "Metales de Santander"	Aprobado	Ninguna
Agregar movimiento	Empresa "Metales de Santander"	Aprobado	Ninguna
Eliminar movimiento	Empresa "Metales de Santander"	Aprobado	Ninguna
Registrar préstamo	Empresa "Metales de Santander"	Aprobado	Ninguna
Aplicar filtro de inventario	Empresa "Metales de Santander"	Aprobado	Ninguna
Generar informe	Empresa "Metales de Santander"	Aprobado	Ninguna
Generar factura	Empresa "Metales de Santander"	Aprobado	Ninguna

Tabla. 29 Prueba de aceptación de usuario("Metales de Santander")

7.4.2.1.2 Cuestionario

Ya registradas las características de las pruebas, se realizó un cuestionario para evaluar las siguientes características: Corrección, Fiabilidad, Eficiencia, Integridad, Facilidad de Uso y Facilidad de Mantenimiento.

1= *Inaceptable*

2= *Bajo el promedio*

3= *Promedio*

4= *Bueno*

5= *Excelente*

Cuestionario		
Característica	Pregunta	Evaluación
Corrección	¿El sistema proporciona los resultados esperados y cumple con las especificaciones establecidas?	5
Fiabilidad	¿El sistema funciona de manera consistente y confiable, sin fallas o errores inesperados?	5
Eficiencia	¿El sistema responde de manera rápida y realiza las tareas de forma eficiente, sin consumir recursos innecesarios?	4
Integridad	¿El sistema garantiza la precisión y la integridad de los datos, evitando la corrupción o pérdida de información?	5
Facilidad de Uso	¿El sistema es intuitivo y fácil de usar, permitiendo a los usuarios navegar y utilizar sus funciones de manera sencilla y sin dificultad?	5
Facilidad de Mantenimiento	¿Cree que el sistema está diseñado de estructurada, facilitando la realización de cambios o actualizaciones sin afectar su funcionamiento global?	5

Tabla. 30 Cuestionario "Metales de Santander"

7.4.2.1.3 Grafico

En apoyo a los resultados obtenidos, se incluye un gráfico que demuestra que el índice de excelencia alcanzado por el software superó ampliamente el 80%. Este gráfico detalla de forma visual el rendimiento de las pruebas de software, mostrando el progreso continuo y la mejora constante en términos de calidad y eficacia. Estos datos respaldan de manera sólida la afirmación de que el software ha cumplido y superado con creces los estándares de excelencia establecidos, consolidándose como una solución de vanguardia en su campo.

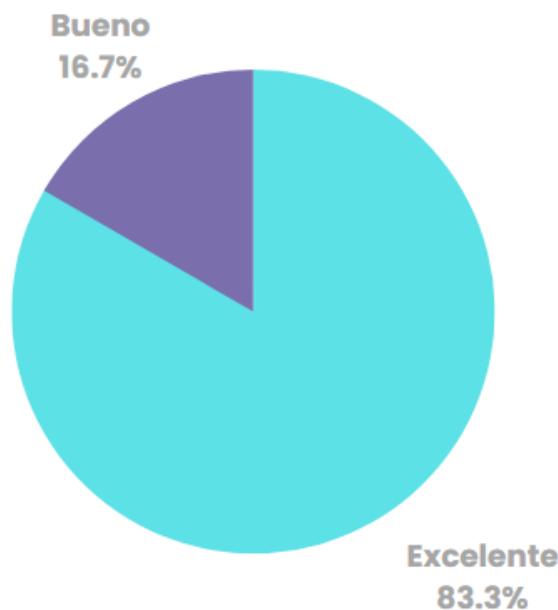


Fig. 74 Grafico cuestionario "Metales de Santander"

7.4.2.1.4 Conclusiones

- ✓ El software ha alcanzado un índice de excelencia superior al 80%, lo que demuestra el nivel de calidad y cumplimiento de los requisitos establecidos.
- ✓ Los usuarios han expresado una satisfacción positiva hacia el aplicativo, destacando su usabilidad, funcionalidad y capacidad de respuesta a sus necesidades.
- ✓ Durante las pruebas realizadas, no se detectaron errores significativos o no deseados en el funcionamiento del software, lo cual refleja un buen nivel de estabilidad y robustez.

