



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA

“SISTEMA INTEGRAL PARA CASAS
DE EMPEÑOS (SICE)”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A N:

ULISES JACOBO ACEVEDO AGUILAR
HÉCTOR SALVADOR ALANIZ GARCÍA
GABRIEL DAVID CORONA ZAMORA
NANCY ELIZABETH PEÑA CRUZ
EZEQUIEL ISAI QUIROZ MARTÍNEZ

DIRECTOR DE TESIS
M.I. JUAN CARLOS ROA BEIZA

MÉXICO, D. F., CIUDAD UNIVERSITARIA, 2015





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos generales

A la Universidad Nacional Autónoma de México

Por abrirnos las puertas a la más importante institución de nivel superior del país; apoyar nuestro desarrollo intelectual, cultural y humanista.

A la Facultad de Ingeniería

Por brindarnos conocimientos sólidos y una formación integral que han sido y serán la base fundamental de nuestro desarrollo profesional.

A nuestros profesores

Por enriquecer las aulas con su experiencia, conocimientos y excelencia.

Al M. I. Juan Carlos Roa Beiza

Por su compromiso en este proyecto de tesis, así como su constancia y experiencia profesional.

A los integrantes del equipo

Por su dedicación y trabajo en equipo durante el desarrollo de la presente tesis.

Agradecimientos

ULISES JACOBO ACEVEDO AGUILAR

A mis padres

Que son y serán una parte muy importante de mi vida, con su apoyo total y absoluto me brindaron las herramientas necesarias para salir adelante y lograr mis objetivos los amo muchas gracias por todo.

A mi hermano y su familia

Gracias por su apoyo en todo momento.

A mi compañeros de tesis

Gracias por su disposición y conocimientos para realización de la misma.

Nancy Elizabeth Peña Cruz gracias por tu apoyo en las buenas y en las malas, eres un gran ser humano.

A mis compañeros y amigos del trabajo

Luis Antonio Gómez Santiago, Fabiola Martínez Mozo gracias por su apoyo e interés en la conclusión de mi titulación.

Agradecimientos

HÉCTOR SALVADOR ALANIZ GARCÍA

A mis padres

Héctor Salvador Alaniz Jiménez y María de Jesús García Palma, por brindarme la mejor educación que alguien pueda tener, su confianza, su cariño, experiencia e inagotable apoyo en todas mis metas.

A mi hermana y sobrino

Fabiola, gracias por tu amistad y unión en los momentos difíciles.
Iván David, gracias por regresarle la vida y alegría a nuestra familia.

A Susana

Gracias por todo el amor, cariño, comprensión, mi novia, mi mejor amiga y mi confidente.

A mi abuelito

Q.E.P.D. Gracias por su aliento, enseñanza y sabiduría que me motivo a seguir estudiando y no claudicar jamás.

A mis compañeros de tesis

Gracias Ezequiel, Gabriel, Nancy y Ulises por su compañerismo y apoyo para lograr una buena conclusión de esta tesis.

A nuestro director de tesis

M.I. Juan Carlos Roa Beiza, gracias por compartir sus vastos conocimientos, su actitud en busca de conocimientos, siempre serán un ejemplo a seguir.

A la Facultad de Ingeniería

Gracias a las instalaciones, los mejores profesores e investigadores, quienes brindaron sus conocimientos y enseñanzas para formar nuevos ingenieros.

A mi alma Mater Universidad Nacional Autónoma de México

Gracias por todo lo que me has dado. Entiendo la grandeza de tu espíritu y fortaleza de tu raza.

Agradecimientos

GABRIEL DAVID CORONA ZAMORA

Agradezco a Dios por los padres que me dio, ya que ellos me dieron sus consejos, apoyo y confianza para cumplir mis metas, por todos los principios y valores que me enseñaron y ayudaron a ser lo que ahora soy.

A mis hermanos porque siempre recibí su apoyo.

A mi familia que siempre me impulsaron a salir adelante y por brindarme su apoyo en algún momento de mi vida.

A Arlene por darme apoyo en los momentos difíciles, alentarme para seguir mi proceso de titulación, además de que es un pilar importante, con su apoyo constante y amor incondicional ha sido amiga, compañera inseparable y consejera en todo momento.

A mis amigos que compartieron conmigo muchas aventuras, y me ayudaron dentro y fuera de lo académico.

A la UNAM y a la Facultad de Ingeniería por proporcionarnos los elementos necesarios en un ambiente competitivo, pacífico e integral que nos permitió ante todo crecer como personas y por brindarnos las herramientas que nos permitieron formarnos como profesionales.

Agradecimientos

NANCY ELIZABETH PEÑA CRUZ

A mis padres Graciela y Porfirio por su amor y apoyo incondicional, así como sus enseñanzas de vida que me han guiado siempre. Por impulsarme a lograr mis sueños y metas; enfrentar con valor las dificultades y ser una persona honorable.

Thania y Jessica, por hacer mi vida muy alegre, por darme ánimos durante mis estudios y seguir compartiendo tantas vivencias juntas.

Gonzalo Emilio por ayudarme a ver la vida con simpleza a través de su ingenio e inocencia.

A Ulises por su cariño y apoyo en todo momento; por tantas alegrías y aventuras compartidas que me hacen muy feliz y por sorprenderme día a día.

A mi familia y amigos que me han acompañado siempre en momentos buenos y difíciles.

Agradecimientos

EZEQUIEL ISAI QUIROZ MARTÍNEZ

Agradezco a Dios, por la vida que me ha prestado y por todas las bendiciones que me da.

Agradezco a mis padres, por su inmenso apoyo incondicional, por sus enseñanzas y consejos. Sabiendo que no existe la forma de agradecer toda una vida de sacrificios y esfuerzos, quiero que sientan que el objetivo logrado es suyo y que la fuerza que me ha ayudado a lograrlo fue su apoyo.

A mis hermanos por sus palabras de aliento, su apoyo y por estar conmigo en todo momento.

A lo largo de todos estos años de estudio y trabajo he conocido a muchas personas que me han apoyado, no sólo en el técnico y profesional, sino también en lo personal y aunque no son muchos, no hace falta dar sus nombres o apellidos, porque de sobra ellos se saben aludidos. A ellos deseo agradecer su esfuerzo y dedicación. Muchas gracias por todo.

ÍNDICE

CAPÍTULO 1.- ANTECEDENTES.....	1
1.1 Introducción.....	1
1.2 Conceptos generales de los términos usados por las casas de empeño.....	4
1.3 Normas de casas de empeños.....	8
1.4 Descripción básica del funcionamiento de una casa de empeño.....	12
1.5 Acotamiento del problema a resolver.....	16
CAPÍTULO 2.- MARCO TEÓRICO	20
2.1 Características, ventajas y desventajas de las Bases de Datos relacionales y metodología Evolutiva-Incremental.....	20
Introducción a las Bases de Datos (BD)	20
Características de las Bases de Datos Relacionales.....	21
Cardinalidad.....	21
Cálculo Relacional	22
Álgebra relacional	22
Operadores de álgebra relacional.....	22
Ventajas de las bases de datos relacionales	25
Desventajas de las bases de datos relacionales	26
Etapas del modelo evolutivo	27
Modelo incremental	29
Metodología evolutiva-incremental	31
Beneficios	32
2.2 Características, ventajas y desventajas de la programación orientada a objetos (POO).....	33
Introducción a la programación orientada a objetos	33
2.3 Características, ventajas y desventajas del lenguaje de programación Hypertext Pre-processor (PHP)	38
Lenguaje de programación PHP	38
Características del lenguaje de programación PHP	39
Ventajas del lenguaje de programación PHP	41

Desventajas del lenguaje de programación PHP	42
2.4 Características, ventajas y desventajas del manejador de Bases de Datos MySQL	43
Introducción al manejador de Base de datos MySQL	43
Características del manejador de Base de datos MySQL.....	44
Ventajas del manejador de Base de datos MySQL	46
Desventajas del manejador de Base de datos MySQL.....	48
2.5 Características, ventajas y desventajas del ambiente Cliente – Servidor	49
Introducción del ambiente Cliente - Servidor	49
Características del modelo Cliente - Servidor.....	50
Ventajas del Modelo Cliente - Servidor.....	51
Desventajas del Modelo Cliente - Servidor.....	51
CAPÍTULO 3.- PLANTEAMIENTO DEL PROBLEMA	54
3.1 Análisis del problema	54
Proceso de valuación de la prenda.....	54
Proceso de préstamo.....	57
Proceso para recuperar la prenda	59
Proceso para refrendar la prenda	61
3.2 Requerimientos generales y particulares de la aplicación.	62
Requerimientos generales	64
Requerimientos particulares	65
3.3 Análisis y posicionamiento de los lenguajes para el Back-End y Front-End	68
Análisis y posicionamiento para Back-End	68
Características fundamentales	69
Tablas temporales y vistas materializadas	70
Front-End.....	72
Comparativa entre PHP, Java y C#	72
C#	72
Java	73
PHP (Hypertext Pre-processor)	74
Conclusiones para haber elegido PHP y MySQL	78

3.4 Aplicación y separación de los módulos según la metodología Evolutiva-Incremental	78
Módulo de Back-End y Front-End	79
Módulo de pruebas	79
Módulo de inicio de sesión.....	80
Módulo de fin de sesión	80
Módulo de administración de usuarios.....	80
Módulo de valuación de prendas	80
Módulo de alta de cliente	81
Módulo de oferta de servicio.....	81
Módulo de control de pagos.....	81
Módulo de mantenimiento.....	81
3.5 Diseño a nivel de bloques de la interfaz del usuario	83
Bloque de menús disponibles	84
Bloque de nombre y logo de la casa de empeño	84
Bloque de fecha de sistema.....	84
Bloque de recepción de prenda	85
Bloque de tipo de empeño	86
Bloque de registro del pignorante	86
CAPÍTULO 4.- DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN.....	87
4.1 Modelo esencial ambiental del Sistema Integral de Casas de Empeños	87
Modelo ambiental	87
Diseño de sistemas	87
4.1.1 Diagrama de flujo de datos	88
Nivel 1.....	88
Nivel 2.....	91
Nivel 3.....	94
Diagrama de flujo de pagos	98
Diagrama de flujo creación de sucursal.....	101
Diagrama de flujo creación de usuario.....	102
Diagrama de flujo usuario - sucursal	102
Diagrama de flujo notificación al usuario sobre sus pagos	103

4.1.2 Diagrama de Entidad-Relación del sistema (SICE)	104
Entidad.....	105
Atributos.....	105
Relación.....	105
Relaciones de cardinalidad.....	106
Claves.....	106
Clave primaria.....	106
Clave externa o clave foránea	106
4.1.3 Diccionario de datos	110
Diccionario de Datos del SICE.....	111
4.1.4 Normalización de la Base de Datos	127
4.2 Creación de la base de datos	131
Implementación de Base de Datos	131
Descarga e instalación de herramientas.....	131
Creación de Base de Datos	136
Script para la creación de una tabla	139
Creación de un trigger	142
Creación de un procedimiento almacenado.....	144
4.3 Desarrollo del front-end del SICE.....	146
Módulo de Catálogos.....	148
Crear sucursal	148
Alta de usuario.....	150
Modificar perfil	151
Empeños.....	152
4.4 Generación de pruebas y mantenimiento	156
Pruebas de caja negra.....	156
Pruebas de caja blanca	159
Pruebas unitarias.....	161
Pruebas de integración.....	163
Pruebas de regresión.....	165
Prueba alfa	166
Prueba beta	166

Prueba de stress.....	166
Prueba de recuperación.....	166
Mantenimiento	166
Mantenimiento preventivo.....	167
Mantenimiento correctivo.....	167
Mantenimiento perfectivo.....	167
Mantenimiento adaptativo.....	167
4.5 Generación de Comprobantes	167
CONCLUSIONES	174
APÉNDICE 1.....	176
APÉNDICE 2.....	176
APÉNDICE 3.....	177
APÉNDICE 4.....	178
BIBLIOGRAFÍA	180

CAPÍTULO 1.- ANTECEDENTES

1.1 Introducción

En la actualidad la Procuraduría Federal del Consumidor (PROFECO) señala que en México existen más de 250 casas de empeño operando entre instituciones de asistencia privada y empresas particulares.

Las casas de empeño están reguladas por la norma oficial mexicana NOM-179-SCFI-2007, la cual establece los requisitos de información comercial que deben proporcionarse en los servicios de mutuo con interés y garantía prendaria, así como los elementos de información que debe contener el contrato que se utilice para formalizar la prestación de estos servicios. El acuerdo A/002/2014, DOF: 04/08/2014 en el cual se establece el funcionamiento de las casas de empeño, del registro público correspondiente a los derechos y obligaciones de los proveedores. Y el DOF: 11/11/2013, acuerdo por el que se establecen las disposiciones de carácter general para la operación, organización y funcionamiento del Registro Público de Casas de Empeño. Además de la Ley Federal de Protección al Consumidor (LFPC) protege a los consumidores que acuden con estos proveedores. En este sentido, las casas de empeño deben tener registrado su contrato ante la PROFECO.

Debido al alto número de casas de empeños que existen en el mercado, nace la necesidad de desarrollar un Sistema Integral de Casa de Empeños (SICE). El cual deberá cumplir con las necesidades de las casas de empeño; en lo administrativo, facilidad de su manejo, generará reportes en el mediano y largo plazo que ayudará en la toma de decisiones.

El objetivo de la presente tesis es el desarrollar, implementar y poner en funcionamiento el Sistema Integral de Casa de Empeños (SICE), en donde el administrador será capaz de gestionar una o más sucursales, registrará los

productos que podrán pignorar los clientes; adicionalmente se generará un recibo del préstamo prendario. Dicho sistema se desarrollará mediante la implementación de la metodología Evolutiva-Incremental, ya que dicha metodología proporciona modelos y herramientas que abarcan todas las fases del desarrollo de un sistema. La cual contempla programación estructurada y orientada a objetos, con la ayuda de diagramas, lo cual permitirá la comprensión de manera más sencilla de todas las partes involucradas. Adicionalmente, permite ir documentando el sistema conforme se va avanzando en el análisis y diseño, de tal manera que se contará con la documentación necesaria del sistema.

El sistema integral se compondrá de un **Back-End** (sistema de fondo) y un **Front-End** (sistema frontal), los cuales serán desarrollados en código abierto para no tener conflicto respecto a licencias. Además el sistema permitirá realizar las siguientes actividades:

- Administrar la operación de una sola casa de empeño y sucursales a través de la misma interfaz.
- Generar diferentes roles / perfiles por sucursal.
- Registrar clientes de manera eficiente.
- Clasificar prendas de empeño por sucursal.
- Generar reportes por sucursal.
- Agilizar el proceso de refrendo, abono a capital y finiquitos de empeños.

La presente tesis está dividida en cuatro capítulos que se describen a continuación.

En el presente capítulo se aborda los antecedentes del problema. Continuando con los conceptos generales usados en las casas de empeño; para dar cabida a la exposición de las principales normativas que rigen a las casas de empeño.

Continuando con la descripción del funcionamiento básico de la casa de empeño. En este punto del capítulo se comprenderán las necesidades del usuario, así como los elementos necesarios para acotar el alcance del problema.

En el capítulo segundo se desarrollará el marco teórico para el Sistema Integral de Casas de Empeños (SICE); de igual manera se analizarán las características, las ventajas y desventajas de los elementos que se necesitarán para el desarrollo del sistema, los cuales son: La metodología Evolutiva-Incremental, la programación orientada a objetos (PHP), el manejador de base de datos MySQL, por último y no menos importante, la arquitectura Cliente-Servidor.

En el capítulo tercero se expondrá la necesidad de desarrollar el Sistema Integral de Casa de Empeños (SICE), los requerimientos generales y particulares de ese sistema, el análisis de los lenguajes de programación para realizar el Front-End (es la parte de un sistema de software que interactúa directamente con el usuario) y el Back-End (comprende los componentes que procesan la salida del Front-End) del sistema, la aplicación y separación de los módulos según la metodología Evolutiva-Incremental y el diseño a nivel de bloques de la interfaz del usuario.

En el cuarto y último capítulo se detalla el proceso de Análisis, diseño y construcción del Sistema Integral de Casas de Empeños (SICE). Con la información obtenida en el capítulo anterior, se procederá a realizar el modelado básico del (SICE), se realizarán los diagramas:

- Contexto del sistema
- Entidad-Relación
- Flujo de datos y procesos
- Diccionario de datos y la normalización de la base de datos.

Una vez finalizadas las tareas mencionadas, se procederá a la creación de la base de datos, el desarrollo del Front-End del sistema, generación de pruebas, mantenimiento; por último reportes y comprobantes.

El desarrollo del sistema pretenderá brindar al usuario una herramienta con un Front-End amigable que le permitirá realizar actividades en forma más eficiente y de manera intuitiva; minimizando los errores humanos al hacer uso de catálogos pre-capturados. Además de que mejorará la administración de sus recursos ya que el sistema le generará el comprobante de la prenda empeñada, en la cual se mostrará el monto que se le otorgó por su prenda e indicará además el tiempo del préstamo y el tipo de empeño que solicitó el pignorante.

1.2 Conceptos generales de los términos usados por las casas de empeño

A continuación se hará mención a los términos comúnmente usados en México por las casas de empeño.

Avalúo

Un avalúo es la estimación del valor comercial de una prenda de acuerdo a sus características, la cual es determinada por el dictamen técnico de un valuador.

Abono

Un abono es un pago que integra capital más intereses, al monto de su préstamo, y de ésta forma terminar de pagarlo más rápido.

Acreeedor

Es un adjetivo que nombra a quien tiene acción o derecho a pedir el cumplimiento de alguna obligación.

