

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

SISTEMA DE INFORMACIÓN BASADO EN UN ENFOQUE DE PROCESOS, PARA LA OPTIMIZACIÓN DE LA OTORGACIÓN DE CRÉDITOS

CASO: EMPRESA COMERBA

Para optar al título de Licenciatura en Ingeniería de Sistemas

Mención: **INFORMÁTICA Y COMUNICACIONES**

Postulante: Roly Rolando Quispe Coarite

Tutor Metodológico: Ing. Marisol Arguedas Balladares

Tutor Especialista: Lic. Mario Torrez Cupiticona

Tutor Revisor: Ing. Yolanda Escobar Mancilla

EL ALTO - BOLIVIA

2020

DEDICATORIA

El presente proyecto de grado está dirigido a Dios por haberme dado una familia maravillosa y por darme la fuerza espiritual para culminar con éxito la meta propuesta.

Con todo mi cariño a mis padres, por ser el pilar fundamental para cumplir mis logros y éxitos, por darme la vida, quererme mucho, creer en mí y por su constante apoyo incondicional.

También a mis amigos y todas las personas que estuvieron presentes en los buenos y malos momentos compartiendo cada fase de mi vida.

AGRADECIMIENTOS

Agradezco a Dios por darme la salud y haberme permitido cumplir una de mis propuestas.

Un agradecimiento especial a mis padres por su apoyo moral y económico en todo momento.

Y de manera especial a la Universidad Pública de El Alto UPEA y a mí estimada carrera Ingeniería de Sistemas por abrirme sus puertas para formar parte de esta prestigiosa institución y darme la oportunidad de superarme. A todos y cada uno de los docentes que la conforman por compartir conmigo sus conocimientos y enseñanzas requeridas para mi formación académica.

RESUMEN

El presente proyecto de grado plasma el desarrollo del sistema de información para la empresa COMERBA¹ de la ciudad de El Alto, el cual está basado en un enfoque de procesos que se adaptan a su realidad y necesidades actuales, con el objetivo de mejorar la operatividad del área de créditos. Por tanto, primero es necesario modelar detalladamente los procesos de la empresa y sobre ello diseñar y construir el sistema de información.

El desarrollo del sistema será de importancia en el manejo de la información, es así que tiene como objetivo el diseñar un sistema que ayude al control y seguimiento de los créditos y control de pagos, para una mejor administración y organización de la empresa. El resultado de este proyecto, será lograr que la empresa tenga una evolución en el ámbito de la información.

El análisis y diseño del sistema se desarrolló con la metodología UWE (UML - Based Web engineering), para evaluar la calidad del software se utilizó la ISO 9126 o ISO/IEC 9126 que permiten conocer el nivel de la calidad del software a través de un proceso de evaluación de acuerdo con las métricas o indicadores que presenta el modelo de calidad, en seguridad de la información se recurrió a la norma ISO 27002 y finalmente para la estimación de costos de uso COSMIC y su modelo de estimación post arquitectura el cual es más detallado y se aplica cuando la arquitectura del proyecto está completamente definida.

El presente proyecto tiene por objeto desarrollar un sistema de información basado en un enfoque orientado a procesos para optimizar la operativización de los créditos que la empresa COMERBA brinda a los trabajadores de transporte público de la ciudad de El Alto.

¹ COMERCIALIZADORA, IMPORTADORA, CRÉDITOS Y SERVICIOS BARRIOS - COMERBA

ÍNDICE GENERAL

PÁG.

MARCO PRELIMINAR

1.1.	INTRODUCCIÓN.....	2
1.2.	ANTECEDENTES.....	3
1.2.1	Antecedentes Institucionales	3
1.2.2	Trabajos Afines al Proyecto	4
1.3.	PLANTEAMIENTO DEL PROBLEMA.....	6
1.3.1	Problema Principal	6
1.3.2	Problemas Secundarios.....	6
1.4	OBJETIVOS	7
1.4.1	Objetivo General.....	7
1.4.2	Objetivos Específicos.....	7
1.5	JUSTIFICACIONES.....	8
1.5.1	Técnica.....	8
1.5.2	Económica.....	8
1.5.3	Social.....	8
1.6	METODOLOGÍA	9
1.6.1	Metodología de Ingeniería.....	10
1.7	HERRAMIENTAS	12
1. 8	LIMITES Y ALCANCES	14
1.8.1	Limites.....	14
1.8.2	Alcances	14
1.9	APORTES.....	17

MARCO TEÓRICO

2.1	INTRODUCCIÓN.....	19
-----	-------------------	----

2.2	RIESGO CREDITICIO	19
2.3	RIESGO OPERACIONAL.....	19
2.4	SISTEMA	19
2.5	INFORMACIÓN	20
2.6	SISTEMAS DE INFORMACIÓN.....	21
2.7	ENFOQUE DE PROCESOS.....	22
2.8	OPTIMIZACIÓN	23
2.9	PROCESOS DE OPTIMIZACIÓN DE CRÉDITOS	23
2.10	INGENIERÍA DEL SOFTWARE.....	24
2.10.2	Ingeniería del Software Orientada a Objetos.....	25
2.10.3	Proceso del Software	25
2.11	METODOLOGÍA DE DESARROLLO RUP	26
2.11.1	Fases del Proceso Unificado Racional.....	27
2.12	METODOLOGÍA UWE.....	28
2.12.1	Fases de UWE	30
2.13	UML (LENGUAJE UNIFICADO DE MODELADO).....	32
2.14	ARQUITECTURA DE SOFTWARE	34
2.14.1	Patrón de Modelo Vista Controlador.....	37
2.15	HERRAMIENTAS DE DESARROLLO.....	40
2.15.1	Gestor de Base de Datos MySQL	40
2.15.2	Lenguaje de Programación	41
2.15.3	Framework Codeigniter.....	41
2.16	MÉTODOS DE PRUEBA.....	42
2.16.1	Tipos De Métodos De Prueba.....	42
2.16.2	Métodos De Caja Blanca.....	42
2.16.2.1	Tipos De Pruebas De Caja Blanca.....	43

2.16.2.2	Prueba Del Camino Básico	44
2.16.3	Métodos De Caja Negra	45
2.16.3.1	Limitaciones	46
2.17	MÉTODOS DE ESTIMACIÓN DE COSTOS DE SOFTWARE (COSMIC)	46
2.18	DESCRIPCIÓN DEL MODELO	50
2.18.1	Características De Un Sistema De Amortización	51
2.18.2	Sistema Alemán De Amortización	53
2.18.3	Sistema De Amortización Francés	55
2.19	MÉTODOS DE CONTROL DE INVENTARIOS	57
MARCO APLICATIVO		
3	INTRODUCCIÓN	60
3.1	METODOLOGÍA UWE	60
3.1.1	Planificación	60
3.1.2	Diseño del Sistema.....	61
3.1.2.1	Situación actual de la Empresa.....	61
3.1.2.2	Rediseño de Procesos y Descripciones de tarea	67
3.1.2.2.1	Gestión de Créditos	67
3.1.2.3	Diseño de la Base de Datos	86
3.1.3	Codificación del Software	89
3.1.4	Pruebas.....	89
3.1.4.1	Pruebas Unitarias	89
3.1.4.2	Pruebas Integrales	90
3.1.5	La instalación o fase de implementación	91
3.1.5.1	Diseño de Interfaces	92
3.1.6	Mantenimiento y seguimiento	104
3.1.6.1	Análisis de los Resultados	104

3.2 MÉTRICAS DE CALIDAD DE SOFTWARE.....	105
3.3 ANÁLISIS DE COSTOS	110
3.3.1 Análisis de Costos por el método COSMIC	111
3.4 SEGURIDAD.....	114
3.4.1 Seguridad a Nivel de Base de Datos	114
3.5 PRUEBAS DE FUNCIONAMIENTO	115
3.5.1 Prueba de Caja Blanca.....	115
3.5.2 Prueba de Caja Negra	117
CONCLUSIONES Y RECOMENDACIONES	
4.1 CONCLUSIONES	119
4.2 RECOMENDACIONES	119
BIBLIOGRAFÍA.....	121

ÍNDICE DE FIGURAS

PÁG.

MARCO PRELIMINAR

Figura No 1.1: ORGANIGRAMA DE LA EMPRESA COMERBA.....3

MARCO TEÓRICO

Figura No 2.1: FASES METODOLOGÍA RUP.....28

Figura No 2.2: ESQUEMA UML32

Figura No 2.3: CICLO DE VIDA DEL MVC40

Figura No 2.4: ESTRATEGIA DE MEDICIÓN (COSMIC).....47

Figura No 2.5: LOS 4 TIPOS DE MOVIMIENTOS DE DATOS49

Figura No 2.6: SISTEMA DE AMORTIZACIÓN ALEMÁN57

Figura No 2.7: SISTEMA DE AMORTIZACIÓN FRANCÉS57

MARCO APLICATIVO

Figura No 3.1: DIAGRAMAS DE METAS Y ESTRATEGIAS60

Figura No 3.2: SITUACIÓN ACTUAL62

Figura No 3.3: REGISTRO DE NUEVOS CLIENTES63

Figura No 3.4: SOLICITUD DE DOCUMENTOS GARANTÍAS64

Figura No 3.5: REGISTRO DE GARANTÍAS65

Figura No 3.6: SOLICITUD DE CRÉDITO65

Figura No 3.7: NUEVO CRÉDITO66

Figura No 3.8: REGISTRO NUEVO CLIENTE69

Figura No 3.9: PROCESO OTORGAR CRÉDITOS70

Figura No 3.10: NUEVO CRÉDITO72

Figura No 3.11: DESEMBOLSO DE CRÉDITO78

Figura No 3.12: RECUPERAR CRÉDITOS80

Figura No 3.13 PROCESO DE CAJA (RECIBIDOR –PAGADOR)81

Figura No 3.14: COBRO DE DEUDA DE CRÉDITO	82
Figura No 3.15: OPERACIONES AUXILIARES DE CRÉDITO	84
Figura No 3.16: CONDONACIONES	86
Figura No 3.17: DISEÑO DE LA BASE DE DATOS CLIENTE Y USUARIOS	87
Figura No 3.18: DISEÑO DE LA BASE DE DATOS CRÉDITOS	88
Figura No 3.19: INTERFAZ VALIDACIÓN DE USUARIO	93
Figura No 3.20: INTERFAZ DEL MENÚ	94
Figura No 3.21: FORMULARIO REGISTRO DE CLIENTES	96
Figura No 3.22: FORMULARIO DE REGISTRO DE GARANTÍAS	97
Figura No 3.23: MENSAJE (GARANTÍA YA EXISTE)	98
Figura No 3.24: REGISTRO DE ARTÍCULOS	99
Figura No 3.25: INGRESO DE ARTÍCULOS AL INVENTARIO	100
Figura No 3.26: VENTA DE ARTÍCULOS	101
Figura No 3.27: SOLICITUD DE CRÉDITOS	102
Figura No 3.28: COBRO DE CRÉDITO	103
Figura No 3.29: PRUEBAS DE CALIDAD	116
Figura No 3.30: TÉCNICA DEL CAMINO BÁSICO	116
Figura No 3.31: PRUEBAS DE CALIDAD	117
Figura No 3.32: PRUEBA DE CAJA NEGRA	118

ÍNDICE DE TABLAS

PÁG.

MARCO APLICATIVO

Tabla No 3.1: REGISTRAR O ACTUALIZAR DATOS DEL CLIENTE	72
Tabla No 3.2: CONSULTAR HISTORIAL CREDITICIO.....	73
Tabla No 3.3: GESTIÓN DE SOLICITUD	75
Tabla No 3.4: CAMBIAR ESTADO DE SOLICITUD.....	76
Tabla No 3.5: REGISTRO Y DESEMBOLSO DE CRÉDITO	78
Tabla No 3.6: REGISTRO DE COBRANZA EN CAJA.....	83
Tabla No 3.7: RELACIÓN ENTRE PERFIL Y LAS OPCIONES DEL SISTEMA.....	95
Tabla No 3.8: FACTORES DE PONDERACIÓN	106
Tabla No 3.9: VALORES DE COMPLEJIDAD	106
Tabla No 3.10: PREGUNTAS DE USABILIDAD.....	109
Tabla No 3.11: ESTIMACIÓN DE COSTOS DEL SOFTWARE.....	110
Tabla No 3.12: CASOS DE PRUEBA.....	117

CAPITULO I

MARCO

PRELIMINAR

1.1. INTRODUCCIÓN

Los sistemas de información son importantes en la gestión de la organización, hoy en día, el entorno cambia constantemente y la competencia es alta, por lo cual, es de vital importancia para las empresas contar con un inventario bien administrado y controlado.

Dentro de toda organización es de vital importancia el manejo de sistemas de información, de aquí la importancia del manejo de inventarios, tanto en empresas como en dependencias gubernamentales, entidades bancarias y otras.

Los sistemas de control han sido adoptados por empresas modernas, para tener una disminución considerable en los costos logísticos y mejorar utilidades. Esto es clave cuando el costo capital es alto y cuando la competitividad del producto se enfrenta a un mercado mundial.

Las funciones que cumplen los sistemas de información en la organizaciones han cambiado a lo largo del tiempo, actualmente no solo representan un apoyo administrativo sino una herramienta que influye en el crecimiento y rendimiento de una organización por ello el desarrollo de un sistema de información en un proceso cada vez más complejo, que implica un conocimiento profundo de la organización, ya que este debe controlar y ejecutar sus procesos de negocio, en los que intervienen personas, aplicaciones y fuentes de información, etc.

El presente proyecto tiene por objeto desarrollar un sistema de información basado en un enfoque orientado a procesos para optimizar la operativización de los créditos que la empresa COMERBA brinda a los trabajadores de transporte público de la ciudad de El Alto.

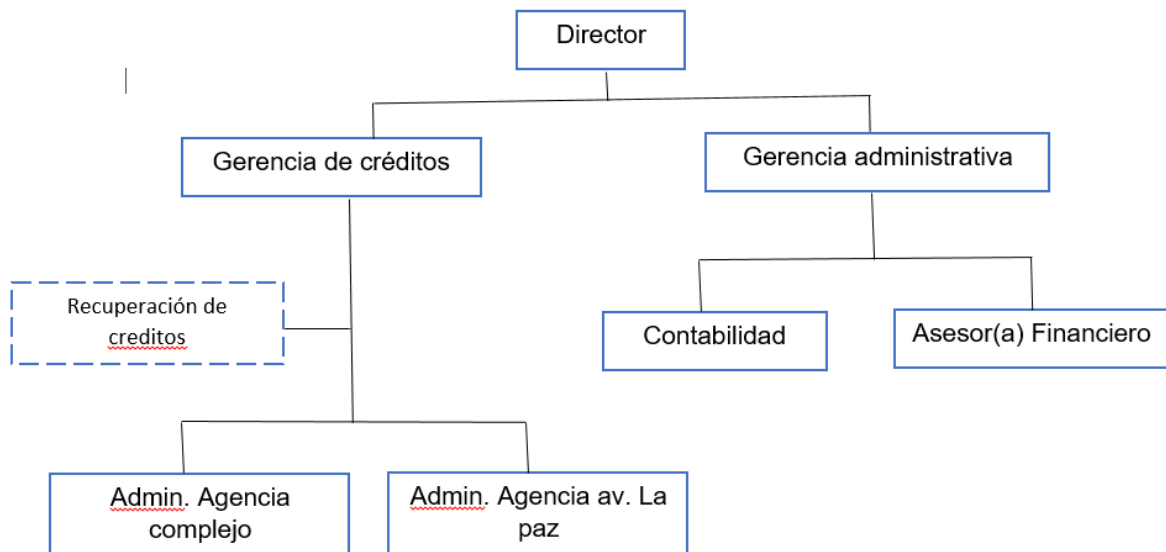
Para el desarrollo del proyecto de apoyo al área de créditos de la empresa, se utilizará la metodología UWE, y las herramientas de desarrollo de aplicación web como lenguaje de programación JavaScript, Framework BootStrap, Framework Codeigniter, JSON, como motor de base de datos MySql, así mismo para determinar la calidad del sistema web desarrollado, se hace uso de los factores de calidad ISO/IEC 25000.

1.2. ANTECEDENTES

1.2.1 Antecedentes Institucionales

La empresa COMERBA de la ciudad de El Alto, es una institución dedicada a brindar apoyo financiero concentrándose básicamente en el otorgación de microcréditos a los trabajadores en transporte público de la ciudad de El Alto.

Figura No 1.1: ORGANIGRAMA DE LA EMPRESA COMERBA



Fuente: (COMERBA, MOF 2011)

MISIÓN : “Brindar el servicio de créditos y venta de productos en la ciudad de El Alto utilizando personal calificado que nos permita garantizar la satisfacción total del cliente, cumpliendo las normas legales y los organismos de control con un recurso humano competente y comprometido con el mejoramiento continuo, generando valor para sus clientes y trabajadores”.

VISIÓN: “Consolidar la empresa COMERBA en el mercado crediticio, posicionarse entre las mejores empresas de créditos a nivel local. La empresa COMERBA será una empresa reconocida por la ética, responsabilidad y la seriedad asumida en todos sus actos por quienes la conformamos, alcanzando el éxito de manera competitiva, cumpliendo estándares de calidad, innovando los servicios y

optimizándolos procesos constantemente de tal forma que permiten satisfacer las necesidades de nuestros clientes.”

1.2.2 Trabajos Afines al Proyecto

Haciendo referencia a trabajos realizados a nivel nacional e internacional con anterioridad y como parte de la investigación bibliográfica podemos citar las siguientes:

INTERNACIONAL

- **“Plataforma Web para la programación y control del recurso Financiero, POA”** El objetivo de la tesis es crear una plataforma web para el área de planeación POA en el instituto tecnológico Colima, la metodología utilizada para el desarrollo del proyecto fue la metodología espiral. Las herramientas tecnológicas que se utilizarán fueron los lenguajes de programación de PHP y lenguaje de marcas de hipertexto HTML, hojas de estilos CSS y JavaScript. Se realizó la utilización de los Framework de Codeigniter, Bootstrap como tal y como motor de base de datos MySQL. Se concluye en que se pudo alcanzar los objetivos trazados y cumplir exitosamente con los indicadores. (Maldonado, 2016):

- **“Sistema de información para el control de inventarios del Almacén ITS”**, El proyecto está dirigido a hacer mejoras en el **Instituto Tecnológico de Saltillo (ITS)** en el departamento de materiales y servicios, en el área de almacén. Desarrollo un sistema de información como un soporte de apoyo administrativo que permite llevar el registro de artículos, el control de entradas y salidas de los mismos con la finalidad de ofrecer rapidez y seguridad al manejo de inventario, la metodología usada en este proyecto es Programación extrema XP, las herramientas que utilizo fueron visual basic 6.0, se diseñó la base de datos en Access 2003; ciudad de Coahuila México (Lic. Marcelino Vargas López, Ing. Blanca Reyes Luna, MC Maricela Sánchez López, MC. Lidia Vidal Vázquez; 2011).

➤ **“Sistema de información web para el manejo de inventarios y actividades de los centros de estudio ingeniería mecánica e ingeniería eléctrica”**

El proyecto de grado desarrolla una herramienta software en ambiente web que permita el fácil manejo y administración de inventario y los procesos de préstamo de libros y demás actividades de los centros de estudio de ingeniería mecánica e ingeniería eléctrica, la metodología que utiliza cascada, el sistema desarrollado e implementado con PHP7 como lenguaje de programación, Maria db como gestor de base de datos y JQuery como lenguaje front-end, Bootstrap como framework de maquetación de sitio. La arquitectura fue implementada bajo el framework codeigniter en su versión 3. Universidad industrial de Santander Bucaramanga (Lic. Jorge Rafael Estrada Duran, Jaime Yesith Valencia Galván, 2009).

NACIONAL

➤ **“Sistema de gestión y administración integral de inventarios para la empresa TERACORP S.R.L. tecnología y consultoría”**, Se desarrolló un sistema de información administrativo que permite un registro y control más eficiente de activos fijos, contabilidad, recursos humanos, almacenes e inventarios, producción, gestión de compra y venta, facilitando así la generación de consultas y reportes que coadyuvan a una buena toma de decisiones por parte de la administración de la empresa. Universidad Pública de El Alto (UPEA) (Univ. Sergio Ramiro Rojas Saire, 2012)

➤ **“Sistema de información para el control de inventarios para el almacén ITS”**, Para el desarrollo de este proyecto se utilizó la metodología de programación extrema (PE). Esta metodología consta de un conjunto de reglas y prácticas comprendidas en cuatro actividades básicas: Planeación, diseño, codificación y pruebas. Inicialmente se conformó la historia del usuario, la planificación del proyecto y la localización del mismo, se utilizó la técnica de construcción del prototipo para realizar las interfaces del sistema. El sistema desarrollado e implementado con PHP7 como lenguaje de programación, MySql como gestor de base de datos, Bootstrap como framework de maquetación de sitio. La

arquitectura fue implementada bajo el framework codeigniter con el patrón conocido modelo, vista, controlador. (Maricela Sánchez López junio 2011, Cochabamba).

1.3. PLANTEAMIENTO DEL PROBLEMA

La empresa COMERBA cuya misión es la de otorgar créditos en dinero y en productos llantas, baterías, entre otros, a las personas de la ciudad de El Alto, en la empresa actualmente se llevan a cabo los procesos de manera manual del ingreso y salida de productos ya que no cuenta con un modelo para el control. De esta manera causa retrasos en el cumplimiento de sus metas.

1.3.1 Problema Principal

Luego de realizar un análisis, en el área de créditos de la empresa se logró identificar que no existe un control del movimiento de la información, que los procesos desde el ingreso hasta la salida de artículos se realizan en una lista manual ya que no cuenta con modelo para el generar listas de artículos, de esta manera causa retrasos en el cumplimiento de sus metas, la empresa no cuenta una lista de actualizada de la cantidad de créditos que ha solicitado un cliente hasta la fecha, no cuenta con datos actualizados sobre las garantías, no se tiene un registro de la cantidad artículos para ser dados a crédito. La lista de clientes en mora de los créditos que ha solicitado debe realizarla cada administrador de agencia puesto que el sistema en excel no cumple la función de generar lista de clientes con mora; además de registrar los pagos al final del día buscando el registro con el número de crédito otorgado, datos de cliente.

Por lo expuesto, se plantea la siguiente pregunta:

¿De qué manera un sistema de información basado en un enfoque de procesos coadyuvaría en la optimización de la operatividad de créditos?

1.3.2 Problemas Secundarios

- Demora en la atención a los clientes con respecto al servicio que otorga la empresa debido a la falta de información centralizada de sus datos personales.

- Falta de información centralizada de clientes y productos debido a la falta de comunicación entre agencias de la empresa.
- Inexistencia de un registro de artículos, por lo cual constantemente se presenta inexactitud entre los inventarios físicos.
- Carencia de un registro de control de garantías, por lo que aparece reclamos por parte de los clientes a la hora de entrega de los mismos.
- Errores frecuentes por códigos mal escritos e inclusive la utilización de referencias de créditos debidos a que antes de registrar el nuevo crédito se debe buscar los códigos del cliente y de la garantía.
- Redundancia de datos no actualizados debido a que el sistema en excel de la empresa no reconoce si un cliente aparece de mucho tiempo atrás, lo registra como nuevo.
- Ausencia de una lista de datos actualizada ni completa de los clientes activos debido a que no se maneja información de manera directa.

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un sistema de información basado en un enfoque de procesos para optimizar la operatividad de créditos, que permita brindar información confiable y oportuna en la toma de decisiones de la empresa COMERBA.

1.4.2 Objetivos Específicos

- Realizar un diagnóstico de la situación actual de la empresa para identificar las tareas y/o procesos que se deben automatizar.
- Definir los módulos de reportes para agilizar el proceso de consulta de inventario disponible.
- Determinar los módulos del sistema a partir de los requerimientos funcionales y no funcionales.
- Establecer módulos para el registro automático de los datos personales, garantías, artículos, inventarios y pagos que brinde una información organizada y confiable.

- Identificar un módulo para el plan de pagos manteniendo actualizado todo el registro de cuotas.
- Elaborar una interfaz gráfica de usuario amigable y bastante intuitivo considerando los criterios de usabilidad y accesibilidad.
- Realizar un análisis de calidad.
- Realizar análisis de costos.

1.5 JUSTIFICACIONES

1.5.1 Técnica

El presente trabajo de justifica desde la perspectiva técnica porque utiliza métodos, técnicas y herramientas, como ser UWE como metodología para el desarrollo además de PHP y Mysql como herramientas para la programación. Ayudará a tener un panorama general de toda la empresa, los problemas y las posibles soluciones.

Es importante mencionar que la empresa cuenta con equipamiento de hardware equipos de computación con acceso a internet para la implementación del sistema a desarrollarse.

1.5.2 Económica

En la empresa cuentan con equipos de computación con acceso a internet por lo que no se realizara en gastos de hardware, en la implementación del sistema se empleara software libre para evitar el hecho de manipulación de licencia, también reducirá costos debido a la sistematización de las tareas, se optimizara los tiempos y se facilitara la administración de información.

El sistema utilizará tecnologías libres, esto implica que el sistema no tendrá costos adicionales en cuanto a compra de licencias, las tecnologías libres no se restringen a un sistema operativo en específico, por el contrario funcionan en la mayoría de los sistemas operativos.

1.5.3 Social

El presente proyecto será útil y beneficioso en el ámbito social porque es una herramienta que facilita las actividades cotidianas del personal encargado,

mejorando el control de ingreso y salida de créditos, beneficiará a los clientes por realizar la atención en un tiempo reducido, realizará un reporte detallado de los productos con los que la empresa cuenta, además de beneficiar al gerente dando una información más confiable para una mejor toma de decisiones.

Los gerentes de agencia tomarían decisiones para mejorar la operatividad de créditos de forma particular, de esta forma los clientes tendrán mayor oportunidad de acceder a un crédito.

El sistema permitirá enlazar automáticamente las sucursales de la empresa, generando un formato único de otorgación de créditos y optimizando el tiempo de espera de un cliente.

1.6 METODOLOGÍA

La metodología que se empleará para el desarrollo del sistema será:

UWE (UML-Based Web Engineering) es un método de ingeniería del software de desarrollo de aplicaciones basado en UML. Cualquier tipo de diagrama UML puede ser usado, porque UWE es una extensión de UML.

UWE es una propuesta basada en el proceso unificado y UML. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

UWE está especializada en la especificación de aplicaciones adaptivas y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptivas de la navegación en función de las preferencias, conocimiento tareas de usuario.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de una meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los métodos. (Koch, 2000).

1.6.1 Metodología de Ingeniería

UML²: Es un lenguaje estándar orientado a objetos para visualizar, especificar, construir y documentar los elementos del sistema de un software. No es una metodología, es usan notación para desarrollar modelos. Beneficios, mejora la comunicación entre los involucrados, reduce la complejidad, mayor calidad de software, Reusó, reduce costos y riesgos al experimentar con os modelos.

Se usara esta metodología porque describe el diseño sistemático, basado en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML cubre todo el ciclo de vida de este tipo de aplicaciones centrando además se atención en aplicaciones personalizadas o adaptivas.

➤ **Métodos de Prueba de Software**

Prueba de caja negra o comportamiento, se centra en los requisitos funcionales del software e intenta probar: funciones incorrectas o ausentes, errores de interfaz, errores de rendimiento y errores de inicialización y terminación. Es aplicado a procesos y módulos del software, es denominado pruebas de gran escala.

Prueba de caja blanca o cristal, se enfoca en las líneas de código y es denominado prueba de pequeña escala. (Pressman, 2002).

Método de Prueba de Rendimiento, se evalúa el rendimiento respecto al tiempo, por cada proceso. Recordando el objetivo de las pruebas, se realizarán para obtener el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible.

➤ **Métricas de Calidad de Software**

La métrica de calidad de software ISO/IEC 9126 para evaluar la calidad de software, este estándar identifica los siguientes atributos clave de calidad: funcionalidad, confiabilidad, eficiencia, facilidad de uso o mantenimiento y portabilidad.

Puesto que la ISO 9126 se limita a la especificación de un modelo de calidad general, esta se debe aplicar en conjunto con la ISO 14598, según esta norma, los

² UML Lenguaje de Modelado Unificado

componentes fundamentales en la evaluación de la calidad de software son: Modelo de calidad, método de evaluación, medidas de software y herramientas de soporte. Motivo por lo cual se usará la métrica de calidad de software **ISO/IEC 25000:2005** conocida como SQUARE (Software Quality Requirement Evaluation), básicamente se trata de una unificación y revisión de los estándares ISO/IEC 9126 e ISO/IEC 14598, guía el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad.