- 
- ✓ Los usuarios han aceptado de manera favorable el software, evidenciando su disposición para adoptarlo como una herramienta útil en su trabajo diario.
 - ✓ La realización de pruebas de campo en las empresas recuperadoras de hierro ha permitido validar la funcionalidad y pertinencia del aplicativo en un entorno real, fortaleciendo su confiabilidad y relevancia para el sector.

En resumen, el software ha logrado alcanzar altos estándares de calidad, ha sido bien recibido por los usuarios y ha demostrado su capacidad para satisfacer las necesidades del negocio de las empresas recuperadoras de hierro.

7.4.2.1.5 Recomendaciones

- Ampliar la cobertura de las pruebas realizadas en las empresas recuperadoras de hierro, incluyendo escenarios y casos de uso adicionales que reflejen de manera más precisa las operaciones y procesos específicos de estas empresas.
- Realizar pruebas de rendimiento adicionales, simulando cargas de trabajo intensivas y situaciones de alto tráfico para evaluar la capacidad y la respuesta del software bajo condiciones de uso extremas. Esto permitirá identificar posibles cuellos de botella o áreas de mejora en términos de eficiencia y rendimiento.
- Establecer un plan de mantenimiento regular para el software, que incluya la revisión periódica de su funcionamiento, actualización de componentes y corrección de posibles fallos. Esto garantizará la estabilidad y confiabilidad a largo plazo del software en el entorno operativo de las empresas recuperadoras de hierro.

7.4.2.2 Pruebas de aceptación “Ekol-Reciclables”

Al igual que “Metales de Santander”, “Ekol-Reciclables” es de bajos recursos, sin embargo, se contó con una mejor iluminación.

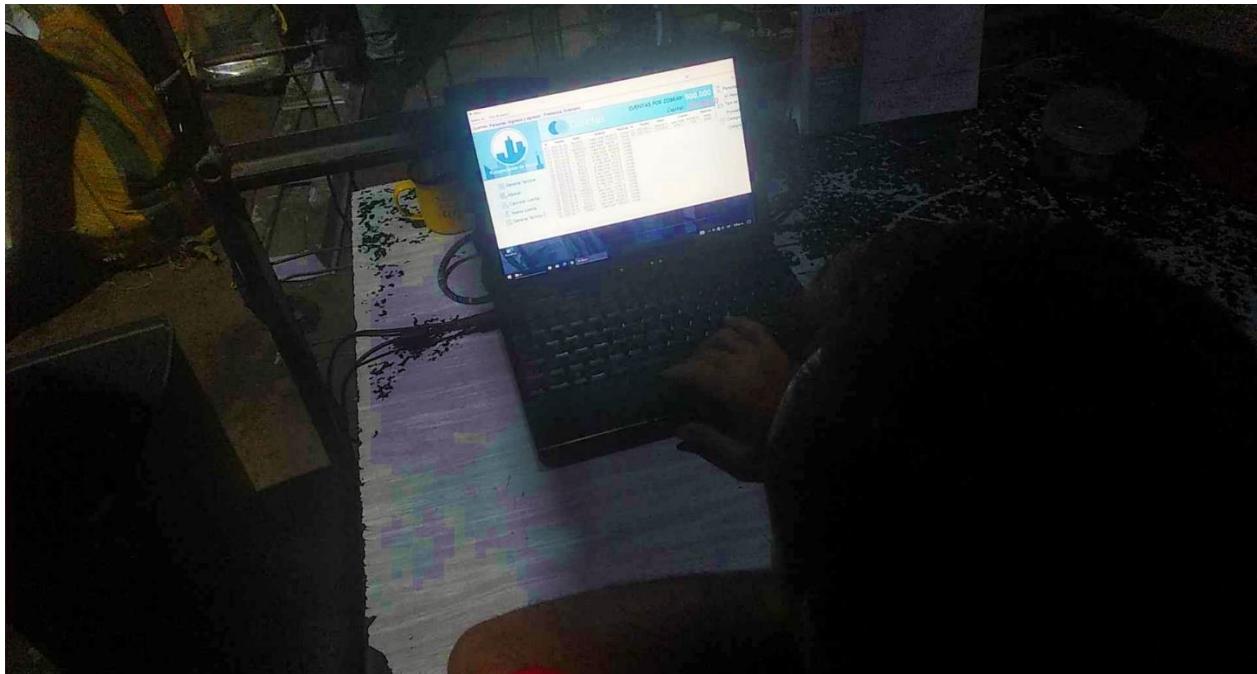


Fig. 75 Pruebas de aceptación “Ekol-Reciclables” (1)