Boleta prendaria / solicitud

El formato requerido y presentado por el proveedor, para iniciar un trámite ante el Registro Público de Casas de Empeño.

Capital

Es una abstracción contable que representa a los bienes y derechos (elementos patrimoniales del Activo) menos las deudas y obligaciones (Pasivo).

Casa de empeño

Todo aquel establecimiento de persona física o jurídica colectiva que otorgue préstamos de dinero al público, mediante la celebración de contratos de mutuo con interés y garantía prendaria.

Contrato prendario

Es un documento único que comprueba la operación prendaria realizada entre la institución y el deudor prendario.

El documento es elaborado unilateralmente por el proveedor, para establecer en formatos uniformes los términos y condiciones aplicables; aún y cuando dicho documento no contenga todas las cláusulas ordinarias de un contrato, por medio del cual el consumidor recibe el préstamo y garantiza su restitución a través de una prenda, el cual deberá estar inscrito y registrado en la Procuraduría Federal del Consumidor.

Cotitular o beneficiario

Persona designada por el titular para que realice en su nombre las operaciones permitidas en el contrato.

Derecho de almacenaje /gastos de almacenaje

Es un porcentaje mensual nominal que se cobra sobre la base del avalúo determinado en el contrato y/o boleta de empeño, por el resguardo de la prenda en depósito.

Desempeño o finiquito

Es el proceso mediante el cual el interesado o pignorante, luego de haber cumplido lo pactado en el contrato prendario y de acuerdo a las condiciones establecidas en

el contrato de empeño puede recuperar la prenda depositada en garantía, mediante el pago del préstamo, los intereses devengados y lo correspondiente a los gastos de almacenaje.

Deudor

Es aquella persona o empresa que está obligada a satisfacer una deuda; partimos de que el origen de la deuda es de carácter voluntario, es decir, que el deudor decidió libremente comprometerse al pago de dicha obligación.

Empeño

Es el proceso mediante el cual el interesado o pignorante, recibe en forma inmediata una suma de dinero en efectivo, a cambio de dejar en depósito y como garantía, una prenda de su propiedad.

Interés prendario

Es el porcentaje que se cobra sobre la base del préstamo determinado en el contrato y/o boleto de empeño.

Interés semanal nominal

Tasas de interés calculado por semanas completas, independientemente de la fecha en que se realice el empeño o refrendo.

NOM

Norma Oficial Mexicana

Papeleta de amarre

Se denomina así, al talón del billete de empeño que es usada para el control de la prenda en el depósito o almacén.

Partida

Se le denomina a la(s) prenda(s) que corresponden a una operación de empeño, sinónimo del número de transacciones realizadas en el empeño, desempeño, refrendo.

Pignorante o deudor prendario

Persona que solicita un préstamo con garantía prendaria.

Pignorar

Acto de dejar en prenda un objeto como garantía de un préstamo.

Plazo de pago

Lapso establecido en el contrato, en el que el consumidor restituirá el préstamo, cubrirá los pagos por intereses y demás gastos.

Prenda

Bien mueble que garantiza la seguridad o el cumplimiento de una obligación.

Préstamo

Un préstamo es una operación por la cual una entidad financiera pone a nuestra disposición una cantidad determinada de dinero mediante un contrato. En un préstamo se adquiere la obligación de devolver ese dinero en un plazo de tiempo establecido y de pagar las comisiones e intereses acordados.

PROFECO

Procuraduría Federal del Consumidor, organismo para la defensa de los derechos del consumidor en México.

Propiedad de la prenda

Derecho legal, legítimo e indiscutible de la prenda y de todo cuanto de hecho y por derecho corresponde.

Producto

Un producto es una opción elegible, viable y repetible que la oferta pone a disposición de la demanda, para satisfacer una necesidad.

Refrendo

Es el proceso mediante el cual el interesado o pignorante, cumpliendo lo pactado en el contrato prendario y de acuerdo a las condiciones del billete de empeño, puede, mediante el pago de los intereses devengados y lo correspondiente al costo del almacenaje, mantener la prenda empeñada

Reposición de contrato

Es la cantidad que se cobra por la reposición de un contrato.

RFC

Registro Federal de Contribuyentes en México.

Servicio de empeño

Lo constituye el préstamo de dinero con interés, que se otorga al consumidor, contra la entrega de una garantía prendaria.

Valuador

Persona física que posee los conocimientos necesarios, habilidades y experiencia, y que se encuentra inscrito en un registro que debe llevar la Secretaría de Finanzas, para emitir el valor de los bienes muebles para transacciones con transferencia de dominio y/o posesión.

1.3 Normas de casas de empeños

Es responsabilidad del Gobierno Federal procurar las medidas que sean necesarias para garantizar que los productos que se comercialicen en territorio nacional contengan los requisitos necesarios con el fin de garantizar los aspectos de

seguridad e información comercial para lograr una efectiva protección del consumidor.

Con fecha 14 de agosto de 2007, el Comité Consultivo Nacional de Normalización de Seguridad al Usuario, Información Comercial y Prácticas de Comercio, aprobó por unanimidad la Norma Oficial Mexicana NOM-179-SCFI-2007, servicios de mutuo con interés y garantía prendaria.

La Ley Federal sobre Metrología y Normalización establece que las normas oficiales mexicanas se constituyen como el instrumento idóneo para la protección de los intereses del consumidor, expide la siguiente:

- **Objetivo**

La presente NOM tiene por objeto establecer los requisitos de información comercial que deben proporcionarse en los servicios de mutuo con interés y garantía prendaria, así como los elementos de información que debe contener el contrato que se utilice para formalizar la prestación de estos servicios.

- **Campo de aplicación**

La presente NOM es de observancia general en la República Mexicana y es aplicable a todas aquellas personas físicas o sociedades mercantiles no reguladas por leyes financieras, que en forma habitual o profesional realicen contrataciones u operaciones de mutuo con interés y garantía prendaria.

- **Definiciones**

El contrato deberá apegarse a los términos definidos en la NOM.

- **Disposiciones generales**

La información que proporcione el proveedor debe ser veraz, comprobable y exenta de textos, diálogos, sonidos, e imágenes u otras descripciones que

induzcan o puedan inducir a error o confusión al consumidor por engañosas o abusivas.

- **De la información al consumidor**

Los proveedores deberán transparentar sus operaciones, por lo que deberán colocar en su publicidad o en todos sus establecimientos abiertos al público, de manera permanente y visible, una pizarra de anuncios o medio electrónico informativo.

- **Del contrato**

Los contratos que utilicen los proveedores deberán, para su validez:

- Estar escritos en idioma español y sus caracteres deben ser legibles a simple vista, sin perjuicio de que también puedan estar escritos en otro u otros idiomas. En caso de controversia, prevalecerá la versión en idioma español.
- Celebrarse en moneda nacional, sin menoscabo de que también pueda hacerse en moneda extranjera. En cuyo caso, el pago deberá ser solventado entregando el equivalente en moneda nacional, al tipo de cambio que rija en el lugar y fecha en que se haga el pago.
- Estar registrados ante la PROFECO.
- Estar a la vista en el establecimiento.

El contrato es el comprobante de la operación y por lo tanto el proveedor debe entregarlo al consumidor al momento de su celebración.

La información que al menos debe contener el contrato en su lado anverso es la siguiente:

- Nombre, denominación o razón social, domicilio, y Registro Federal de Contribuyentes del proveedor del servicio.
- Nombre, domicilio, número del documento oficial con que se identifica el consumidor, beneficiarios y, en su caso, el nombre del cotitular.
- Descripción de la prenda.
- Nombre o clave interna del valuador y monto del avalúo.

- Fecha en que se realiza la operación, y número de referencia.
- El monto del préstamo expresado en números.
- Tasa de interés en términos anuales, en su caso, gasto de almacenaje, Impuesto al Valor Agregado, y demás gastos necesarios y útiles que hiciera el proveedor para conservar la prenda, que el consumidor debe cubrir al desempeño.
- Plazo máximo para desempeño, forma de pago y opciones de refrendo. En caso de que el vencimiento corresponda a un día inhábil, se considerará el día hábil siguiente.
- Información completa sobre la fecha de inicio de comercialización de la prenda no desempeñada, y fecha límite para el finiquito. Así como, procedimiento y términos para finiquito, y en su caso, remanente.
- Firma del consumidor al empeñar y al desempeñar y firma del proveedor o representante legal, o por el encargado o responsable del establecimiento abierto al público. En este caso, dichas firmas podrán estar contenidas en el reverso del contrato.

- **Verificación y vigilancia**

La vigilancia de lo dispuesto en la presente NOM está a cargo de la Procuraduría, conforme a lo dispuesto por la Ley y demás ordenamientos legales aplicables.

- **Bibliografía indicada en la NOM**

- Ley Federal sobre Metrología y Normalización, Diario Oficial de la Federación, 1 de julio de 1992.sobre Metrología y Normalización, Diario Oficial de la Federación, 14 de enero de 1999.
- NMX-Z-13-1977, Guía para la Redacción, Estructuración y Presentación de las Normas Oficiales Mexicanas, declaratoria de vigencia publicada en el Diario Oficial de la Federación, 31 de octubre de 1977.

- **Concordancia con normas internacionales**

La NOM-179-SCFI-2007 no coincide con norma internacional alguna, por no existir referencia al momento de su elaboración.

1.4 Descripción básica del funcionamiento de una casa de empeño

Las personas (pignorantes) interesadas en conseguir un préstamo deberán llevar sus prendas a la casa de empeño de su elección, una vez ahí el pignorante entregará sus prendas a un valuador, para que este realice una valoración de los bienes.

El valuador determinará el monto del préstamo en relación del precio de oro y/o plata del mercado para el caso de las joyas, cabe mencionar que cada casa de empeño manejará sus propios estándares y reglas para la valoración de las prendas ya mencionadas.

Una vez establecido el valor de la prenda por parte del valuador, éste ofrecerá el monto del préstamo al pignorante y se le explicarán los términos de pago, el cual decidirá si acepta o no la propuesta económica. En el caso en que la respuesta sea positiva por parte del pignorante, el valuador realizará el proceso de registro del cliente y de la prenda. La prenda deberá ser almacenada en condiciones óptimas; posteriormente se entregará por escrito las cláusulas de la boleta y/o contrato de prenda, que indicará el monto del préstamo, tiempo de vigencia para finiquitar la deuda, así como los intereses y gastos extras que deberá pagar el deudor para recuperar la prenda. Vea figura 1.4.1

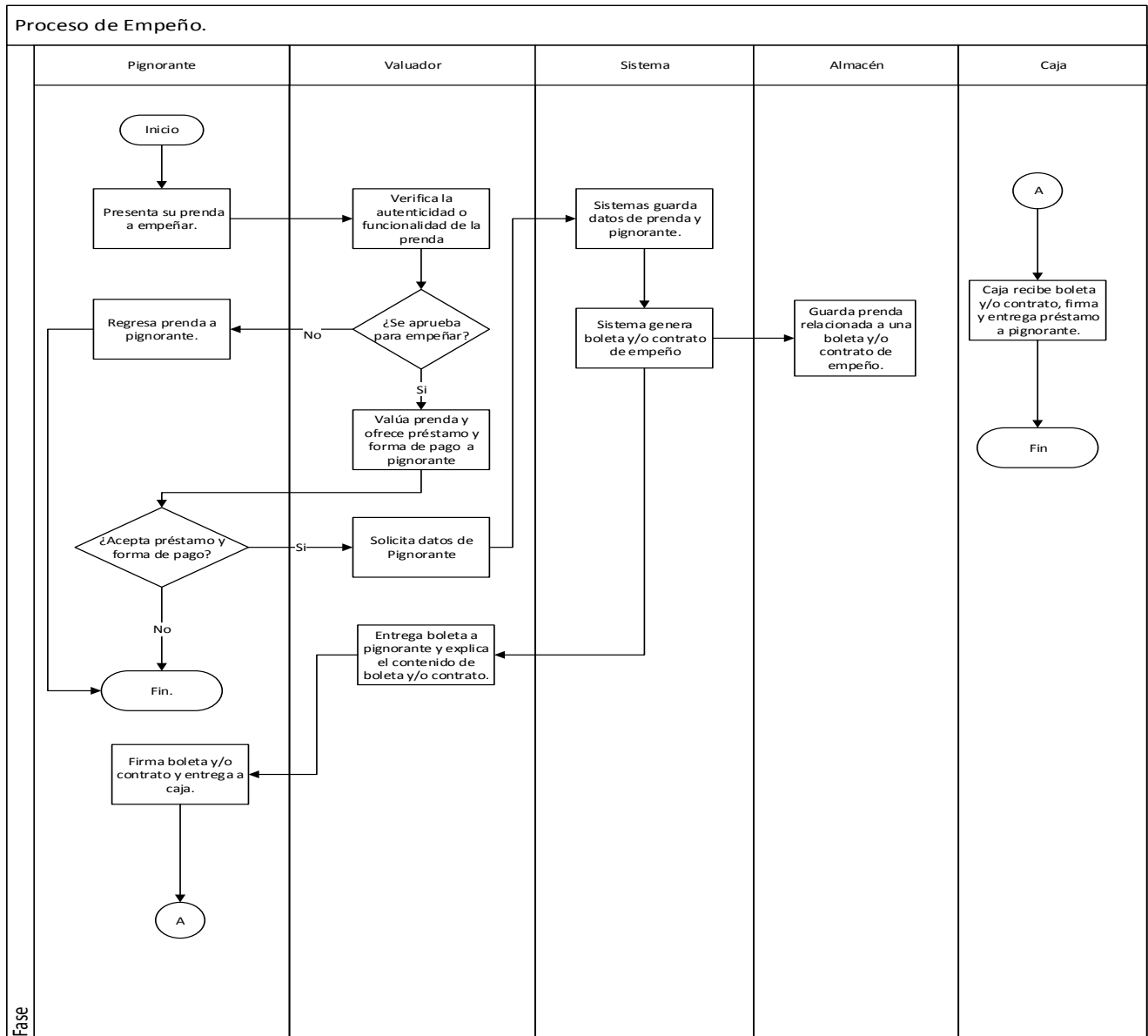


Figura 1.4.1 Proceso de empeño

Si el pignorante desea recuperar su prenda, deberá acudir a la casa de empeño, la cual le solicitará su boleta y/o contrato, identificación y el finiquito de la deuda; con lo cual podrá realizar el proceso de desemeño. Una vez validada la información solicitada, la institución (casa de empeño) deberá entregar la prenda en las mismas condiciones en las cuales fue recibida. Vea figura 1.4.2

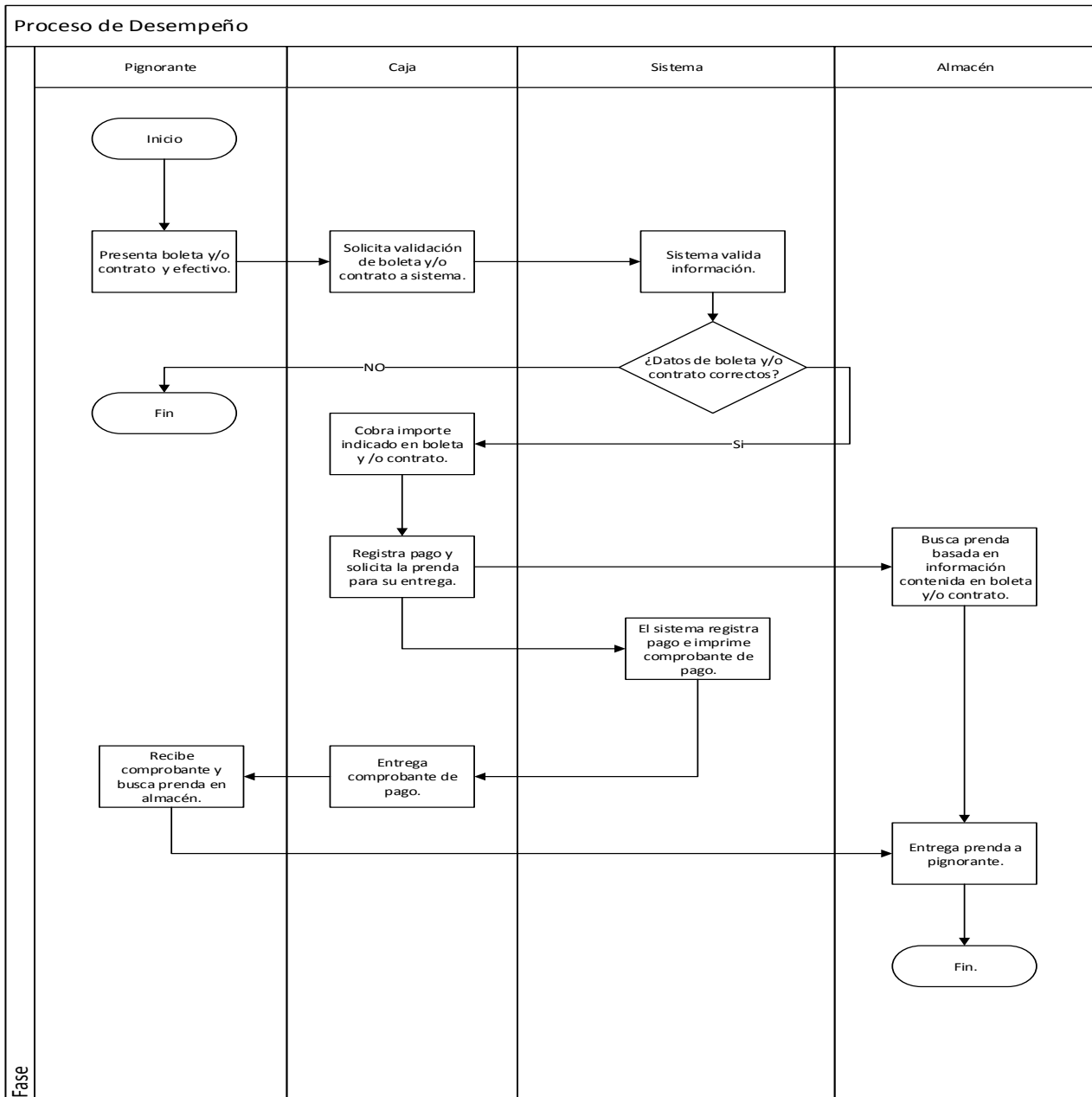


Figura 1.4.2 Proceso de desempeño

Si el deudor por algún motivo no puede pagar el total de la deuda, este podrá elegir la opción de refrendar su prenda, con el motivo de obtener más tiempo para la recuperación de la misma. Esto implicará pagar un costo adicional al monto de su préstamo. Las casas de empeño determinarán el número de veces y el costo del refrendo; la opción estará indicada en la boleta y/o contrato prendario el cual fue entregado en el proceso de empeño. Vea figura 1.4.3