➤ **Métodos de estimación de costo de Software**

La medición y estimación de software utilizando **COSMIC** (Common Software Measurement International Consortium), es un método de segunda generación que determina el tamaño del software a partir del número de interacciones entre los componentes de los requerimientos funcionales.

Tiene la ventaja que no establece límites arbitrarios al tamaño funcional, por lo cual pueden medirse componentes de software independientemente de si son muy grandes o pequeños. Adicionalmente, el análisis para la medición está basado en el desglose funcional de los componentes de software, por lo que está alineado con las prácticas de Ingeniería de software. Es un método de análisis de puntos de función de segunda generación, el grupo se nombró a sí mismo como el Common Software Measurement International Consortium (COSMIC, 1998).

El tamaño de un software es la principal variable necesaria para determinar el esfuerzo de desarrollo que deberá invertirse para implementarlo. La medición y estimación de software utilizando COSMIC, es un método de segunda generación que determina el tamaño del software a partir del número de interacciones entre los componentes de los requerimientos funcionales.

Estandarizado bajo la ISO 19761, el método COSMIC puede aplicarse a diversos tipos de software, incluyendo aplicaciones de negocios, sistemas de información gerencial, software en tiempo real, infraestructura, e inclusive software científico y de ingeniería.

1.7 HERRAMIENTAS

Las herramientas que se utilizaran en el proyecto son:

- **Sistema Operativo**, Linux Mint es una distribución de GNU/Linux comunitaria de origen franco-irlandesa basada en Debian y Ubuntu que tiene por objeto proveer un sistema operativo moderno, elegante y cómodo que sea tanto poderoso como fácil de usar. Al igual que otros sistemas operativos GNU/Linux, está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto. (Linux Team, 2019)
- **PHP (PHP Hypertext Pre-processor)**. Lenguaje de programación usado generalmente en la creación de contenidos para sitios web. PHP es un lenguaje de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también interfaz de línea de comandos que puede ser usada en aplicaciones graficas independientes. (Rasmus Lerdorf, 2002)
- **Framework CODEIGNITER** Como cualquier otro framework, Codeigniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. Esto es, marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. Codeigniter implementa el proceso de desarrollo llamado Model View Controller (MVC³), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales. **CodeIgniter** es un framework para aplicaciones web de código abierto para

³ MVC Model View Controller “Modelo Vista Controlador”

crear sitios web dinámicos con PHP. Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas.

También hay que destacar que CodeIgniter es más rápido que muchos otros entornos. Incluso en una discusión sobre entornos de desarrollo con PHP, Rasmus Lerdorf, el creador de PHP, expresó que le gustaba CodeIgniter porque es rápido, ligero y parece poco un entorno. (EllisLab, 2007)

- **Gestor de base de datos (MySQL)**, es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. En cuanto a la definición general, MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional. Tus datos deberían ser la principal preocupación al elegir el software RDBMS correcto. Con su sistema de privilegios de acceso y la administración de cuentas de usuario, MySQL establece un alto estándar de seguridad. La verificación basada en el host y el cifrado de contraseña están disponibles. (Oracle Corporation, 2010)
- **Servidor HTTP Apache** Es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPs 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo firme y energético pero no agresivo. Apache consistía solamente en un conjunto de parches a aplicar al servidor NCSA. En inglés a patchy server

(un servidor “parcheado”) suena igual que Apache Server se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. La mayoría de las vulnerabilidades de la seguridad descubiertas y resultas tan solo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones. **(Robert McCool, 1995)**

1. 8 LIMITES Y ALCANCES

1.8.1 Limites

El sistema de información para la empresa COMERBA esta específicamente orientado a los procesos de la administración de información, la formulación, ejecución, seguimiento y evaluación, siendo un desarrollo Web, su utilización está dirigido a la colaboración de las unidades de créditos al momento de dar créditos y realizar seguimiento de las mismas.

1.8.2 Alcances

El sistema debe ser capaz de controlar los procesos de préstamo y cobros de la unidad como la información de clientes, mantener planillas de información de forma automatizada.

El sistema contemplará los siguientes módulos:

Modulo administrador: el siguiente modulo tendrá acceso a todo el sistema sin restricciones.

- Administrar usuarios
- Registro y manejo de la información de almacén.
- Administrar clientes (registro y modificación).
- Administrar créditos.
- Administrar ingresos y egresos.
- Seguimiento de préstamos y cobros.
- Búsqueda de clientes.
- Búsqueda de productos en almacén

- Seguimiento de la ejecución de créditos, actividades y evaluación del plan operativo anual.
- Generar reportes que coadyuve el control y administración.

Modulo asesor financiero: el siguiente modulo tendrá acceso a la lista de clientes y almacén correspondiente.

- Registro y manejo de la información de almacén.
- Administrar clientes (registro y modificación).
- Administrar créditos.
- Administrar ingresos y egresos.
- Seguimiento de préstamos y cobros.
- Búsqueda de clientes.
- Búsqueda de productos en almacén.
- Seguimiento de la ejecución de créditos, actividades y evaluación del plan operativo anual.
- Generar reportes que coadyuve el control y administración.

Módulo Garantías: el siguiente modulo presentara la lista de clientes con la garantía respectiva:

- Búsqueda de documento de identidad del cliente registrado en el sistema.
- Registro y recepción de los documentos dados en garantía a la empresa de tipo: Movilidad o Convenio.
- Registro de datos del vehículo que esta como garantía.
- Lista de garantías divididas en Movilidad y Convenio.
- Seguimiento en la ejecución de recepción y devolución de garantías.
- Reportes que ayude a los procesos de recepción y devolución de documentos.

Modulo créditos: el siguiente modulo presentara la lista de clientes que solicitaron un crédito con estado en deuda:

- Búsqueda de cliente por el documento de identidad.
- Relación entre con el modulo garantías.

- Generar tabla de créditos por tiempo de acuerdo al monto solicitado, plazos y número de cuotas.
- Generar cuotas de tipo fijo o gradual en la tabla de créditos.
- Generar reportes de apoyo con el propósito de validar el tipo de crédito y la forma de pago.
- Generar reporte por cliente mostrando la fecha de pago de las cuotas.

Modulo gestor de cobranza: el siguiente modulo presentara la lista de clientes con mora del crédito activo:

- Relación con el modulo créditos.
- Búsqueda de clientes con deuda de cuotas.
- Lista de cuotas con estado en “Cobrar”.
- Estado de la cuenta.
- Lista de tabla de las cuotas del crédito.

Modulo lista de Deudores: el siguiente modulo presentara la lista de clientes con mora del crédito activo:

- Lista de clientes con crédito en Mora de más de 5 semanas de retraso.
- Búsqueda por documento de identidad de clientes con mora.
- Lista de dirección de los clientes en mora lugares de trabajo y domicilio.
- Reporte de clientes en mora.

Módulo Reportes: el siguiente modulo se presentará la lista de clientes y almacén correspondiente.

- Registro y manejo de la información de almacén.
- Control de entrada y salida de documentos (garantías) del cliente.
- Lista de clientes generalizada.
- Lista de clientes en mora.
- Lista de precios de artículos.
- Libro diario.

1.9 APORTES

El sistema permitirá cubrir las necesidades de mejora para los procesos que realizan en la unidad de créditos, haciendo que se emitan resultados más rápidos y completos.

Por otra parte en el beneficio de la unidad de créditos por el cual permitirá centralizar las informaciones de las solicitudes y requerimientos de las diferentes sucursales, en una sola base de datos y facilitando al administrador del sistema recabar toda la información necesaria y para el control de toma de decisiones respectiva y oportuna.

CAPITULO II

MARCO TEÓRICO

2.1 INTRODUCCIÓN

En este capítulo se pretende desarrollar los antecedentes y las consideraciones teóricas del proyecto de grado.

2.2 RIESGO CREDITICIO

Viene representado por la posibilidad de pérdidas por la incapacidad o falta de voluntad de los deudores para cumplir sus obligaciones, estos pueden ser originados por diversos factores: evaluación inadecuada del crédito, confianza excesiva en las garantías del deudor, deficiente seguimiento, entre otros. La falta de información actualizada y oportuna sobre los clientes solicitantes de un crédito, sus expedientes anteriores y comportamiento crediticio conlleva a una mala evaluación por parte de los analistas de créditos y se cae en un nivel muy alto de riesgo crediticio.

2.3 RIESGO OPERACIONAL

Es la posibilidad de ocurrencia de pérdidas debido a procesos inadecuados, políticas y procedimientos, fallas del personal a causa de una inadecuada capacitación, negligencia, sabotaje, fraude y apropiación de información sensible, de la tecnología de información (errores de desarrollo e implementación de los sistemas informáticos) o acontecimientos y eventos externos.

2.4 SISTEMA

Es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones. (Peña, 2006).

Otros autores como Peralta (2008), de una manera más acertada define sistema como: conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Un sistema tiene la propiedad de que toda acción que produce cambios en una de las partes de los sistemas, también estos cambios se dan en el resto del sistema. El sistema también reaccionara ante cualquier evento o estímulo producido en cualquier parte de la unidad.

Los "**sistemas Web**" o también conocido como "**Aplicaciones Web**" son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos (Windows, Linux). Sino que se alojan en un servidor en Internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los sistemas Web tienen funcionalidades muy potentes que brindan respuestas a casos particulares.

Los sistemas Web se pueden utilizar en cualquier navegador Web (**Chrome, Firefox, Internet Explorer**, etc.) sin importar el sistema operativo. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el sistema.

Las aplicaciones Web trabajan con bases de datos que permiten procesar y mostrar información de forma dinámica para el usuario. Los sistemas desarrollados en plataformas Web, tienen marcadas diferencias con otros tipos de sistemas, lo que lo hacen muy beneficioso tanto para las empresas que lo utilizan, como para los usuarios que operan en el sistema. (Sergio Báez, 2012)

2.5 INFORMACIÓN

Está constituida por un grupo de datos ya supervisados y ordenados, que sirven para construir un mensaje basado en un cierto fenómeno o ente. La información permite resolver problemas y tomar decisiones, ya que su aprovechamiento racional es la base del conocimiento. Por lo tanto, otra perspectiva nos indica que la información es un recurso que otorga significado o sentido a la realidad, ya que mediante códigos y conjuntos de datos, da origen a los modelos de pensamiento humano.

Idalberto Chiavenato afirmaba que la información consiste en un conjunto de datos que poseen significado, de tal modo que reducen la incertidumbre y aumentan el

conocimiento de quien se acerca a contemplarlos. Estos datos se encuentran disponibles para su uso inmediato y sirven para clarificar incertidumbres sobre determinados temas. (Chiavenato Idalberto, 2006)

Ferrel y Hirt, por su parte, dicen que esos datos y conocimientos están estrictamente ligados con mejorar nuestra toma de decisiones. Si un individuo se encuentra bien informado sobre un aspecto, seguramente su decisión al respecto podrá ser más acertada que uno que no lo este. (Ferrell O. C. y Hirt Geoffrey, 2004)

Otros autores que han definido la información son Czinkota y Kotabe, que dicen que consiste en un conjunto de datos que han sido clasificados y ordenados con un propósito determinado. Uniendo todas las teorías sobre el concepto llegamos a la conclusión de que son datos sobre un suceso o fenómeno particular que al ser ordenados en un contexto sirven para disminuir la incertidumbre y aumentar el conocimiento sobre un tema específico. (Czinkota Michael y Kotabe Masaaki, 2000)

2.6 SISTEMAS DE INFORMACIÓN

Conjunto de informaciones que afectan a una o más entidades en alguna de sus actividades, unido a las normas, recursos y procedimientos de que se disponen para recoger, elaborar y permitir el acceso a esas informaciones. Conjunto formal de procesos que operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan, elaboran y distribuyen la información necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes para desempeñar su actividad a su estrategia de negocio (Andreu, 1991).

Un sistema de información es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones. (Peña, 2006).

Una manera más acertada define sistema de información como: conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que

interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema (Peralta, 2008).

2.7 ENFOQUE DE PROCESOS

Las organizaciones tienen una misión y para cumplirla desarrollan variadas actividades, definimos un proceso como “un conjunto de actividades que recibe elementos o recursos de entrada y los transforma en salidas, o sea, resultados”. En las organizaciones existen variados procesos interrelacionados entre sí, y en múltiples ocasiones las salidas o resultados de un proceso constituyen la entrada de otro.

Podemos expresar, en sentido general, que el conjunto de actividades para la realización de un producto, la prestación de un servicio o el desarrollo de una actividad comercial constituye un proceso. Los procesos se definen como las diversas actividades que se requieren para generar un resultado, dichas actividades se alimentan de varios componentes (proveedores, insumos, servicios, etc.) que agregan valor al resultado final. Cuando en una organización aplicamos un sistema de procesos los que se identifican, se interrelacionan y se gestionan para garantizar un resultado deseado y que satisfaga las necesidades del cliente decimos que contamos con un enfoque basado en procesos. En el diseño de los procesos el talento humano tiene un rol muy importante debido al conocimiento que tiene sobre las particularidades que señalan los mejores pasos o fases que permite la construcción de un sistema. La mejor manera de emplearla es obteniendo un impacto positivo en los clientes, intermediarios y hasta el medio ambiente laboral. Un enfoque basado en procesos permite un mejor y continuo control sobre los procesos y las interrelaciones entre ellos, lo cual sin lugar a dudas representa una ventaja competitiva para la organización. Permite además un desempeño mejor y la obtención de mejores resultados no sólo en los procesos sino en los productos y servicios, así como la posibilidad de un mejoramiento continuo de manera integral. (Alhama, R., Alonso, F., & Martínez, 2005)

2.8 OPTIMIZACIÓN

Es la acción y efecto de mejorar. Este verbo hace referencia a buscar la mejor manera de realizar una actividad. El término se utiliza mucho en el ámbito de la informática. La optimización de software busca adaptar los programas informáticos para que realicen sus tareas de la forma más eficiente posible. Virtualmente, existen infinitas maneras de desarrollar una misma aplicación y uno de los factores más influyentes a la hora de crear el diseño es la arquitectura de hardware con la cual se desea trabajar. En pocas palabras, conseguir el mejor rendimiento en una plataforma enfocada en el tipo y la cantidad de memoria es muy diferente a hacerlo en una cuyo fuerte es la velocidad de los procesadores.

La optimización de consultas, por otra parte, consiste en mejorar los tiempos de respuesta en la gestión de base de datos. En el lenguaje de consultas SQL, uno de los más utilizados por los desarrolladores, la optimización busca simplificar las operaciones más complejas, aquellas que suelen requerir de mucho tiempo para su resolución.

A nivel general, la optimización puede realizarse en el ámbito del desarrollo de un sistema, pero siempre con el mismo objetivo: mejorar el funcionamiento de algo o el desarrollo de un proyecto a través de una gestión perfeccionada de los recursos. La optimización puede realizarse en distintos niveles, aunque lo recomendable es concretarla hacia el final de un proceso. (Julián Pérez Porto y Ana Gardey, 2009)

2.9 PROCESOS DE OPTIMIZACIÓN DE CRÉDITOS

La principal función del crédito es la financiación de la producción con anticipación a la demanda, es decir, unir la propiedad de recursos productivos con las demandas de sus productores. El mecanismo del crédito economiza también el uso del dinero, permite el planteamiento más flexible de los gastos a lo largo del tiempo y actúa como canal recolector de los pequeños ahorros de la sociedad.

Definir la forma en que se propone optimizar el proceso. Crédito y Riesgo El riesgo que presenta cualquier operación de crédito reside en que el cliente no devuelva el monto prestado y los intereses devengados; que lo haga de forma distinta a lo convenido, sobre todo en cuanto al plazo o que el proceso inflacionario haya hecho

que el rendimiento de la operación disminuya o sea negativo. El riesgo, contingencia o proximidad de un daño o perjuicio según la acepción gramatical, está presente en toda actividad económica. Sin embargo quizá a ninguna afecte tanto como a la banca, al punto de constituir un elemento esencial en su gestión. Debido a que la operatoria bancaria se basa en el empleo de recursos de terceros en una elevada proporción, y en la medida que los retiene van devengando intereses que aumentan del endeudamiento (Banco Nacional de Fomento, 1983)

2.10 INGENIERÍA DEL SOFTWARE

Según la definición del IEEE⁴ menciona que la ingeniería de software es la aplicación de un enfoque sistemático disciplinado, una tecnología estratificada y cuantificable al desarrollo, funcionamiento y mantenimiento del software (Pressman, 2006).

Software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo. Según la definición de IEEE, (Lewis, 1994)

(Cota, 1994) define a la ingeniería de software como la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software, es decir “permite elaborar consistentemente productos correctos, utilizables y costo-efectivos”. Existen diferentes definiciones de la ingeniería del software, sin embargo estas coinciden que es una disciplina que coadyuva al desarrollo del software de calidad. A continuación se describen algunas definiciones:

La ingeniería del software es una disciplina tecnológica preocupada de la producción sistemática y mantenimiento de los productos software que son desarrollados y modificados en tiempo y dentro de un presupuesto definido. (Fairley, 1998)

⁴ Abreviación del Inglés “Institute of Electrical and Electronic Engineers”.

Se menciona que la ingeniería del software es la rama de la ingeniería que la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas relacionados con el desarrollo del software. (Zabala, 2002)

En la actualidad el uso de la ingeniería del software como disciplina empleado para el desarrollo de sistemas de información, para lograr un producto eficaz y económico.

La ingeniería del software es una tecnología multicapa. Los cimientos que son la base de la ingeniería del software están orientados hacia la calidad. La gestión de calidad total y las filosofías similares fomentan una cultura continua de mejora de procesos y es esta cultura la que conduce últimamente al desarrollo de enfoques cada vez más robustos para la ingeniería del software (Choque, 2002).

2.10.2 Ingeniería del Software Orientada a Objetos

(Zabala, 2002) menciona que se debe distinguir a la POO como un paradigma y una metodología, no son la misma cosa. La POO como paradigma es una postura ontológica: el universo computacional está poblado por objetos, cada uno responsabilizándose de sí mismo y comunicándose por medio de mensajes con los demás.

La ingeniería del software orientada a objetos (ISOO⁵) introduce las mismas fases de la ingeniería convencional que son: Análisis, diseño, codificación, pruebas, mantenimiento. La ISOO también está dirigida a un enfoque de calidad, proceso, los métodos y las herramientas, además de la aplicación de conceptos tales como: herencia, polimorfismo, abstracción, clase, objeto, etc.

2.10.3 Proceso del Software

El proceso de ingeniería del software se define como “un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad”.

⁵ ISOO del término original “Object Oriented Software Engineering”.

(Jacobson, 1998). El proceso de desarrollo de software “es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos son transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo”. Concretamente “define quien está haciendo que, cuando hacerlo y como alcanzar un cierto objetivo”. (Jacobson, 1998)

La ingeniería del software tiene como base al enfoque de calidad, sustentado en un compromiso de calidad del producto final que es el software. (Choque, 2002) menciona que el fundamento de la ingeniería del software es la capa de proceso. El proceso es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería del software. Las áreas clave del proceso forman la base del control de gestión de proyectos del software de manera adecuada.

2.11 METODOLOGÍA DE DESARROLLO RUP

Los métodos indican como construir de manera técnica el software. Los métodos abarcan una gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Los métodos dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas (Choque, 2002).

RUP⁶ Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades al interior de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga la necesidad del usuario final en un tiempo y presupuesto previsible. El RUP es un método de desarrollo iterativo enfocado hacia los casos de uso, manejo de riesgos y manejo de la arquitectura.

El ciclo de vida RUP es una implementación del desarrolló en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones. RUP divide el proceso en cuatro fases, dentro de las cuales se realizan pocas pero grandes y formales iteraciones en

⁶ Del término original “Rational Unified Process”.

número variable según el proyecto. En la Figura muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requisitos.

- En la fase de elaboración, las iteraciones se orientan al desarrollo de la base line de la arquitectura, abarcan más los flujos de trabajo de requisitos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la base line de la arquitectura.
- En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se seleccionan algunos Casos de Uso, se refinan su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan iteraciones hasta que se termine la implementación de la nueva versión del producto.

- En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

2.11.1 Fases del Proceso Unificado Racional

➤ Fase de Inicio

Durante esta fase de inicio las iteraciones se centran con mayor énfasis en las actividades de modelado de la empresa y en sus requerimientos.

➤ Fase de elaboración

Durante esta fase de elaboración, las iteraciones se centran en el desarrollo de la base del diseño, encierran los flujos de trabajo de requerimientos, modelo de la

organización, análisis, diseño y una parte de implementación orientada a la base de la construcción.

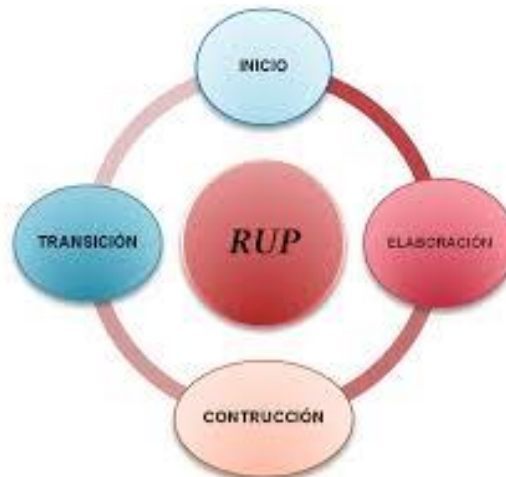
➤ **Fase de construcción**

Durante esta fase de instrucción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones las cuales seleccionan algunos casos de uso, referidas a su análisis y diseño además se procede a su implementación y prueba, en esta fase se realiza una pequeña cascada para cada ciclo, se realizan tantas iteraciones hasta que se termine la nueva implementación del producto.

➤ **Fase de transición**

Esta fase de transición busca garantizar que se tiene un producto preparado para su entrega al usuario.

Figura No 2.1: FASES METODOLOGÍA RUP



Fuente: (Metodología RUP, UNAD 2015)

2.12 METODOLOGÍA UWE

UWE está basado en estándares de la OMG como UML, Model Driven Architecture de OMG (MDA), Object Constraint Language (OCL) y eXtensible Markup Language (XML), asegurando su seguimiento mediante guías y especificaciones para el uso de tecnologías orientadas a objetos.

El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático; y herramientas de soporte para la generación semi-automática de Aplicaciones Web.

La notación de UWE se define como una ligera extensión de UML, proporcionando un perfil UML para el dominio específico de la web.

El método UWE consiste en la construcción de seis modelos de análisis y diseño. Dicha construcción se realiza dentro del marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos, diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación.

Los principales artefactos que produce el método de diseño de UWE son los siguientes:

- Un Modelo de Requerimientos que captura los requerimientos del sistema.
- Un Modelo Conceptual para el contenido (modelo de contenido).
- Un Modelo de Usuario.
- UN Modelo de Navegación que comprende la estructura de la navegación.
- Un Modelo de Presentación que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo del flujo de la presentación, modelo de interface abstracta de usuario, y modelo de ciclo de vida del objeto).
- Un modelo de adaptación.

➤ **Modelo de Requerimientos**

El primer paso para el desarrollo de un sistema web que se especificará con UWE, es realizar la identificación de los requerimientos y plasmarlos en un modelo de requerimientos.

Los requerimientos pueden ser documentados en diferentes niveles de detalle, para este caso, UWE propone dos niveles de granularidad. En primera instancia se deben describir detalladamente las funcionalidades del sistema, las cuales son

modeladas con casos de uso UML. Como segundo paso, se debe elaborar una descripción de los casos de uso más detallada, por ejemplo, realizando diagramas de actividad UML donde se delimiten las responsabilidades y acciones de los actores involucrados.

Los **casos de uso** fueron propuestos por el Proceso de Desarrollo de Software Unificado (RUP) para capturar los requerimientos del sistema. Es una técnica centrada en el usuario que obliga a definir quiénes son los usuarios (actores) de la aplicación y ofrece una forma intuitiva de representar la funcionalidad que una aplicación tiene que cumplir para cada actor.

2.12.1 Fases de UWE

- **UWE distingue tres tipos de casos de uso: navegación, proceso, y casos de uso personalizados.**

Los casos de uso de navegación, se distinguen con el estereotipo y se utilizan para modelar el comportamiento típico del usuario cuando interactúa con una aplicación web, tal como navegar a través del contenido de la WebApp o buscar información por medio de palabras claves.

Los casos de uso de *proceso*, se utilizan para describir las tareas del negocio que los usuarios finales realizarán con el sistema, tales como acciones transaccionales sobre la Base de Datos. No se denota con ningún estereotipo específico, por lo tanto se utiliza en este caso la notación pura de UML.

Los casos de usos *personalizados*, implican la personalización de un sistema web, la cual es desencadenada por el comportamiento del usuario. Estos casos de uso se denotan con el estereotipo.

- **Modelo de Contenido (Conceptual) y Modelo de Usuario**

El diseño conceptual se basa en el modelo de análisis e incluye los objetos involucrados en las actividades típicas que los usuarios realizan con la aplicación.

El propósito del *modelo de contenido* es proporcionar una especificación visual de la información relevante para el dominio del sistema web, que comprende principalmente el contenido de la aplicación Web.

➤ **Modelo de Navegación:**

El modelo de estructura de navegación define la estructura de nodos y links de una WebApp mostrando cómo se puede realizar la navegación utilizando elementos de acceso tales como índices, visitas guiadas, consultas y menús.

Los elementos de modelado son:

- **Clases de navegación**, que se denotan con 0, representan los nodos navegables de la estructura de hipertexto.
- **Links de navegación**, que muestran el vínculo directo entre las clases de navegación.
- **Caminos de navegación alternativos**, los cuales son visualizados con el estereotipo.
- **Primitivas de acceso**, las cuales se utilizan ya sea para llegar a múltiples instancias de una clase de navegación para seleccionar ítems.
- **Clases de procesos** las cuales modelan los puntos de entrada y de salida de los procesos de negocio. Cada clase de proceso está asociada a un caso de uso de proceso.
- **Links de procesos**, que representan el vínculo entre las clases de proceso y de navegación.

El modelo de *estructura de navegación* se representa mediante diagramas de clases UML estereotipados con las clases de navegación y procesos, menús y primitivas de acceso y así también los links de navegación y proceso.

➤ **Modelo de Presentación**

El modelo de presentación proporciona una vista abstracta de la interfaz de usuario (UI) de la aplicación web. Se basa en el modelo de navegación y describe qué elementos (por ejemplo texto, elementos, links, formularios) se utilizarán para presentar los nodos de navegación.

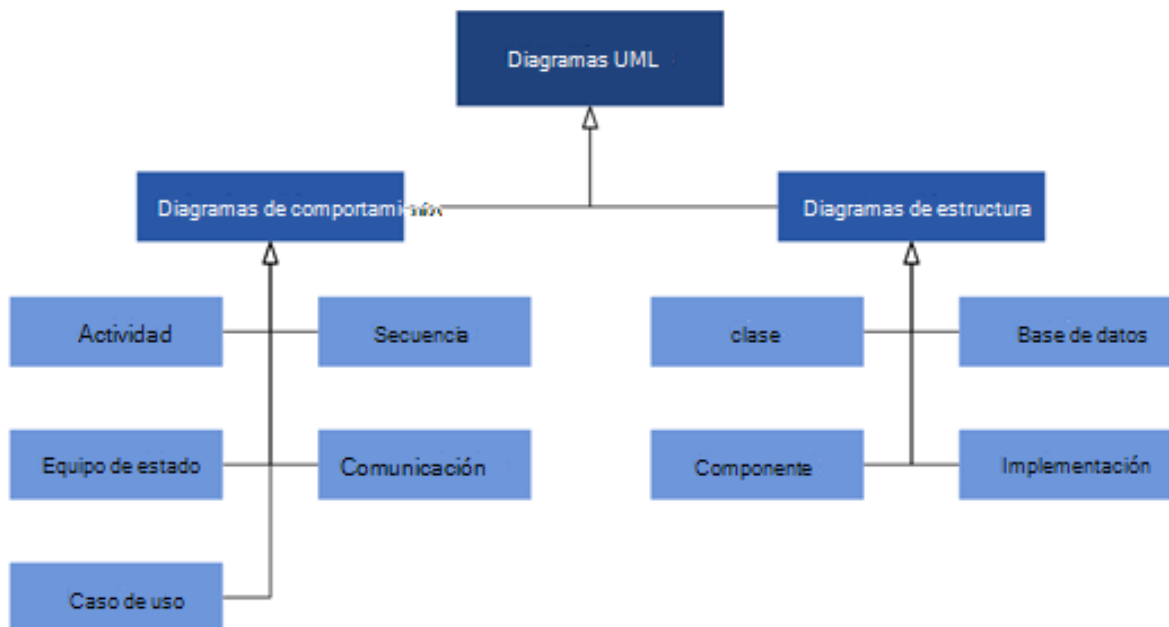
Los elementos básicos del modelo de presentación son:

- **Clases de presentación**, las cuales se basan directamente en los nodos del modelo de navegación. Una clase de presentación está compuesta por elementos de UI tales como, texto, vínculo, botón, imagen, formulario, y colección de vínculos.
- **Páginas web**, que se utilizan para modelar la información proveniente de varios nodos de navegación y que se presentan en una misma página web.
- **Grupo de presentación**, el cual es un contenedor de clases de presentación, y a su vez de otros grupos de presentación.(OMG, 2003)

2.13 UML (LENGUAJE UNIFICADO DE MODELADO)

UML, es el lenguaje grafico de modelado de sistemas de software, para visualizar, especificar, construir y documentar un sistema UML, consta de distintos tipos de diagramas: diagramas de estructura de comportamiento y de interacción como se ve la figura No 3.