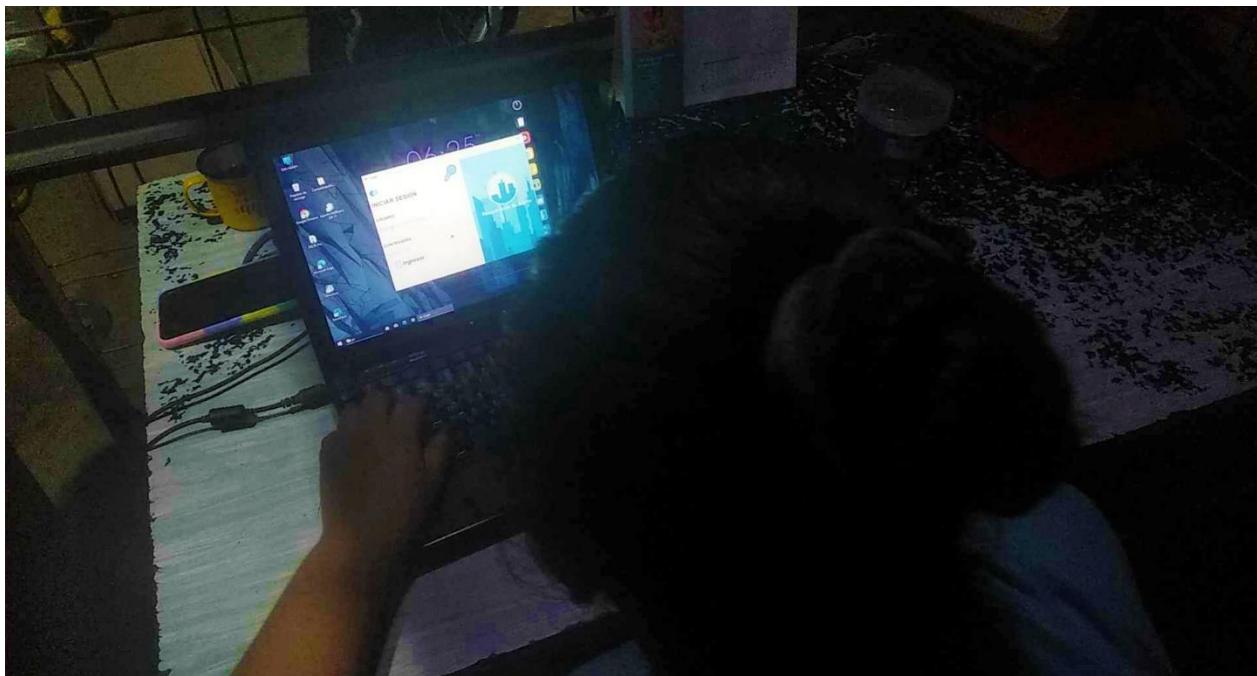


Fig. 76 Pruebas de aceptación “Ekol-Reciclables” (2)

7.4.2.2.1 Registro de características de pruebas

De igual manera en se registraron las principales características que se consideraron importantes.

Prueba de aceptación de usuario("Ekol-Reciclables")			
Encargado : Jonathan Baquero Rodríguez		Fecha: 26-Junio-2023	
Funcionalidad	Entorno de prueba	Resultado Prueba	Observaciones
Registro de usuario	Empresa "Ekol-Reciclables"	Aprobado	Ninguna
Login	Empresa "Ekol-Reciclables"	Aprobado	Ninguna
Abonar	Empresa "Ekol-Reciclables"	Aprobado	Ninguna
Agregar Proveedor	Empresa "Ekol-Reciclables"	Aprobado	Ninguna
Agregar movimiento	Empresa "Ekol-Reciclables"	Aprobado	Ninguna
Eliminar movimiento	Empresa "Ekol-Reciclables"	Aprobado	Ninguna
Registrar préstamo	Empresa "Ekol-Reciclables"	Aprobado	Ninguna
Aplicar filtro de inventario	Empresa "Ekol-Reciclables"	Aprobado	Ninguna
Generar informe	Empresa "Ekol-Reciclables"	Aprobado	Ninguna
Generar factura	Empresa "Ekol-Reciclables"	Aprobado	Ninguna

Tabla. 31 Prueba de aceptación de usuario("Ekol-Reciclables")

7.4.2.2.2 Cuestionario

Ya registradas las características de las pruebas, se realizó un cuestionario para evaluar las siguientes características: Corrección, Fiabilidad, Eficiencia, Integridad, Facilidad de Uso y Facilidad de Mantenimiento.