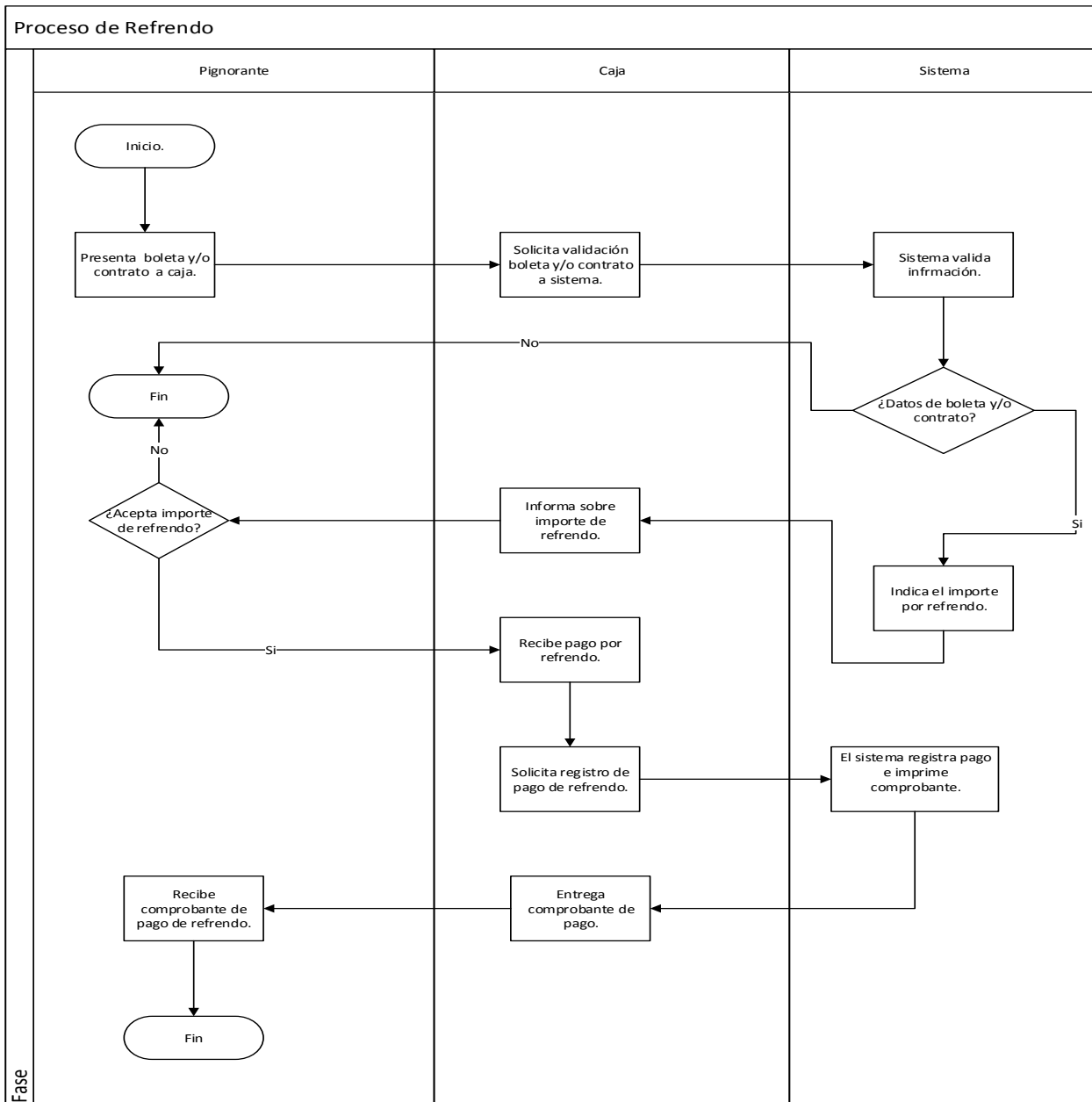


Figura 1.4.3 Proceso de refrendo

1.5 Acotamiento del problema a resolver

Se desarrollará un sistema de software que permitirá administrar las tareas y procesos de una casa de empeño.

El cual deberá cumplir con las siguientes características:

- Permitir administrar la operación de una casa de empeño y sucursales utilizando la misma interfaz.
- Acelerar el proceso de registro de clientes de una manera eficiente.
- Mantener una buena imagen del negocio.
- Inventariar prendas de empeño por sucursal.
- Generar reportes de inventario por sucursal.
- Disminuir el tiempo de registro de las características y especificaciones de las prendas.
- Realizar el proceso de refrendo, pago y desempeño/finiquito.
- Entregar el avalúo de las prendas que se desean empeñar bajo estándares y precios establecidos por la casa de empeños.
- Identificar oportunidades de mejora en las necesidades de los clientes.

Debido a la creciente demanda en el mercado de empeño, se pretende ofrecer un sistema integral, el cual esté desarrollado bajo un software de libre licenciamiento, el cual tenga un costo-beneficio favorable para el cliente, con un sistema Front-End amigable para el usuario y que de esta manera puedan interactuar de una forma intuitiva con el sistema y sea a su vez eficiente.

Para minimizar los posibles errores por parte del usuario se tratará de solventar, mediante los catálogos de pre-captura, los cuales harán más eficiente y ágil la generación de datos, almacenamiento, resguardo y actualización del inventario.

Se pretende que el sistema genere el comprobante de la prenda empeñada, monto otorgo por la prenda, los pagos que deberá realizar el pignorante, tiempo de préstamo, monto total a pagar y el tipo de empeño.

El alcance del sistema será el administrar las operaciones de empeño, la venta de las prendas no está contemplada y solo se limitara a brindar las herramientas al usuario para informar al pignorante sobre la venta de su prenda por incumplimiento de pago. **Vea apéndice 1.**

El sistema utilizará el mismo proceso de empeño y pagos para cualquier prenda, el sistema abarca en una primer versión sólo el ramo de joyería, sin embargo el cálculo para un diferente, ramo como electrónicos o electrodomésticos será diferente ya que las características cambian sin embargo el sistema deberá permitir actualizaciones y fácil adaptación de estos al sistema. **Vea apéndice 2.**

El sistema ofrecerá diversos productos de empeño, y en éstos variara la forma de capital que recibirá el cliente y de la forma en que va a pagar. **Vea apéndice 3.**

Los cálculos que se necesitarán para obtener las cantidades en efectivo que se manejarán en la casa de empeño son las fórmulas indicadas. **Vea apéndice 4.**

Los principales elementos que deberá ofrecer el sistema son:

- Empeños
- Refrendos
- Pagos
- Administración del sistema

Cotización y avalúo

Es la valoración de la prenda de acuerdo a sus características. Dependiendo de las condiciones de la misma y de la oferta comercial que realice la casa de empeño de acuerdo a las necesidades del cliente. Vea figura 1.5.2.



Pignorante acude a casa de empeño.



El valuador recibe y analiza las características de la prenda.



El valuador realiza una oferta económica al pignorante.

Figura 1.5.2 Valuación de una prenda

REFRENDO

Cada casa de empeño podrá ofrecer un nuevo plazo para recuperar la prenda, únicamente se deberá pagar los intereses generados, indicados en el contrato de empeño. Este proceso se podrá realizar tantas veces este especificado en el contrato prendario. Vea figura 1.5.3.



El pignorante acudirá a la sucursal.



Presentará la boleta de empeño en la caja.



Solicitará refrendo, pagará los intereses y gastos indicados en la boleta de empeño.



Recibirá el comprobante de pago.



Se generará la nueva calendarización de los pagos y se le entregará la boleta de empeño.

Figura 1.5.3 Refrendo de una prenda

CAPÍTULO 2.- MARCO TEÓRICO

2.1 Características, ventajas y desventajas de las Bases de Datos relacionales y metodología Evolutiva-Incremental

Introducción a las Bases de Datos (BD)

Actualmente para las empresas la información es un elemento fundamental para su desarrollo, operación y crecimiento.

Es por ello que se requiere almacenar dicha información, de tal forma que de un mejor entendimiento del negocio.

Desde el punto de vista informático, una base de datos relacional es el conjunto de datos relativos a diversas temáticas, categorizados de diversas maneras, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

Una base de datos relacional deberá cumplir con un modelo relacional, el cual permiten establecer interconexiones (relaciones) y restricciones entre los datos almacenados en dichas tablas.

Debido a la necesidad de contar con un sistema de administración para controlar datos, aplicaciones, usuarios, se han desarrollado sistemas denominados **DBMS** (Data Base Management System) o Sistema de Gestión de Base de Datos (SGBD).¹ Es un software dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

¹ Nevado, V. (2010). INTRODUCCIÓN A LAS BASES DE DATOS RELACIONALES. Visión Libros.

El diseño de una base de datos consiste en definir la estructura de los datos que deberá tener un sistema de información determinado. Para ello se requerirá diseñar el modelo Entidad - Relación.

Características de las Bases de Datos Relacionales

- Una BD se compone de “n” tablas.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es, a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las llaves primarias y foráneas.
- Las llaves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.
- Las llaves foráneas se colocan en la tabla hija, contienen el mismo valor que la llave primaria del registro padre; por medio de éstas se hacen las formas relacionales.
- Existe una independencia entre la representación lógica y física de los datos.

Cardinalidad

Es la forma en que se relacionan las entidades, o expresa cuantas entidades se relacionan con otras entidades. Cabe mencionar que una entidad es la representación de un objeto o concepto del mundo real que se describe en una base de datos.

Existen cuatro tipos de relaciones que pueden establecerse entre entidades, las cuales establecen con cuántas ocurrencias de entidad de tipo B se puede relacionar una ocurrencia de entidad de tipo A:

- **Uno a Uno: (1:1)**

Un registro de una entidad A se relaciona con solo un registro en una entidad B.

- **Uno a Muchos: (1:N)**

Un registro en una entidad en A se relaciona con muchos registros en una entidad B. Pero los registros de B solamente se relacionan con un registro en A.

- **Muchos a Uno: (N:1)**

Una entidad en A se relaciona exclusivamente con una entidad en B. Pero una entidad en B se puede relacionar con muchas entidades en A.

- **Muchos a Muchos: (N:M)**

Una entidad en A se puede relacionar con muchas entidades en B y viceversa.

Cálculo Relacional

El cálculo relacional es un lenguaje de consulta que describe la respuesta deseada sobre una Base de datos, sin especificar cómo obtenerla.

Álgebra relacional

Es un conjunto de operaciones que describen paso a paso cómo computar una respuesta sobre las relaciones, tal y como éstas son definidas en el modelo relacional.

Describe el aspecto de la manipulación de datos. Estas operaciones se usan como una representación intermedia de una consulta a una base de datos y, debido a sus propiedades algebraicas, sirven para obtener una versión optimizada y eficiente de dicha consulta.

Operadores de álgebra relacional

El álgebra relacional es un conjunto de operaciones simples sobre tablas relacionales, a partir de las cuales se definen operaciones más complejas mediante composición. Definen, por tanto, un pequeño lenguaje de manipulación de datos.

Acepta una o dos relaciones y retorna una relación como resultado. σ y π son operadores unarios, el resto de los operadores son binarios. Las operaciones básicas del álgebra relacional son:

- **Selección (σ)**

Podrá seleccionar un subconjunto de tuplas de una relación (R), todas aquellas que cumplan la(s) condición(es) P, esto es: $\sigma_P(R)$.

Una condición puede ser una combinación booleana, donde se pueden usar operadores como: \vee , \wedge combinándolos con operadores $<$, $>$, \leq , \geq , $=$, \neq .

- **Proyección (π)**

Permitirá extraer columnas (atributos) de una relación, dando como resultado un subconjunto vertical de atributos de la relación, esto es: $\Pi_{A_1, A_2, \dots, A_n}$.

Donde A_1, A_2, \dots, A_n son atributos de la relación R.

- **Producto cartesiano (\times)**

El producto cartesiano de dos relaciones se escribe como: $R \times S$.

Entregará una relación, cuyo esquema corresponde a una combinación de todas las tuplas de R con cada una de las tuplas de S, y sus atributos corresponden a los de R seguidos por los de S.

- **Unión (\cup)**

La operación: $R \cup S$.

Retorna el conjunto de tuplas que están en R, o en S, o en ambas. R y S deben ser uniones compatibles.

- **Diferencia ($-$)**

La diferencia de dos relaciones, R y S denotada por: $R - S$.

Generará todas aquellas tuplas que están en R, pero no en S. R y S deben ser uniones compatibles. Estas operaciones son fundamentales en el sentido en que:

- Todas las demás operaciones pueden ser expresadas como una combinación de éstas.
- Ninguna de estas operaciones pueden ser omitidas sin que con ello se pierda información.

No básicas o derivadas

Entre los operadores no básicos tenemos:

- **Intersección (\cap)**

La intersección de dos relaciones se puede especificar en función de otros operadores básicos: $R \cap S = R - (R - S)$.

La intersección, corresponde al conjunto de todas las tuplas que están en R y en S, siendo R y S uniones compatibles.

- **Unión natural (\bowtie) (Natural Join)**

Permitirá reconstruir las tablas originales previas al proceso de normalización. Consiste en combinar la proyección, selección y producto cartesiano en una sola operación, donde la condición θ es la igualdad Clave Primaria = Clave Externa (o Foránea), y la proyección elimina la columna duplicada (clave externa).

La expresión es: $R \bowtie S = \pi_{A1, A2, \dots, An}(\sigma_{\theta}(R \times S))$, una reunión theta (θ -Join) de dos relaciones es equivalente a: $R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$; donde la condición θ es libre. Si la condición θ es una igualdad se denomina EquiJoin.

- **División (/)**

Partiendo de que se tienen dos relaciones $A(x, y)$ y $B(y)$, donde el dominio de y, en A y B es el mismo.

El operador división A / B retorna todos los distintos valores de x tales que para todo valor y en B existe una tupla (x, y) en A.

- **Agrupación (G)**

Permite agrupar conjuntos de valores en función de un campo determinado y hacer operaciones con otros campos.²

Ventajas de las bases de datos relacionales

- **Control sobre la redundancia de datos**

En los sistemas de BD todos los ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una BD no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

- **Consistencia de datos**

Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes. Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias.

- **Compartir datos**

En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de BD, la BD pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

- **Mejora en la integridad de datos**

Se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa por restricciones o reglas que no se

² Gabillaud, J. (2010) SQL y Álgebra relacional. Ediciones ENI.

pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

- **Mejora en el mantenimiento**

En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan. Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Desventajas de las bases de datos relacionales

- **Complejidad**

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

- **Costo de equipamiento adicional**

Tanto el SGBD como la propia BD, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de BD sea más cara.

- **Vulnerable a los fallos**

El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad en disco espejo.

Modelo evolutivo

Es una representación abstracta de un proceso de desarrollo de software de una manera en particular.

La idea detrás de este modelo, es el desarrollo de una implantación del sistema inicial, exponerla a los comentarios y evaluación del usuario, mejorarla en N versiones hasta que se desarrolle el sistema adecuado.

Una ventaja de este modelo es que se obtiene una rápida realimentación del usuario, ya que las actividades de especificación, desarrollo y pruebas se ejecutan en cada iteración.

El modelo evolutivo asume que los requisitos pueden cambiar en cualquier momento del ciclo de vida y no sólo en la etapa de análisis.

Comúnmente se tienen los siguientes objetivos en esta metodología:

1. Desarrollo exploratorio

Consiste en analizar los requerimientos de los usuarios hasta llegar a un sistema final.

2. Uso de prototipos de desarrollo

Desarrollar los requisitos del usuario y trabajar para mejorar la calidad de los requisitos.

Etapas del modelo evolutivo

- **Planeación**

En esta etapa evalúa la función y el rendimiento que se asignaron al Software durante la Ingeniería del Sistema de Computadora para establecer un ámbito de proyecto que no sea ambiguo, e incomprensible.

- **Análisis de riesgos**

Esta etapa el analista se encarga de analizar los riesgos que el software a crear estará expuesto y así encontrar la manera de corregirlos.

- **Construcción y adaptación de la Ingeniería**

Esta etapa se construye el software, se prueba si no tiene algún problema o para detectar errores, se instala, y luego se le brinda soporte al cliente.

- **Evaluación del cliente**

El cliente tiene la tarea de evaluar el software para verificar si este cumple con los requisitos que este proporcione y verifica el desarrollo del sistema para aprobar o rechazar el software.