Figura No 2.2: ESQUEMA UML



Fuente: (Almudéver, 1999)

➤ **Diagrama de clases**

El diagrama de clases, es un tipo de diagrama que describe la estructura de un sistema que muestran sus clases, es la propiedad que tiene operaciones en un contexto, estos incluyen información entre la relación de objetos, la herencia de propiedades den objeto a otro.

➤ **Diagrama de Objetos**

El diagrama de objetos es un gráfico de instancias, que incluyen objetos y datos, el mismo es una instancia de un diagrama de clases.

➤ **Diagrama de componentes**

El diagrama de componentes, es un diagrama que representa como un sistema, está dividido en componentes con sus respectivas dependencias entre ellos.

➤ **Diagrama de estructura**

El diagrama, muestra la estructura interna de una clase y las colaboraciones que este tiene, donde las partes interactúan con cada una de las otras, y las instancias externas, en síntesis la estructura compuesta es un conjunto de elementos interconectados que colaboran con cada elemento.

➤ **Diagrama de paquetes**

Un diagrama de paquetes muestra como un sistema está dividido en agrupaciones lógicas, donde se muestran las dependencias entre ellas. Los paquetes normalmente están organizados para maximizar la coherencia interna de cada paquete y minimizar el acoplamiento externo entre los paquetes.

➤ **Diagrama de despliegue**

El diagrama de despliegue es un tipo de diagrama que es utilizado para modelar la disposición física de los dispositivos de software en nodos. Algunos de estos diagramas muestran el despliegue de modelados de sistemas empotrados, sistemas cliente – servidor, sistemas completamente distribuidos.

➤ **Diagrama de casos de uso**

El caso de uso es la notación gráfica para representar las acciones de un sistema desde el punto de vista del usuario. Los diagramas de casos de uso modelan la funcionalidad del sistema donde intervienen los actores y los casos de uso.

➤ **Diagrama de actividades**

Muestra la naturaleza dinámica de un sistema mediante el modelado del flujo ocurrenciente de actividad en actividad, donde una actividad representa una operación en alguna clase del sistema y que este resulta en un cambio en el estado del sistema.

➤ **Diagrama de estado**

Representa situaciones durante la vida de un objeto, el mismo se representa mediante rectángulos de esquina ovaladas.

➤ **Diagrama de secuencias**

Muestran la mecánica de la interacción de las clases y los objetos respecto del tiempo, donde las clases y los objetos representan la información.

➤ **Diagrama de colaboraciones**

Describe las interacciones entre los objetos en términos de mensajes secuenciados, los diagramas de colaboración representan una combinación de información tomada de los diagramas de clases, secuencias y casos de uso, mostrando el comportamiento tanto de la estructura estática, como de la estructura dinámica de un sistema.

2.14 ARQUITECTURA DE SOFTWARE

En los inicios de la informática, la programación se consideraba un arte y se desarrollaba como tal debido a la dificultad que resultaba para la mayoría de las personas, pero con el tiempo se han ido descubriendo y desarrollando formas y guías generales, con base a las cuales se puedan resolver los problemas. A estas, se les ha denominado arquitectura de software, porque, a semejanza de los planos

de un edificio o construcción, estas indican la estructura, funcionamiento e interacción entre las partes del software. En el libro "An introduction to Software Architecture", David Garlan y Mary Shaw definen que la arquitectura es un nivel de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema". (David Garlan y Mary Shaw, 1994)

La arquitectura de software es el diseño de más alto nivel de la estructura de un sistema.

- Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software.
- Una arquitectura de software se selecciona y diseña con base en objetivos (requisitos) y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como el mantenimiento, la auditoria, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real.
- La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

➤ **Modelos o vistas**

Toda arquitectura de software debe describir diversos aspectos del software. Generalmente, cada uno de estos aspectos se describe de una manera más

comprensible si se utilizan distintos modelos o vistas. Es importante destacar que cada uno de ellos constituye una descripción parcial de una misma arquitectura y es deseable que exista cierto solapamiento entre ellos. Esto es así porque todas las vistas deben ser coherentes entre sí, evidente dado que describen la misma cosa.

Cada paradigma de desarrollo exige diferente número y tipo de vistas o modelos para describir una arquitectura. No obstante, existen al menos tres vistas absolutamente fundamentales en cualquier arquitectura:

- La visión **estática**: describe qué componentes tiene la arquitectura.
- La visión **funcional**: describe qué hace cada componente.
- La visión **dinámica**: describe cómo se comportan los componentes a lo largo del tiempo y cómo interactúan entre sí.

Las vistas o modelos de una arquitectura de software pueden expresarse mediante uno o varios lenguajes. El más obvio es el lenguaje natural, pero existen otros lenguajes tales como los diagramas de estado, los diagramas de flujo de datos, etc. Estos lenguajes son apropiados únicamente para un modelo o vista. Afortunadamente existe cierto consenso en adoptar UML (*Unified Modeling Language*, lenguaje unificado de modelado) como lenguaje único para todos los modelos o vistas. Sin embargo, un lenguaje generalista corre el peligro de no ser capaz de describir determinadas restricciones de un sistema de información (o expresarlas de manera incomprensible).

➤ **Arquitecturas más comunes**

Generalmente, no es necesario inventar una nueva arquitectura de software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Así, las arquitecturas más universales son:

- Descomposición Modular. Donde el software se estructura en grupos funcionales muy acoplados.
- Cliente-servidor. Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.

- Arquitectura de tres niveles. Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente. (Zapata Sánchez, Andrés Felipe. Arquitectura de Software, 2019)

2.14.1 Patrón de Modelo Vista Controlador

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento

➤ Descripción del patrón

De manera genérica, los componentes de MVC se podrían definir como sigue:

- El **Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- El **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un

cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

- La **Vista**: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto requiere de dicho 'modelo' la información que debe representar como salida.

➤ **Interacción de los componentes**

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo de control que se sigue generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a

los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.

- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

➤ **Ventajas de MVC**

Las principales ventajas del uso del patrón MVC son (4):

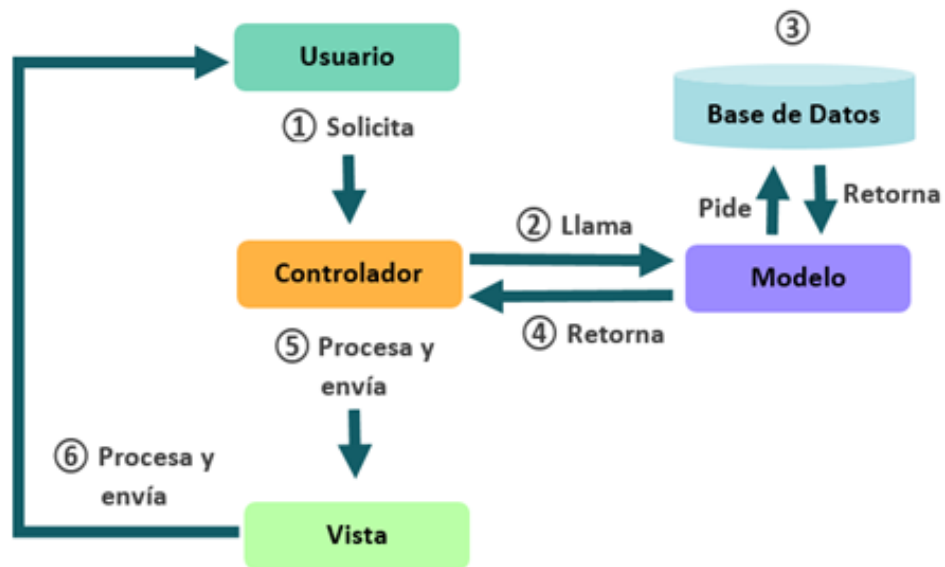
1. La separación del Modelo y la Vista, lo cual logra separar los datos, de su representación visual.
2. Facilita el manejo de errores.
3. Permite que el sistema sea escalable si es requerido.
4. Es posible agregar múltiples representaciones de los datos.

➤ **Desventajas de MVC**

Las principales desventajas del uso del patrón MVC son (3):

1. La cantidad de archivos que se deben mantener incrementa considerablemente.
2. La curva de aprendizaje es más alta que utilizando otros modelos.
3. Su separación en capas, aumenta la complejidad del sistema.

Figura No 2.3: CICLO DE VIDA DEL MVC



Fuente: (Rodrigo Gómez, 2015)

2.15 HERRAMIENTAS DE DESARROLLO

2.15.1 Gestor de Base de Datos MySQL

Gestor de base de datos (MySQL), es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. En cuanto a la definición general, MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional. Tus datos deberían ser la principal preocupación al elegir el software RDBMS correcto. Con su sistema de privilegios de acceso y la administración de cuentas de usuario, MySQL establece un alto estándar de seguridad. La verificación basada en el host y el cifrado de contraseña están disponibles. (Oracle Corporation, 2010)

2.15.2 Lenguaje de Programación

PHP (PHP Hypertext Pre-processor). Lenguaje de programación usado generalmente en la creación de contenidos para sitios web. PHP es un lenguaje de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

Fue uno de los primeros lenguajes del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también interfaz de línea de comandos que puede ser usada en aplicaciones graficas independientes. (Rasmus Lerdorf, 2002)

2.15.3 Framework Codeigniter

Como cualquier otro framework, Codeigniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. Esto es, marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. Codeigniter implementa el proceso de desarrollo llamado Model View Controller (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales

Codeigniter es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas.

También hay que destacar que Codeigniter es más rápido que muchos otros entornos. Incluso en una discusión sobre entornos de desarrollo con PHP, Rasmus Lerdorf, el creador de PHP, expresó que le gustaba Codeigniter porque es rápido, ligero y parece poco un entorno.(EllisLab, 2007)

Las herramientas de la ingeniería de software proporcionan el soporte automatizado o semi-automatizado para el proceso y los métodos. Cuando las herramientas se integran de forma que la información que cree una de ellas pueda usarla otra, se dice que se ha establecido un sistema para el soporte del desarrollo del software, que con frecuencia se denomina ingeniería del software asistida por computadora (Pressman, 2006).

2.16 MÉTODOS DE PRUEBA

Al finalizar las fases del ciclo de vida para la entrega de un programa y su utilización, está la fase de pruebas.

La fase de prueba es un conjunto de actividades que se planean con anticipación y se realiza de manera sistemática, el cual debe incluir técnicas y métodos de diseño de casos de prueba.

2.16.1 Tipos De Métodos De Prueba

Es el diseño de pruebas que tiene la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y de tiempo.

Los productos de ingeniería de software se pueden probar de dos formas:

- Métodos de caja blanca
- Métodos de caja negra

2.16.2 Métodos De Caja Blanca

La realización de pruebas para verificar que líneas específicas de código funcionan tal como está definido, se le conoce como prueba de caja-transparente (Pressman, 2005).

La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba.

Las pruebas de caja blanca intenta garantizar que:

- a) Se ejecutan al menos una vez todos los caminos independientes de cada modulo
- b) Se utilizan las decisiones en su parte verdadera y en su parte falsa

- c) Se ejecutan todos los bucles en sus límites
- d) Se utilizan todas las estructuras de datos internas.
- e) Para esta prueba se consideran tres importantes puntos.
- f) Conocer el desarrollo interno del programa, determinante en el análisis de coherencia y consistencia del código.
- g) Considerar las reglas predefinidas por cada algoritmo.
- h) Comparar el desarrollo del programa en su código con la documentación pertinente.

2.16.2.1 Tipos De Pruebas De Caja Blanca

De estructura de datos locales: Se centran en el estudio de las variables del programa. Busca que toda variable esté declarada y que no existan con el mismo nombre, ni declaradas local y globalmente, que haya referencias a todas las variables y para cada variable, analiza su comportamiento en comparaciones.

- De Cobertura de Sentencias: Comprueba que todas las sentencias se ejecuten al menos una vez.
- De Cobertura de Decisión: Ejecuta casos de prueba de modo que cada decisión se pruebe al menos una vez a Verdadero (True) y otra a Falso (False).
- De Cobertura de Condición: Ejecuta un caso de prueba a True y otro a False por cada condición, teniendo en cuenta que una decisión puede estar formada por varias condiciones.
- De Cobertura de Condición/Decisión: Se realizan las pruebas de cobertura de condición y las de decisión a la vez.
- De Condición Múltiple: Cada decisión multicondición se traduce a una condición simple, aplicando posteriormente la cobertura de decisión.
- De Cobertura de Caminos: Se escriben casos de prueba suficientes para que se ejecuten todos los caminos de un programa. Entendiéndose camino como una secuencia de sentencias encadenadas desde la entrada del programa hasta su salida.

2.16.2.2 Prueba Del Camino Básico

Buscando una mejor comprensión de los contenidos, se hace importante definir primeramente algunos conceptos fundamentales:

Camino: “Secuencia de todas las instrucciones de un programa de principio a fin”. Un camino se puede definir como la ruta de secuencias que se siguen dentro del código fuente de un programa, desde la entrada de valores al sistema hasta la devolución de resultados que arroja, respetando la estructura de código.

Camino Básico: Es una técnica de prueba de Caja Blanca que permite obtener una medida de complejidad lógica para generar un conjunto básico de caminos que se ejecutan por lo menos una vez durante la ejecución del programa.

La prueba del camino básico es una técnica de pruebas de Caja Blanca propuesta por Tom MacCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control.

Camino independiente: “Es cualquier camino del programa que incluye nuevas instrucciones de un proceso o una nueva condición”.

El conjunto de caminos independientes se obtiene construyendo el Grafo de Flujo asociado y se calcula su complejidad ciclomática. Por último se diseñan los casos de prueba y se ejecutan los mismos.

Complejidad: Es proporcional al número de errores en un segmento de código. “Entre más complejo, más susceptible a errores”. Se relaciona con el esfuerzo requerido para probar. “Entre más complejo, mayor atención para probar”.

Complejidad ciclomática: Es la “medida de la complejidad lógica de un módulo” y el esfuerzo mínimo necesario para calificarlo. Es el número de rutas lineales independientes de un módulo, por lo tanto es el número mínimo de rutas que deben probarse”.

Esta técnica ofrece una gran ventaja con respecto a las otras técnicas, ya que el número mínimo requerido de pruebas se sabe por adelantado y por tanto el proceso de prueba se puede planear y supervisar en mayor detalle.

Los pasos a seguir para aplicar esta técnica son:

1. Representar el programa en un grafo de flujo.
2. Calcular la complejidad ciclomática.
3. Determinar el conjunto básico de caminos independientes.
4. Derivar los casos de prueba que fuerzan la ejecución de cada camino.

A continuación, se detallan cada uno de estos pasos.

1. **Representación de un grafo de flujo:**

El grafo de flujo se utiliza para representar el flujo de control lógico de un programa. Este emplea los tres elementos siguientes:

- **Nodos:** Representan cero, una o varias sentencias en secuencia. Cada uno comprende como máximo una sentencia de decisión (bifurcación).
- **Aristas:** Líneas que unen dos nodos.
- **Regiones:** Áreas delimitadas por aristas y nodos. Cuando se contabilizan las regiones de un programa debe incluirse el área externa como una región más. Así, cada construcción lógica de un programa tiene una representación. Un grafo de flujo del diagrama de módulos correspondiente.

2.16.3 Métodos De Caja Negra

Se refiere a que las entradas apropiadas producen resultados esperados. Los datos de prueba son escogidos en función a las especificaciones del problema, sin importar los detalles internos del programa a fin de verificar que el programa tenga la funcionalidad correcta.

Las pruebas de caja negra se refieren por algunos de los siguientes métodos:

- **Prueba del camino básico:** le permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizaran que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

- **Prueba de la estructura del control:** En este tipo de prueba se contempla el método del proceso básico mencionado que permiten ampliar la cobertura de la prueba y mejorar su calidad.

Con este equipo de pruebas de intenta encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

2.16.3.1 Limitaciones

Lograr una buena cobertura con pruebas de caja negra es un objetivo deseable; pero no suficiente a todos los efectos. Un programa puede pasar con holgura millones de pruebas de especificación y sin embargo tener defectos internos que surgen en el momento más inoportuno.

Las pruebas de caja negra nos convencen de que un programa puede realizar bien sus funcionalidades programadas, pero no de que haga otras cosas menos aceptables.

2.17 MÉTODOS DE ESTIMACIÓN DE COSTOS DE SOFTWARE (COSMIC)

La medición y estimación de software utilizando COSMIC⁷, es un método de segunda generación que determina el tamaño del software a partir del número de interacciones entre los componentes de los requerimientos funcionales.

Tiene la ventaja que no establece límites arbitrarios al tamaño funcional, por lo cual pueden medirse componentes de software independientemente de si son muy grandes o pequeños. Adicionalmente, el análisis para la medición está basado en el desglose funcional de los componentes de software, por lo que está alineado con las prácticas de Ingeniería de software. Es un método de análisis de puntos de función de segunda generación, el grupo se nombró a sí mismo como el Common Software Measurement International Consortiom (COSMIC, 1998).

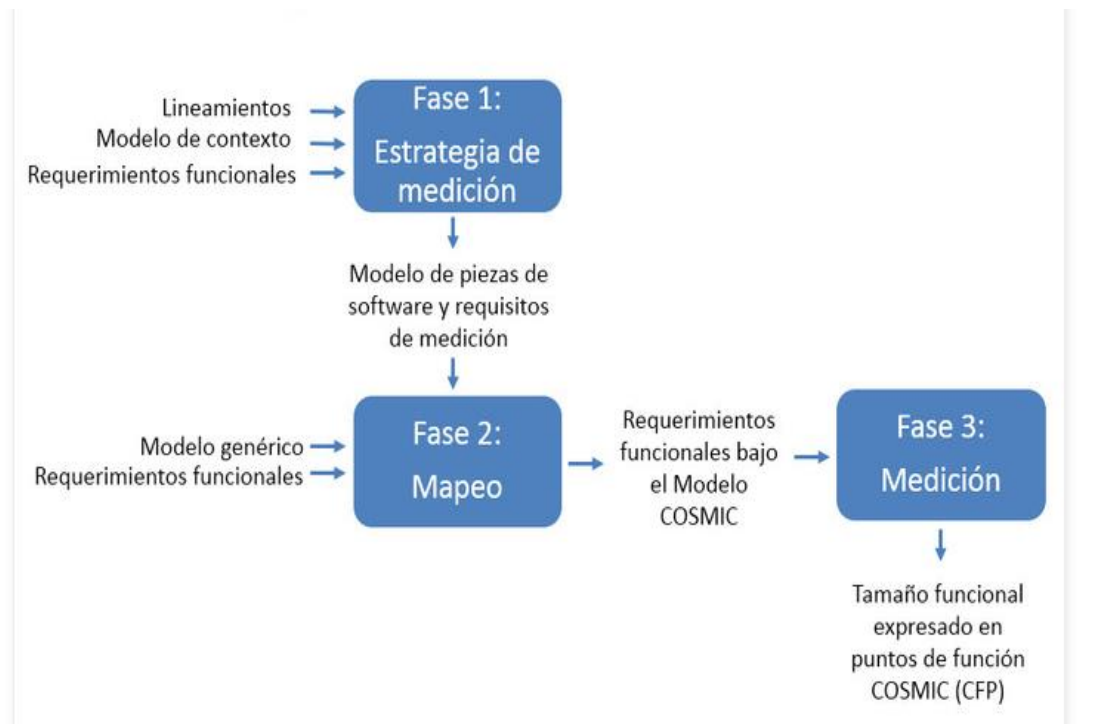
⁷ Cosmic “Common Software Measurement International Consortiom”

El tamaño de un software es la principal variable necesaria para determinar el esfuerzo de desarrollo que deberá invertirse para implementarlo. La medición y estimación de software utilizando COSMIC, es un método de segunda generación que determina el tamaño del software a partir del número de interacciones entre los componentes de los requerimientos funcionales.

Estandarizado bajo la ISO 19761, el método COSMIC puede aplicarse a diversos tipos de software, incluyendo aplicaciones de negocios, sistemas de información gerencial, software en tiempo real, infraestructura, e inclusive software científico y de ingeniería.

ISO estableció un grupo de trabajo para estudiar y definir los principios de la medición del tamaño funcional, se publicó un primer estándar, el ISO 14143/1 que si bien ayudo a mejorar el entendimiento de los principios, no logró resolver las oportunidades de mejora que tenían los métodos de puntos de función de ese momento.

Figura No 2.4: ESTRATEGIA DE MEDICIÓN (COSMIC)



Fuente: (COSMIC, 1998)

➤ **Método COSMIC. Fase 1: Estrategia de medición**

Lo primero que se realiza en una medición y estimación de software con COSMIC, es determinar qué es lo que se va a medir.

Una medición de software depende del punto de vista de lo que definimos como usuarios funcionales, tales como personas, dispositivos de hardware u otros sistemas que interactúan con el software.

En esta primera fase se define el propósito y alcance de la medición de software, que incluye cuales son los requerimientos funcionales de usuario que se van a medir, quienes son los usuarios funcionales y otros parámetros. Previo a esto, es necesario haber aplicado **técnicas para el levantamiento de requerimientos de software**.

Es importante dejar documentados los parámetros de la medición de software, para asegurar que esta pueda ser interpretada adecuadamente por quienes harán uso de ella para realizar las estimaciones y presupuestos.

➤ **Método COSMIC. Fase 2: Mapeo**

En una medición COSMIC, el mapeo se realiza para crear un modelo COSMIC de los requerimientos funcionales de usuario.

El punto de partida para el mapeo son los artefactos disponibles, como por ejemplo un esquema o especificación de requerimientos detallada, modelos de diseño como por ejemplo los **casos de uso**, software que está instalado físicamente, entre otros.

Para elaborar este modelo, se utilizan los principios del Modelo genérico de software COSMIC, aplicados a los requerimientos de software que se van a medir.

El modelo de requerimientos de software COSMIC tiene 4 principios:

1.- La funcionalidad de software está comprendida de procesos funcionales. La tarea de cada proceso funcional es responder a un evento ocurrido fuera de la frontera del sistema (el mundo de los usuarios funcionales).

2.- Los procesos funcionales están compuestos de sub-procesos:

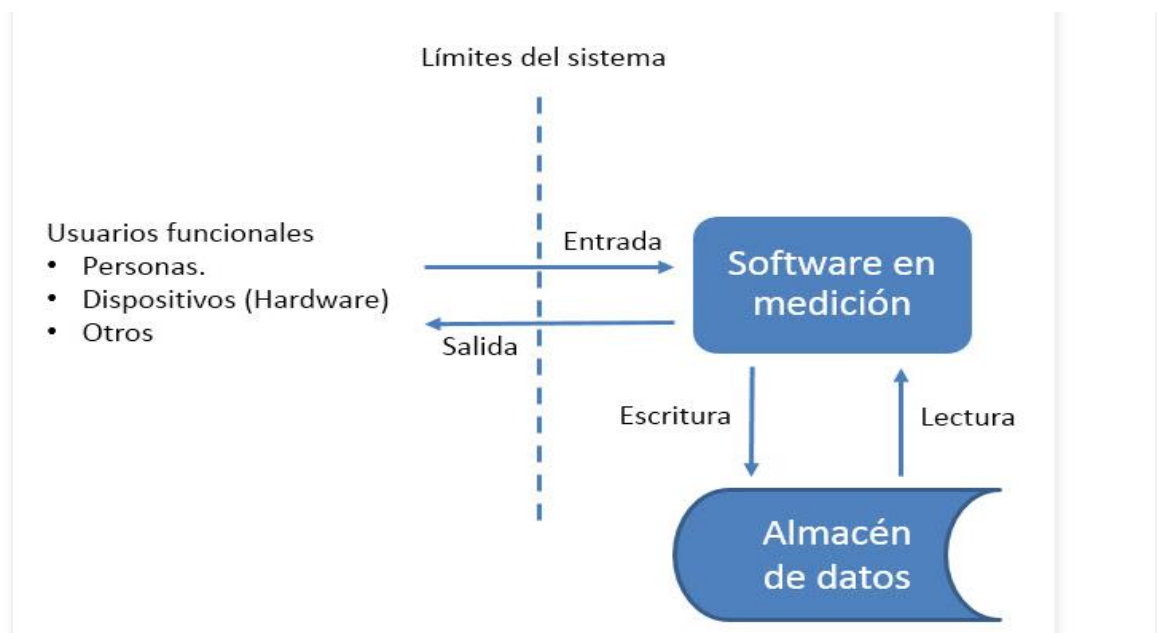
- Cada sub-proceso puede mover datos o manipular datos.
- Los sub-procesos de movimiento de datos que mueven datos de un usuario funcional a un proceso funcional se les llaman “Entradas”.
- Los sub-procesos que mueven datos desde un proceso funcional hacia el exterior se les llaman salidas.
- Los sub-procesos que mueven datos hacia un almacén de datos se les llaman “Escrituras” mientras que a los que mueven datos desde dichos almacenes se les conoce como "lecturas".

3.- Cada movimiento de datos (Entrada, salida, lectura o escritura) moviliza solamente un grupo de datos, cuyos atributos describen un solo objeto de interés.

4.- Se asume que la manipulación de datos forma parte de las entradas, salidas, lecturas o escrituras, por lo tanto, estas no se miden por separado (En la medición solo se cuentan los movimientos de datos).

La siguiente figura ilustra los tipos de sub-procesos definidos por el modelo COSMIC.

Figura No 2.5: LOS 4 TIPOS DE MOVIMIENTOS DE DATOS



Fuente: (COSMIC, 1998)

Se entiende que un proceso funcional termina su ejecución cuando ha realizado todos los sub-procesos necesarios para responder a los datos que recibió del evento.

➤ **Método COSMIC. Fase 3: Medición**

La unidad de medida del método COSMIC es el “punto de función COSMIC” (CFP⁸).

- La medición de la nueva pieza de software se realiza identificando todos los movimientos de datos, es decir todas las entradas, salidas, lecturas y escrituras de cada proceso funcional. Luego sumándolas todas.
- Todo proceso funcional debe tener al menos dos movimientos de datos (al menos una entrada y una salida o una escritura). Solo de esta forma se garantiza que el proceso funcional modelado proporciona un servicio completo. Por lo tanto, el tamaño funcional mínimo de un proceso es de 2 CFP.
- No existe un límite superior al tamaño de un proceso funcional.
- Para realizar mediciones sobre mejoras a piezas de software existente, se identifican todos los movimientos de datos que se van a agregar, modificar o eliminar, sumándolos todos en cada uno de sus procesos funcionales. El tamaño mínimo de una modificación es de un CFP.

2.18 DESCRIPCIÓN DEL MODELO

Un sistema de amortización es un método por el cual un capital cedido en préstamo es devuelto por una sucesión de pagos o cuotas. Estas cuotas periódicas constituyen una renta cuyo valor actual debe ser igual al préstamo otorgado. Se puede suponer que cualquier sistema de amortización es una renta con pagos vencidos, ya que si la primera cuota se pagara al momento del préstamo sería equivalente a considerar un préstamo de menor valor con cuotas vencidas. Entonces si el préstamo es por un monto V y las cuotas de devolución son C_1, C_2, \dots, C_n entonces el valor actual de dicha renta al momento del préstamo deberá ser V .

⁸ CFP unidad de medida método COSMIC “punto de función **COSMIC**”.

Existen diferentes sistemas de amortización. Dos de los más conocidos son el sistema alemán, que utiliza cuotas variables, decrecientes en forma aritmética y el sistema francés en el cual la deuda se amortiza en cuotas constantes.

2.18.1 Características De Un Sistema De Amortización

Considerando el interés de la empresa que es: de 1 a 8 cuotas 0,07%; de 9 a 12 cuotas 0,05 y de 13 cuotas en adelante es de 0.035%.