1= *Inaceptable*

2= *Bajo el promedio*

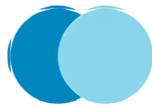
3= *Promedio*

4= *Bueno*

5= *Excelente*

Cuestionario		
Característica	Pregunta	Evaluación
Corrección	¿El sistema proporciona los resultados esperados y cumple con las especificaciones establecidas?	5
Fiabilidad	¿El sistema funciona de manera consistente y confiable, sin fallas o errores inesperados?	5
Eficiencia	¿El sistema responde de manera rápida y realiza las tareas de forma eficiente, sin consumir recursos innecesarios?	5
Integridad	¿El sistema garantiza la precisión y la integridad de los datos, evitando la corrupción o pérdida de información?	5
Facilidad de Uso	¿El sistema es intuitivo y fácil de usar, permitiendo a los usuarios navegar y utilizar sus funciones de manera sencilla y sin dificultad?	5
Facilidad de Mantenimiento	¿Cree que el sistema está diseñado de estructurada, facilitando la realización de cambios o actualizaciones sin afectar su funcionamiento global?	4

Tabla. 32 Cuestionario "Ekol-Reciclables"



7.4.2.2.3 Grafico

De igual forma que en “Metales de Santander,” se incluye un gráfico que demuestra que el índice de excelencia alcanzado por el software superó ampliamente el 80%. Estos datos respaldan de manera sólida la afirmación de que el software ha cumplido y superado con creces los estándares de excelencia establecidos, consolidándose como una solución de vanguardia en su campo.

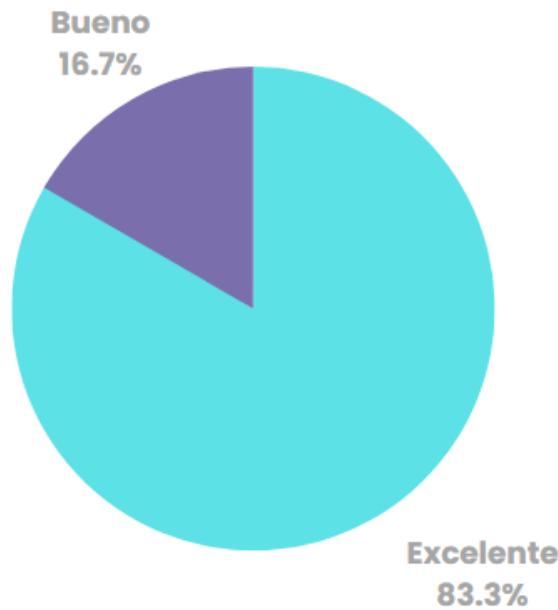


Fig. 77 Grafico cuestionario "Ekol-Reciclables"

7.4.2.2.4 Conclusiones

- ✓ La implementación exitosa del software en “Ekol-Reciclables” ha demostrado su adaptabilidad y utilidad en un entorno real, brindando soluciones efectivas a los desafíos y procesos específicos de la industria.
- ✓ El alto nivel de aceptación por parte de los usuarios refuerza la confianza en el software como una herramienta efectiva para optimizar la gestión y el seguimiento de las operaciones relacionadas con la recuperación de hierro.
- ✓ Durante las pruebas realizadas, se ha verificado la corrección y fiabilidad del software, al no detectarse errores significativos que afecten su desempeño y operatividad en el entorno empresarial.

- 
- ✓ La implementación de pruebas de aceptación de usuario ha permitido validar la funcionalidad del software de acuerdo con las expectativas y necesidades de los usuarios finales, asegurando así su satisfacción y adopción exitosa.
 - ✓ En general, el software ha demostrado ser una herramienta integral y confiable para la gestión eficiente de las actividades relacionadas con la recuperación de hierro, contribuyendo al aumento de la productividad y la optimización de los procesos en las empresas del sector.

Tanto las conclusiones de “Metales de Santander” y “Ekol-Reciclables” respaldan la efectividad y pertinencia del software desarrollado para las empresas recuperadoras de hierro, reafirmando su valor como una solución tecnológica que impulsa el crecimiento y la eficiencia en la industria.