A continuación se indican algunas ventajas y desventajas del uso del Modelo Evolutivo.³ Vea tabla 2.1.1

Ventajas	Desventajas
<ul style="list-style-type: none"> • La especificación de los requerimientos puede desarrollarse de manera evolutiva. • Usuarios y desarrolladores tienen un mejor entendimiento del sistema y se refleja la mejora de la calidad del software desarrollado. • Procura satisfacer las necesidades inmediatas del cliente. 	<ul style="list-style-type: none"> • Es necesario un control de la administración de las entregar para realizar una medición del avance del proyecto. • Si el sistema se necesita desarrollar rápido, no es efectivo producir documentos que reflejen cada versión del sistema.

Tabla. 2.1.1 Ventajas y desventajas del modelo evolutivo

³ Fernández, R. EDUCACION Y TECNOLOGIA: Un binomio excepcional. Grupo Editor K.

El desarrollo de software en forma evolutiva requiere un especial cuidado en la manipulación de documentos, programas, datos de test, etc.; desarrollados para distintas versiones del software. Cada paso debe ser registrado, la documentación debe ser recuperada con facilidad, los cambios deben ser efectuados de una manera controlada. Vea figura 2.1.1.

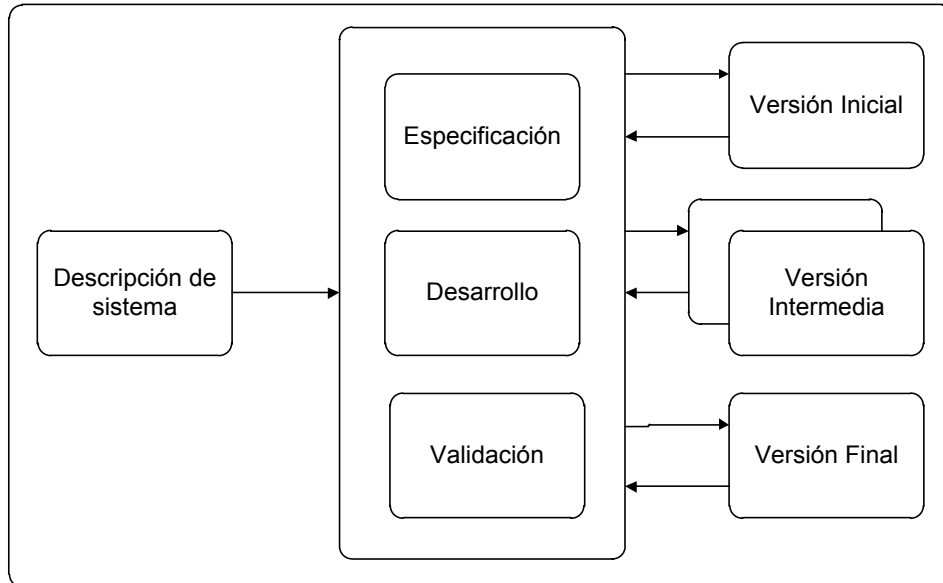


Figura 2.1.1 Modelo evolutivo

Modelo incremental

En este modelo se desarrolla el sistema para satisfacer un subconjunto de requisitos específicos; y en posteriores versiones se incrementa con nuevas funcionalidades que satisfagan nuevos requerimientos de negocio. Vea figura 2.1.2.

Se caracteriza por:

- Ser un modelo en cascada (ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior) ejecutando de manera iterativa.
- Cada secuencia produce un incremento:
 - Primero: Producto esencial.
 - Siguiendo: Características suplementarias.
- Entrega versiones “incompletas” pero funcionales.

- Se evitan proyectos largos y se entrega algo de valor a los usuarios con cierta frecuencia.
- El usuario tiene una mayor participación en el proyecto.
- Difícil de evaluar el coste total.
- Requiere gestores experimentados.
- Los errores en los requisitos se detectan tarde.

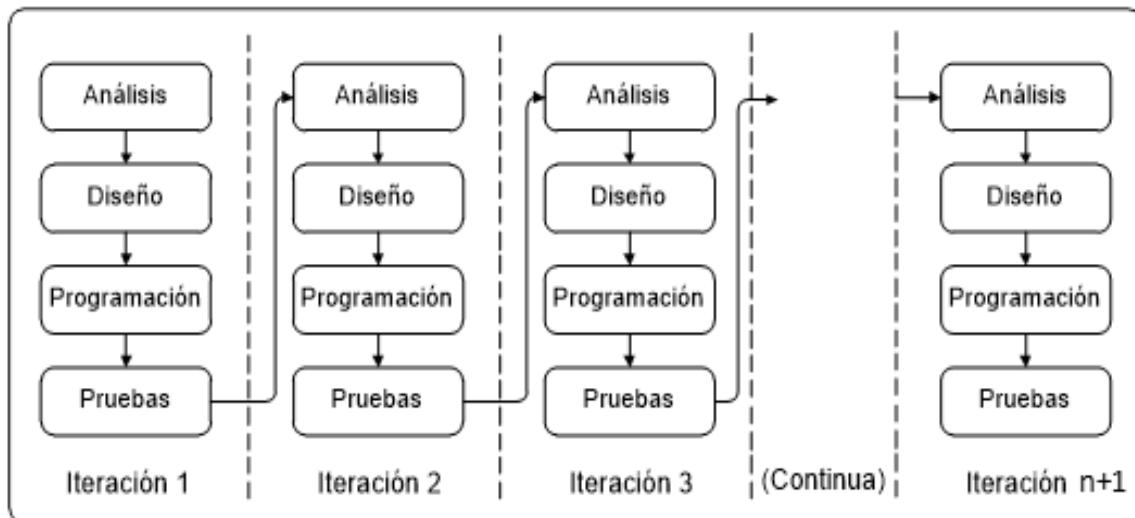


Figura 2.1.2 Modelo incremental

La arquitectura en pipeline (basada en filtros) consiste en ir transformando un flujo de datos en un proceso comprendido por varias fases secuenciales, siendo la entrada de cada una la salida de la anterior.

Esta arquitectura es muy común en el desarrollo de programas para el intérprete de comandos, ya que se pueden concatenar comandos fácilmente con tuberías (pipe). También es una arquitectura muy natural en el paradigma de programación funcional, ya que equivale a la composición de funciones matemáticas. Vea tabla 2.1.2.

Ventajas	Desventajas
<ul style="list-style-type: none"> • Con un paradigma incremental se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial. • Provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del Software. • El modelo proporciona todas las ventajas del modelo en cascada realimentado, reduciendo sus desventajas sólo al ámbito de cada incremento. • Permite entregar al cliente un producto más rápido en comparación del modelo de cascada. • Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos. • Por su versatilidad requiere de una planeación cuidadosa tanto a nivel administrativo como técnico. 	<ul style="list-style-type: none"> • No es recomendable para casos de: <ul style="list-style-type: none"> • Sistemas de tiempo real. • Alto nivel de seguridad. • Procesamiento distribuido. • Alto índice de riesgos. • Requiere de mucha planeación, tanto administrativa como técnica. • Requiere de metas claras para conocer el estado del proyecto.

Tabla. 2.1.2 Ventajas y desventajas del modelo incremental (pipeline)

Metodología evolutiva-incremental

El software como los requerimientos establecidos para desarrollarlos han evolucionado con el tiempo, es decir, estos requerimientos de usuario y de productos cambian a medida que se desarrolla el software y puede que se tenga que esperar largo tiempo hasta que el software esté listo.

El modelo del proceso evolutivo-incremental, aminora esta situación debido a la liberación de manera paulatina de versiones funcionales limitadas. Vea figura 2.1.3.

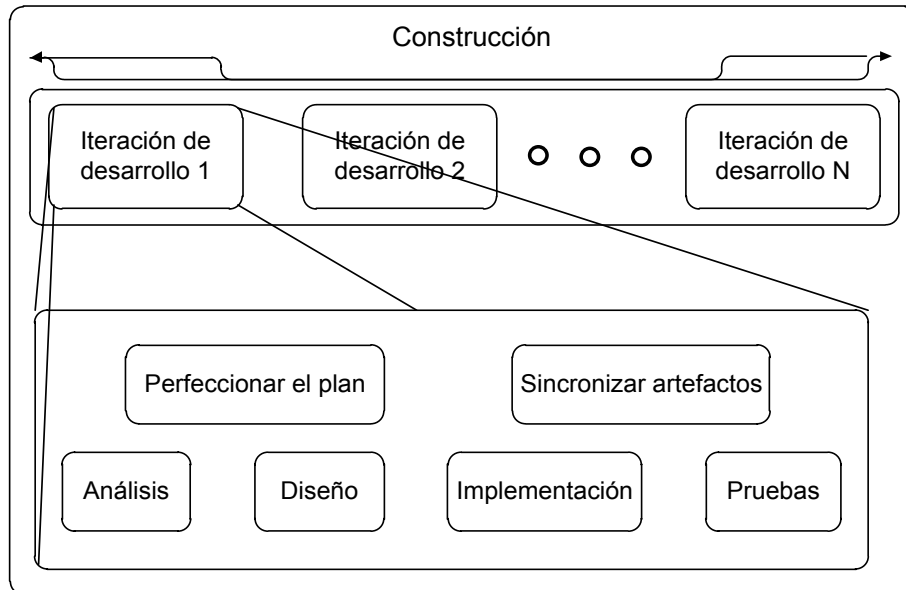


Figura 2.1.3 Fases del ciclo de vida

Beneficios

- Construir un sistema pequeño es siempre menos riesgoso que construir un sistema grande.
- Al ir desarrollando parte de las funcionalidades, es más fácil determinar si los requerimientos planeados para los niveles subsiguientes son correctos.
- Si un error importante es realizado, sólo la última iteración necesita ser descartada.
- Reduciendo el tiempo de desarrollo de un sistema (en este caso en incremento del sistema) decrecen las probabilidades que esos requerimientos de usuarios puedan cambiar durante el desarrollo.
- Si un error importante es realizado, el incremento previo puede ser usado. Los errores de desarrollo realizados en un incremento, pueden ser arreglados antes del comienzo del próximo incremento. Vea figura 2.1.4.

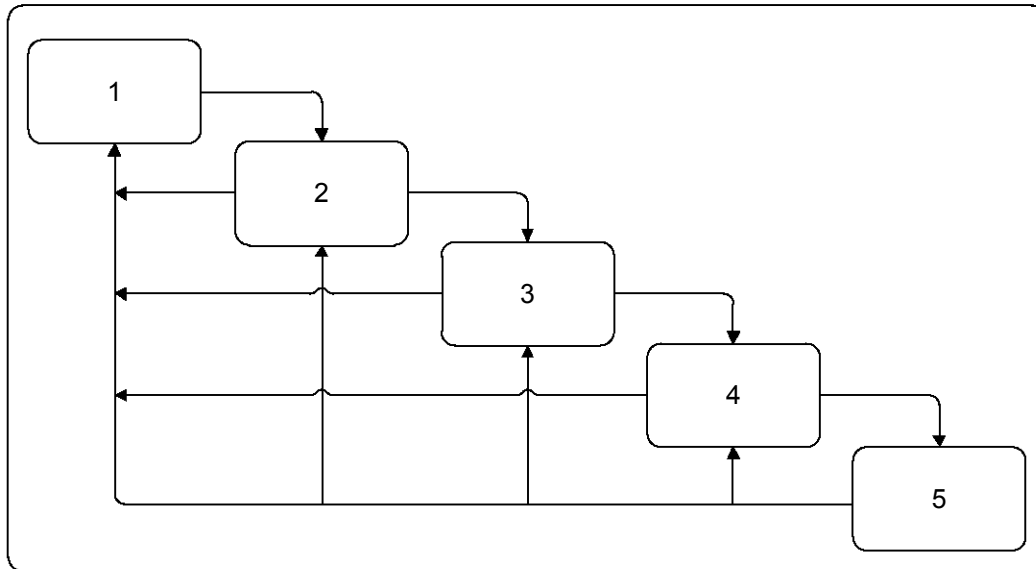


Figura 2.1.4 Esquema de una metodología Evolutiva-Incremental

2.2 Características, ventajas y desventajas de la programación orientada a objetos (POO)

Introducción a la programación orientada a objetos

Existen tres importantes conceptos en la programación orientada a objetos: utiliza objetos, no algoritmos, como bloques de construcción lógicos; cada objeto es una instancia de una clase; las clases se relacionan unas con otras por medio de relaciones de herencia. A continuación se describe cada uno de estos conceptos brevemente:

Objeto

Un objeto es la unidad atómica que encapsula un estado y comportamiento, éste puede caracterizar una entidad física o abstracta. En el mundo cotidiano un objeto representa un elemento identificable que está constituido por ciertas características o atributos y es capaz de realizar un conjunto de acciones u operaciones. En este sentido, un objeto posee:

- Estado: El estado de un objeto lo constituyen todos los datos que encapsula en un momento determinado.
- Comportamiento: la manera en que actúa y reacciona un objeto, esto en función de sus cambios de estado y el paso de mensajes.
- Identidad: La idea es que los objetos no se definen por los valores de sus atributos en un momento determinado, un objeto tiene una existencia continua.

En la programación orientada a objetos, un objeto es una instancia de una clase. Los objetos le ofrecen al programador una herramienta para implementar: modularidad y simulación.

Clase

Una clase es un conjunto de objetos que comparten una estructura y un comportamiento común, describe un conjunto de objetos que tienen un rol o roles equivalentes en un sistema.

En otras palabras una clase es un grupo de objetos con características comunes (atributos), comportamiento común (operaciones) y relaciones comunes con otros objetos (asociaciones y agregaciones). Una clase es una definición abstracta de un objeto que:

- Enfatiza las características relevantes de los objetos.
- Define la estructura y comportamiento de un objeto.
- Sirve como "plantilla" para crear nuevos objetos.

Método

Como se mencionó anteriormente, los objetos pueden realizar operaciones, dichas operaciones se conocen como métodos. Un método es un algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje".

Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Características de la programación orientada a objetos

La POO cuenta con una serie de técnicas o características que incluyen: abstracción, herencia, jerarquía, polimorfismo, encapsulamiento y modularidad.

Abstracción

Es la propiedad que nos permite representar las características esenciales de un objeto, permitiéndonos omitir las propiedades y acciones de un objeto que no son de nuestro interés y dejar sólo aquellas que nos interesan para una situación en particular.

Herencia

Consiste en trasladar al contexto de objetos el más puro significado de la palabra "herencia", es decir el hecho de que una clase "padre" herede características, estructuras y comportamiento a todas las clases "hijas". En sentido estricto se reutilizan las clases para definir nuevas.

La clase general o "padre" es llamada superclase y a la clase especializada o "hija" es llamada subclase. La superclase define un comportamiento (protocolo) que todas las subclases heredan. Asimismo, las subclases pueden agregar nuevos atributos y operaciones. Las subclases pueden sobre-escribir operaciones conservando la firma, es decir la manera en la que está declarado, pero proporcionando distintas implementaciones.

Jerarquía

Permite clasificar y ordenar a los objetos de dos maneras:

- Generalización/Especialización. Organización establecida de lo particular a lo general, conceptualmente se utiliza la frase "es un" para establecer la generalización.
- Agregación. Organización establecida de las partes hacia el "todo", conceptualmente se utiliza la frase "es parte de" para establecer la agregación.

Polimorfismo

Es la propiedad por la cual dos o más clases responden al mismo mensaje cada uno de manera diferente, lo que permite realizar la misma operación de forma diferente en cada objeto.

Encapsulamiento

Consiste en un mecanismo que reúne los elementos como datos y métodos de una estructura a una misma entidad en un mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema, es decir, evita el acceso a datos por cualquier otro medio distinto a los especificados. Por lo tanto, el encapsulamiento garantiza la integridad de los datos que contiene un objeto.

Modularidad

A través de la modularidad, se propone al programador dividir su aplicación en varios módulos diferentes (ya sea en forma de clases, paquetes o bibliotecas), cada uno de ellos con un sentido propio.

Esta división reduce el grado de complejidad del problema al que dará respuesta el programa, además de facilitar la comprensión del programa.

Ventajas de la programación orientada a objetos

- La reutilización de código, así como el manejo de clases y objetos permite un desarrollo más rápido, flexible y modular de sistemas y aplicaciones.
- La POO favorece la creación de mejores estructuras de información.
- Facilita el mantenimiento y escalabilidad de una aplicación, ya que entre otras cosas permite la incorporación de otras aplicaciones o componentes externos.
- Disminución de los costos de desarrollo.
- Incremento de la calidad del sistema.
- El uso y aplicación de la metodología orientada a objetos facilita el diseño e implementación de los sistemas con interfaces gráficas de usuario, así como nos permiten desarrollar sistemas inmersos de tiempo real con una mejor calidad y flexibilidad.