Consideramos sistemas de amortización en el cual el préstamo V es devuelto en N cuotas en el tiempo C_1, C_2, \dots, C_n . Tomaremos como unidad de tiempo el lapso entre dos cuotas consecutivas y como origen del tiempo al momento del préstamo. Así el préstamo está ubicado en $T=0$.

Cada cuota del sistema de amortización se compone de dos partes:

$$C_k = V_k + S_k$$

Donde V_k se denomina cuota de amortización real y S_k es la cuota de interés. La suma de las n cuotas de amortización real es igual al valor del préstamo:

$$V = V_1 + V_2 + \dots + V_n$$

Mientras que las cuotas de interés se calculan como el interés sobre las cuotas de amortización aún no pagadas: $S_k = i * (V_k + \dots + C_n)$.

Esto significa que en cada cuota el deudor paga una parte del capital prestado, V_k y los intereses sobre el capital aun adeudado, S_k .

En particular, al momento de haber pagado la K -ésima cuota, el monto adeudado del préstamo es:

$$V - (V_1 + V_2 + \dots + V_k) = V_{k+1} + \dots + V_n \quad 1 \leq K < N.$$

Y esto coincide con el valor actual de las cuotas que restan pagar:

$$VA_k = c_{k+1} \frac{1}{1+i} + c_{k+2} \left(\frac{1}{1+i} \right)^2 + \dots + c_n \left(\frac{1}{1+i} \right)^{n-k}.$$

Cabe destacar que la renta finaliza al momento del pago de la última cuota de amortización real, ya que de esta manera se completa el pago del préstamo V y por ende no hay más interés por cobrar.

Ejemplo: Un préstamo de Bs. 1000 que se amortiza en tres cuotas cada 30 días, y cuyas cuotas de amortización real son de bs 300, bs 300, y bs 400 respectivamente y la tasa de interés efectiva mensual es del 2 %.

Las cuotas a pagar estarán conformadas de la siguiente manera:

Cuota K	Amortización real V_k	Cuota de interés S_k	Cuota C_k	Saldo adeudado VA_k
1	Bs 300	Bs 20 (1000 * 0,02)	Bs 320	Bs 700,00
2	Bs 300	Bs 14 (700 * 0,02)	Bs 314	Bs 400,00
3	Bs 400	Bs 8 (400 * 0,02)	Bs 408	Bs 0,00

Como ejemplo, la fila correspondiente a la cuota 2 debe leerse así: en la cuota 2, paga Bs 300 de amortización real más Bs 14 de interés, por lo que la cuota es de Bs 314. El saldo adeudado luego de pagar la cuota resulta de Bs 400.

El valor actual de la renta al momento del préstamo es:

$$VA_0 = 320 \cdot \frac{1}{1,02} + 314 \cdot \frac{1}{1,02^2} + 408 \cdot \frac{1}{1,02^3} = 1000,$$

Es decir el monto total del préstamo. El valor actual de la renta calculado inmediatamente después de pagar la primera cuota es:

$$VA_1 = \frac{314}{1,02} + \frac{408}{1,02^2} = 700,$$

Es decir, el saldo adeudado a ese momento.

2.18.2 Sistema Alemán De Amortización

El sistema alemán es un sistema de amortización donde las cuotas de amortización reales son todas iguales. Es decir, si se prevén n cuotas, entonces cada cuota de amortización real es igual a

$$v_k = \frac{V}{n}, \quad k = 1, 2, \dots, n.$$

Por lo tanto, el interés que se paga en cada cuota está dado por:

$$s_k = i \cdot (v_k + \dots + v_n) = i \cdot (n + 1 - k) \cdot \frac{V}{n}.$$

Como se puede observar, los valores S_1, S_2, \dots , decrecen en forma aritmética:

$$s_{k+1} = s_k - i \frac{V}{n}.$$

Como las cuotas de amortización son constantes, esto implica que las cuotas del sistema de amortización C_k , también decrecen en forma aritmética: $C_{k+1} = C_k - iV/n$. Dado que la primera cuota es igual a la cuota de amortización más el interés sobre todo el préstamo:

$$c_1 = \frac{V}{n} + i \cdot V,$$

Y las cuotas disminuyen en $i = \frac{V}{n}$, se sigue que las siguientes cuotas están dadas por

$$\begin{aligned} c_k &= c_1 - (k-1) \cdot i \frac{V}{n} \\ &= \frac{V}{n} (1 + i(n-k+1)), \quad k = 2, 3, \dots, n. \end{aligned}$$

Se puede ver además para este caso particular, que la renta determinada por las cuotas C_1, C_2, \dots, C_n es un sistema de amortización, es decir que el valor actual de las mismas al momento del préstamo es igual al préstamo V . En efecto, teniendo en cuenta la fórmula para el cálculo del valor actual de una renta en progresión aritmética con cuotas vencidas con cuota inicial $c = V$ $c = \frac{V}{n} + iV$ y razón de progresión $h = -i + \frac{V}{n}$ se tiene que

$$\begin{aligned}
VA_0 &= \left(\frac{V}{n} + iV\right) a_{\overline{n}|i} - i \frac{V}{n} \left(\frac{a_{\overline{n}|i} - n \cdot (1+i)^{-n}}{i}\right) \\
&= \frac{V}{n} a_{\overline{n}|i} + iV a_{\overline{n}|i} - \frac{V}{n} a_{\overline{n}|i} + V(1+i)^{-n} \\
&= V((1+i)^{-n} + i a_{\overline{n}|i}) = V.
\end{aligned}$$

Es decir, el valor actual de la renta es igual al préstamo otorgado.

Con el mismo razonamiento se puede ver que el valor actual en $t = k$ de la anualidad compuesta por las cuotas $C_{k+1}, C_{k+2}, \dots, C_n$ es igual a la parte del préstamo aun no amortizado.

Esto es

$$VA_k = v_{k+1} + v_{k+2} + \dots + v_n.$$

En efecto, en $t = k$ resta el pago de $n - k$ cuotas que conforman una renta en progresión aritméticas, con razón $h = -iV/n$ y primer término C_{k+1} . Así.

$$\begin{aligned}
VA_k &= c_{k+1} \cdot a_{\overline{n-k}|i} - i \frac{V}{n} \left(\frac{a_{\overline{n-k}|i} - (n-k)(1+i)^{-(n-k)}}{i}\right) \\
&= \left(\frac{V}{n} + i(n-k) \frac{V}{n}\right) a_{\overline{n-k}|i} - \frac{V}{n} a_{\overline{n-k}|i} + \frac{V}{n} (n-k)(1+i)^{-(n-k)} \\
&= (n-k) \frac{V}{n} (a_{\overline{n-k}|i} i + (1+i)^{-(n-k)}) \\
&= (n-k) \frac{V}{n}
\end{aligned}$$

Tenemos entonces que el sistema alemán cumple con las propiedades de un sistema de amortización. Una característica de este sistema es que las cuotas son decrecientes. Esto tiene la desventaja de que las primeras cuotas son de mayor valor monetario, y por lo tanto más difíciles de afrontar para el deudor. Una alternativa que suele usarse es modificar el sistema alemán variando la tasa de interés. Así, se calculan primero los valores de las cuotas para una tasa baja de interés y luego de pagar algunas cuotas se refinancia la deuda con una tasa de interés más alta.

Ejemplo: un préstamo por Bs 10.000 a pagar en cuatro cuotas, aplicando el sistema alemán con una tasa de interés de 2% para las dos primera cuotas y del 4% para las dos últimas.

Para las dos primeras se tiene:

$$C_1 = 2.500 + 200 = 2.700$$

$$C_2 = 2.700 - 0.02 * 2.500 = 2.650$$

El saldo adeudado al finalizar la segunda cuota es de Bs 5.000 y la refinanciación implica que las próximas cuotas serán:

$$C_3 = 2.500 + 0.04 * 5.000 = 2.700$$

$$C_4 = 2.700 - 0.04 * 2.500 = 2.600$$

La anualidad de cuotas Bs 2.700, Bs 2.650, Bs 2.700, Bs 2.600 tiene un valor próximo a Bs 10.000 si se utiliza una tasa de interés constante del 2.6 %. Sin embargo si se aplica el sistema alemán con una tasa fija de 2.6%, las dos primeras cuotas serian iguales a:

$$C_1 = 2.500 + 260 = 2.760$$

$$C_2 = 2.760 - 0.026 * 2.500 = 2.695$$

Algo superior al sistema que utiliza dos tasas.

2.18.3 Sistema De Amortización Francés

El sistema francés es un sistema de amortización en el cual las n cuotas a pagar son todas iguales: es decir, $C_1 = C_2 = \dots = C_n = C$.

Para determinar el valor de C, tendremos en cuenta que el valor actual de la renta debe ser igual al préstamo V. Por otro lado, utilizando la fórmula de actualización de una renta con cuotas constantes y vencidas, este valor actual debe ser $c * a_{\overline{n}|i}$. Por lo tanto:

$$c = \frac{V}{a_{\overline{n}|i}} = V \frac{i}{1 - (1 + i)^{-n}}$$

Como hemos visto, cada cuota de la renta se compone de una amortización real V_k y una cuota de interés S_k . La cuota V_k es la parte del capital adeudado que se salda en el instante $t = k$. Así, si denotamos con VA_k el valor actual de la renta en $t = k$, entonces:

$$\begin{aligned} v_k &= VA_{k-1} - VA_k = c \cdot (a_{\overline{n-k+1}|i} - a_{\overline{n-k}|i}) \\ &= \frac{V \cdot i}{1 - (1 + i)^{-n}} \frac{(1 - (1 + i)^{-(n+1-i)}) - (1 - (1 + i)^{-(n-i)})}{i} \\ &= \frac{V}{1 - (1 + i)^{-n}} (1 + i)^{-(n-i)} (1 - (1 + i)^{-1}) \end{aligned}$$

Y usando que $1 - (1 + i)^{-1} = \frac{i}{1+i}$ concluimos que:

$$v_k = \frac{V \cdot i}{1 - (1 + i)^{-n}} (1 + i)^{-(n+1-k)} = c \cdot (1 + i)^{-(n+1-k)}.$$

Además puesto que $C = V_k + S_k$ se sigue que

$$s_k = c \cdot (1 - (1 + i)^{-(n+1-k)}).$$

Notemos que la sucesión de cuotas de amortización reales V_k es creciente ($1 \leq k \leq n$) mientras que S_k es decreciente.

Naturalmente, el valor actual de esta renta es V (pues de esa manera ha sido elegido C), y el valor final $V F$ al momento del pago de la última cuota es

$$VF = c \cdot s_{nr} = \frac{V \cdot r}{1 - (1 + i)^{-n}} \frac{(1 + i)^n - 1}{i} = V \cdot (1 + i)^n.$$

Ejemplo: un préstamo de Bs 1000000 es amortizable en 5 años, con el 15 % de interés anual sobre saldos. Los siguientes cuadros resumen los pagos a efectuar según los sistemas alemán y francés respectivamente.

Cuadros de amortización. Los siguientes cuadros muestran el valor de las cuotas a pagar según cada sistema, la composición de las mismas, y el saldo adeudado al comienzo del periodo.

Puede observarse que las primeras cuotas son mayores para el caso del sistema alemán y esta relación se invierte en las últimas cuotas.

Figura No 2.6: SISTEMA DE AMORTIZACIÓN ALEMÁN

Período	Capital adeudado		Intereses a		Amortización real		Cuota	
	Al comienzo del período	Fines del período	A fines del período	A fines del período	A fines del período	A fines del período		
1	1000000	150000	200000	350000				
2	800000	120000	200000	320000				
3	600000	90000	200000	290000				
4	400000	60000	200000	260000				
5	200000	30000	200000	230000				
suma		450000	1000000	1450000				

Fuente: Matemática financiera, 2003

Figura No 2.7: SISTEMA DE AMORTIZACIÓN FRANCÉS

Período	Capital adeudado		Intereses a		Amortización real		Cuota	
	Al comienzo del período	Fines del período	A fines del período	A fines del período	A fines del período	A fines del período		
1	1000000	150000	148315,55	298315,55				
2	851684,45	127752,67	170562,89	298315,55				
3	681121,56	102168,23	196147,32	298315,55				
4	484974,24	72746,14	225569,42	298315,55				
5	259404,83	38910,72	259404,83	298315,55				
suma		491577,76	1000000	1491577,76				

Fuente: Matemática financiera, 2003

2.19 MÉTODOS DE CONTROL DE INVENTARIOS

Para el control de inventarios se tiene diversos métodos de control, entre los cuales se puede describir:

- a) Método ABC, conocido como método 80/20, que consiste en dividir los productos en tres categorías según su importancia, cantidad y valor, así es más fácil identificar los productos más valiosos (ESERP, 2020).
- b) Método PEPS, conocido como el método que sigue el criterio de primero en entrar primero en salir, consiste en identificar los primeros artículos en entrar al almacén para que sean los primeros en salir a la venta o ser distribuidos y utilizados en la producción o provisión de un servicio (ESERP, 2020).
- c) Método EOQ o de cantidad económica de pedido, se utiliza cuando la empresa tiene una demanda y frecuencia de uso de inventario constantes en el tiempo (ESERP, 2020).

CAPITULO III

MARCO

APLICATIVO

3 INTRODUCCIÓN

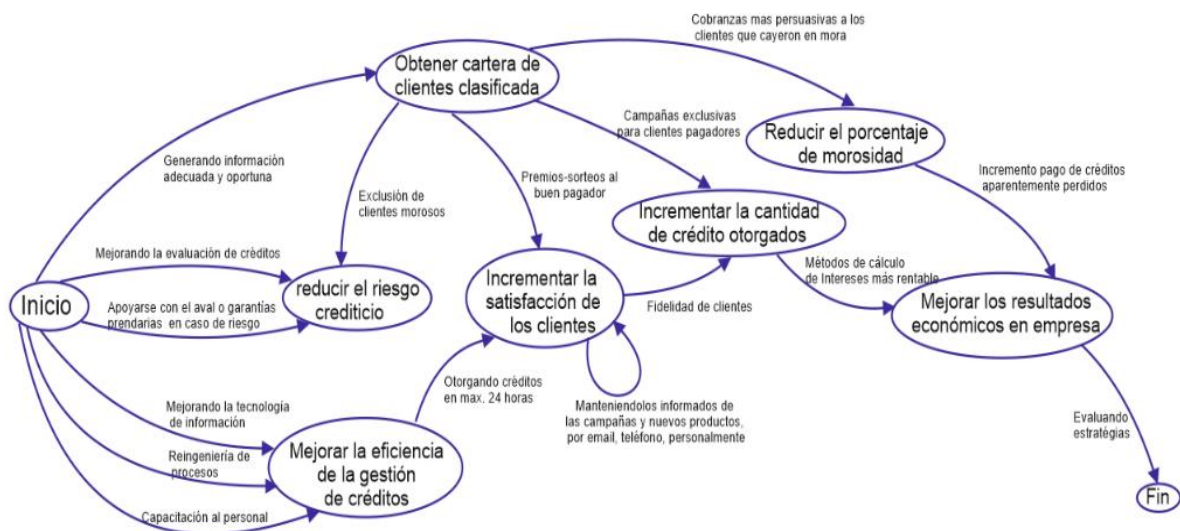
El propósito de capítulo es poner en práctica lo mencionado en los anteriores capítulos, que se analiza, desarrolla la metodología UWE, que satisfaga las necesidades en el análisis de créditos de la base de datos de la empresa COMERBA.

3.1 METODOLOGÍA UWE

3.1.1 Planificación

Lo primero que se realiza en este proceso es el funcionamiento del sistema, ya que tomando en cuenta todas las características que se han mencionado anteriormente, se ha encontrado varias soluciones al problema se consulta y verifica toda la información encontrada sobre el área de créditos. Luego se presenta modelos finales sobre cómo se elaborara el programa, los usuarios finales son los que evaluarán si el programa que se presenta es correcto o no.

Figura No 3.1: Diagramas de metas y estrategias



Fuente: (Elaboración propia)

En la figura No 3.1 se observa 2 metas iniciales: mejorar la eficiencia de la calidad de créditos, la primera meta se logra aplicando estrategias como: mejorar la tecnología de información, capacitar al personal y la segunda meta se logra al generar información adecuada y oportuna; también existen metas intermedias

como: reducir el riesgo crediticio la cual se obtiene mejorando la evaluación de créditos, apoyándose con avales o garantías prendarias y la exclusión de clientes morosos basándose en estas metas logradas; otras metas intermedias muy importantes para la empresa son reducir el porcentaje de morosidad a través de cobranzas más persuasivas a los clientes que cayeron en mora; por último se presenta el objetivo final el cual es mejorar los resultados económicos de la empresa, ello a través de 2 estrategias: incrementar el pago de los créditos aparentemente perdidos y nuevos métodos de cálculo de interés más rentables.

3.1.2 Diseño del Sistema

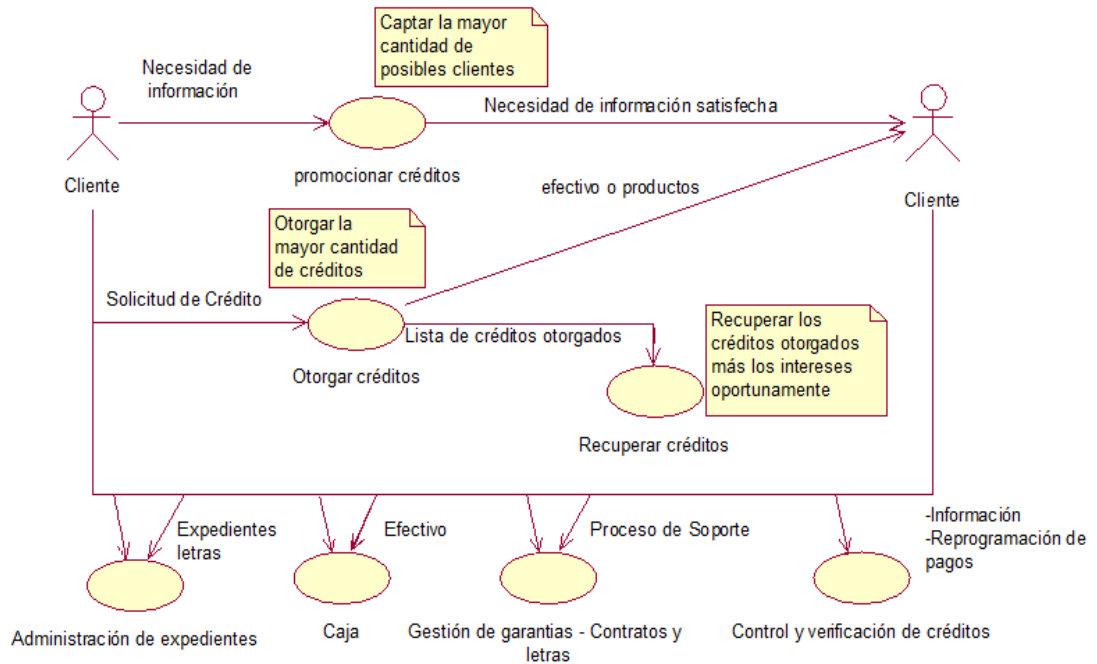
En el desarrollo del presente proyecto “SISTEMA DE INFORMACIÓN BASADO EN UN ENFOQUE DE PROCESOS PARA LA OPTIMIZACIÓN DE LA OTORGACIÓN DE CRÉDITOS” se debe seguir y ejecutar un conjunto de actividades que se encuentran en el proceso de desarrollo que se ha optado por emplear. Haciendo la referencia a la figura 3.2 que esta adelante.

3.1.2.1 Situación actual de la Empresa

Actualmente en la empresa existe un gran movimiento de activos, ya que se encuentra en una etapa de crecimiento, esto está haciendo que el sistema con el que la empresa cuenta actualmente (realizado excel) no cumpla con las funciones que la empresa necesita. El manejo de documentos dentro la empresa teniendo como consecuencia lo siguiente:

- Pérdida de tiempo: A la hora de buscar manualmente en excel los códigos de cliente y de garantía.
- Documentos duplicados: garantías que han sido entregados en dos agencias en el mismo mes.
- Mal servicio al cliente: al momento de buscar o llamar a las otras dos agencias para verificar si el cliente ya se encuentra registrado en alguna agencia.
- Documento inexistentes: algunos clientes solo registran sus datos en una agencia y al momento de solicitar un crédito en otra agencia el sistema en excel no reconoce si el cliente posee una garantía.

Figura No 3.2: SITUACIÓN ACTUAL

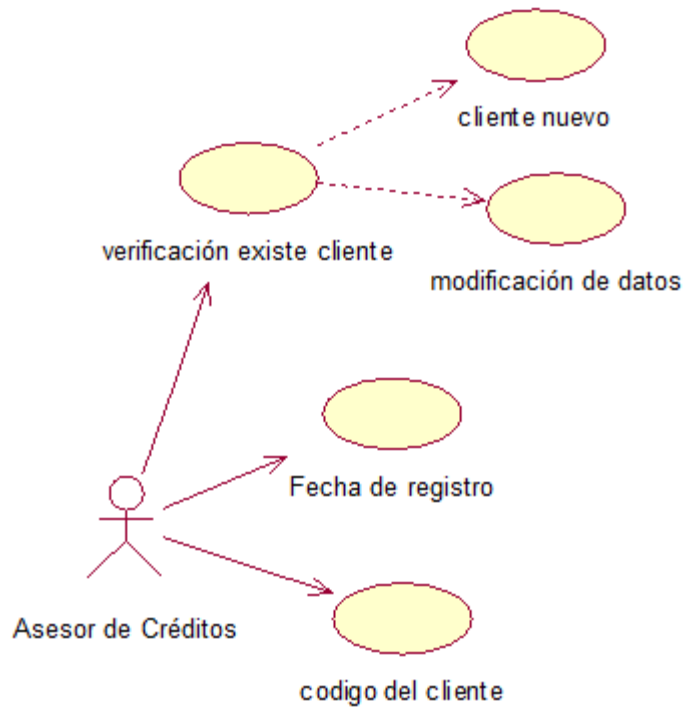


Fuente: (Elaboración propia)

Se describe la figura No 3.2 para tener un panorama general sobre los procesos del área de créditos de la empresa COMERBA. También se muestra como se inicia el proceso de gestión de créditos, este se inicia de 2 maneras: cuando el cliente solicita directamente un crédito, o cuando un cliente presenta interés por conocer los servicios que otorga la empresa, los tipos de créditos, las tasas, etc.

Los procesos antes mencionados se relacionan entre si estableciendo de esta manera flujos de información, de materiales; como se aprecia en la figura los procesos brindan a lineamientos de abastecimiento de información y control financiero, lineamientos de clasificación, lineamientos de ventas y promociones; y los procesos de expedientes, dinero en efectivo, recibos, reprogramaciones de pago, información diversa, nuevos crédito.

Figura No 3.3: REGISTRO DE NUEVOS CLIENTES

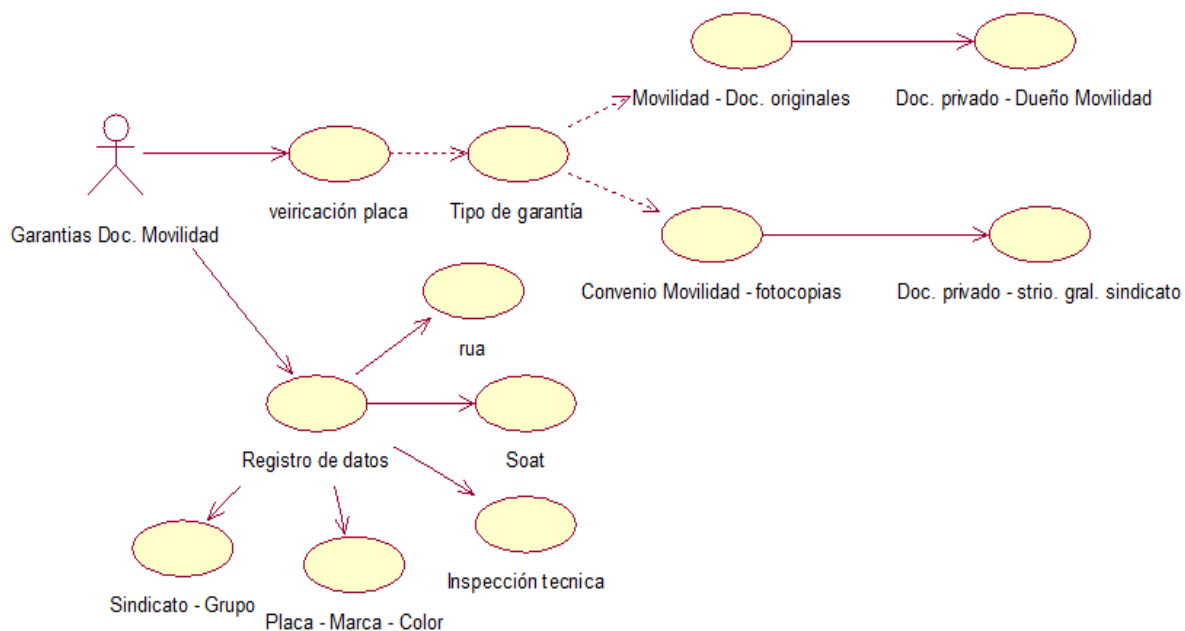


Fuente: (Elaboración propia)

En la figura No 3.3 se presenta el registro de nuevos clientes o actualización de sus datos en caso de que el cliente sea antiguo, se debe registrar los datos personales, la fecha de registro, en este caso el que realiza la modificación de datos es el asesor de créditos, se debe tener en cuenta los vínculos que existen entre clientes o el personal de la empresa, como conyugues, hermanos, padres, etc. El representante de una institución no debe ser menor de edad.

El asesor de créditos debe tener en información de los clientes utilizando un código de cliente que utiliza la empresa actualmente, de los créditos solicitados anteriormente, plan de pagos, los pagos realizados, las garantías vinculadas al crédito.

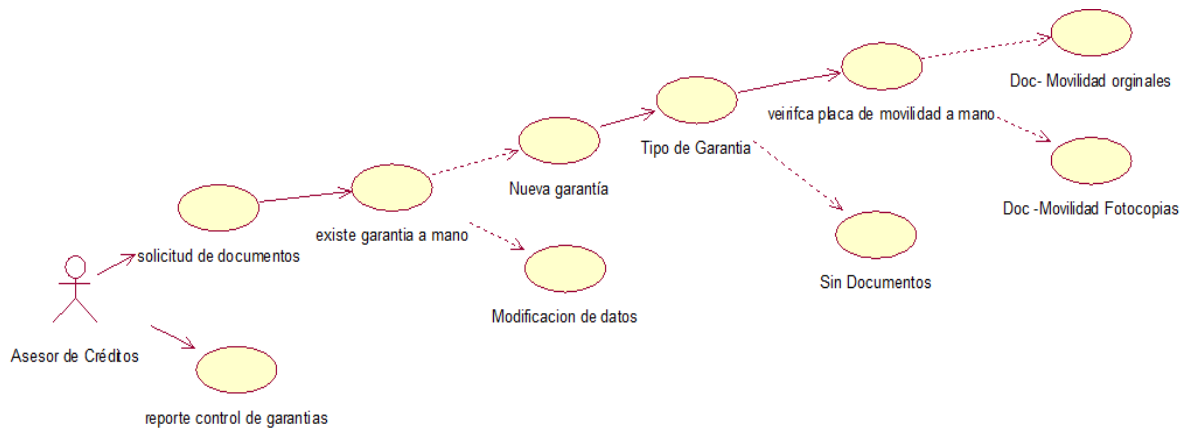
Figura No 3.4: SOLICITUD DE DOCUMENTOS GARANTÍAS



Fuente: (Elaboración propia)

En la figura 3.4 se presenta como la empresa maneja actualmente la parte de garantías, documentación de los clientes. Desde el tipo de garantías que va a proporcionar el cliente para solicitar un crédito. La empresa presenta dos opciones: uno es mediante documentos originales del vehículo y dos estar en un sindicato donde el representante deja documentos originales de su vehículo para que los socios puedan dejar fotocopias de los documentos del vehículo. El cliente debe tener una sola garantía, el asesor de créditos debe verificar que la documentación entregada a la empresa debe ser actualizada; para la parte de convenio se debe verificar que el representante del sindicato deba seguir trabajando en la institución y que la empresa COMERBA aun tenga convenios con dicha institución.