7.4.2.2.5 Recomendaciones

- Realizar pruebas de compatibilidad con diferentes dispositivos y sistemas operativos utilizados por las empresas recuperadoras de hierro. Esto asegurará que el software funcione correctamente en los diferentes entornos tecnológicos utilizados por los usuarios, brindando una experiencia consistente y sin problemas en todas las plataformas.
- Realizar pruebas de escalabilidad para evaluar cómo el software responde y se adapta al crecimiento de la empresa y al aumento en la cantidad de datos manejados. Esto permitirá garantizar que el sistema pueda manejar eficientemente un mayor volumen de transacciones y usuarios sin comprometer su rendimiento.
- Realizar pruebas de actualización y migración para evaluar la compatibilidad y la correcta migración de datos al realizar actualizaciones del software. Esto permitirá asegurar que las empresas recuperadoras de hierro puedan beneficiarse de las mejoras y nuevas funcionalidades sin interrupciones ni pérdida de datos.



8. Conclusiones finales

1. Se cumplieron las metas y objetivos establecidos satisfactoriamente en el presente proyecto. A lo largo del proceso, el equipo de trabajo demostró un compromiso sólido y una dedicación constante para lograr los resultados esperados.
2. Se definieron los requerimientos funcionales y no funcionales correspondientes al aplicativo, determinando las necesidades de las empresas recuperadoras de hierro.
3. Se logró llevar a cabo la implementación de los diagramas propuestos, los cuales se diseñaron de acuerdo con los requerimientos establecidos. Estos diagramas sirvieron como base para la creación del aplicativo, siguiendo los lineamientos y estándares previamente establecidos.
4. Se implementó el aplicativo propuesto, siguiendo buenas prácticas de desarrollo de software y despliegue de aplicaciones. Se aplicaron criterios de seguridad en el manejo de los datos de la organización, garantizando la integridad y confidencialidad de la información.
5. Se realizaron pruebas de campo del aplicativo en dos empresas recuperadoras de hierro, con el objetivo de validar su funcionalidad y pertinencia en un entorno real. Estas pruebas permitieron evaluar el rendimiento del aplicativo y recopilar información relevante para documentar los resultados obtenidos.
6. Según la opinión de las personas que probaron el software en "Metales de Santander" y "Ekol-Reciclables" el aplicativo cumplió con las metas estipuladas.
7. La elección de utilizar herramientas de uso libre fue una decisión estratégica que demostró ser altamente beneficiosa. Al adoptar soluciones de software de código abierto, se ha logrado mantener un bajo presupuesto sin comprometer la calidad y funcionalidad del sistema

En conclusión, la presente tesis representó un destacado logro en el ámbito de la aplicación de tecnologías en las empresas recuperadoras de hierro. A lo largo de este proyecto, se llevó a cabo una exhaustiva investigación, desarrollo e implementación de un software especializado, diseñado para facilitar y optimizar las operaciones y procesos de estas organizaciones.



Durante el proceso de desarrollo del software, se aplicó un enfoque riguroso basado en buenas prácticas de desarrollo de software, como el modelo-vista-controlador (MVC), lo que permitió la creación de un sistema robusto, modular y fácilmente escalable. Además, se implementaron medidas de seguridad adecuadas para garantizar la protección de los datos sensibles de las empresas y usuarios.

Para validar la funcionalidad y pertinencia del aplicativo, se realizaron pruebas de campo en dos empresas recuperadoras de hierro, lo que permitió obtener valiosos comentarios y retroalimentación de los usuarios. Estas pruebas, junto con las pruebas unitarias realizadas durante el desarrollo, contribuyeron a la calidad y confiabilidad del software, asegurando que cumple con los requisitos establecidos y las necesidades de las empresas.

La implementación del software en las empresas recuperadoras de hierro demostró ser exitosa, con una aceptación positiva por parte de los usuarios. Se logró facilitar la gestión de información, agilizar los procesos internos y mejorar la eficiencia operativa de estas organizaciones. Además, el software contribuyó a la reducción de errores y la optimización de recursos, lo que ha impactado positivamente en los resultados y el desempeño de las empresas.

En resumen, el presente proyecto ha dado un importante paso hacia la modernización y optimización de las empresas recuperadoras de hierro, a través de la implementación de un software especializado. La combinación de una sólida investigación, la aplicación de buenas prácticas de desarrollo de software y la validación en entornos reales permitió obtener un resultado exitoso. Se espera que este software continúe evolucionando y mejorando en futuras versiones, adaptándose a las necesidades cambiantes de las empresas y brindando un soporte continuo para la gestión eficiente de sus operaciones.