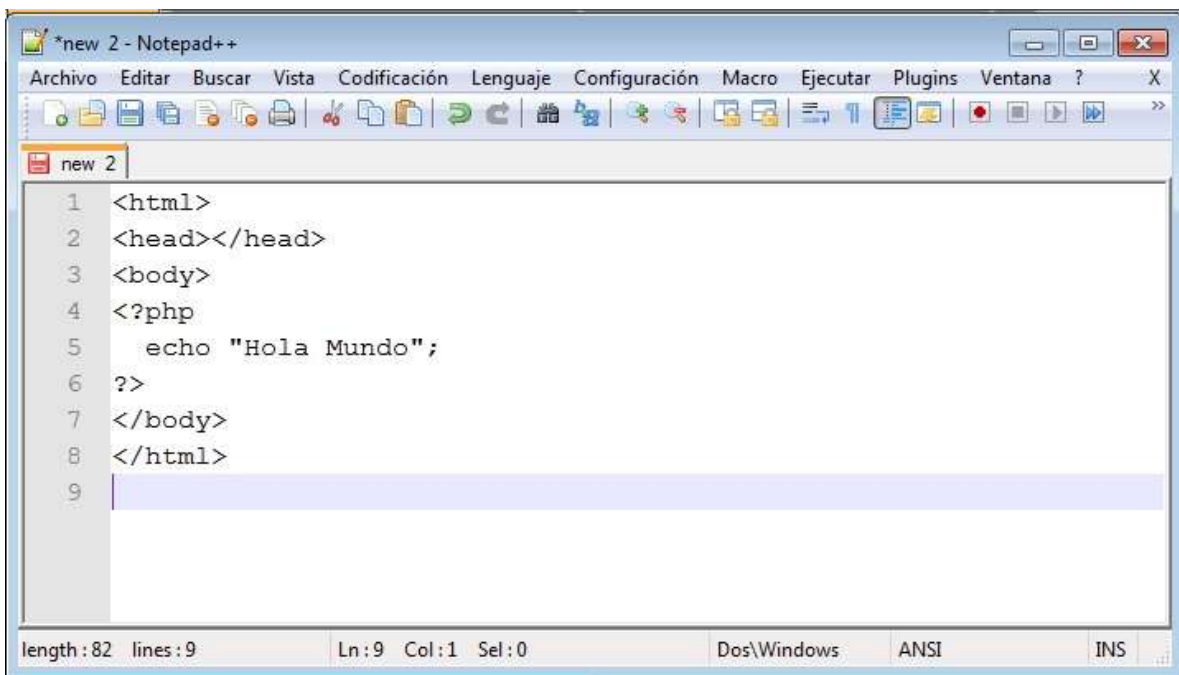
Desventajas de la programación orientada a objetos

- No todos los programas pueden ser modelados con exactitud por el modelo de objetos.
- Si lo que se desea es leer algunos datos, hacerles algo simple y escribir de nuevo, no es necesario definir clases y objetos. Sin embargo, en algunos lenguajes de POO, puede que haya que realizar este paso extra.
- Si se fuerza el lenguaje en el concepto de programación orientada a objetos, se pierden algunas de las características de lenguajes útiles, como los "lenguajes funcionales".
- El concepto que un programador tiene de lo que constituye un objeto abstracto puede no coincidir con la visión de otro programador.
- Los objetos a menudo requieren una extensa documentación.

2.3 Características, ventajas y desventajas del lenguaje de programación Hypertext Pre-processor (PHP)

Lenguaje de programación PHP

PHP es un lenguaje de código abierto especialmente adecuado para el desarrollo web y que puede ser incrustado dentro del código HTML. En las páginas con PHP en lugar de existir una gran cantidad de comandos de HTML de salida (como sería en el caso de los lenguajes C o Perl), contienen HTML con código incrustado PHP, que hace "ejecutar una acción" (Vea figura 2.3.1). El código PHP se incluye entre las instrucciones especiales "<? php" y "?>" Que le permiten a las páginas HTML entrar y salir del "modo PHP".⁴

A screenshot of the Notepad++ text editor window. The title bar reads '*new 2 - Notepad++'. The menu bar includes 'Archivo', 'Editar', 'Buscar', 'Vista', 'Codificación', 'Lenguaje', 'Configuración', 'Macro', 'Ejecutar', 'Plugins', 'Ventana', and '?'. The toolbar contains various icons for file operations and editing. The main text area shows the following code:

```
1 <html>
2 <head></head>
3 <body>
4 <?php
5     echo "Hola Mundo";
6 ?>
7 </body>
8 </html>
9
```

The status bar at the bottom indicates 'length : 82 lines : 9', 'Ln : 9 Col : 1 Sel : 0', 'Dos\Windows', 'ANSI', and 'INS'.

Figura 2.3.1 Ejemplo básico PHP

PHP permite a los desarrolladores web realizar páginas dinámicas. Esto se logra cuando se hace la petición de un cliente a un servidor web, de una página HTML con PHP embebido, el servidor pasa el control al intérprete de PHP para que ejecute

⁴ <http://php.net/manual/en/intro-what-is.php>

la acción del contenido del código en la página, el intérprete devuelve el resultado al servidor web y éste despliega el resultado al cliente en código HTML.

Características del lenguaje de programación PHP⁵

- **Permite autenticación HTTP con PHP**

Es posible usar un mensaje "Autenticación Requerida" al navegador del cliente causando que se abra una ventana para ingresar usuario y contraseña. Una vez se ha llenado el usuario y la contraseña, la URL contenida dentro del script PHP será llamada nuevamente con las variables predefinidas, el nombre del usuario, la contraseña y el tipo de autenticación respectivamente.

- **Soporta el uso de Cookies**

PHP soporta cookies HTTP de forma transparente. Se puede configurar el uso de cookies usando la función `setcookie()` o `setrawcookie()`. Las cookies son parte de la cabecera HTTP, la función será invocada antes de que cualquier otra salida sea enviada al navegador.

- **Manejo de sesiones**

El soporte de sesiones en PHP consiste en una manera de guardar ciertos datos a través de diferentes accesos web.

- **Manejo de XForms**

PHP permite el manejo de XForms, que implementa una variación de los webforms tradicionales. Los XForms son usados en una gran variedad de plataformas y navegadores, permitiendo la implementación de formularios más dinámicos y ofrecen más funcionalidades.

⁵ <http://php.net/manual/es/features.php>

- **Permite cargar archivos con el método POST**

Esta característica permite que los usuarios envíen tanto archivos de texto como binarios. Con la autenticación de PHP y las funciones de manipulación de archivos, se tiene completo control sobre quién está autorizado a cargar y qué hay que hacer con el archivo una vez que se ha cargado. PHP es capaz de recibir cargas de archivos de cualquier navegador compatible con RFC-1867. También soporta el método PUT para la carga como lo utilizan los clientes Netscape Composer y Amaya del W3C.

- **Subida de múltiples archivos**

PHP permite la subida de múltiples archivos utilizando diferentes nombres para los archivos de entrada. También es posible subir múltiples archivos simultáneamente y tener la información organizada automáticamente en arreglos.

- **Manipular archivos remotos**

Siempre que se tenga habilitada la opción `allow_url_fopen` dentro del archivo de configuración en `php.ini`, se pueden usar URLs HTTP y FTP con la mayoría de las funciones que toman un nombre de archivo como parámetro.

- **Manejo de conexiones**

Internamente, en PHP se mantiene un estado de conexión. Cuando un script de PHP está ejecutándose normalmente, está activo el estado NORMAL. Si el cliente remoto se desconecta, se activa el indicador de estado ABORTED. Si se alcanza el tiempo límite impuesto por PHP, se activa el indicador de estado TIMEOUT.

- **Conexiones persistentes a bases de datos**

Las conexiones persistentes son enlaces que no se cierran cuando la ejecución del script termina. Cuando una conexión persistente es solicitada, PHP valida si ya existe una conexión persistente idéntica (que fuera abierta antes) y si existe,

la usa. Si no existe, crea el enlace. Una conexión "Idéntica" es una conexión que fue abierta por el mismo host, con el mismo usuario y la misma contraseña.

- **Recolección de basura**

Tradicionalmente, los mecanismos que contabilizan las referencias en memoria, fallaban al manejar las fugas de memoria en referencias cíclicas. Sin embargo, PHP ha implementado un algoritmo síncrono para la recolección de ciclos concurrentes en sistemas de contabilidad de referencias que resuelve éste asunto.

- **Utilizar extensiones PECL**

PECL es el repositorio oficial de extensiones de PHP disponible mediante el sistema de paquetes denominado PEAR. Para poder usar una extensión ésta debe de ser descargada, instalada y cargada en el archivo php.ini.

Ventajas del lenguaje de programación PHP

- Es un lenguaje multiplataforma. PHP puede ser instalado en servidores basados en Linux, incluyendo aquellos sistemas operativos OS X y también en los servidores con sistema operativo Windows. Además el código PHP se ejecuta directamente en el servidor y no en el cliente, solo se envía al cliente el resultado en HTML para que sea mostrado en el navegador.
- Orientado al desarrollo de aplicaciones web dinámicas. Debido a que permite el Manejo de XForms, sesiones, el uso de cookies y conexiones a bases de datos.
- Permite el acceso a información almacenada en una base de datos. Debido a que se pueden implementar el uso de conexiones persistentes a bases de datos y cuenta con la capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, entre ellos MySQL y PostgreSQL.
- La programación en PHP es más segura y confiable. Debido a que el código PHP se ejecuta directamente en el servidor y no en el cliente y solo se envía al cliente el resultado en HTML al navegador.

- Permite expandir su potencial utilizando módulos (llamados extensiones). Las extensiones pueden obtenerse desde la página pecl.php.net/. Que es el repositorio oficial de extensiones de PHP.
- Cuenta con documentación en su sitio web oficial.
- Es de código abierto, por lo tanto es alternativa más redituable para el usuario.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Soporta varios servidores web.
- La sintaxis de PHP es muy similar a la del lenguaje de programación C. Por lo que cualquiera que conozca el lenguaje de programación C puede aprender mucho más fácilmente el lenguaje de programación PHP.

Desventajas del lenguaje de programación PHP

- Necesita tener instalado un servidor web.
- Consume más recursos.
PHP se ejecuta en el servidor, provocando que se tarde más tiempo en mostrar una página Web, de lo que se tardaría mostrando una página que ha sido creada con HTML.
- No nació como un lenguaje de programación orientado a objetos.
Lo cual no puede permitir explotar al máximo las ventajas de la programación orientada a objetos en aplicaciones muy complejas.
- Se puede presentar el caso de que el código fuente sea poco legible.
Debido a que se tiene que incrustar el código PHP dentro del código HTML puede verse afectada la legibilidad al no documentar con comentarios el código.
- No está estandarizado el acceso a las base de datos.
Existen diferentes sintaxis para los diferentes tipos de manejadores de bases de datos.

2.4 Características, ventajas y desventajas del manejador de Bases de Datos MySQL

Introducción al manejador de Base de datos MySQL

My Structured Query Language mejor conocido como MySQL, SGBD ampliamente usado por su simplicidad y notable rendimiento.

Fue creado por la empresa MySQL AB como software de código abierto, este hecho se remonta a principios de la década de 1980. Hoy en día MySQL es subsidiaria de Oracle Corporation.

Es licenciado bajo (GPL) Licencia Pública General de la GNU. Sin embargo cuenta con una versión comercial la cual ofrece soporte técnico y la posibilidad de integrar este gestor en un software propietario, ya que de lo contrario se vulneraría la licencia GPL.

Si un desarrollador desea incorporar MySQL en su producto pero desea distribuirlo bajo otra licencia que no sea la GNU GPL, puede adquirir una licencia comercial de MySQL que le permite hacer justamente eso.

Hace uso del lenguaje de programación Structured Query Language (SQL) que fue desarrollado por IBM 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

MySQL está disponible para múltiples plataformas; sin embargo, las diferencias con cualquier otra plataforma son prácticamente nulas.

Aunque MySQL ha evolucionado gradualmente para apoyar las necesidades de mayor escala, comúnmente se utiliza en pequeña escala para implementaciones de servidor único medio; ya sea como un componente de aplicación web basada en LAMP o como servidor de base de datos independiente.

LAMP es la plataforma para desarrollo y ejecución de aplicaciones web de alto performance, dada por el uso en conjunto de Linux, Apache, MySQL y PHP. Gran parte del atractivo de MySQL se origina en su relativa simplicidad y facilidad de uso, que se activa mediante un conjunto de herramientas de código abierto. En el rango medio, MySQL se puede escalar mediante la implementación en hardware más potente, como un servidor multi-procesador con Gigabytes o Terabytes de memoria.⁶ Vea figura 2.4.1.



Figura 2.4.1 Logo de MySQL

Características del manejador de Base de datos MySQL

MySQL tiene una serie de características que se generaron evolutivamente, entre ellas son:

- Está desarrollado en C/C++.
- Soporte Multi-Usuario. Múltiples clientes tienen acceso concurrente a una o más bases de datos simultáneamente.
- Sistema de privilegios de usuario potente y flexible.

⁶ <http://dev.mysql.com/doc/refman/5.6/en/what-is-mysql.html>

- Esquema de autenticación basado en usuario-máquina.
- Incluye un potente entorno gráfico de administración, que permite el uso de comandos DDL (Data Definition Language) y DML (Data Manipulation Language) gráficamente.
- Cumplimiento de estándares SQL
- Soporte de varios lenguajes y juego de caracteres.
- Procedimientos y funciones almacenados.
- Contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.
- Planificador de Eventos.
- Se distribuyen ejecutables para diversas plataformas entre las que destacan AIX, Mac OS X, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8 y Windows Server (2000, 2003, 2008 y 2012), BSD, Free BSD, HP-UX, Open Solaris, GNU/Linux, Sun OS, SCO OpenServer, SCO UnixWare, Perl y Python por mencionar algunas.
- La API se encuentra disponible en C, C++, Eiffel, Java, Perl, PHP, Ruby, TCL, etcétera.
- Está optimizado para equipos de múltiples procesadores.
- Es destacable su velocidad de respuesta.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Sus opciones de conectividad abarcan TCP/IP, sockets UNIX y sockets NT; soporte Secure Sockets Layer (SSL). Además de soportar completamente Open Data Base Connectivity (ODBC).
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Un amplio subconjunto de ANSI SQL 99, así como extensiones.
- Procedimientos almacenados, utilizando un lenguaje de procedimientos que se adhiere estrechamente al estándar ISO, SQL/ Persistent Stored Modules.
- Permite el uso de Disparadores (Triggers), Cursores y Vistas actualizables.

- Almacenamiento en caché de consultas.
- Sub-selecciones; permite realizar SELECTs anidados.
- Soporte de réplica con un maestro por esclavo, muchos esclavos por cada maestro. Replicación multi-master se proporciona en MySQL Cluster, y soporte multi-master se puede agregar a configuraciones no agrupados utilizando Galera Cluster.
- Indización de texto completo y búsqueda usando el mecanismo de almacenamiento de datos MyISAM motor.
- Permite almacenar tablas con particiones.
- Copia de seguridad en caliente bajo ciertas condiciones.
- La interfaz gráfica de usuario, conocida también como GUI (Graphical User Interface)
- Biblioteca de Base de datos integrada. Cumplimiento de **ACID** (Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad), utilizando motores de almacenamiento.
- MySQL soporta varios esquemas para tablas y permite la incorporación de esquemas definidos por el usuario.
- Motores de almacenamiento independientes.
- Apoyo al proceso de transacciones distribuidas XA; confirmación en dos fases como parte de esto, utilizando el motor InnoDB de Oracle.
- Analizar, repasar y reparar tablas.
- Administrador Visual para la gestión de usuarios y sus privilegios.
- Permite Importar y Exportar datos desde MS Excel y MS Access.
- Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.

Ventajas del manejador de Base de datos MySQL

- Facilidad de configuración, instalación y administración.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.

Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- A través del examinador de consultas permite construir y testear consultas.
- Conectividad y seguridad robusta.
- Es altamente confiable en cuanto a estabilidad se refiere. Ninguna característica es implementada en MySQL si antes no se tiene la certeza que funcionará con la mejor velocidad de respuesta y, por supuesto, sin causar problemas de estabilidad.
- MySQL es el manejador de bases de datos más utilizado en el desarrollo web de gran tráfico por su motor de consultas de alto rendimiento, su posibilidad de insertar datos a gran velocidad, y un buen soporte para funciones web especializadas como las búsquedas fulltext. Estas mismas fortalezas también se aplican a entornos de data warehousing, donde MySQL escala hasta el rango de los terabytes tanto para un solo servidor, como para varios. Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.
- Compatibilidad con XML.
- Permite informes con carácter Unicode.
- Importación y exportación de conexiones.
- Escritura predictiva para los editores de SQL.
- Vista del diagrama Entidad-Relación.
- Velocidad al realizar las operaciones: realizando gestiones en milisegundos y en algunos casos, microsegundos, mostrándose como uno de los gestores con mejor rendimiento.
- La flexibilidad de plataforma es una característica clásica de MySQL, soportando distintas versiones de Linux, UNIX y Windows y, por supuesto, la naturaleza

open source de MySQL permite una personalización completa para aquellos que deseen añadir características al servidor.

- Existe una estructura de privilegios que permite que los usuarios sólo puedan acceder a los datos que se les permite, así como potentes funciones de cifrado y descifrado para asegurarse de que los datos están protegidos. Finalmente, se ofrecen utilidades de backup y recuperación por parte de MySQL y terceros, que permiten copias completas, tanto lógicas como físicas, así como recuperación point-in-time.

Desventajas del manejador de Base de datos MySQL

- Al comprender sus principios de diseño, se puede explicar mejor las razones de algunas de sus carencias.
El soporte de transacciones o la integridad referencial (la gestión de claves foráneas) en MySQL está condicionado a un esquema de almacenamiento de tabla concreto, de forma que si el usuario no va a usar transacciones, puede usar el esquema de almacenamiento “tradicional” (MyISAM) y obtendrá mayor rendimiento, mientras que si su aplicación requiere transacciones, deberá usar el esquema que lo permite (InnoDB), sin ninguna otra restricción o implicación.
- No incluye características de objetos como tipos de datos estructurados definidos por el usuario, herencia etc.
- Para el control de acceso, le hace falta: protección ante Ataques de Fuerza bruta, compatibilidad con directorios corporativos, reglas para la complejidad de password y certificación de seguridad.
- Lentitud en el manejo de grandes volúmenes de información.
- Puede provocar problemas de integridad en entornos de alta concurrencia.
- La función de conversión CAST, no es funcional para INT y BIGINT.
- Debe usarse explícitamente un comando REVOKE para quitar los privilegios de una tabla.
- MySQL no tiene un panel de control gráfico y detallado.
- No permite el modo de autenticación local, sólo el modo estándar.
- Un gran porcentaje de las utilidades de MySQL no están documentadas.