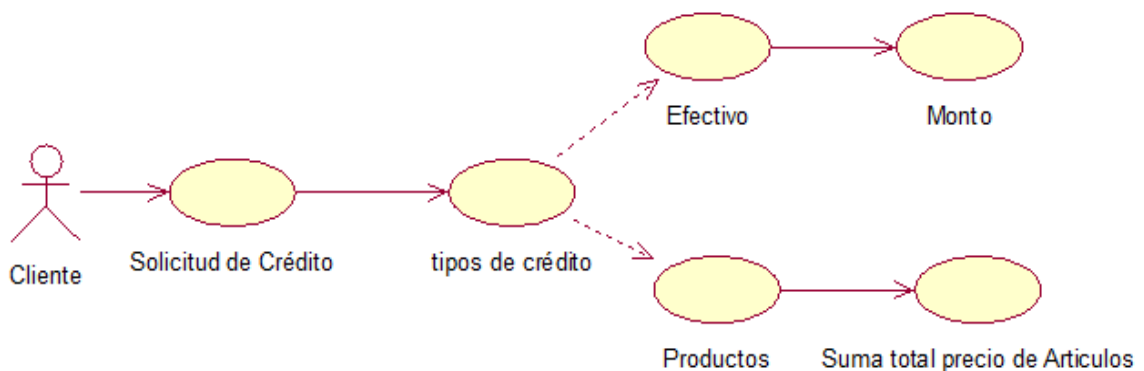
Figura No 3.5: REGISTRO DE GARANTÍAS



Fuente: (Elaboración propia)

La figura No 3.5 describe el proceso de registro de las garantías que se deben almacenar en la empresa, divididos por códigos, según la fecha de registro. El asesor de créditos revisa con el número de placa del vehículo antes de registrarlo como nuevo. En la parte de clientes sin documentos se refiere a clientes que son familiares o conocidos del gerente de la empresa. Al finalizar el registro el asesor de créditos realiza un reporte con los datos del cliente fecha de entrega de documentos y la agencia en la cual están siendo almacenadas.

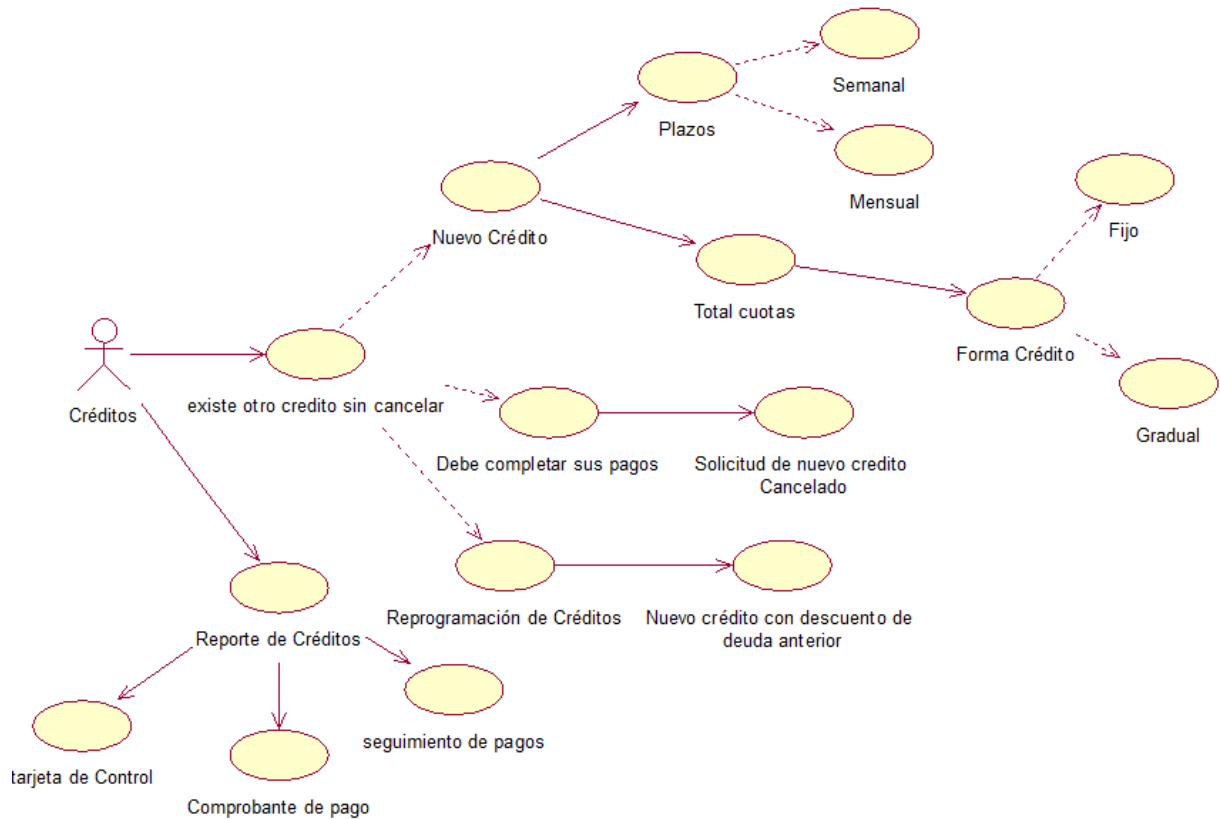
Figura No 3.6: SOLICITUD DE CRÉDITO



Fuente: (Elaboración propia)

En la figura 3.6 se muestra la solicitud de crédito que realiza el cliente, la empresa debe brindar información sobre los tipos de crédito, las tasas, los plazos, etc. En el caso de que el cliente este de acuerdo con el monto y el plazo del crédito negociado con el asesor, se inicia el proceso de nuevo crédito presentado a continuación.

Figura No 3.7: NUEVO CRÉDITO



Fuente: (Elaboración propia)

En la figura No 3.7 muestra el diagrama de procesos de nuevo crédito, si todo es conforme el asesor de créditos debe armar el expediente y exponerlo ante el gerente, el cual deberá tomar la decisión de aprobar o rechazar el crédito, el asesor de créditos debe verificar que el cliente no tenga una deuda de otro crédito solicitado anteriormente; en el caso de reprogramación de créditos se verifica que los datos de las garantías sean actuales, finalmente se entrega un expediente con el monto y los plazos acordados.

3.1.2.2 Rediseño de Procesos y Descripciones de tarea

Para realizar el rediseño de los procesos del área de créditos de la empresa COMERBA, se usaron conjuntamente los diagramas de casos de uso que plasman la realidad de cómo se llevan a cabo las actividades, además de ello se tomó en cuenta los conceptos y principios de la gestión de calidad especificados en la norma ISO 9000:2005, para realizar un modelo de procesos adecuado, que cumpla con el propósito de la organización, que ayude a la realización de sus metas y que asegure la calidad de los procesos a realizar, teniendo así una base sólida para la construcción del sistema de información que cumpla con los requisitos reales de la empresa.

A continuación se presenta los mapas y diagramas de casos de uso, los cuales fueron diseñados bajo notaciones de Erickson Penker (extensión de UML); la(s) tabla(s) de descripción de tarea a las que el sistema de información (SI) debe dar soporte y de las cuales se obtendrá los requisitos del sistema que ya son fácilmente visibles en la tabla, en ella se encuentra de manera resumida los componentes que intervienen en una determinada tarea, que la realizara un solo usuario el cual interactuara directamente con el SI, también se especifica detalladamente como se da la interacción usuario-SI y se especifica las reglas de negocio que se deberá tener un cuenta para dicha tarea.

3.1.2.2.1 Gestión de Créditos

Está compuesto por procesos estratégicos de soporte que interactúan entre sí para cumplir con los objetivos trazados por la empresa descritos anteriormente, entre estos procesos tenemos:

- Procesos estratégicos
 - Planificación estratégica
 - Gestión financiera
 - Gestión integral de riesgos
 - Gestión legal
 - Gestión de marketing

- Procesos misionales
 - Promocionar créditos
 - Otorgar créditos
 - Recuperar créditos
- Procesos de soporte
 - Administración de expedientes
 - Caja (recibidor-Pagador)
 - Gestión de garantías, contratos y letras
 - Operaciones auxiliares de créditos
 - Control y verificación de créditos

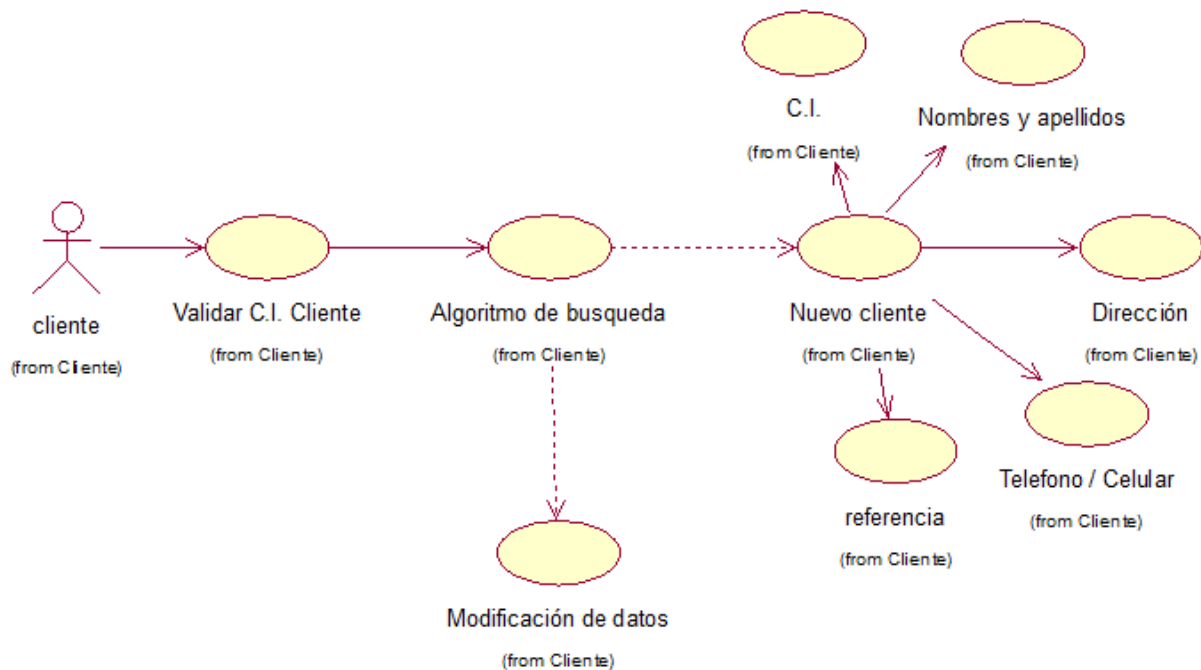
Todos los procesos antes mencionados se relacionan entre si estableciendo de esta manera flujos de información, materiales, políticas, etc. Los procesos estratégicos brindan a los procesos legales los lineamientos estratégicos, abastecimiento y control financiero, lineamientos de clasificación de cartera, lineamientos legales, lineamientos de ventas y promociones a los procesos misionales; y los procesos de soporte brindan los expedientes, dinero en efectivo, recibos, reprogramaciones de pago, nuevos créditos e información diversa de la misma manera los procesos de soporte reciben de los procesos misionales los expedientes actualizados, letras de pago y dinero en efectivo.

La manera como se inicia la gestión de créditos; este se inicia de 2 maneras: cuando el cliente (actor) solicita directamente un crédito a la empresa, en este caso se inicia inmediatamente el proceso de otorgamiento de crédito (cuyo objetivo es otorgar la mayor cantidad de créditos manteniendo un bajo riesgo crediticio), o cuando un cliente presenta interés por conocer los servicios que otorga la empresa, los tipos de créditos, las tasas, etc. En ese caso se inicia el proceso de promoción de créditos (su objetivo es captar la mayor cantidad de posibles clientes potenciales), el cual interactúa con el proceso gestión logística (proceso de soporte), ya que recibe materiales logísticos de este para que pueda iniciarse, posteriormente este proceso emite una lista de solicitudes de créditos y así da inicio al proceso de otorgamiento de créditos y este último entrega una lista de créditos aprobados al proceso de

recuperación de créditos (su objetivo es recuperar los créditos otorgados más los intereses oportunamente) dando fin a la gestión de créditos.

A continuación se detallaran mediante los diagramas de casos de usos todos aquellos procesos y subprocesos que interactúan con el sistema de información y necesitan soporte de el para su realización.

Figura No 3.8: REGISTRO NUEVO CLIENTE



Fuente: (Elaboración propia)

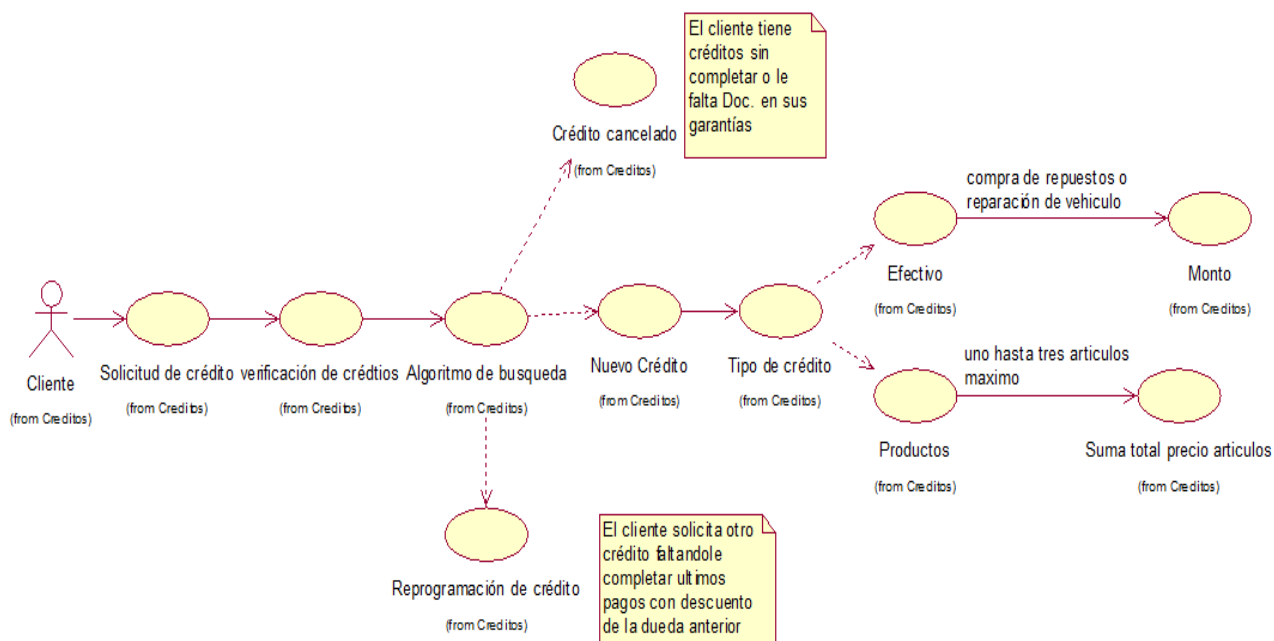
Proceso: otorgar créditos

Este es un proceso misional que tiene por objetivo otorgar la mayor cantidad de créditos pero manteniendo un riesgo crediticio bajo, por ello se enfatiza en la evaluación de créditos ya que a través de este proceso se decidirá si se brindara o no el crédito solicitado.

Como se observa en la Figura No 3.8 el sistema presenta un algoritmo de búsqueda de clientes mediante el documento de identidad del cliente, si es nuevo o los datos ya han sido registrados anteriormente para ser actualizados. En la figura 3.8 se

muestra el proceso de otorgar créditos se inicia cuando el cliente solicita un crédito o cuando el proceso de promoción de créditos entrega la solicitud de crédito de un cliente captado, en ese momento se inicia el subproceso evaluación de créditos (su objetivo es identificar correctamente el riesgo crediticio de cada posible cliente), al finalizar este subproceso, en caso de que se haya resuelto los documentos de crédito aprobado al subproceso de desembolso (generar documentos y entregar el efectivo al cliente), para que este pueda efectuar la entrega del efectivo y enviar la información necesaria para la recuperación del crédito.

Figura No 3.9: PROCESO OTORGAR CRÉDITOS



Fuente: (Elaboración propia)

Subproceso: evaluación de crédito

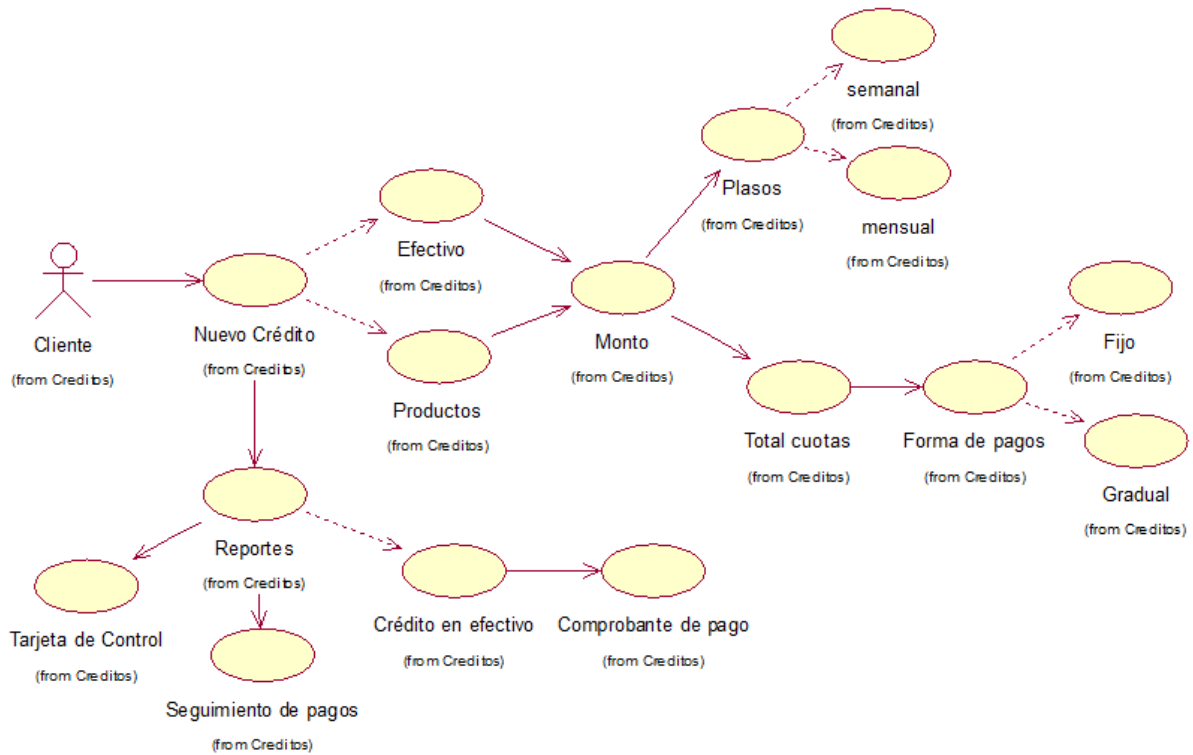
El subproceso de evaluación de créditos se inicia con una solicitud de crédito captada en el proceso de promoción de créditos o hecha directamente por un cliente, en primer lugar se registran los datos del cliente; datos generales, laborales, direcciones y vinculación que pueda tener con el personal que trabaja en la institución u otros clientes de la empresa, el asesor de créditos analiza rápidamente toda la información sobre el cliente especialmente el comportamiento crediticio que

tuvo (en caso de ser un cliente antiguo), y emite un veredicto, en caso de que haya posibilidades para la aprobación de esta solicitud, se continua el procedimiento, el asesor negocia con el cliente el monto y plazo en el que se le puede otorgar el crédito apoyándose en un simulador de pagos.

En el caso de que le cliente este de acuerdo con el monto y plazo del crédito negociado con el asesor, el cliente deberá conseguir y entregar los documentos necesarios para la evaluación correspondiente, seguidamente el asesor deberá recibir los documentos e información brindada por el cliente a través de una visita al domicilio del cliente (si lo cree necesario).

Si es todo conforme el asesor deberá armar el expediente y presentarlo al gerente, el cual tomara la decisión de aprobar o rechazar la solicitud, esta decisión deberá ser registrada en el sistema, en caso de ser positiva la respuesta, el asesor de créditos prepara el expediente y todo los documentos necesarios para el desembolso correspondiente y lo envía al operador (inicio del subproceso de desembolso).

Figura No 3.10: NUEVO CRÉDITO



Fuente: (Elaboración propia)

Del procedimiento anterior Figura 3.10 se obtiene 4 tareas fundamentales en las cuales interactúan los actores con el sistema de información, estas tareas son: registra o actualizar datos del cliente, consultar historial crediticio, gestión de la solicitud y cambiar el estado a la solicitud; las descripciones de cada una de las tareas serán la base para la posterior captura de requisitos.

Tabla No 3.1: REGISTRAR O ACTUALIZAR DATOS DEL CLIENTE

Nombre: Registrar o actualizar datos del cliente			
Tareas: Registrar datos del cliente, actualizar datos del cliente			
Proceso de negocio: Evaluación de crédito		Rol: Asesor de crédito	
Disparadores: Solicitud de crédito			
Entrada		Salida	
Objeto de datos	estado	Objeto de datos	estado
Registro de datos	Desactualizado o nulo	Registro de datos	Actualizado

cliente	-		
Intención del usuario		Responsabilidad del sistema	
1. Buscar los datos almacenados del cliente 3. Ingresar los datos actuales del cliente		2. Mostrar la información del cliente solicitado (datos generales, direcciones y vinculaciones). 4. Validar la información ingresada 5. Registrar vinculaciones 6. Almacenar la información	
Variantes			
		2. a. Si el sistema no encuentra al cliente solicitado, entonces entregara el formulario vacío, con la opción de ingresar un registro nuevo	
Reglas de negocio			
<ul style="list-style-type: none"> - Si se registra a una persona jurídica (empresa), también se deberá registrar todos los datos de un representante (como mínimo). - El representante de una persona jurídica no debe ser menor de edad. 			

Fuente: (Elaboración propia)

En la tabla 3.1 se muestra la descripción de la tarea registrar o actualizar datos del cliente, el cual es llevado a cabo por el asesor de créditos; el disparador o el evento que da inicio a esta tarea es una solicitud de crédito; los objetos de datos que intervienen son: el registro de datos (desactualizados o nulo al inicio de la tarea y actualizado al final de la tarea) y el cliente; la tarea consiste en ubicar al cliente en el sistema o crear un nuevo registro en caso de un cliente nuevo, ingresar los datos actuales del cliente, validar la información, guardar los datos y las vinculaciones que exista con otros clientes; por último se detallan algunas reglas de negocio establecida por la empresa como registrar obligatoriamente a un representante mayor de edad en caso de que le cliente fuera una persona jurídica.

Tabla No 3.2: CONSULTAR HISTORIAL CREDITICIO

Nombre: Consultar historial crediticio	
Tareas: Consultar historial crediticio	
Proceso de negocio: Evaluación de crédito	Rol: asesor de crédito
Disparadores: Solicitud de crédito	

Entrada		Salida	
Objeto de datos	Estado	Objeto de datos	Estado
Solicitud de crédito	Propuesta	-	-
Cliente			
Intención del Usuario		Responsabilidad del sistema	
1. Buscar la información sobre los créditos otorgados al cliente. 3. Seleccionar alguno de los créditos mostrados. 5. Solicitar imprimir Plan de pagos o Kardex de pago de algún crédito otorgado.		2. Mostrar una lista detallada de los créditos otorgados al cliente solicitado, sean activos o cancelados 4. Mostrar información sobre el plan de pago, los pagos realizados, los intervinientes del crédito (conyugue, aval, garante, etc.), las garantías vinculadas al crédito. 6. Mostrar un reporte con formato adecuado para la impresión con la información solicitada.	
Variantes			
		2. a. Si el cliente solicitado no cuenta con ningún crédito otorgado anteriormente, entonces el sistema no deberá mostrar información alguna	

Fuente: (Elaboración propia)

La tabla 3.2 presenta la descripción de tarea consultar historial crediticio, esta tarea también es llevada a cabo por el asesor de créditos e iniciada por una solicitud de crédito; los objetos de datos de entrada son: propuesta de solicitud de crédito, y el cliente ; la tarea consiste en buscar información concerniente a los créditos otorgados anteriormente al cliente y/o conyugue (si hubiese), en especial los cronogramas de pago para analizar el comportamiento de pago del cliente, la tarea finaliza al imprimir estos cronogramas. Para esta tarea la empresa no reporto reglas de negocio para tener en cuenta en la funcionalidad del sistema.

Tabla No 3.3: GESTIÓN DE SOLICITUD

Nombre: Gestión de la solicitud de crédito			
Tareas: Registro de solicitud de crédito, registro de intervinientes, registro de garantías.			
Proceso de negocio: Evaluación de crédito		Rol: asesor de crédito	
Disparador: Revisar los documentos entregados por el cliente.			
Entrada		Salida	
Solicitud	Pre-Solicitada	Solicitud	Solicitada
Documentos recep.			
Intención del usuario		Responsabilidad del sistema	
<p>1. Recolectar o renegociar las condiciones del crédito con el cliente (producto, plazo, monto, la tasa de interés, garantías, garantes, etc.).</p> <p>2.Registrar la solicitud de crédito</p> <p>5. Registrar a todos los intervinientes asociados al posible crédito (conyugue, garante, etc.).</p> <p>8. Registrar todas las garantías que posee el cliente (solo si hubiese).</p> <p>11. Vincular las garantías con la solicitud de crédito.</p>		<p>3.Validar la información ingresada</p> <p>4. Almacenar la inf. De la solicitud de crédito.</p> <p>6.Validar la información ingresada</p> <p>7.Almacenar información de los intervinientes</p> <p>9. Validar la información ingresada.</p> <p>10. Almacenar información de las garantías.</p> <p>12. Generar vinculación entre garantías y solicitud, almacenar esta información.</p>	
Reglas de negocio			
<ul style="list-style-type: none"> - El monto solicitado deberá fluctuar entre 100 y 4000 bolivianos. - Los montos solicitados solo se efectuaran en moneda nacional (bolivianos). - El plazo de los créditos fluctuara entre 1y 2 meses (generalmente 30, 45 y 60 días). 			

- La frecuencia de pago será semanal a excepción de los casos especiales que se realiza de forma mensual.
- Las garantías deben ser estrictamente con papeles originales de los vehículos motorizados o fotocopias en caso de convenio y acuerdo entre una institución (sindicato) y la empresa.
- Los créditos deben otorgarse únicamente a personas entre los 20 y 65 años.

Fuente: (Elaboración propia)

En la tabla 3.3 se describe la tarea de gestión de la solicitud, la cual se puede disgregar en: registro de solicitud de crédito, registro de intervinientes y registro de garantías, esta tarea también está a cargo el asesor de créditos, se inicia después de revisar los documentos entregados por el cliente, sus objetos de datos son: la solicitud de crédito (pre-solicitada al inicio y solicitada al final de la tarea) y los documentos entregados, esta tarea consiste en recordar o renegociar con el cliente las condiciones del crédito, registrar la solicitud y los datos de todos los intervinientes (garante, conyugue, etc.) y/o garantías en el sistema, toda la información deberá ser debidamente validada y cumplir con las reglas de negocio determinadas por la empresa para evitar inconsistencias posteriores, después de ello se deberá vincular el crédito con los intervinientes y las garantías correspondientes para así almacenar en el sistema una información completa sobre el crédito solicitado.

Tabla No 3.4: CAMBIAR ESTADO DE SOLICITUD

Nombre: cambia estado de solicitud			
Tareas: Aprobar solicitud, rechazar solicitud.			
Proceso de Negocio: Evaluación de crédito		Rol: asesor de créditos	
Disparadores: respuesta del gerente general.			
Entrada		Salida	
Objeto de datos	Estado	Objeto de datos	Estado
Solicitud de crédito	Solicitada	Solicitud de crédito	- Aprobada - Rechazada
Intención del usuario		Responsabilidad del sistema	
1. Buscar solicitud de crédito de un cliente determinado.			

3. Cambiar el estado de la solicitud (aprobada o rechazada), dependiendo de la respuesta del gerente general después de la evaluación respectiva.	3. Mostrar información sobre la solicitud de crédito requerido. 4. Cambiar la solicitud al nuevo estado.
Reglas del negocio	
<ul style="list-style-type: none"> - Para aprobar una solicitud con una tasa diferente a la pactada con el cliente, ello lo debería hacer una persona con un nivel de autorización mayor al asesor de créditos. 	