9. Recomendaciones finales

En vista de futuras versiones del aplicativo, se formulan las siguientes recomendaciones:

1. Realizar mejoras en el rendimiento del aplicativo, como la optimización de consultas a la base de datos, la reducción de tiempos de respuesta y la gestión eficiente de los recursos del sistema.
2. Adaptar el aplicativo para ser compatible con diferentes plataformas, dispositivos y sistemas operativos. Esto garantizará que un mayor número de usuarios pueda acceder y utilizar el software, ampliando así su alcance y utilidad.
3. Identificar oportunidades para agregar nuevas funcionalidades y características al software. Recopilar comentarios y sugerencias de los usuarios y las empresas recuperadoras de hierro para determinar qué aspectos se pueden mejorar y qué funcionalidades adicionales podrían agregar valor al aplicativo.

- 
- Continuar fortaleciendo la seguridad del aplicativo para proteger los datos sensibles de los usuarios y garantizar la integridad y confidencialidad de la información. Actualizar regularmente las medidas de seguridad, realizar pruebas de penetración y estar al tanto de las últimas amenazas y vulnerabilidades.

10. Bibliografía

- E. Actis, “La era de la globalización de riesgos,” *Cent. Bras.*, vol. 1, no. 2, pp. 91–111, 2022.
- R. Colombiana, “Website of the Colombian Journal of Cardiology: Path to modernization,” *Rev. Colomb. Cardiol.*, vol. 22, no. 4, pp. 159–161, 2015, doi: 10.1016/j.rccar.2015.07.001.
- D. Stanojević, D. Cenić, and S. Cenić, “Application of computers in modernization of teaching science,” *Int. J. Cogn. Res. Sci. Eng. Educ.*, vol. 6, no. 2, pp. 89–104, 2018, doi: 10.5937/ijcrsee1802089S.
- Austin Johnson, “web site of Uppsala University,” *Simple search*. [Online]. Available: <https://uu.diva-portal.org/smash/get/diva2:1680105/FULLTEXT01.pdf>. [Accessed: 15-Feb-2023].
- P. Kashyap and C. Visvanathan, “Formalization of informal recycling in low-income countries,” *SpringerLink*, 01-Jan-1970. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-4451-73-4_3. [Accessed: 28-Mar-2023].
- A. C. Kusters, “Relating digitization, digitalization and digital transformation: a maturity model and roadmap for Dutch logistics companies,” 2022, [Online]. Available: http://essay.utwente.nl/89432/0Ahttp://essay.utwente.nl/89432/1/Kusters_BA_EEMCS.pdf
- S. Delgado, L. Cruz, and E. Lince, “El uso de software libre en el control de inventarios: caso de estudio (Free software in inventory control: case study) Sandra E. Delgado Soto * , Lidilia Cruz Rivero ** y Ernesto Lince Olgún *** Recibido: 19/02/19,” p. 52, 2019.
- H. F. Efendi and A. Aditya, “Procedia of Social Sciences and Humanities Business Process Analysis and Implementation of Odoo Open Source ERP System in Inventory , Purchasing and Sales Activities (Case Study : Captain Gadget Store) Analisis Proses Bisnis dan Penerapan Sistem ERP Odo,” vol. 0672, no. c, pp. 349–357, 2022.
- bista, “odoo inventory management module,” *bistaSolutions*, 2022. <https://www.bistasolutions.com/resources/blogs/odoo-inventory/> (accessed Nov. 14, 2022).
- Odoo, “Odoo inventory,” *Odoo*, 2022. https://www.odoo.com/es_ES (accessed Nov. 14, 2022).
- J. F. y B. Ripoll, “Holded,” *Holded*, 2022. <https://www.holded.com/es> (accessed Nov. 14, 2022).
- Delsol, “Delsol,” *sdelsol*, 2022. <https://www.sdelsol.com/programa-facturacion-factusol/> (accessed Nov. 14, 2022).
- A. A. Dubey, “DESIGN AND DEVELOPMENT OF SMART JUNKYARD SYSTEM BASED ON MACHINE LEARNING.” [Online]. Available: https://www.irjmets.com/uploadedfiles/paper/issue_6_june_2022/25606/final/fin_irjmets1654861006.pdf. [Accessed: 18-Feb-2023].
- “Industry 4.0 sustainable supply chains: An application of an IoT enabled scrap metal management solution,” *ScienceDirect*. [Online]. Available: <https://doi.org/10.1016/j.jclepro.2020.122377>. [Accessed: 18-Feb-2023].