- No posee vistas materializadas, pero se pueden simular mediante los procedimientos almacenados y disparadores.
- MySQL, como la mayoría de bases de datos relacionales transaccionales, está fuertemente limitado por el rendimiento del disco duro. Esto es especialmente cierto en términos de latencia de escritura. Dada la reciente aparición de grado discos de estado sólido muy accesible al consumidor, tales como la interfaz SATA que ofrecen latencia mecánica cero, cinco veces más veloz incluso sobre una matriz RAID (Redundant Array of Inexpensive Disks) el cual puede obtenerse por una menor inversión.

2.5 Características, ventajas y desventajas del ambiente Cliente – Servidor

Introducción del ambiente Cliente - Servidor

En los ambientes cliente-servidor, los clientes realizan diferentes tareas desde terminales Front-End (sistemas frontales) en interacción con sistemas servidores denominados Back-End (servicio de fondo) el cual proporciona acceso a una base de datos, gestión de archivos y lógica de negocio de un sistema.

En una relación cliente-servidor, el procesamiento se divide en dos partes. El sistema cliente realiza una petición a través de una interfaz de usuario, en donde la aplicación da formato a las peticiones y muestra la información o los mensajes enviados al servidor. El servidor realiza el procesamiento de la información recibida y comienza la extracción de información de una base de datos, todo aquello para lo que este dedicado el servidor, una vez realizada la tarea, el cliente recibe una respuesta satisfactoria o no satisfactoria a su petición, dependiendo de la aplicación (cliente) como interprete la respuesta recibida. Vea figura 2.5.1.

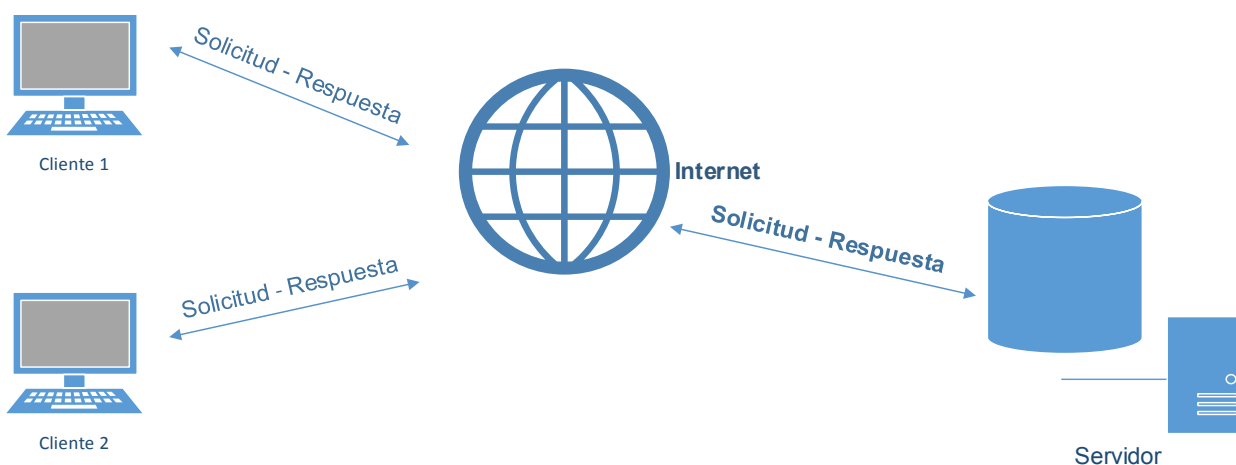


Figura 2.5.1 Cliente-servidor

El software de un sistema cliente/servidor habitualmente consiste de un sistema SGBD instalado en un servidor Back-End, hacia el que los clientes dirigen sus peticiones a través de SQL, Structured Query Language (Lenguaje de consulta estructurado), por lo general disponen de un sistema de OLTP, On-Line Transaction Processing (Procesamiento de transacciones en línea).

Características del modelo Cliente - Servidor

A continuación se describen las características más relevantes del modelo cliente/servidor.

- El cliente siempre inicia las solicitudes al servidor, y espera una respuesta de este mismo.
- El servidor puede procesar solicitudes de uno o más clientes.
- Los clientes utilizan los mismos servidores y, a través de ellos, comparten tanto recursos lógicos como físicos.
- El cliente es quien interactúa con el usuario final.
- El cliente no tiene por qué saber dónde se encuentra situado el recurso que desea utilizar.
- Independencia de la plataforma de Hardware y Software que se emplee en el cliente y el servidor.

- El servidor se puede comunicar con otros servidores para la atención de una solicitud.
- Los detalles de la implementación de un servicio son transparentes al cliente.
- La integridad de datos y programas centralizados en servidores facilitan su mantenimiento y actualización.

Ventajas del Modelo Cliente - Servidor

A continuación se describen las algunas de las ventajas del modelo cliente – servidor:

- El modelo cliente-servidor facilita el procesamiento de información independiente del sistema operativo en que se ejecuten las aplicaciones (clientes) que solicitan peticiones al servidor.
- Escalabilidad de acuerdo a la demanda de información de forma horizontal (agregando/eliminando clientes) y de forma vertical (migración de servidores más robustos o el aumento en su número).
- Evita la concurrencia de datos.
- El servidor puede ser optimizado para el procesamiento y almacenamiento de la información.
- Con datos centralizados, los administradores pueden aplicar controles de seguridad para restringir los accesos a los datos y pueden utilizar mecanismos de rastreo para monitorear los accesos a los datos.
- Cifrado de la información del cliente al servidor o viceversa.
- Facilidad para actualizar la información por estar centralizada.
- El cliente no es responsable del procesamiento de datos.

Desventajas del Modelo Cliente - Servidor

A continuación se describen las algunas de las desventajas del modelo Cliente - Servidor:

- Costo elevado en los equipos servidores.
- Congestión de la red debido al número alto de peticiones al servidor.

- Se necesita un sistema extra de balanceo de carga.
- Se necesita una réplica del servidor, para en caso de contingencia no se detenga la producción.

Arquitectura de una capa

Las aplicaciones de una capa tienen la característica, de concentrar la lógica de presentación, aplicación y fuente de datos en una sola entidad, por lo que la aplicación funge como cliente y servidor simultáneamente.

Arquitectura de dos capas

En el modelo de dos capas, el cliente mantiene la lógica de la aplicación, y se conecta de manera directa al servidor de aplicaciones, el cual administra las peticiones que realiza hacia la base de datos. Este tipo de arquitecturas es ideal para entornos pequeños (50-70 usuarios) por el bajo volumen de procesamiento de datos, ya que en el servidor de aplicaciones reside la base de datos centralizada.

Arquitectura de tres capas

El modelo de tres capas, está compuesto por clientes y servidores de aplicaciones, la capa intermedia que existe entre ellos, es denominada capa de presentación, la cual se encarga de proveer de servicios, regula el manejo de transacciones y reglas de negocio. En el lado del servidor de aplicaciones, tiene la tarea de establecer comunicación con el servidor de base de datos el cual mediante procedimientos almacenados extrae la información de la base de datos. Vea figura 2.5.2.

Capa de usuario

Los componentes del nivel de usuario, proporcionan la interfaz visual que los clientes utilizarán para ver la información y los datos. En este nivel, los componentes son responsables de solicitar y recibir servicios de otros componentes del mismo nivel o del nivel de servicios de negocio.

Capa de presentación – aplicación

Los servicios de usuario no pueden contactar directamente con el nivel de capa de datos, es responsabilidad de los servicios de aplicación hacer de puente entre estos. Los objetos de presentación proporcionan servicios que completan las tareas de negocio tales como verificar los datos enviados por el cliente. Antes de llevar a cabo una transacción en la base de datos.

Los componentes de los servicios de aplicación, también nos sirven para evitar que el usuario tenga acceso directo a la base de datos por medio de la autenticación, lo cual proporciona mayor seguridad en la integridad de ésta.

Capa de datos

La capa de datos, se encarga de la tarea que se realizan con los datos como son la inserción, actualización, consulta y borrado. El servicio de datos es responsable de la gestión de las peticiones realizadas por un objeto de negocio.

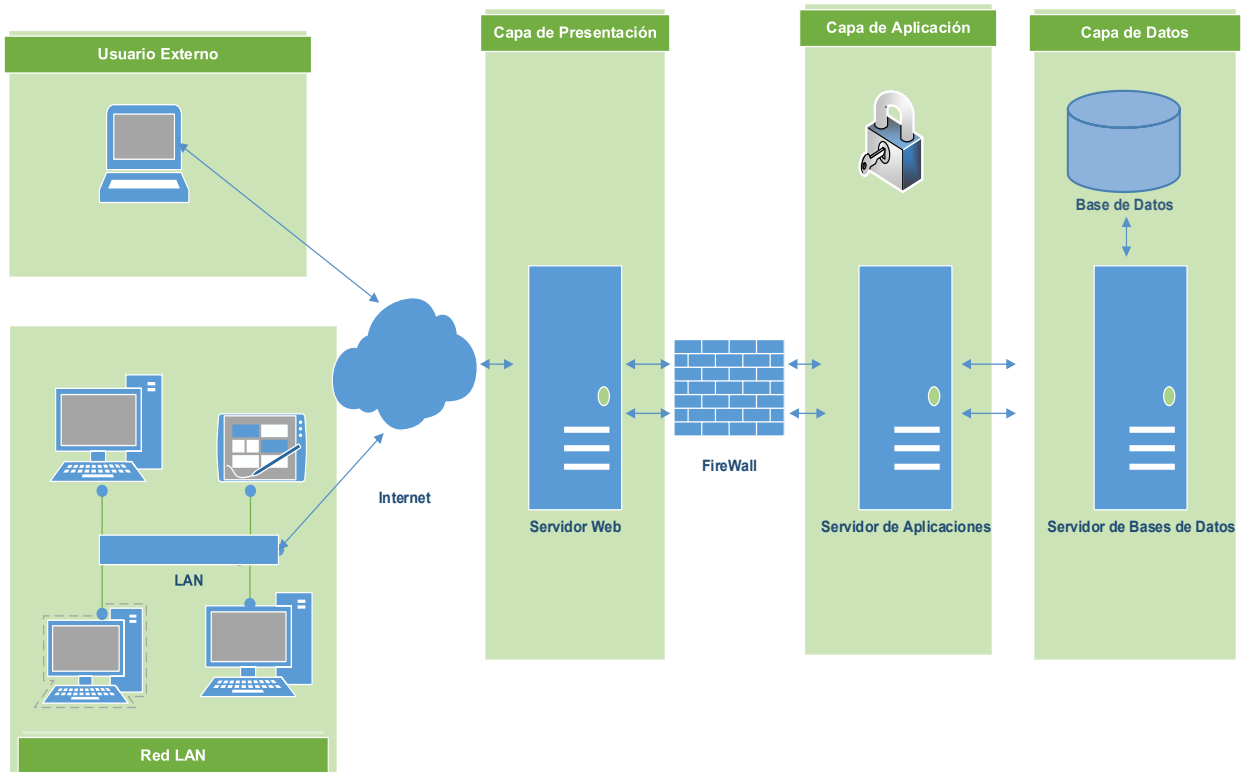


Figura 2.5.2 Modelo de tres capas

CAPÍTULO 3.- PLANTEAMIENTO DEL PROBLEMA

3.1 Análisis del problema

Debido a la creciente demanda en el mercado de empeño de joyas, en la presente tesis se pretende desarrollar un sistema integral el cual esté construido con software libre, que tenga un costo-beneficio favorable para el cliente, con un sistema Front-End amigable para que los empleados se adapten rápidamente a él. Permitiendo de esta manera, puedan interactuar de una forma intuitiva con el sistema y sea a su vez eficiente. Con la implementación de éste sistema se podrá evitar los errores humanos (captura de la información del producto mediante el llenado manual de boletas).

Para comprender mejor el funcionamiento de una casa de empeño, se han identificado cuatro procesos: proceso de valuación de la prenda, proceso de préstamo, proceso para recuperar la prenda y proceso para refrendar la prenda. Estos procesos son descritos a continuación, se debe de hacer notar que estos procesos se identificaron para realizar el análisis del problema, y que en el transcurso del desarrollo del presente tema pueden variar.

Proceso de valuación de la prenda

El funcionamiento de la casa de empeño es el siguiente: la persona (pignorante) interesada en conseguir un préstamo lleva su o sus piezas de valor a la casa de empeño, una vez ahí el pignorante entregará su prenda a un valuador, para que este realice la valoración de los bienes. El valuador determina el monto del préstamo (dicho es determinado en relación del precio de oro y/o plata). Una vez establecido el monto del préstamo por parte del valuador, éste ofrece dicho monto al pignorante, el cual decide si acepta o no la propuesta económica. Vea figura 3.1.1

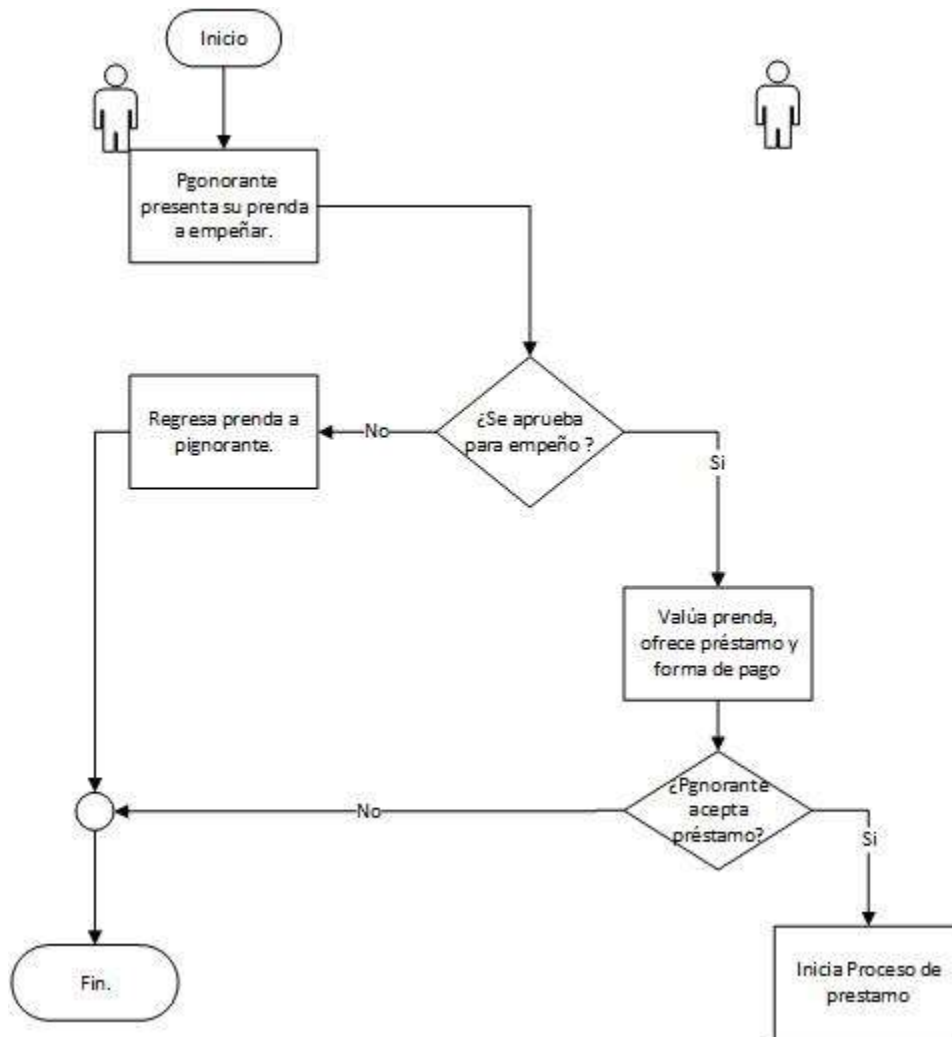


Figura 3.1.1 Proceso de valuación de la prenda

En ésta fase se evitaran los errores humanos ya que el valuador solo elegirá las características de la prenda a empeñar y el sistema determinará el monto del préstamo. Vea figura 3.1.2.