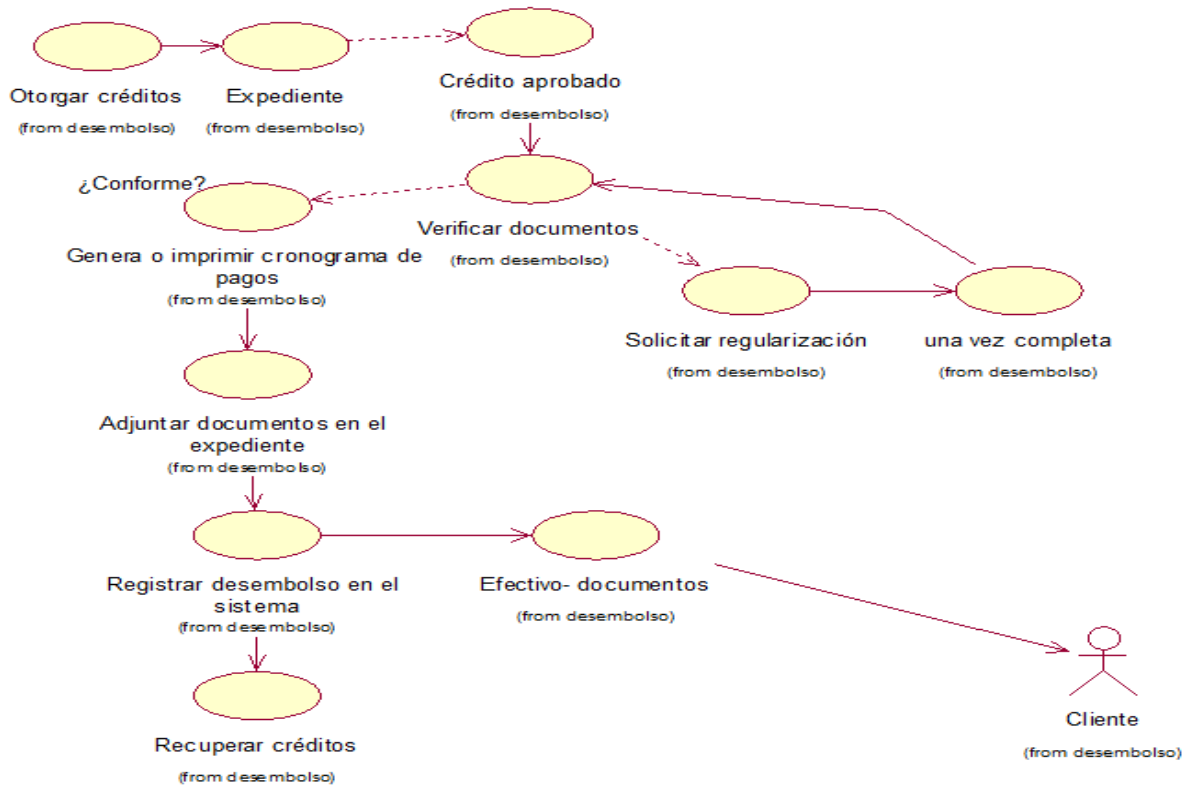
Fuente: (Elaboración propia)

Por último la tabla 3.4 detalla la descripción de la tarea cambiar estado de la solicitud, realizada por el asesor de créditos, iniciada inmediatamente después de saber la respuesta del gerente general, cuyo objeto de datos es la solicitud de crédito (solicitada la inicio y aprobada o rechazada la final de la tarea), esta tarea consiste en ubicar el registro de solicitud de crédito en el sistema y cambiar su estado ha aprobado o rechazado según sea el caso, para aprobar una solicitud con una tasa diferente a la pactada con el cliente, se necesitaría ingresar que la tarea lo realice una persona con un nivel de autorización mayor al asesor de créditos.

Subproceso: desembolso

El subproceso desembolso se inicia con un crédito aprobado, el operador es el único responsable de realizar el proceso, primero verifica los documentos entregados por el asesor de créditos (expediente, letra, contrato documento privado , etc.), genera e imprime un plan de pagos o cronograma de pagos, de acuerdo a las especificadores de la solicitud aprobada, gestiona las firmas de todos los documentos pertinentes, entrega el dinero en efectivo al cliente y lo registra debidamente en el sistema de información para que así automáticamente el crédito se convierta en un crédito desembolsado y listo para proceder a la recuperación de este; posteriormente devuelve el expediente con todos los documentos al asesor de para su archivamiento.

Figura No 3.11: DESEMBOLSO DE CRÉDITO



Fuente: (Elaboración propia)

De este procedimiento figura 3.11 se obtiene una tarea concreta en la cual interviene el sistema de información, esta tarea se denomina Registro y desembolso del crédito, en la tabla 3.5 se presenta un resumen de la descripción de la tarea y sus aspectos más relevantes que se deberán tener en cuenta para la captura de requerimientos.

Tabla No 3.5: REGISTRO Y DESEMBOLSO DE CRÉDITO

Nombre: Registro y desembolso del crédito.			
Tareas: Generar plan de pagos, registrar desembolso.			
Proceso de negocio: Desembolso		Rol: Operador	
Disparadores: Crédito aprobado			
Entrada		Salida	
Objeto de datos	Estado	Objeto de datos	Estado
Solicitud de crédito	Aprobada	Crédito	Otorgado

Intención del usuario	Responsabilidad del sistema
1. Revisar y verificar documentos. 2. Generar e imprimir calendario de pagos (Kardex de pago), según lo pactado con el cliente. 6. Registrar el desembolso.	3. Generar plan de pagos. 4. almacenar el plan de pagos. 5. Mostrar un reporte con el plan de pagos, listo para su impresión. 7. Almacenar la información del desembolso. 8. Generar registros contables que afecten a la caja.
Reglas de negocio	
<ul style="list-style-type: none"> - Todos los desembolsos pueden ser en efectivo y productos. - La entrega del dinero en efectivo o la cantidad de productos solicitado se deberá realizar en las oficinas de la empresa. 	

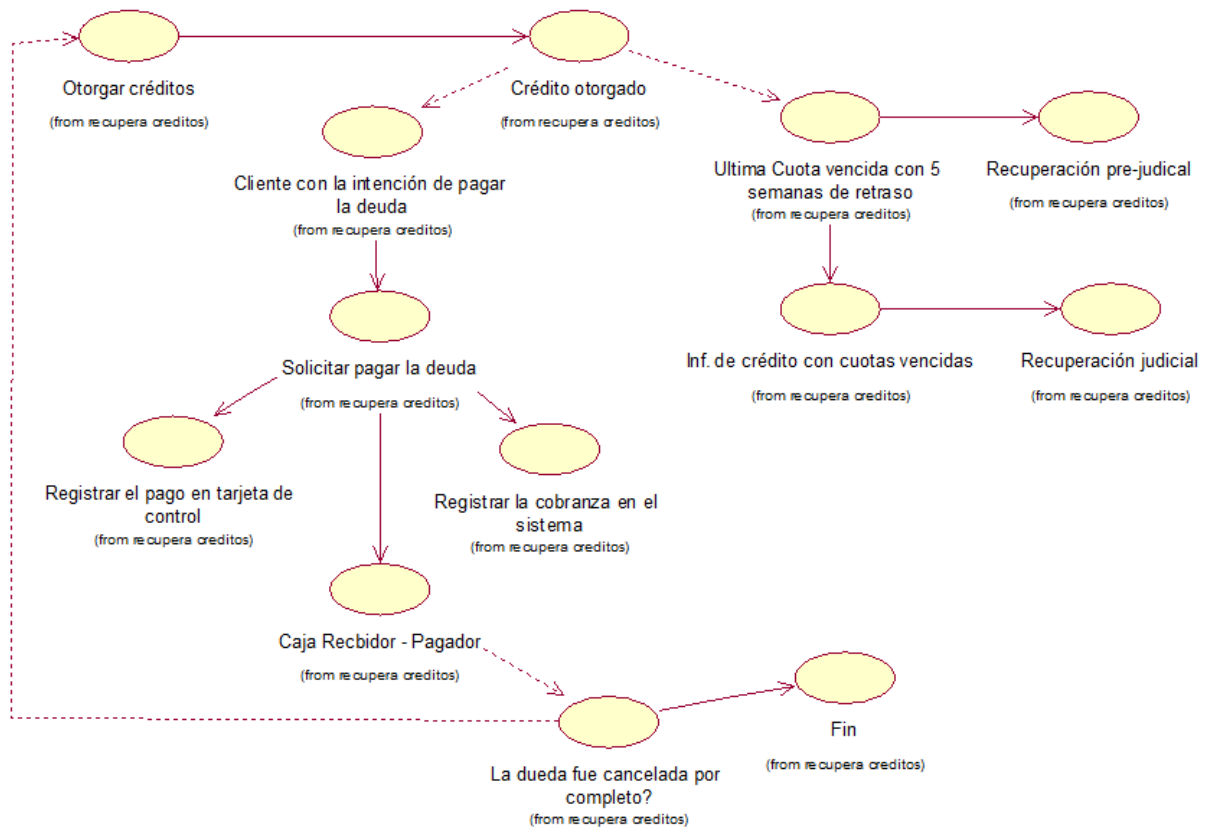
Fuente: (Elaboración propia)

La tabla 3.5 contiene la descripción de la tarea Registro y desembolso del crédito, esta es llevada a cabo por el Operador, se inicia con un crédito aprobado, el objeto de datos es la solicitud de créditos (aprobada en un inicio y otorgada al final de la tarea), esta tarea consiste en revisar y verificar los documentos pertinentes, generar el calendario de pagos con ayuda del sistema, guardarlo en el sistema de información e imprimirlo, registrar el desembolso en efectivo y por ultimo generar los registros de caja para el control contable de la empresa.

Proceso de recuperación de créditos

Este proceso hace referencia al cobro del monto de los créditos otorgados más los intereses respectivos, existen 2 maneras de cobranza, la primera a través del asesor de créditos, quien visita al cliente en su domicilio y realiza la cobranza, registrando esta operación en la tarjeta de control del cliente como conformidad de la cobranza realizada, la otra manera es a través del operador, se da cuando un cliente ingresa a las oficinas de la empresa a solicitar realizar el pago de la deuda, para ello el proceso se apoya de otro llamado caja (Recibidor - Pagador) que se detallara posteriormente.

Figura No 3.12: RECUPERAR CRÉDITOS



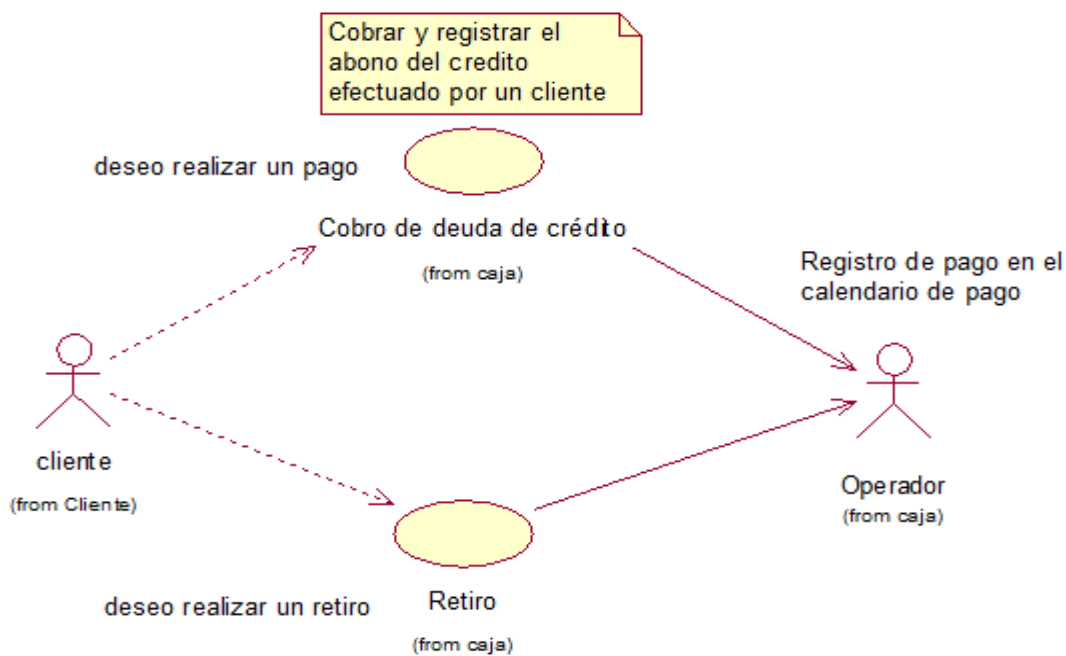
Fuente: (Elaboración propia)

En la figura No 3.12 muestra el recuperar créditos, en este diagrama se observa todo el procedimiento que se sigue para la realización de este subproceso y la interacción que tiene con otros subprocesos al intercambiar información y documentos, por ejemplo con el subproceso de otorgar créditos (recibe la información de que el crédito ya fue otorgado), con el subproceso de recuperación Prejudicial (envía una nota que sus cuotas ya fueron vencidas con 5 semanas de retraso), con el subproceso de recuperación judicial y con el subproceso de caja recibidor – pagador (intercambia información, documentos y recibos).

Proceso: caja (Recibidor- Pagador)

El proceso de soporte denominado caja (recibidor - pagador) engloba a 4 sub procesos muy importantes todos ellos involucrados con el movimiento de dinero y la interacción entre el operador y el cliente, entre estos subprocesos tenemos a uno fundamental para la gestión de créditos, denominado cobro de deuda de crédito, el cual se detallará más adelante.

Figura No 3.13: PROCESO DE CAJA (RECIBIDOR –PAGADOR)



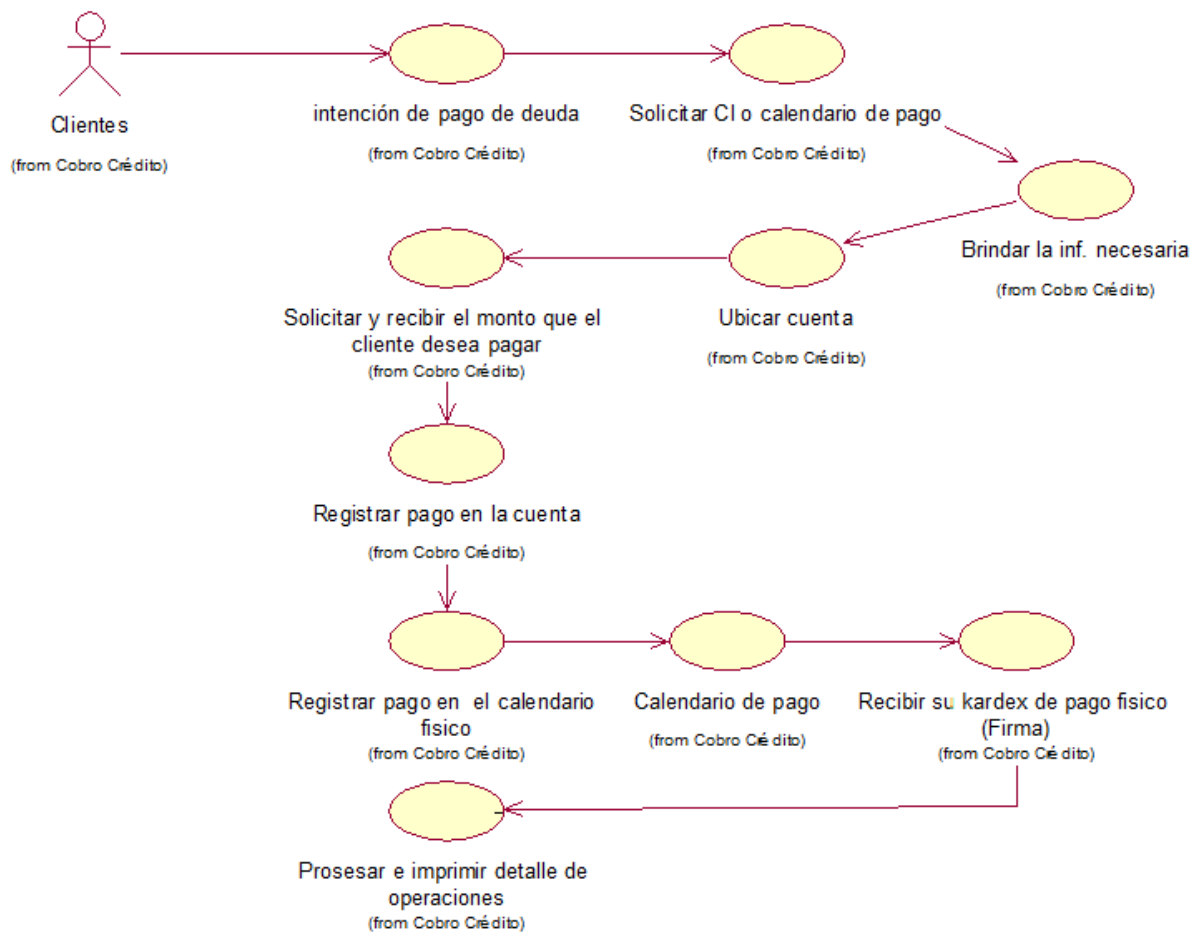
Fuente: (Elaboración propia)

En la figura 3.13 se detalla las entradas y salidas de cada uno de los subprocesos, existe un subproceso involucrado directamente con el área de créditos, el cual es el cobro de deuda de crédito, el cual tiene por objetivo cobrar y registrar el abono del crédito efectuado por un cliente, este se inicia con la tarea de realizar el pago por parte del cliente y culmina con el registro de pago en el calendario de pagos que será devuelto al cliente.

Sub-proceso: Cobro de Deuda de crédito

El subproceso cobro de deuda de crédito se inicia con la intención de pago del cliente, el operador debe solicitar el calendario de pago del cliente o el C.I., para obtener la información necesaria del crédito y poder ubicar la cuenta a la que se abonará una o varias cuotas, el operador también deberá solicitar el monto a pagar y verificar el dinero de alguna presunta falsificación, si todo es conforme se procederá a registrar el pago efectuado en el sistema y registrarlo en el calendario de pago físico que custodia el cliente, al final del día se deberá procesar mediante el sistema un consolidado de todas las cobranzas del día para su análisis y archivamiento.

Figura No 3.14: COBRO DE DEUDA DE CRÉDITO



Fuente: (Elaboración propia)

En la figura 3.14 se plasma el diagrama de procesos de cobro de deuda de crédito, en este se detalla todo el procedimiento implicado en este subproceso mediante, el cual se inicia un evento de mensaje (intención de pago de deuda) y finaliza con la impresión del detalle de operaciones diarias.

Tabla No 3.6: REGISTRO DE COBRANZA EN CAJA

Nombre: Registro de cobranza en caja			
Tareas: Ubicar cuenta, registrar pago a la cuenta			
Proceso de negocio: cobro de deuda de crédito.		Rol: operador	
Disparador: Intención de pago de deuda por parte del cliente			
Entrada		Salida	
Objeto de datos	Estado	Objeto de datos	Estado
Cuota(s)	Pendiente	Cuota(s)	Cancelada
Intención del usuario		Responsabilidad del sistema	
1. Ubicar el crédito en el registro de cobranza. 3. Registrar monto de la cobranza. 7. Registrar el pago en el Kardex de pagos físico.		2. Mostrar formulario de cobranza, con todos los datos del cliente solicitado. 4. Almacenar la información registrada. 5. Modificar el Kardex de pago y recalcular el saldo de la deuda. 6. Generar reg. Contables que afecten a la caja.	
Reglas de Negocio			
<ul style="list-style-type: none"> - Un cliente si lo desea puede adelantar sus pagos (la cantidad de cuotas que desea). - Si un cliente se retrasa, se empezara a cobrar mora. (el monto varía dependiendo del tipo de crédito y monto prestado). 			

Fuente: (Elaboración propia)

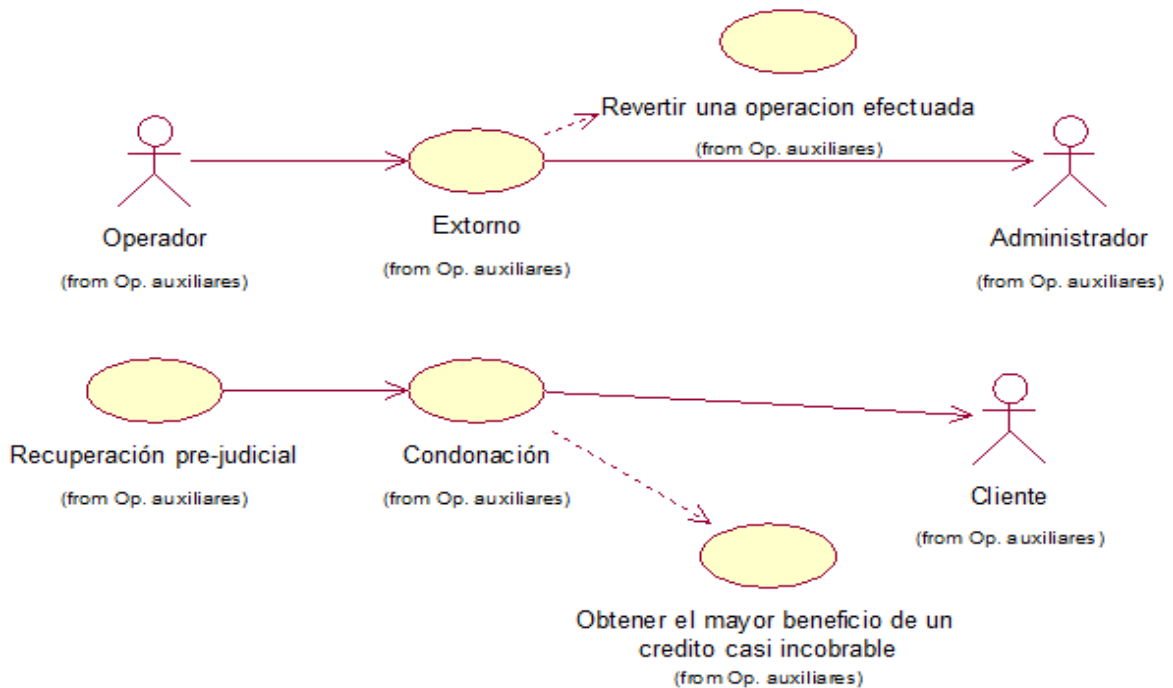
En la tabla 3.6 se presenta la descripción de tarea de registro de cobranza en caja, esta tarea es realizada por el operador, se inicia con la intención de pago de deuda por parte del cliente, los objetos de datos son las cuotas (Pendientes al inicio y canceladas al final de la tarea), esta tarea consiste en ubicar el crédito en cuestión

en el sistema, registrar la cobranza tanto en el sistema, para que este calcule el saldo de la deuda y genere los registros contables; también es indispensable que se registre la cobranza en el Kardex de pago físico que custodia el cliente, entre las reglas de negocio asociados a esta tarea tenemos las siguientes: un cliente puede adelantar la cantidad de cuotas que desee y si un cliente se retrasa en sus pagos se procederá a recargarlo con los montos correspondientes a la mora.

Proceso: Operaciones auxiliares de Crédito

El proceso de soporte denominado operaciones auxiliares de crédito, contiene 2 subprocesos totalmente independientes entre sí y no consecutivos, que pueden ser solicitados en cualquier momento y de manera separada con la intención de ayudar a los procesos. A continuación se presenta un diagrama de procesos sobre las operaciones auxiliares.

Figura No 3.15: OPERACIONES AUXILIARES DE CRÉDITO



Fuente: (Elaboración propia)

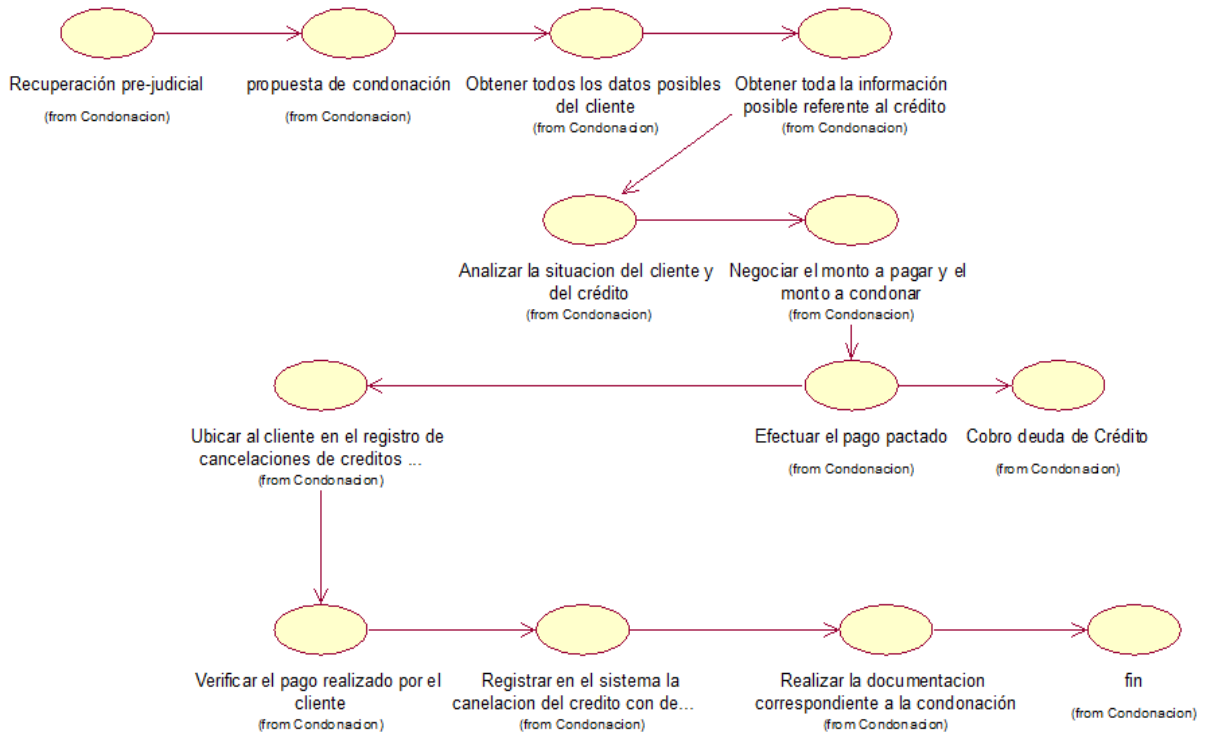
La figura 3.15 refleja el diagrama de operaciones auxiliares de crédito, este proceso está compuesto por 2 subprocesos muy importantes para la gestión de créditos: el

extorno, cuyo objetivo es revertir alguna operación realizada erróneamente y la condonación de un crédito, para tratar de recuperar la mayor cantidad posible de un crédito otorgado el cual ha sido clasificado como crédito irrecuperable, ambos procesos se ejecutan de manera independiente, cada uno cuenta con entradas y salidas diferentes, tal y como se muestra en el gráfico, el extorno se inicia con una solicitud por parte de un operador y termina con la operación anulada, y la condonación se inicia con una propuesta por parte del subproceso recuperación pre-judicial y finaliza con el crédito condonado.

Sub- proceso: Condonación

Tiene la finalidad de recuperar un porcentaje de crédito otorgado y condonar o perdonar el otro restante, en vez de recurrir al proceso de recuperación judicial donde se perdería muchísimo tiempo, recursos y posiblemente no se recupere el crédito en su totalidad. La persona responsable de realizar este proceso es el administrador de la empresa; este proceso se inicia con una propuesta de condonación por parte de los implicados en la recuperación de créditos, el administrador debe evaluar la situación del cliente, los motivos por el cual no ha cumplido con el pago oportuno de las cuotas y la situación del crédito en sí y negociar con el cliente sobre los montos a pagar y a condonar (eliminando de esta manera: mora, intereses), después de llegar a un acuerdo con el cliente, este último deberá realizar inmediatamente el pago del monto acordado, el administrador al verificar ese procedimiento realizará la cancelación de la cuenta indicando que existe un saldo pendiente, posterior a ello se elaborará y guardará la documentación correspondiente.