- [15] M. A. Mesquita and J. V. Tomotani, “Simulation-optimization of inventory control of multiple products on a single machine with sequence-dependent setup times,” *Comput. Ind. Eng.*, vol. 174, p. 108793, Dec. 2022, doi: 10.1016/J.CIE.2022.108793.
- [16] E. Tello-Leal, O. Chiotti, and P. D. Villarreal, “Software agent architecture for managing inter-organizational collaborations,” *J. Appl. Res. Technol.*, vol. 12, no. 3, pp. 514–526, 2014, doi: 10.1016/S1665-6423(14)71631-1.
- [17] 123RF, “123RF,” *Jemastock*, 2022. https://es.123rf.com/photo_87061268_carpeta-de-archivos-conectada-al-diseño-del-ejemplo-del-vector-de-la-imagen-del-icono-de-los-servidores.html (accessed Nov. 14, 2022).
- [18] ionos, .“NET Framework:,” *Ionos Digital Guide*, 2022. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/net-framework/#:~:text=NET%20Framework%20es%20uno%20de%20colección%20de%20herramientas%20para%20desarrolladores>. (accessed Nov. 14, 2022).
- [19] D. M. Finkelstein, *A Beginner ’ s Guide*, vol. 47, no. 4. 2005. [Online]. Available: <http://pubs.amstat.org/doi/abs/10.1198/tech.2005.s328>
- [20] “Beneficios del prototipado y técnicas más habituales,” *INFINITIA Industrial Consulting*, 16-May-2022. [Online]. Available: <https://www.infinitiaresearch.com/noticias/prototipado-definicion-fases-tecnica/>. [Accessed: 21-Apr-2023].
- [21] “Modelo de prototipos,” *EcuRed*. [Online]. Available: https://www.ecured.cu/Modelo_de_prototipos. [Accessed: 21-Apr-2023].
- [22] Sun Microsystems, “Apache NetBeans,” *Apache*, 2020. <https://netbeans.apache.org/about/index.html%0Ahttps://netbeans.apache.org/> (accessed Nov. 14, 2022).
- [23] Software Advice, “Netbeans,” *Software Advice*, 2022. <https://www.softwareadvice.es/software/207940/apache-netbeans> (accessed Nov. 14, 2022).
- [24] Oracle Colombia, “Oracle,” *Oracle Java*, 2022. <https://www.oracle.com/co/java/> (accessed Nov. 14, 2022).
- [25] A. Gutiérrez Díaz, “Bases de Datos Bases de Datos,” *Cent. Cult. Itaca S.C*, vol. 17, p. 108, 2013, [Online]. Available: <https://repositorio.uchile.cl/handle/2250/151632>
- [26] “MySQL database for free,” MySQL Database for free, <https://www.db4free.net/> (accessed Jul. 2, 2023).
- [27] “MySQL database for free,” MySQL Database for free, <https://www.db4free.net/images/db4free-logo.png> (accessed Jul. 2, 2023).
- [28] S. UML, “A sophisticated software modeler for agile and concise modeling,” *Star UML*, 2022. <https://staruml.io/> (accessed Nov. 14, 2022).
- [29] A. Delgado, A. L. Mesquida, and A. Mas, “Utilización de Trello Para realizar el seguimiento del aprendizaje de equipos de trabajo,” Pàgina inicial de UPCommons, <https://upcommons.upc.edu/handle/2099/15518> (accessed May 23, 2023).
- [30] N., “Las 10+ Mejores Alternativas a Trello,” Salesdorado, <https://salesdorado.com/es/automatizacion/alternativas-trello/> (accessed May 24, 2023).
- [31] “A first look at bug report templates on GitHub,” sciencedirect, <https://doi.org/10.1016/j.jss.2023.111709> (accessed Jun. 27, 2023).

- 
- [32] “Mastering scrum with a focus on team maturity and key components of scrum,” sciencedirect, <https://doi.org/10.1016/j.infsos.2022.107079> (accessed Jun. 29, 2023).
 - [33] E. Bedoya Alzate, “Implementación de Pruebas unitarias,” Repositorio Institucional Universidad de Antioquia: Página de inicio, <https://bibliotecadigital.udea.edu.co/handle/10495/20225> (accessed Jun. 29, 2023).