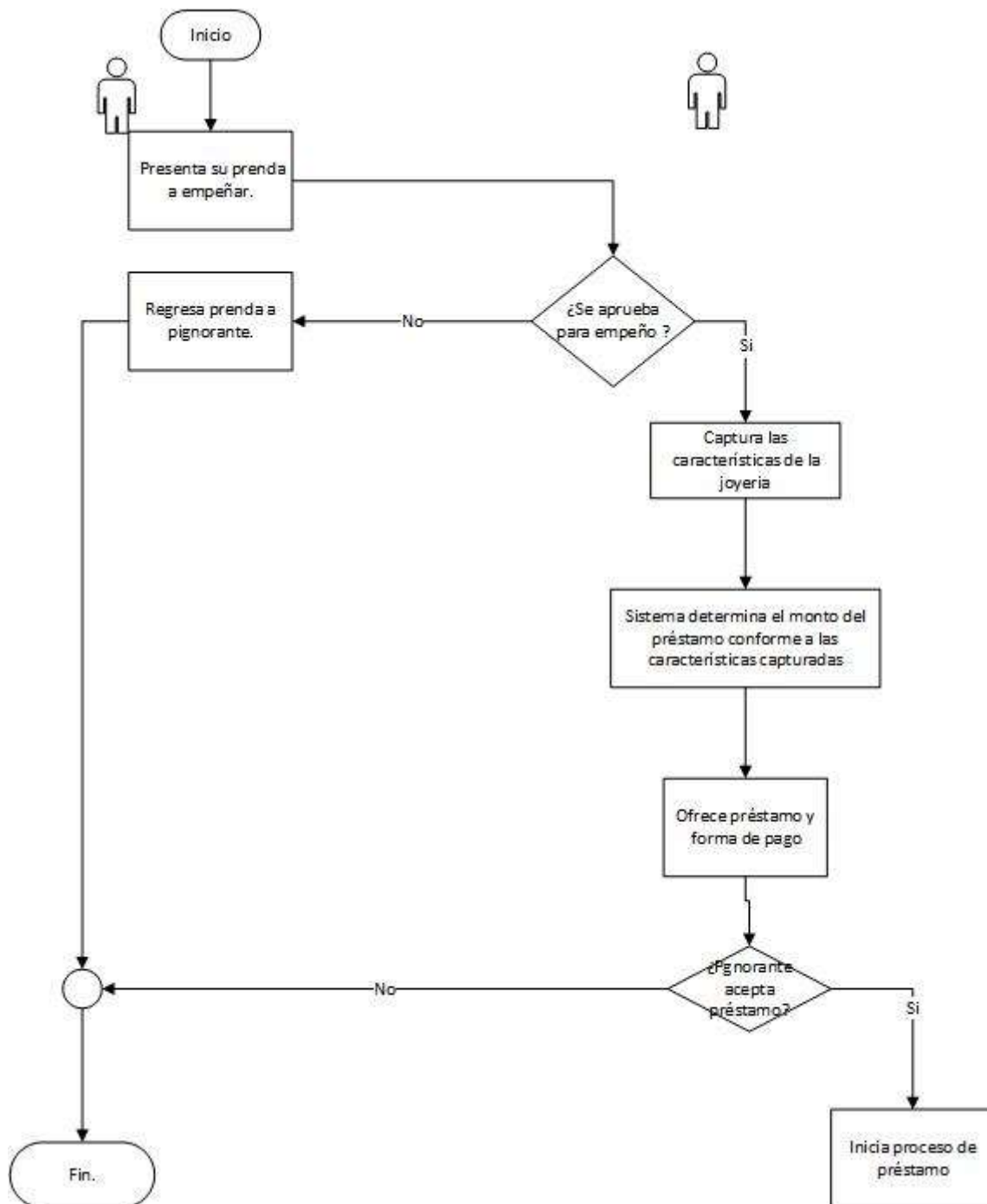


Figura 3.1.2 Proceso de valuación con el sistema implementado

Proceso de préstamo

En caso de aceptar la propuesta, el evaluador realiza el proceso de registro del cliente y de la prenda, para posteriormente explicar y entregar por escrito las cláusulas de la boleta y/o contrato de la prenda, monto del préstamo, tiempo de vigencia para finiquitar la deuda, y los intereses a pagar por parte del pignorante. Vea figura 3.1.3

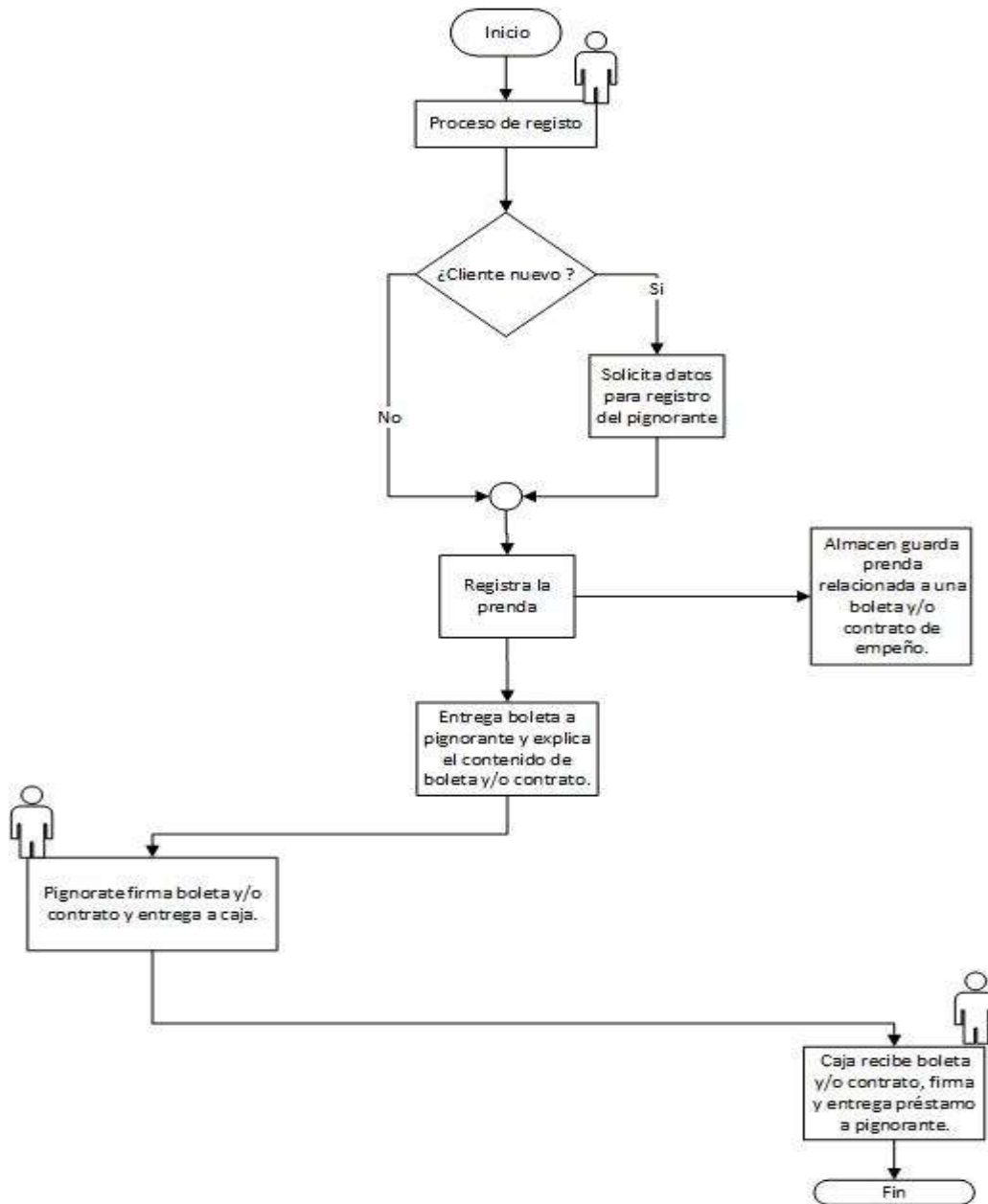


Figura 3.1.3 Proceso de préstamo

En éste proceso se reducirá los errores humanos ya que el contrato y/o boleta serán generados por el sistema evitando errores al ser llenados. Vea figura 3.1.4.

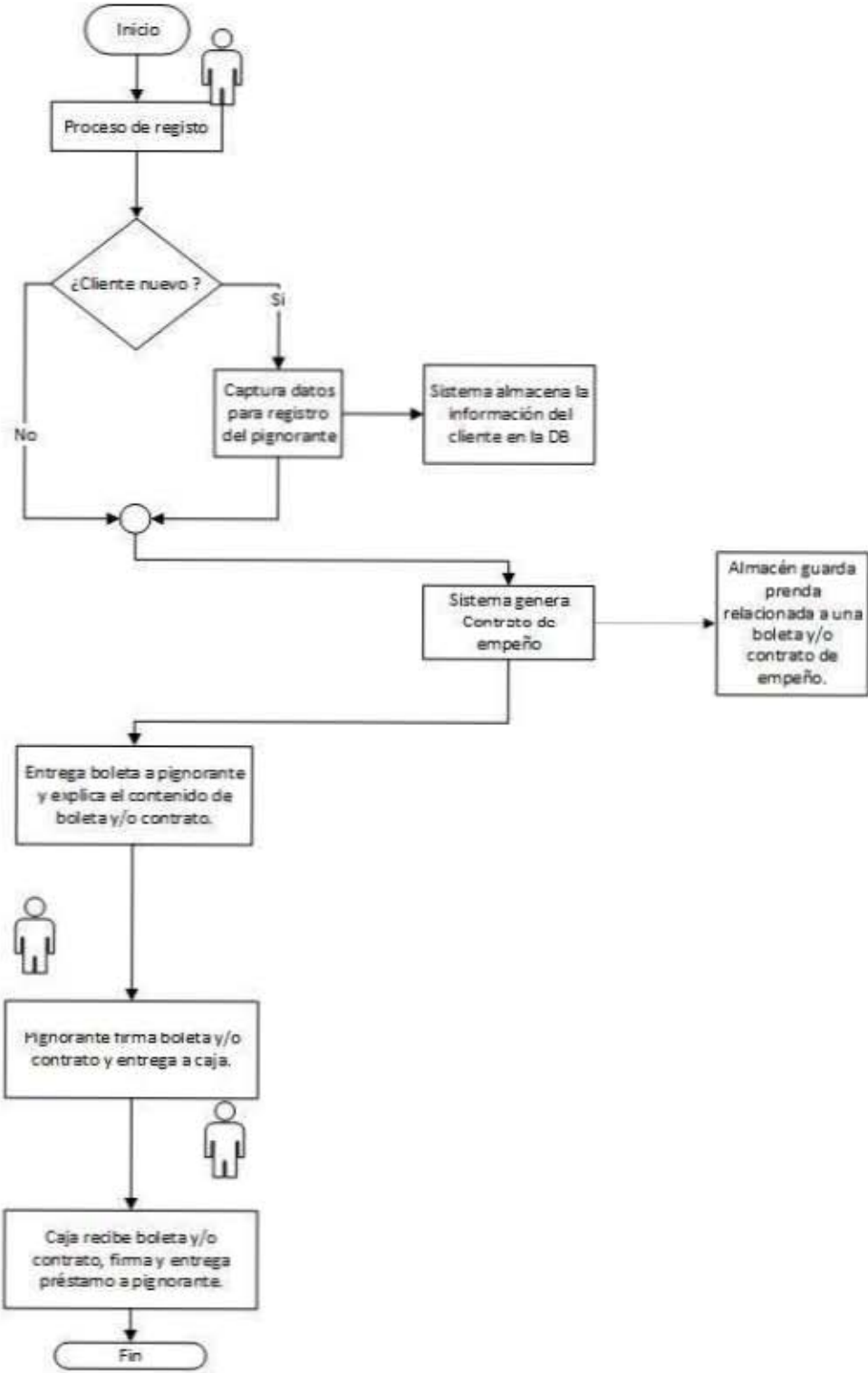


Figura 3.1.4 Proceso de préstamo con el sistema implementado

Proceso para recuperar la prenda

Si el pignorante desea recuperar su prenda, deberá acudir a la casa de empeño, la cual le solicitará su boleta y/o contrato, identificación y el finiquito de la deuda, con lo cual podrá realizar el proceso de desempeño. Una vez validada la información solicitada, la institución deberá entregar la prenda en las mismas condiciones en las cuales fue recibida. Vea figura 3.1.5

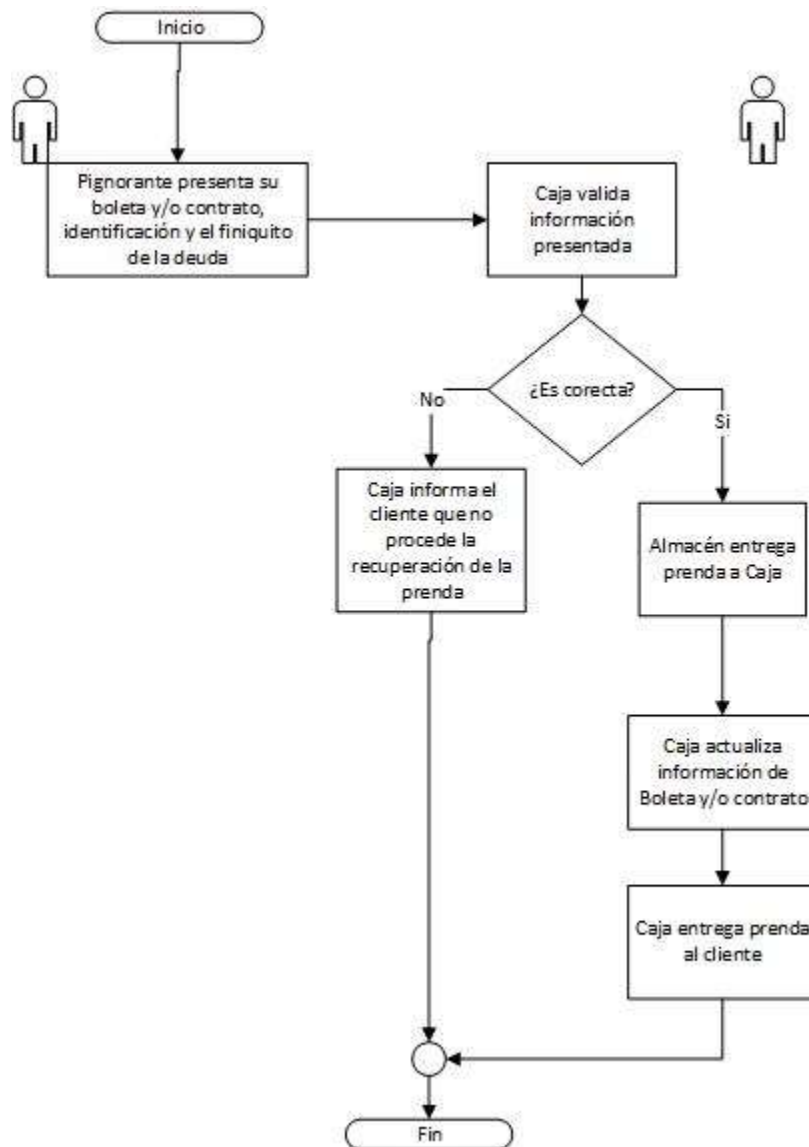


Figura 3.1.5 Proceso para recuperar la prenda

En éste proceso se reducirán los errores humanos ya que el contrato y/o boleta serán validados por el sistema evitando posibles errores. Vea figura 3.1.6.

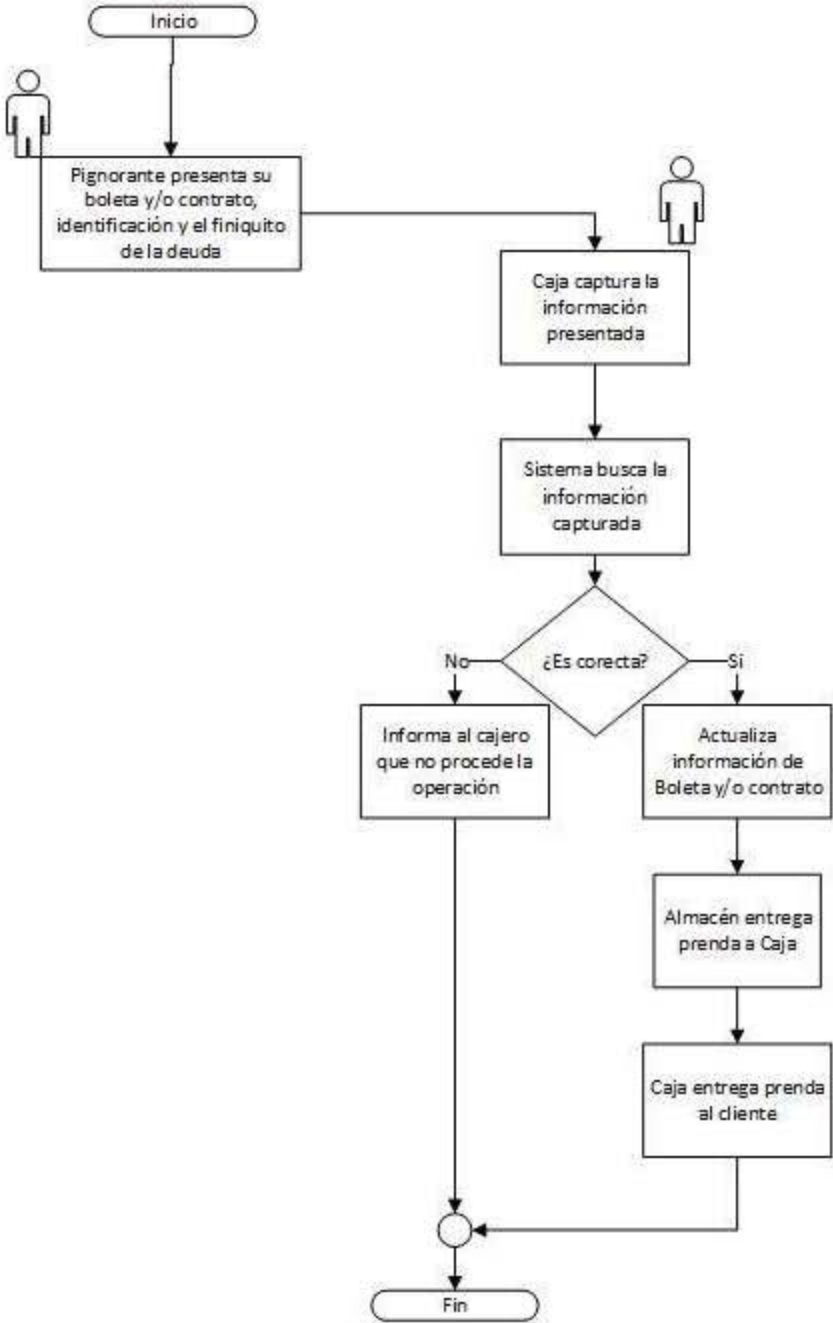


Figura 3.1.6 Proceso para recuperar la prenda con sistema implementado

Proceso para refrendar la prenda

Si el deudor por algún motivo no puede pagar el total de la deuda, éste puede elegir la opción de refrendar su prenda, con el motivo de obtener más tiempo para la recuperación de la misma. Esto implica el tener que pagar un costo adicional al monto de su préstamo, las casas de empeño determinan el número de veces y el costo del referendo, la opción del referendo está indicada en la boleta y/o contrato prendario el cual fue entregado en el proceso de empeño. Vea figura 3.1.7