Figura No 3.16: CONDONACIONES



Fuente: (Elaboración propia)

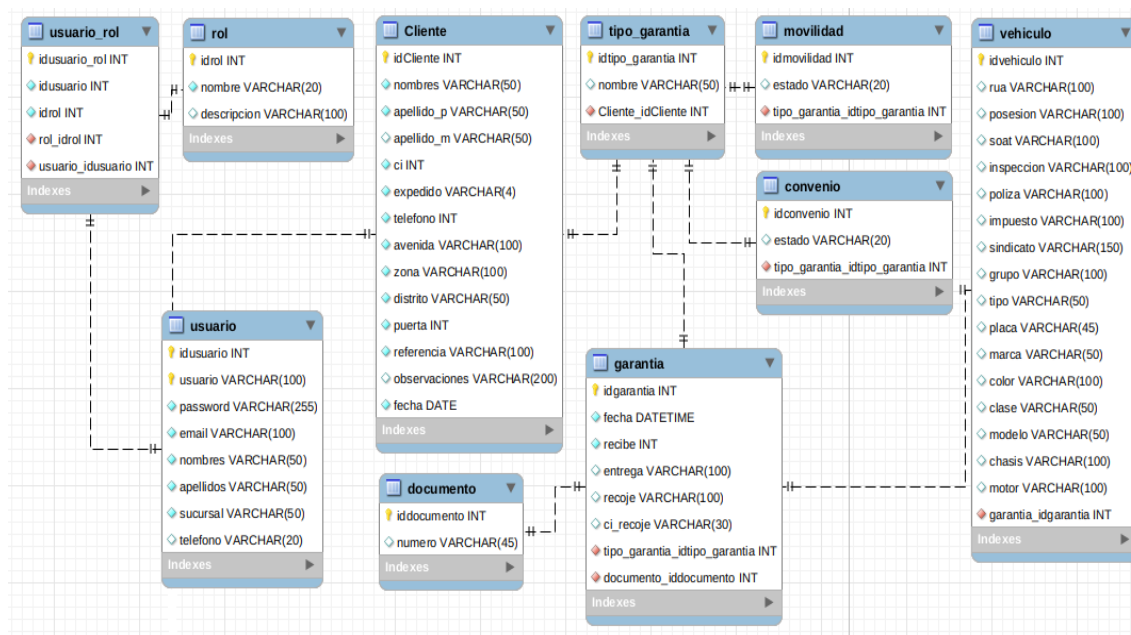
En la figura No 3.16 se muestra el diagrama de condonaciones, en este caso se muestra el procedimiento que realiza el administrador y el cliente para condonar una deuda, la relación que tiene este subproceso con el de recuperación pre-judicial (de quien recibe la propuesta de condonación) y con el subproceso de cobro de deuda de crédito (Entrega de dinero y los documentos que avalan el pago).

3.1.2.3 Diseño de la Base de Datos

La base de datos se diseñó bajo el modelo relacional, este permitió establecer interconexiones entre las diferentes tablas de la base de datos vinculadas entre sí por un campo en común (uso de foreign keys), permitió también evitar la duplicidad de registros, garantizar la integridad referencial, ya que al eliminar un registro elimina todos los registros relacionados dependientes y favorecer la normalización por ser más comprensible y aplicable.

Las tablas utilizadas están divididas en 2 grupos, las concernientes a la información de los clientes, también de los usuarios y las concernientes a los créditos, la nomenclatura usada para las tablas es la siguiente: todas inician con un nombre propio de la tabla, el que refleja los datos que esta contiene y la nomenclatura usada para los nombres de los campos es: el nombre propio de cada campo, nombre que representa la información a guardar; seguido del tipo de dato (varchar, numeric , datetime).

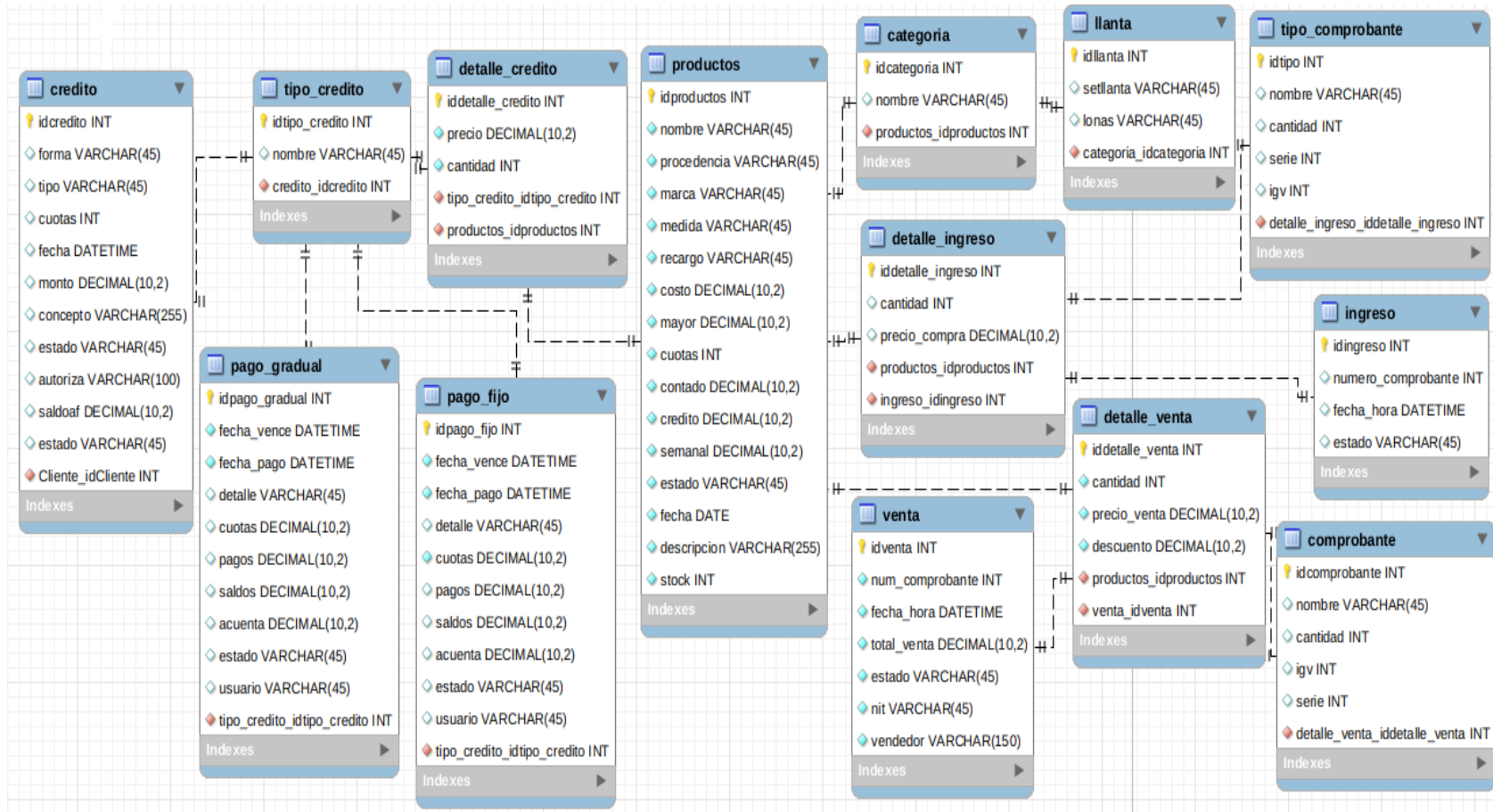
Figura No 3.17: DISEÑO DE LA BASE DE DATOS CLIENTE Y USUARIOS



Fuente: (Elaboración propia)

En la figura 3.17 se muestran todas las tablas relacionadas a la información de los clientes y de los roles de usuario, cada una de las tablas contiene por lo menos una llave primaria y una llave foránea.

Figura No 3.18: DISEÑO DE LA BASE DE DATOS CRÉDITOS



Fuente: (Elaboración propia)

3.1.3 Codificación del Software

La codificación del programa se ha realizado con lenguajes de programación; se parte del diseño del sistema, como está realizado el diseño deberá ser realizado la codificación del sistema, se ha utilizado programación relativamente sencilla y puede hacerse con generadores de código. En el diseño los requisitos del software se traducen a una serie de diagramas que representan la estructura del sistema teniendo como resultado el modelo propuesto del sistema. Por ultimo pasa la ejecución del programa para ver el funcionamiento. Se ha trabajado en dos casos: el diseño y la codificación ya que sin el diseño del sistema no se habría realizado la codificación correcta del programa. Se ha compilado el programa para aclarar dudas que han surgido en el proceso del programa y hacer que el modelo propuesto se ha mejorado para su fácil manipulación. Durante el proceso de codificación se han encontrado errores que se han ido resolviendo mediante los lenguajes de programación, se ha ido avanzando poco a poco para no encontrarse con serios problemas; al final el diseño que se ha presentado mostraba algunas fallas que se ha visto en el proceso, se ha tenido que modificar algunos cosas para que el programa funcione de acuerdo a lo requerido.

3.1.4 Pruebas

Las pruebas se han realizado con el gerente y asesores de crédito ya quienes serán encargados de manejar el sistema, para la prueba se ha elegido a los administradores de agencia para ver cómo va a ser utilizado el sistema si se encuentra alguna falla se realiza la modificación correspondiente. Se verifica si se ha encontrado errores en el sistema desde el registro de clientes hasta los reportes de crédito, si se ha encontrado errores se debe anotar de que tipo son y resolverlos.

3.1.4.1 Pruebas Unitarias

Las pruebas unitarias fueron ejecutadas en paralelo con la codificación, para lograr el correcto funcionamiento del código fuente y corregir errores en fases tempranas sin esperar la realización de pruebas integrales, que demandan mayor gasto de recursos, tiempo y esfuerzo.

Para ello se usaron 2 técnicas: pruebas de caja negra, con la finalidad de comprobar la funcionalidad, el comportamiento en la entrada y salida de datos y la integridad de la información enviada y recibida; y pruebas de caja blanca, para revisar y corregir la estructura de un código fuente según la lógica implementada evaluando la ejecución correcta a nivel de sentencia.

3.1.4.2 Pruebas Integrales

Para realizar las pruebas integrales de tipo incremental se construyó un ambiente especial de pruebas, el cual estaba exento de información y datos usados en el ambiente de desarrollo. Para el correcto desarrollo de estas pruebas se tenían en cuenta los siguientes puntos de importancia:

- Se probaron todas las implementaciones en el sistema, distribuidas en los siguientes módulos: clientes, créditos, reportes y Garantías.
- Se verificaron las siguientes características en cada uno de los formularios:
 - Todos los formularios deben mostrar su nombre en la parte superior bajo el estándar establecido.
 - El diseño de todos los formularios deben cumplir el estándar establecido.
 - Los elementos de todos los formularios: colores, tipo de letra, tamaños, botones, controles, etc. Deben cumplir el estándar establecido.
 - Todos los formularios deben tener un botón de salir.
 - Los formularios deben cumplir la funcionalidad para la cual fueron creados.
 - Los cálculos matemáticos realizados deben ser correctos.
 - La información registrada debe almacenarse correctamente en la Base de Datos.
 - El foco en los elementos de los formularios debe deslizarse en orden a través de la tecla TAB.

- Los cuadros de texto numéricos solo deben permitir el ingreso de números.
- Los cuadros de dialogo deben mostrar mensajes claros, entendibles y cortos, además deben contar con el icono de información adecuado.
- El nombre del cuadro de dialogo debe ser de acuerdo a la operación que se está realizando.
- Se verificarán las siguientes características en cada uno de los reportes:
 - La información que se muestre en los reportes deberá ser correcta y confiable.
 - Los reportes deben cuadra en una hoja de formato A4 para su impresión.

Al finalizar las pruebas integrales y obtener una lista de los errores y observaciones encontrados, se armó un documento de observaciones donde se especifican cada uno de ellos, para ser subsanados y devueltos, para que los integre nuevamente al sistema y realice las pruebas de regresión, similares a las integrales realizadas en un primer momento, pero poniendo mayor énfasis en la observaciones capturadas anteriormente.

Se tiene un plan de trabajo en esta fase, se formuló un programa de medidas correctivas de los cambios que requieren los usuarios, propios de la empresa.

3.1.5 La instalación o fase de implementación

Una vez superada la fase de pruebas se realiza la entrega del programa al cliente y comienza la vida útil del mismo. Esto permite que los usuarios finales sean los encargados de evaluar el programa, el gerente es quien registra a los usuarios verificando que no haya errores en sus datos. Se realiza estrategias para que el usuario final aprenda a manejar el sistema de una manera correcta ya es el que está involucrado en el manejo del programa, también se realiza el funcionamiento del sistema en tiempos adecuados. Se analiza el sistema de información para evaluar el desempeño del programa y como funcionara en los primeros días de uso y conforme vaya avanzando el tiempo.

3.1.5.1 Diseño de Interfaces

Las interfaces de todas las opciones del sistema se desarrollaron bajo un patrón estándar para mantener la armonía visual tanto en color, formato, botones, ubicación del control de búsqueda, etc. Además estas interfaces son intuitivas y amigables, lo cual apoya al rápido entendimiento y uso de este por parte por parte del personal. A continuación se observa las principales opciones del sistema y un breve resumen sobre su funcionamiento:

- **validación de usuario**

Para ingresar al sistema, este valida si el usuario que desea ingresar está autorizado o no, figura No 3.18 para ello se ingresa el correo electrónico del usuario (nombre de usuario) y la contraseña, el sistema comparara que la información brindada sea correcta, que el usuario se encuentre registrado en la base de Datos, todo ello es el primer mecanismo de seguridad.

Al validar la información y permitir el acceso al sistema, este mostrará el perfil o perfiles que se otorgó al usuario dependiendo del cargo que ocupa en la empresa y para cada perfil está habilitado diversas opciones (las que se crea necesarias para el desempeño de su labor), de esta manera se restringirá el ingreso de usuarios a algunas opciones para proteger la confidencialidad de información y mala manipulación de estos.

Figura No 3.19: INTERFAZ VALIDACIÓN DE USUARIO

COMERBA

Comerba Prodillantas

Sistema de Créditos.

Ingrese Usuario

Ingrese Contraseña

INGRESAR

Recuerda: [Olvidó su contraseña?](#)

INFORMATE

Fuente: (Elaboración propia)

- **Ventana Principal – Menú (Administrador)**

El menú es la ventana principal del sistema, esta ventana permite acceder a todas las opciones implementadas en el sistema de información las cuales se encuentran ordenadas por módulos para una mejor organización, además cuenta con herramientas y la información básica concerniente al sistema.

Figura No 3.20: INTERFAZ DEL MENÚ



Fuente: (Elaboración propia)

En la figura 3.20 se muestra la ventana principal del sistema, en la parte superior se muestra en usuario conectado, el perfil; al lado izquierdo se encuentran las opciones agrupadas por módulos (cliente, Garantías, Inventario, créditos y consultas), las opciones se encuentran ordenadas de acuerdo al flujo de los procesos y subprocesos especificados en los diagramas mostrados anteriormente.

Las opciones del menú son accesibles al usuario dependiendo del perfil que se le haya otorgado a se presenta la tabla donde se aprecia la relación entre el perfil y las opciones del sistema.

Tabla No 3.7: RELACIÓN ENTRE PERFIL Y LAS OPCIONES DEL SISTEMA

OPCIÓN \ PERFIL	Administrador	Asesor de créditos
Registro de Usuarios	x	
Registro y control de clientes	x	x
Garantía Movilidad Doc. Originales		
Lista de Garantías (activas)	x	x
Lista de Garantías (baja)	x	x
Garantía Convenio Doc. Fotocopias		
Lista de Garantías (activas)	x	x
Lista de Garantías (baja)	x	x
Inventario		
Lista de artículos	x	x
Ingreso de artículos	x	
Venta de artículos	x	x
Créditos		
Lista de Créditos	x	x
Lista de Créditos Pagados (Cancelados)	x	x
Consultas		
Consultas diarias	x	x

Fuente: (Elaboración propia)

En la tabla 3.7 se listan las opciones con la que cuenta el sistema agrupados por módulos; clientes, créditos y mantenimiento y la accesibilidad de estas opciones respecto a los perfiles manejados: administrador y asesor de créditos.

- Registro de clientes

Este formulario está diseñado para el registro o actualización de datos de todos los cliente o posibles clientes de la empresa, permite el registro de datos delas personas naturales y la relación o vinculación que existe, a continuación se presenta la interfaz de esta ventana.

Figura No 3.21: FORMULARIO REGISTRO DE CLIENTES

The image shows a web form titled "Registrar Nuevo Cliente." with a close button (X) in the top right corner. The form is organized into several rows of input fields and dropdown menus. The first row contains: "Cedula de Identidad:" with a text input "Carnet de Identidad..", "Complemento:" with a text input "complemento", "Expedido:" with a dropdown menu showing "La paz", and "Tipo de Garantia:" with a dropdown menu showing "movilidad". The second row is labeled "Documento de identidad" and contains: "Nombre(s):" with a text input "Ingrese Nombre(s)..", "Primer Apellido:" with a text input "Primer Apellido...", "Segundo Apellido:" with a text input "Segundo Apellido...", and "Telefono/Celular:" with a text input "#Contacto..". The third row contains: "Ciudad / Distrito:" with a text input "Ingrese Ciudad", "Avenida / Calle:" with a text input "Ingrese Avenida...", "Zona:" with a text input "Ingrese Zona...", and "Número:" with a text input "# puerta...". The fourth row contains three text areas: "Referencia:" with "Referencia en caso de emergencia", "Autoriza Recojo:" with "Autoriza recojo de documentos", and "Observaciones:" with "Observaciones". At the bottom right, there are two buttons: a red "Cancelar" button and a green "Guardar" button.

Fuente: (Elaboración propia)

En la figura No 3.21 se presenta la interfaz del formulario usado para el registro de clientes en la parte de cedula se valida el campo con búsqueda en tiempo real en la base de datos por si número de documento ya ha sido registrado. También el tipo de garantía con la cual el cliente va a ser registrado, los nombre apellidos, teléfono, dirección, referencia (nombre y número de contacto de la persona) que tiene el cliente con alguno de los trabajadores de la empresa o algún otro cliente antiguo, autoriza recojo de documentos de las garantías (o recoge personalmente), observaciones se registra algún otro dato faltante del cliente o se anota si se debe bloquear al cliente en un futuro.

- **Registro de Garantías**

Esta opción es muy importante para el proceso de evaluación de créditos, ya que permite tener un amplio panorama sobre los documentos recepcionados y debidamente almacenados por la empresa, relacionarlas con el proceso de créditos involucrado.

Figura No 3.22: FORMULARIO DE REGISTRO DE GARANTÍAS

Registrar Garantía.

Cedula de Identidad: 13001452 Fecha: 2020-07-30 Nombre(s): Juan Apellido(s): alarcon apaza

Tipo de Garantia: movilidad RUA03.: Ingrese RUA_03... Documento posesión: Documento posesión...

SOAT.: Ingrese SOAT... Inspección técnica: Inspeccion tecnica... Poliza.: Ingrese Poliza... Impuesto.: Ingrese impuesto...

Tipo.: Ingrese tipo... Placa.: Ingrese placa... Marca.: Ingrese marca... Color.: Ingrese color...

Motor.: Ingrese motor... Modelo.: Ingrese modelo... chasis.: Ingrese chasis... Clase.: Ingrese clase...

Sindicato.: Ingrese sindicato... Grupo.: Ingrese grupo...

Cancelar **Guardar**

Fuente: (Elaboración propia)

En la figura No 3.22 se muestra el formulario de registro de garantías, el cual permite guardar número de registros de los documentos de movilidad de un cliente, en la parte de cedula de identidad con una búsqueda en tiempo real a la base de datos verificando que el cliente este registrado y autocompletando campos como: nombres, apellidos, y tipo de garantía en caso de que la garantía ya se encuentre registrado aparece un mensaje de garantía ya existe como se muestra a continuación.

Figura No 3.23: MENSAJE (GARANTÍA YA EXISTE)



Fuente: (Elaboración propia)

Una vez finalizado el registro se debe imprimir el formulario de que acredite la recepción del documento la fecha, la hora y que es el responsable de la entrega de los documentos.

- **Registro de Nuevos artículos**

Mediante esta opción registra toda la información necesaria de los artículos que tiene la empresa tanto para la venta como a crédito, siendo el fuerte las llantas que la misma empresa realiza la importación.

Figura No 3.24: REGISTRO DE ARTÍCULOS



Registrar Producto.

Fecha : 30/07/2020

Producto.: Ingrese Marca...

Procedencia.: Ingrese Procedencia..

Marca.: Ingrese Marca...

Categoria : llantas

Medida.: Ingrese Medida..

Set.:

Lonas.:

Precio al costo.: Costo...

Precio mayor.: Precio x Mayor...

Precio al contado.: Precio al contado...

Precio al crédito.: Precio a crédito...

Recargo.: Ingrese recargo...

Nº Cuotas.: # Cuotas...

Cuota inicial.: C. Inicial...

Cuota semanal.: C.Semanal...

Cancelar Guardar

Fuente: (Elaboración propia)

Como se muestra en la figura No 3.24 se registra el tipo de producto, marca, procedencia precio al costo, precio por mayor, precio al contado, precio al crédito, número de cuotas, cuota inicial y cuota semanal para poder brindar un reporte detallado de la lista de precios de acuerdo al mes de registro.

- **Lista de ingresos de artículos**

Este formulario permite al usuario ingresar una cantidad de artículos para ser almacenados en el inventario de la empresa, stock de productos.

Figura No 3.25: INGRESO DE ARTÍCULOS AL INVENTARIO

Nuevo Ingreso.

Tipo Comprobante: boleta Serie comprobante.: 1 Número comprobante.: 000002

Artículo : Seleccione.... Cantidad.: Cantidad... Precio.Compra.: P.Compra... En almacen.: Stock... **Agregar**

OPCIONES	ARTICULO	CANTIDAD	PRECIO COMPRA	SUBTOTAL
X	llanta PEARL RIVER 15/40	4	200.00	800
X	llanta chengsang 15/40	5	200.00	1000
TOTAL				Bs/. 1800

Cancelar **Guardar**

Fuente: (Elaboración propia)

En la figura No 3.25 la primera parte del formulario de selecciona el tipo de comprobante de ingreso que genera un número automático de serie y su correlativa, en la segunda parte se selecciona el producto al cual desea ingresar una cantidad específica, en la tercera parte se tiene una tabla con los registros que se va almacenar y en la parte de subtotal una sumatoria de la cantidad de artículos por el precio de compra y en la parte final el total de la cantidad en Bs que ha sido el ingreso a la empresa. En la parte izquierda de la tabla se tiene un botón de eliminar el cual elimina de la lista un artículo que ha sido seleccionado erróneamente.

- Venta de Artículos

Este formulario permite al usuario registrar las ventas que realiza por día, manejando la fecha actual de la computadora y el vendedor en este caso es el usuario, que ha ingresado al sistema, se debe realizar las ventas a través del

sistema para así poder controlar la cantidad de artículos que se tiene en la empresa, además por que brinda un comprobante de venta al cliente.

Figura No 3.26: VENTA DE ARTÍCULOS

✎ Nueva Venta. ✕

Tipo Comprobante:

Serie comprobante:.

Número comprobante:.

Artículo :

Cantidad:.

Stock:.

Precio Venta:.

Descuento:.

Fecha :

Vendedor:.

NIT / C.I.:

OPCIONES	ARTICULO	CANTIDAD	PRECIO VENTA	DESCUENTO	SUBTOTAL
<input type="button" value="X"/>	llanta PEARL RIVER 15/40	2	200.00	0	400
<input type="button" value="X"/>	llanta chensang 19/40	2	250.00	50	450
TOTAL					Bs/. 850

Fuente: (Elaboración propia)

En la figura No 3.26 se muestra un formulario de registro de ventas en la parte superior el tipo de comprobante de la venta con un número de serie y su correlativa, en la segunda parte se elige un artículo y se ingresa la cantidad a vender y el descuento que se va a tener por cada producto, en la tercera parte se tiene la fecha de la venta, el vendedor y el NIT o CI del comprador y por último y una tabla indicando la cantidad de productos a vender con la cantidad, precio de venta y si hay un descuento en la parte final la sumatoria total de la venta incluidos descuento.

- **Solicitud de crédito**

El formulario de créditos sirve tanto para registrar una solicitud de crédito como para consultar la información de las solicitudes hechas anteriormente; para registrar una solicitud de crédito es necesario proporcionar la siguiente información previamente acordada con el cliente: datos del cliente, numero de cuotas, fecha de desembolso, productos, y los plazos de pago.

Figura No 3.27: SOLICITUD DE CRÉDITOS

✎ Nuevo Crédito. ✕

Cedula de Identidad:	Nombre(s):	Apellido(s):	Tipo de Garantia:
<input type="text" value="13001452"/>	<input type="text" value="Juan"/>	<input type="text" value="alarcon apaza"/>	<input type="text" value="movilidad"/>
Crédito :	Forma Pago :	Tipo crédito:	Monto.:
<input type="text" value="Efectivo"/>	<input type="text" value="Semanal"/>	<input type="text" value="Fijo"/>	<input type="text" value="1200"/>
Fecha desembolso.:	Número de cuotas.:	Concepto de.:	
<input type="text" value="2020/07/30"/>	<input type="text" value="2"/>	<input type="text" value="Reparación de caja y compra de repuestos"/>	

+ Generar

FECHA VCTO.	FECHA PAGO	___DETALLE___	CUOTAS	_PAGOS_	_SALDOS_
	2020/07/30	Sin cuota inici	S/C.I.	----	1332
2020/8/6			1°	666.0	
2020/8/13			2°	666.0	

✕ Cancelar
📄 Guardar

Fuente: (Elaboración propia)

En la parte superior del formulario de la figura No 3.27 datos del cliente, en la segunda parte el tipo de crédito, los plazos a cancelar la deuda el tipo de plazo y el

monto, en la tercera parte la fecha de desembolso el número de cuotas que es importante para generar la tabla de cobros de recuperación de créditos y el concepto por el cual el cliente está solicitando el crédito. En la parte final se tiene una tabla generando el calendario de pagos que cliente deberá realizar incluyendo las fechas y los pagos a realizar en cada una.

- **Cobro de créditos**

Esta opción es estrechamente ligada al módulo de caja, ya que sirve de soporte al subproceso de cobro de los créditos otorgados, en él se registra el pago de la cuota o las cuotas que realiza el cliente como abono de su deuda, a continuación la interfaz de este formulario.

Figura No 3.28: COBRO DE CRÉDITO

The screenshot shows a web interface for 'Lista pagos.' with the following fields and values:

- Fecha desembolso.: 2020-05-10
- Estado cuenta.: Deuda
- Monto.: 1200
- A pagar.: 1332
- Días atraso.: 6
- Fecha.: 2020-07-30
- Mora.: 5
- Total a pagar.: 1337
- Descuento.: 0

There is a 'Reprogramacion' button. Below the form is a table with the following data:

ACCIÓN	FECHA VCTO.	FECHA DE PAGO	__DETALLE__	CUOTAS	PAGOS	SALDOS
		2020-05-10	Sin cuota inicial	S/C.I.		1332.00
Pagado	2020-05-17	2020-05-11	Pagado	1°	266.40	1065.60
Pagado	2020-05-24	2020-05-11	Pagado	2°	266.40	799.20
Pagado	2020-05-31	2020-05-11	Pagado	3°	266.40	532.80
Cobrar	2020-06-07			4°	266.40	0.00
Cobrar	2020-06-14			5°	266.40	0.00
					Saldos	532.80

At the bottom right, there are 'Salir' and 'Guardar' buttons.

Fuente: (Elaboración propia)

En la primera parte de la figura 3.28 se muestra la fecha de desembolso del crédito es estado, el monto, la cantidad a cancelar incluidos los intereses por parte de la

empresa, y si es que la cuenta tuviera días de atraso, en la segunda parte se encuentra la cantidad en bs de la mora que es 5 Bs de recargo por semana. En la parte final la tabla con los registro de las cuotas del crédito detallados por cuotas canceladas y cuotas por cancelar.

3.1.6 Mantenimiento y seguimiento

Una vez puesta la Pagina Web a Disposición de los usuarios hay que ir cambiando datos y mantener este sitio actualizado, ya que esta página no puede permanecer estática. Los problemas de uso no detectados durante el proceso de desarrollo pueden descubrirse a través de varios métodos, principalmente a través de los mensajes, opiniones de los usuarios, el comportamiento y uso del sitio.

3.1.6.1 Análisis de los Resultados

Para la empresa COMERBA la perspectiva misional cambio después de la presentación de este proyecto con la inclusión de un sistema de información basado en un enfoque de procesos para la optimización de la operatividad de créditos dentro su planeación estratégica, pues con el desarrollo del proyecto se fueron evidenciando las grandes falencias que la empresa tiene para el nuevo reto que le espera con la otorgación de nuevos créditos.

Durante el desarrollo del proyecto se identificaron problemas administrativos y operativos que permitieron identificar problemáticas internas para intervenir de inmediato, como la gestión y control de inventario, la gestión y control de créditos, desde que no hubo un constante crecimiento de clientes registrados en la empresa no había sido un tema contemplado por la gerencia, al no conocer los múltiples beneficios que trae consigo la implementación de un sistema de información y aún más cuando espera el crecimiento de clientes y créditos.

Entre otras problemáticas a intervenir se identificó el proceso de registro de garantías, el flujo de información en tiempo real y la estandarización de actividades, todo lo anterior se verá solucionado con el desarrollo del sistema de información puesto que permite la organización tanto administrativa como operativa de la empresa facilitando el flujo de información y el manejo de reportes al momento de controlar la entrada y salida de documentos de los clientes, además con la

clasificación ABC se logró observar que de la totalidad de garantías son referencias obsoletas, por lo anterior se sugiere a la gerencia tomar decisiones que le permitan deshacer las garantías obsoletas. Con las capacitaciones realizadas al personal sobre el funcionamiento del software adquirido la gerencia se encuentra comprometida con un sistema de información como eje principal para la actividad de su empresa.

3.2 MÉTRICAS DE CALIDAD DE SOFTWARE

Se verá el desarrollo de la medición de la calidad del software mediante las métricas de ISO-9126 que ayudará a demostrar la confiabilidad del sistema.

FUNCIONALIDAD

No se puede medir directamente porque corresponde derivar medidas directas, como el punto función que cuantifica el tamaño y la complejidad del sistema e términos de las funciones del usuario, determinando las características del dominio, como ser:

- Número de entrada Usuario
- Número salida Usuario
- Número petición Usuario
- Número de archivos
- Número interfaces externas

Para realizar el cálculo de la cuenta total con factores de ponderación se toma en cuenta la siguiente tabla.

Tabla No 3.8: FACTORES DE PONDERACIÓN

	Parámetro de Medición	Cuenta	Factor medio	Total
1	Número de Entrada del Usuario	48	4	192
2	Número de Salidas del Usuario	11	5	55
3	Número de Petición del Usuario	8	4	32
4	Número de archivos	8	10	80
5	Número de interfaces externas	1	7	7
CT	Cuenta Total			336

Fuente: (Elaboración propia en base a criterios: R. Pressman, 2007).

Valores de ajuste de complejidad de las siguientes preguntas.

Tabla No 3.9: VALORES DE COMPLEJIDAD

	FACTOR	0	1	2	3	4	5	Fi
1	¿Requiere el sistema copias de seguridad y recuperación?						X	5
2	¿Se requiere comunicación de datos?						X	5
3	¿Existen funciones de procesos distribuido?				X			3
4	¿Es crítico el rendimiento?				X			3
5	¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?				X			3
6	¿Requiere el sistema entrada de datos interactiva?						X	5
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas operaciones?			X				2
8	¿Se actualizan los archivos maestros de forma interactiva?						X	5
9	¿Son complejas las entradas, salidas, archivos o peticiones?					X		4

10	¿Es complejo el procesamiento interno?					X		4
11	¿Se diseña el código para ser reutilizable?						X	5
12	¿Están incluidas en el diseño la conversión y la instalación?					X		4
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones?	X						0
14	¿Se ha diseñado para facilitar cambios y ser fácilmente utilizada por el usuario?						X	5
FACTOR DE COMPLEJIDAD (FCT)								53

Fuente: (Elaboración propia en base a criterios: R. Pressman, 2007).

Para calcular Punto Función (PF), usamos la siguiente relación:

$$PF = cuenta\ total * (0.65 + 0.01 * \sum Fi)$$

Calculando el punto función real:

$$PF = 336 * (0.65 + 0.01 * 50)$$

$$PF = 386$$

Al 100% el nivel de confianza consideramos la sumatoria de $F_i = 70$ como valor máximo de ajuste de complejidad sería:

$$PF_{max} = 336 * (0.65 + 0.01 * 70)$$

$$PF_{max} = 454$$

Por tanto, la funcionalidad es:

$$Funcionalidad = 386/454 = 0.85$$

La funcionalidad del sistema es de 85% sin riesgo a fallo y operatividad y 15% de colapso del sistema.

CONFIABILIDAD

Para determinar la confiabilidad de un software especificamos desde el instante que empieza a funcionar es decir $t_0 = 0$, a partir de este momento se realiza las observaciones pertinentes. En son de encontrar una falla en el sistema

considerando el tiempo de falla como t_1 , como intervalo entre ambos tiempos es una variable continua se vio la necesidad del uso de una función continua, que nos da la confiabilidad en términos probabilísticos.

$$P(T \leq t) = F(t) \quad \text{Probabilidad de fallos}$$

$$P(T > t) = 1 - F(t) \quad \text{Probabilidad de éxito}$$

Para el cálculo de las probabilidades se tomó la distribución exponencial, por la existencia de intervalos continuos.

$$F(t) = 1 - e^{-\lambda * t}$$

Para calcular el índice de error se define de cada 100 ejecuciones dos fueron fallidas en un periodo de una semana durante un mes y medio, y reemplazando tenemos:

$$F(t) = 1 - e^{-[0.02 * 6]}$$

$$F(t) = 0,12$$

Reemplazando en las fórmulas de probabilidades:

$$P(T \leq t) = F(t)$$

$$P(T \leq t) = 0,12 \quad \text{Probabilidad de fallos}$$

$$P(T > t) = 1 - F(t)$$

$$P(T > t) = 1 - 0,12$$

$$P(T > t) = 0,88 \quad \text{Probabilidad de éxito}$$

Siendo la probabilidad de fallo del 12% y la probabilidad de éxito de un 88%.

MANTENIBILIDAD

Se desarrolla para mejorar el sistema a los nuevos requerimientos que tengan, el estándar IEE94 sugiere un índice de madurez del sistema que proporciona un indicador en la estabilidad de un producto y se tiene que:

$$M = \frac{(Mt - (Fa + Fc + Fd))}{Mt}$$

Mt = Número de módulos actual

Fa = Número de módulos que se añadió

Fc = Número de módulos actual que se cambió

Fd = Número de módulos anterior que se borró en lo actual

Por tanto, se calcula la M (madurez)

$$M = (6 - (0 + 1 + 0)) / 6$$

$$M = 0.83$$

Se puede aclarar que el software tiene una estabilidad de 83 %, fácil mantenimiento y el 17% restante sería el margen de error por los cambios y modificaciones.

USABILIDAD

Nos muestra la facilidad de uso del producto y se calcula con la siguiente fórmula y tabla:

$$U = ((\text{sum}(xi)/n)*100)$$

Tabla No 3.10: PREGUNTAS DE USABILIDAD

Preguntas		SI	NO	R
1	¿Puede utilizar con facilidad el sistema?	5	1	0.86
2	¿Puede controlar operaciones que el sistema solicita?	5	1	0.86
3	¿Las respuestas del sistema son complicadas?	1	5	0.71
4	¿El sistema permitió la retroalimentación de información?	5	1	0.86
5	¿El sistema cuenta con interfaces agradables a la vista?	6	1	0.86
6	¿La respuesta del sistema es satisfactorio?	5	1	0.86
7	¿Los resultados que proporciona el sistema facilitan el trabajo?	5	1	0.86
USABILIDAD				0.84

Fuente: (Elaboración propia en base a criterios: R. Pressman, 2007).

Sé concluye que el sistema tiene una usabilidad del 84% de acuerdo a los datos obtenidos.

PORTABILIDAD

Se toma en cuenta dos aspectos muy importantes, como ser; nivel de aplicación y nivel de hardware.

El software desarrollado como se trata de un sistema con tecnología web es fácilmente implementado en cualquier plataforma con servidor web y gestor de base de datos mysql y puede ser ejecutado en cualquier computadora con acceso a internet con cualquier navegador web como ser Explorer, Firefox, Opera, Chrome, etc.

3.3 ANÁLISIS DE COSTOS

En el análisis de costos del presente trabajo se ha considerado lo siguiente:

- Se obtuvo fuentes de información en línea y física, por el costo distribuido en la información.
- Costo de infraestructura el uso de servidor de la Universidad, es despreciable porque cuenta con propios servicios.
- Costos de recursos humanos por hora en la aplicación del modelo teórico y el desarrollo mismo, calculando 4 horas laborales por día.
- Se utilizó las herramientas para su implementación el software, cuyo costo de licencia es gratuita además con programas y servicios.

El proyecto no se apoyó en terceros económicamente, solo la inversión es de tiempo tanto en el sistema y en el modelo teórico, los resultados del proyecto son una justa inversión.

Tabla No 3.11: ESTIMACIÓN DE COSTOS DEL SOFTWARE

ACTIVIDADES	COSTO
INVESTIGACIÓN	1421 Bs
ANÁLISIS, MODELO Y DISEÑO	1000 Bs
DESARROLLO DEL SISTEMA	9000 Bs
TOTAL	11421Bs

Fuente: (Elaboración propia).

3.3.1 Análisis de Costos por el método COSMIC

➤ **Proceso funcional: Mostrar lista de la cantidad de artículos**

Movimientos de datos

- Entrada: seleccionar en el menú la opción de lista de artículos.
- Lectura: obtener la lista de cantidad de artículos en almacén.
- Salida: Mostrar en pantalla la cantidad actual de artículos en almacén.
- Entrada: Seleccionar lista de precios de artículos.
- Lectura: Obtener la lista de precios actualizados a la fecha.
- Salida: Imprimir lista de precios a la fecha.
- Entrada: especificar fecha de ingreso de productos.
- Lectura: Obtener lista de ingresos según fecha y artículo.
- Salida: Mostrar en pantalla la cantidad de artículos ingresados en determinada fecha.
- Entrada: Especificar fecha de venta de artículos.
- Lectura: Mostrar en pantalla la cantidad de artículos vendidos según parámetros de búsqueda.
- Salida: Imprimir la lista de artículos vendidos según criterios filtrados.

Puntos de función COSMIC: 12 CFP.

➤ **Proceso funcional: Mostrar lista de los clientes registrados**

Movimientos de datos

- Entrada: Seleccionar en el menú la opción de lista de Clientes.
- Lectura: obtener lista de clientes y su referencia.
- Salida: Mostrar lista de clientes filtrados según parámetros de búsqueda.
- Entrada: registrar datos generales del cliente.
- Lectura: obtener datos de referencia del cliente (dirección, teléfono, etc.).
- Salida: Modificar datos generales del cliente cada vez que se solicite un nuevo crédito.

Puntos de función COSMIC: 6 CFP.

➤ **Proceso funcional: Mostrar lista de garantías de los clientes**

Movimientos de datos

- Entrada: Seleccionar en el menú de opción lista de garantías
- Lectura: Obtener la lista de todos los clientes registrados.
- Salida: Mostrar en pantalla el tipo de garantía que ha recepcionado la empresa.
- Entrada: Especificar fecha de registro de garantía en Kardex.
- Lectura: Obtener lista de las garantías filtrados según parámetros de búsqueda.
- Salida: mostrar en pantalla la lista de garantías activas.
- Entrada: Seleccionar en el menú de opción la lista de garantías inactivas (devueltas).
- Lectura: Obtener lista de garantías inactivas (devueltas).
- Salida: Mostrar en pantalla la lista de garantías devueltas filtradas según parámetros de búsqueda.
- Entrada: registrar datos del vehículo del cliente.
- Lectura: Obtener lista de datos del vehículos activo del cliente.
- Salida: Mostrar en pantalla datos del vehículo puestos en garantía.
- Salida: Imprimir reporte en recepción de garantías.
- Salida: Imprimir reporte de Devolución de garantías.

Puntos de función COSMIC: 15 CFP.

➤ **Proceso funcional: Mostrar lista de créditos**

Movimientos de datos

- Entrada: Seleccionar en el menú la opción de lista de Créditos.
- Lectura: Obtener todas la lista de créditos de los clientes.
- Salida: Mostrar en pantalla la lista de créditos de otorgados según parámetros de búsqueda.
- Entrada: Especificar cliente y su garantía.
- Entrada: Especificar fecha de desembolso de crédito y última fecha de cancelaciones de cuota.
- Lectura: Obtener datos de referencia del cliente.

- Salida: Mostrar en pantalla lista de créditos otorgados según parámetros de búsqueda.
- Entrada: Presentar lista de pagos de los clientes.
- Lectura: Obtener lista de pagos por cliente según plazo de crédito.
- Salida: mostrar cantidad de cuotas canceladas y por cancelar según parámetros de búsqueda.
- Salida: imprimir lista de tarjeta de control de pagos.
- Salida: imprimir documento privado firma de garantía.
- Salida: Imprimir lista de clientes en mora.

Puntos de función COSMIC: 14 CFP.

Costo del equipo de trabajo de desarrollo de software

El costo mensual del equipo de software es de 54000 bs

El proyecto se desarrolló con tres personas: desarrollador, analista de pruebas, diseñador.

Se determina que el equipo en los últimos 6 meses de trabajo ha producido 47 puntos de función COSMIC.

➤ **Determinar el costo por unidad de medida**

Para determinar cuánto cuesta desarrollar cada punto de función se utiliza la siguiente formula:

Costo por punto de función= Costo mes del equipo de trabajo / puntos de función del mes

Costo por punto de función = 5400 Bs / 47 puntos de función = 253.8 Bs punto de función

Una vez que contamos con la medición del tamaño del software y el costo por unidad de medida, podemos determinar el costo del proyecto de software usando la siguiente formula:

Costo de un proyecto de software= Tamaño del software x costo de función

Costo del proyecto de software = 45 CFP x 253.8 Bs. = 11421 Bs.

➤ **Tiempo que durará el proyecto de desarrollo de software**

Los puntos de función de COSMIC los podemos utilizar también para determinar cuánto tiempo durará el proyecto de software.

Duración del proyecto = 47 puntos de función COSMIC / 12 puntos de función COSMIC mes

Duración del proyecto = 3.92 Meses

Conclusión

De esta forma hemos determinado que nuestro proyecto de software;

Durará: 3.92 Meses (3 meses, casi cuatro)

Costará: 11421Bs (bolivianos moneda nacional)

3.4 SEGURIDAD

3.4.1 Seguridad a Nivel de Base de Datos

La seguridad de datos, también conocida como seguridad de la información o seguridad informática, es un aspecto esencial de TI en organizaciones de cualquier tamaño y tipo. Se trata de un aspecto que tiene que ver con la protección de datos contra accesos no autorizados y para protegerlos de una posible corrupción durante todo su ciclo de vida.

Seguridad de datos incluye conceptos como encriptación de datos, tokenización y prácticas de gestión de claves que ayudan a proteger los datos en todas las aplicaciones y plataformas de una organización.

Hoy en día, organizaciones de todo el mundo invierten fuertemente en la tecnología de información relacionada con la ciberdefensa con el fin de proteger sus activos críticos: su marca, capital intelectual y la información de sus clientes.

En todos los temas de seguridad de datos existen elementos comunes que todas las organizaciones deben tener en cuenta a la hora de aplicar sus medidas: las personas, los procesos y la tecnología.

La seguridad de datos es un tema de suma importancia que nos afecta a casi todos nosotros. Cada vez son más los productos tecnológicos que de una u otra forma deben ser tenidos en cuenta para temas de seguridad y que se están introduciendo en nuestra vida cotidiana, desde smartwatches hasta vehículos sin conductor. Ya ha llegado la era del Internet de las Cosas (IoT) y, por supuesto, de los hacks relacionados con IoT. Todos estos dispositivos conectados crean nuevas “conversaciones” entre dispositivos, interfaces, infraestructuras privadas y la nube, lo que a su vez crea más oportunidades para que los hackers puedan escuchar. Todo esto ha impulsado una demanda de soluciones y expertos en seguridad de datos que sean capaces de construir redes más fuertes y menos vulnerables.

Tendencias recientes han demostrado que los ataques de ransomware están aumentando en frecuencia y en gravedad. Se ha convertido en un negocio en auge para ladrones cibernéticos y hackers, que acceden a la red y secuestran datos y sistemas. En los últimos meses, grandes empresas y otras organizaciones, así como también usuarios particulares, han caído víctimas de este tipo de ataques y han tenido que pagar el rescate o correr el riesgo de perder datos importantes.

3.5 PRUEBAS DE FUNCIONAMIENTO

La fase de pruebas es una de las más costosas del ciclo de vida del software en sentido estricto deben realizarse pruebas de todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y el código fuente y los productos que forman parte de la aplicación de esta forma se aplican diferentes técnicas de prueba de cada tipo de producto de software.

3.5.1 Prueba de Caja Blanca

En las pruebas de caja blanca se dispone el código fuente y se intentan analizar tantos fragmentos como sea posible.

Técnica del camino básico

Consiste en derivar casos de uso de prueba a partir de un conjunto dado de caminos independientes por los que circula el flujo de programas, el número de camino viene por la complejidad ciclomática $v(G)$ del grafo de flujo asociado al código.

Figura No 3.29: PRUEBAS DE CALIDAD

```
public function guardar_nuevo_cliente(){
    $ci= $this->input->post('carnet');
    $expedido= $this->input->post('expedido');

    $ver=$this->Modelo_cliente->validar_usuario1($ci);
    if ($ver==false) {
        echo 1;
        $nombre= $this->input->post('nombre');
        $paterno= $this->input->post('paterno');
        $materno= $this->input->post('materno');
        $telefono= $this->input->post('telefono');
        $ciudad= $this->input->post('ciudad');
        $avenida= $this->input->post('avenida');
        $zona= $this->input->post('zona');
        $puerta= $this->input->post('puerta');
        $garantia= $this->input->post('idtipo_garantia');
        $recojo= $this->input->post('recojo');
        $referencia= $this->input->post('referencia');
        $observaciones= $this->input->post('observaciones');
        $numero= $this->input->post('num_doc');

        $idcliente=$this->Modelo_cliente->guardar_nuevo_cliente($ci,$expedido,$nombre,$paterno,$materno,$telefono,$ciudad,$avenida,$zona,$puerta,$garantia,$recojo,$referencia,$observaciones);
        $idnum_doc=$this->Modelo_cliente->guardar_nuevo_documento($numero,$sucursal,$idcliente);
    }else{
        echo 0;
    }
}
```

Fuente: (Elaboración propia).

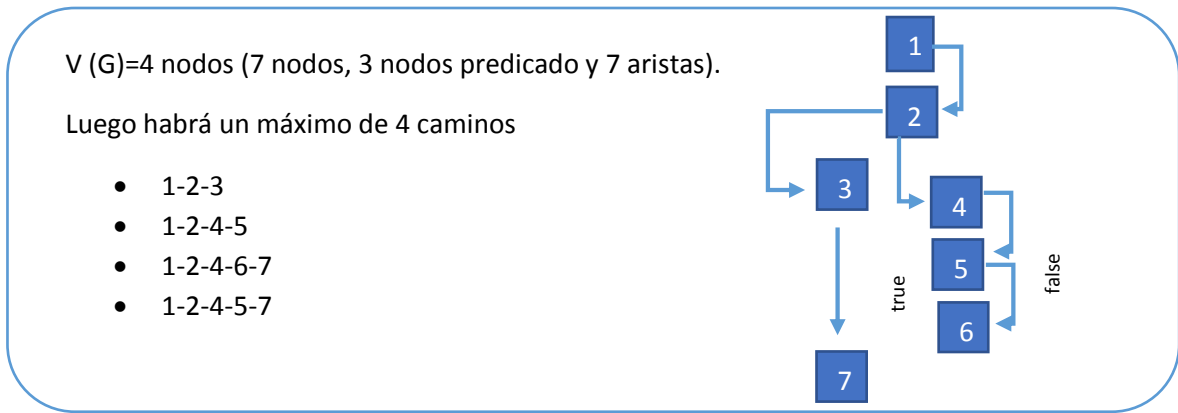
Figura No 3.30: TÉCNICA DEL CAMINO BÁSICO

```
public function guardar_nuevo_cliente(){
    $ci= $this->input->post('carnet');
    $expedido= $this->input->post('expedido');
    $ver=$this->Modelo_cliente->validar_usuario1($ci);
    if ($ver==false) {
        echo 1;
        $nombre= $this->input->post('nombre');
        $paterno= $this->input->post('paterno');
        $materno= $this->input->post('materno');
        $telefono= $this->input->post('telefono');
        $ciudad= $this->input->post('ciudad');
        $avenida= $this->input->post('avenida');
        $zona= $this->input->post('zona');
        $puerta= $this->input->post('puerta');
        $garantia= $this->input->post('idtipo_garantia');
        $recojo= $this->input->post('recojo');
        $referencia= $this->input->post('referencia');
        $observaciones= $this->input->post('observaciones');
        $numero= $this->input->post('num_doc');

        $idcliente=$this->Modelo_cliente->guardar_nuevo_cliente($ci,$expedido,$nombre,$paterno,$materno,$telefono,$ciudad,$avenida,$zona,$puerta,$garantia,$recojo,$referencia,$observaciones);
        $idnum_doc=$this->Modelo_cliente->guardar_nuevo_documento($numero,$sucursal,$idcliente);
    }else{
        echo 0;
    }
}
```

Fuente: (Elaboración propia).

Figura No 3.31: PRUEBAS DE CALIDAD



Fuente: (Elaboración propia).

Tabla No 3.12: CASOS DE PRUEBA

Nro.	Camino	Cadena	Letra	N
Independiente				
1	1-2-3	""	"a"	1
2	1-2-4-5	"a"	"a"	1
3	1-2-4-6-7	"b"	"a"	1
4	1-2-4-5-7	"c"	"a"	1

Fuente: (Elaboración propia).

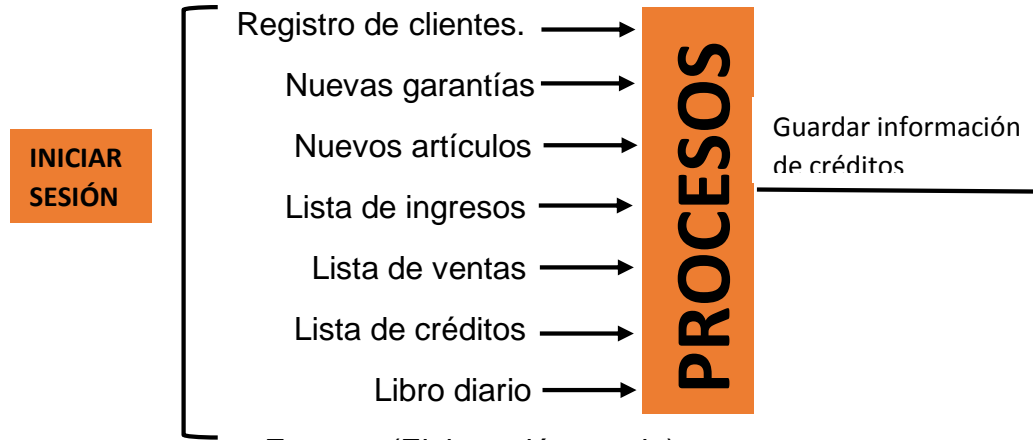
3.5.2 Prueba de Caja Negra

La caja negra se utiliza para representar a los sistemas cuando no sabemos que elemento o cosas componen al sistema o proceso, pero sabemos que a determinadas entradas corresponden determinadas salidas y con ello poder inducir, presumiendo que a determinados estímulos, las variables funcionaran en cierto sentido.

En la entrada puede considerarse toda la información para el registro de nuevos garantías y se produce una salida que nos visualiza toda la información compuesta por varias clases. En estos casos solo nos preocupamos por las entradas y salidas que produce no por lo que sucede dentro del sistema, es decir la forma en la que

operan los mecanismos y procesos internos del sistema y mediante los cuales se producen las salidas.

Figura No 3.32: PRUEBA DE CAJA NEGRA



Fuente: (Elaboración propia).

Las condiciones de entrada inicialmente son:

- Inicio de sesión: condición de entrada
- Registro de datos para créditos: condición múltiple

Las condiciones de salida inicialmente son:

Visualizar: condición de salida, visualiza toda la información introducida en el proceso de entrada

CAPITULO IV

CONCLUSIONES Y

RECOMENDACIONES

4.1 CONCLUSIONES

Realizando el respectivo análisis, diseño, programación, pruebas de la plataforma “SISTEMA DE INFORMACIÓN BASADO EN UN ENFOQUE DE PROCESOS PARA LA OPTIMIZACIÓN DE LA OTORGACIÓN DE CRÉDITOS”, se obtuvo como producto final un sistema que cumple con el objetivo general y los objetivos específicos planteados en el presente proyecto de grado.

- La incorporación de un sistema de información basado en un enfoque de procesos permitió abordar y dar soporte a todos y cada uno de los procesos realizados en la gestión de créditos de la empresa COMERBA.
- Se desarrolló una interfaz atractiva, funcional y amigable para los usuarios.
- La implementación del sistema de información basado en un enfoque de procesos permitió abordar y dar soporte al control de garantías de los clientes brindando información organizada y confiable.
- La creación del módulo de control de inventarios agilizo el proceso de consulta en tiempo real al inventario actual con el que cuenta la empresa.
- Mediante el desarrollo del sistema de información, se identificó el plan de pagos de cada cliente manteniendo actualizado el registro de cuotas.
- Se realizó un análisis de calidad mediante las métricas de ISO – 9126, la funcionalidad del sistema es de 85% sin riesgo de fallo de operatividad, con la probabilidad de éxito de un 88%, se puede aclarar que el software tiene una estabilidad de 83% fácil mantenimiento y se concluye que el sistema tiene una usabilidad de 84% de acuerdo a los datos obtenidos.
- La estimación de costos utilizando COSMIC y su modelo de estimación post arquitectura, el cual es más detallado, con la utilización de los puntos de función, permitió determinar el tiempo de duración del proyecto de software en meses y el costo en bolivianos (moneda nacional) .

4.2 RECOMENDACIONES

Como consecuencia del desarrollo del presente proyecto de grado y en base a los logros obtenidos, surgen algunas recomendaciones:

- Crear normas y políticas para fomentar al personal a realizar todos los procedimientos establecidos por el sistema de información de manera correcta para obtener el mayor beneficio y administración de la información.
- El personal de la empresa COMERBA debe almacenar en el sistema la mayor cantidad de información de los clientes, créditos, solicitados, créditos otorgados, etc. Para contar con un amplio historial y facilitar la evaluación de créditos posteriores.
- Se recomienda a la empresa implementar un sistema de gestión de calidad para complementar el diseño de todos los procesos implicados en el área de créditos y así mejorar continuamente esta área.
- Implementar un módulo de facturación enlazado con impuestos nacionales.

BIBLIOGRAFÍA

1. “Análisis y Diseño de Sistemas de Información” México, McGrawHill, (SENN, James A., 1992).
2. “Análisis y Diseño de Procesos de datos usando IDEF y UML” Editorial GraphPeru S.A.C,Perú (Taboada Jiménez, Alberto, 2005).
3. "Architectural Blueprints--The 4+1 View Model of Software Architecture". IEEE Software, Institute of Electrical and Electronics Engineers. November 1995, pp. 42-50. (Kruchten, Philippe, 1995).
4. Ciencias económicas y empresariales “Sistemas de amortización de préstamos con cuota constante”. (Alfonso Pérez Sacristán, 2015).
5. Calculo financiero aplicado “Matemática financiera”, un enfoque profesional contable (Dr. Guillermo López Dumrauf, 2003).
6. “El Proceso Unificado de Desarrollo de Software”. México: Addison-Wesley, 1999. (Jacobson, Ivar, Grady Booch, and James Rumbaugh, 1999)
7. “Ingeniería del Software” Séptima edición; Pearson Educación S.A. Madrid Ian (Sommerville, 2005).
8. “Ingeniería del software un enfoque práctico”, Editorial McGraw-Hill, México, (Roger S. Pressman, 2006).
9. “Introduccion a la teoría general de sistemas Editorial Limusa” (Bertologio, O.J. & Johanse, 1982).
10. “Ingeniería de Software”. Soluciones avanzadas. (Cota A, 1994).
11. Matemática financiera “Sistema Alemán”, un enfoque práctico. (Prof. Carlos Mario Frías, 2008).
12. “Metología de la Investigación”, cuarta edición; Editorial McGraw-Hill, 2006. N Gaither, G Frazier (2000) Administración de producción de operaciones (Roberto Hernández Sampieri, Carlos Fernández Collado, Pilar Baptista Lucio, 2006).
13. “Metodología y Diseños en la Investigación Científica“, Editorial Mantaro, Perú 1998. Pérez Huebe, M. Ingeniería de requerimientos (Hugo Sánchez Carlessi y Carlos Reyes Meza, 2005);

14. "NORMA INTERNACIONAL ISO 9000:2005 – Sistema de gestión de la calidad – fundamentos y vocabularios". (Sipper, D., y Bulfin, R. , 2003).
15. "NORMA INTERNACIONAL ISO 9001:2008 Sistema de gestión de calidad – Requisitos" (Ginebra, Suiza, 2008).
16. "Object-Oriented Analysis and Design". Second Edition. Benjamin/Cummings, Redwood (Booch, Grady, 1994).
17. "Principios de microeconomía". México: Prentice Hall (Case y Ray F., 1997).
18. "Planeación y Gestión de la producción y operaciones". (Thomson Learning, 2000).