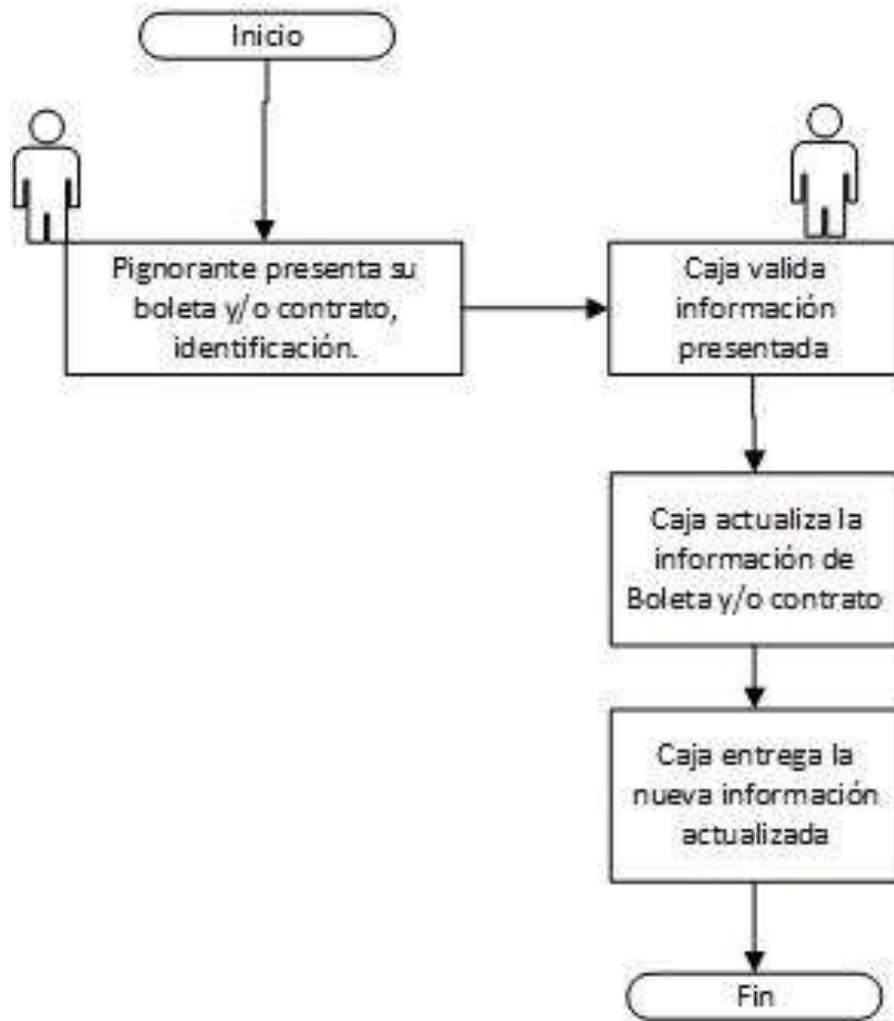


Figura 3.1.7 Proceso para refrendar la prenda

En éste proceso se evitaran los errores humanos ya que el contrato y/o boleta será validados por el sistema. Vea figura 3.1.8.

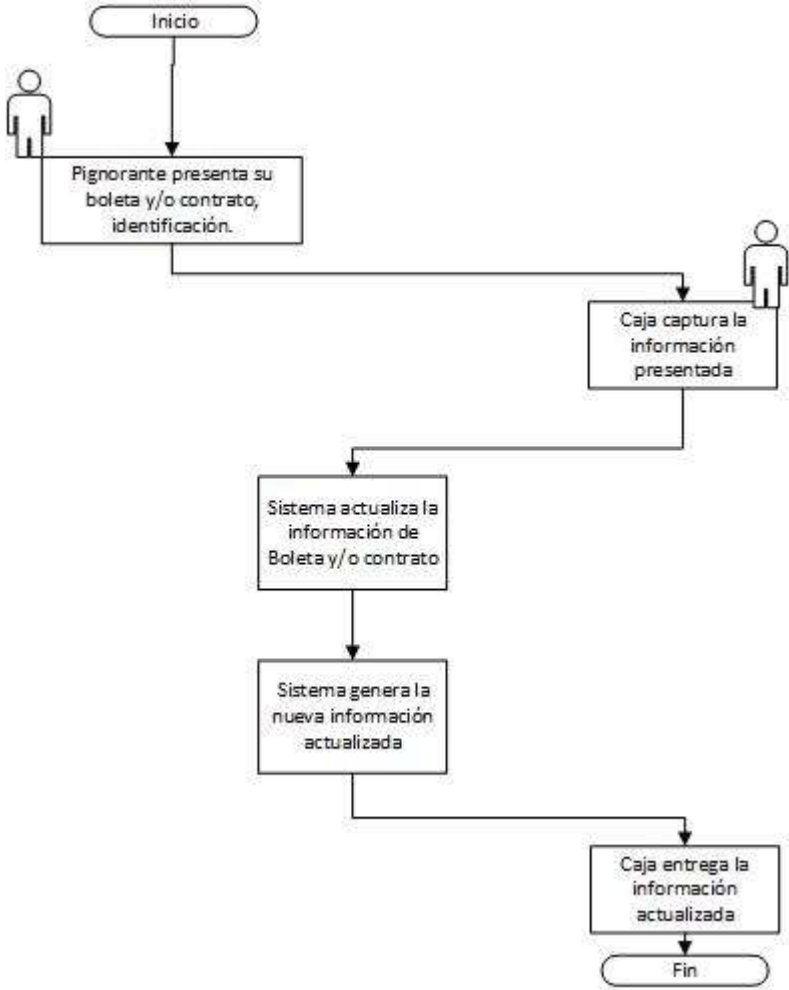


Figura 3.1.8 Proceso para refrendar la prenda con el sistema implementado

3.2 Requerimientos generales y particulares de la aplicación.

Se requiere un sistema, con interfaz amigable y moderna que le permita al usuario administrar casas de empeño y llevar un registro por cada sucursal de los movimientos realizados por los servicios de empeño además de la gestión de usuarios. Los requerimientos únicamente especifican lo que debe tener el sistema

y que debe de poder hacer el usuario, estos no indican como diseñar o desarrollar el sistema para que incluya las características, funciones y contenidos.

El sistema SICE está dividido en las siguientes categorías:

- Requerimientos de negocio. Define los objetivos y los problemas que se intentan desarrollar en el sistema SICE.
- Requerimientos del usuario. Describe las funciones del usuario y como este interactúa con el sistema. En la planeación se deben enfatizar las tareas que el usuario quiere realizar en el sistema mediante perfiles.
- Requerimientos funcionales. Proporcionan los detalles de cómo el sistema se debe comportar y especifica lo necesario para su desarrollo. **Vea apéndice 4.**
- Requerimientos que conciernen a la calidad del servicio. Detallan las características que debe contener el sistema para mantener su integridad y sus restricciones.
- Requerimientos de implementación. Son usados para detallar los cambios en el proceso, migración de un sistema a otro, etc.

El sistema SICE debe contar con características propias de una página web, de una aplicación que debe llevar la gestión de diferentes tipos de usuario con diferentes perfiles. Para cumplir con esto el sistema necesita tener información detallada que debe de ser fácilmente leída a través del uso efectivo de los colores, la tipografía, el espaciado, la disposición espacial de las imágenes, etc.

En pocas palabras no se puede contradecir las convenciones actualmente usadas para el diseño de páginas web.

El mercado de casas de empeño se encuentra ampliamente establecido, por lo que es fácil identificar los errores a los que se enfrenta una casa de empeño y así poder corregirlos y proporcionar una solución más viable en este sistema. Uno de los errores más comunes es el error humano, ya que los usuarios y valuadores tienen una baja credibilidad por parte de los clientes cuando están en desacuerdo con la

cantidad que se les ofrece por sus prendas, ya que pueden haber errores de medición y/o cálculo, o una desactualización de los precios, también sobre la balanza o rapidez de los cálculos. El sistema SICE tendrá la labor de automatizar las tareas y transparentar los cambios al usuario por lo que se asegura la integridad y confiabilidad de las casas de empeño.

Requerimientos generales

El sistema SICE debe cumplir con lo siguiente:

- Minimizar los tiempos de carga en la página.
- Funcionamiento en los navegadores de Internet más populares.
- Reducir el número de menús.
- Interfaz de fácil interacción.
- Incluir solo las imágenes necesarias con la resolución adecuada.
- El texto y la tipografía debe ser simple y agradable para su lectura.
- Se emplearan pocas fuentes y estas serán de uso común.
- El contenido debe ser fácil de entender.
- Eliminar la información que no sea necesaria.
- Evitar el uso de enunciados y párrafos extensos.
- Para el manejo de cifras, utilizar números en lugar de deletrearlas.
- Utilizar lo más posible un lenguaje coloquial.
- Mantener la información organizada.
- Manejo adecuado de listas y el formato de la información.
- Información actualizada.

- Buen contraste entre el texto y el fondo.
- Todas las ligas serán funcionales.
- El uso del color en la página no debe ser motivo de distracción.
- Cada imagen tiene un claro propósito.
- La información será útil y significativa.
- Información organizada de forma consistente.
- Información fácil de encontrar.
- Libre de errores tipográficos y gramaticales.
- Manejo confiable de la información personal.
- Fácilmente administrable.

Requerimientos particulares

Se deberá garantizar que posee lo siguiente:

- Requerimientos del sistema SICE.
 - Contenido propio del sitio.
 - Logo que identifique al sistema de casas de empeño SICE.
 - Estará en línea las 24 horas del día los 365 días del año.
 - Pie de página que incluya el nombre del sistema.
 - Contará con una página de atención sobre el sistema.
 - Se deberá mostrar por separado la función de empeño y la función de pagos.
 - Deberá permitir elegir algún tipo de producto preestablecido.

- Al valuar la prenda se deberá mostrar la cantidad de capital que se ofrecerá en cada producto en moneda nacional (pesos).
- Se mostrará la amortización de pagos.
- Se especificará el tipo de prenda a empeñar.
- Se informará al cliente sobre los vencimientos de las fechas de pago por correo.
- Una vez elegido el producto se podrá elegir el número de semanas a pagar.
- Deberá mostrar en colores las fechas de pago, color verde cuando está vigente y color roja cuando esté vencida.
- Deberá contar con catálogos para añadir y hacer crecer la base de datos.
- Deberá contar con reportes por sucursal y concentradora.
- Se podrá empeñar más de una prenda a la vez.
- Registrar a los clientes que empeñarán sus prendas.
- Realizar búsquedas en los catálogos.
- Actualizar los precios de onza y dólar.
- Poder crear nuevos productos.
- Imprimir el contrato del préstamo con sus fechas de pago.
- No permitir realizar empeños si la tienda no cuenta con capital disponible.
- Deberá aplicar intereses cuando no se cumpla con los pagos establecidos.
- Deberá de permitir refrendar una prenda.
- Permitir escribir comentarios respecto a la prenda.
- Permitir hacer dotaciones a sucursales.

- Permitir la gestión de los gastos de oficina.
- Realizar las respectivas operaciones tomando las características de cada producto. **Vea apéndice 3.**
- Realizar los cálculos claros y precisos. **Vea apéndice 4.**
- Requerimientos para usuarios del sistema SICE.
 - Deberá de contar con un menú fijo en el que los usuarios se puedan desplazar a cualquier sitio del sistema en cualquier momento.
 - El acceso al sistema será por medio de un nombre de usuario y contraseña para cada usuario, mismo que serán asignado por el administrador del sistema.
 - Al cerrar la sesión no se podrá consultar ningún menú ni realizar operaciones.
 - El administrador del sistema será el único que podrá tener acceso directo a la base de datos y agregar artículos a los catálogos.
 - Deberá poder asignar usuarios a sucursales.
 - Se contará con un registro de movimientos de usuario.
 - Se deberá poder gestionar usuarios y asignar perfiles.
 - Se deberá restringir el acceso a los usuarios de acuerdo a su perfil.
 - Deberá poder existir un cotitular en cada contrato.

3.3 Análisis y posicionamiento de los lenguajes para el Back-End y Front-End

Análisis y posicionamiento para Back-End

Comparativo entre el soporte de SGBD y Sistema Operativo (SO). Vea Tabla 3.3.1.

SGBD / SO	Windows	OS X	Linux	BSD	UNIX	z/OS	iOS	Android	OpenVMS
MySQL	SI	SI	SI	SI	SI	SI	Desc.	SI	No
PostgreSQL	SI	SI	SI	SI	SI	SI	No	SI	No
Informix Dynamic Server	SI	SI	SI	SI	SI	No	No	No	No
DB2	SI	SI	SI	No	SI	SI	SI	No	No
Oracle	SI	SI	SI	No	SI	SI	No	No	SI
Microsoft Access (JET)	SI	No	No	No	No	No	No	No	No
Microsoft Visual Foxpro	SI	No	No	No	No	No	No	No	No
Microsoft SQL Server	SI	No	No	No	No	No	No	No	No
Oracle Rdb	No	No	No	No	No	No	No	No	SI
Teradata	Yes	No	Yes	No	Yes	No	No	No	No

Tabla 3.3.1 Soporte de sistema operativo

Características fundamentales

Información acerca de características fundamentales comparado con varios SGBD son implementados nativamente. Vea tabla 3.3.2.

SGBD / Características Fundamentales	ACID	Integridad Referencial	Transacciones	Unicode	Interfaz
MySQL	Si	Partial	Si	Si	GUI & SQL
PostgreSQL	Si	Si	Si	Si	API & GUI & SQL
Informix Dynamic Server	Si	Si	Si	Si	SQL & JSON
DB2	Si	Si	Si	Si	GUI & SQL
Oracle	Si	Si	Si	Si	API & GUI & SQL
Microsoft Access (JET)	Si	Si	Si	Si	GUI & SQL
Microsoft Visual FoxPro	No	Si	Si	No	GUI & SQL
Microsoft SQL Server	Si	Si	Si	Si	GUI & SQL
Oracle Rdb	Si	Si	Si	Si	SQL
Teradata	Si	Si	Si	Si	SQL

Tabla 3.3.2 Características fundamentales

Tablas temporales y vistas materializadas

Información acerca de que tablas y vistas (unas más básicos que otras) son soportadas nativamente. Vea tabla 3.3.3.

SGBD/ Tablas,Vistas	TablasTemporales	Vistas Materializadas
MySQL	Si	No
PostgreSQL	Si	Si
Informix Dynamic Server	Si	No
DB2	Si	Si
Oracle	Si	Si
Microsoft Access (JET)	No	No
Microsoft Visual Foxpro	Si	Si
Microsoft SQL Server	Si	Si
Oracle Rdb	Si	Si
Teradata	Si	Si

Tabla 3.3.3 Información sobre tablas y vistas

Operadores permitidos

Los operadores involucran a aquellas operaciones que son aplicables en el SGBD hacia los datos, los usuarios deberán basar estas operaciones a través de las relaciones establecidas en el modelo Entidad - Relación. Vea tabla 3.3.4.

SGBD/ Operaciones	Union	Intersect	Left Join	Inner joins	Outer joins	Inner selects	Merge joins	Parallel Query
MySQL	Si	No	No	Si	Si	Si	No	No
PostgreSQL	Si	Si	Si	Si	Si	Si	Si	Si
Informix Dynamic Server	Si	Si	Si	Si	Si	Si	Si	Si
DB2	Si	Si	Si	Si	Si	Si	Si	Si
Oracle	Si	Si	Si,	Si	Si	Si	Si	Si
Microsoft Access (JET)	Si	No	No	Si	Si	Si	No	--
Microsoft Visual Foxpro	Si	--	--	Si	Si	Si	--	--
Microsoft SQL Server	Si	Si	Si	Si	Si	Si	Si	Si
Oracle Rdb	Si	Si	Si	Si	Si	Si	Si	--
Teradata	Si	Si	Si	Si	Si	Si	Si	Si

Tabla 3.3.4 Información sobre operaciones permitidas por el SGBD

Front-End

Comparativa entre PHP, Java y C#

C#

Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. Y sus características se marcan a continuación:

- Sencillez: Elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en NET.
- Modernidad: Incorpora en el propio lenguaje elementos que a lo largo de los años ha demostrado que son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular.
- Orientación a componentes: La sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).
- Instrucciones seguras: Para evitar errores muy comunes, se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes.
- Eficiente: En principio, todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros.
- Compatible: Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir

directamente en código escrito en C# fragmentos de código escrito en estos lenguajes.

Java

La tecnología Java se usa para desarrollar aplicaciones para un amplio alcance de entornos, desde dispositivos del consumidor hasta sistemas empresariales heterogéneos. En esta sección, se presenta una vista de alto nivel de la plataforma Java y sus componentes.

- API de Java. Como cualquier lenguaje de programación, el lenguaje Java tiene su propia estructura, reglas de sintaxis y paradigma de programación. El paradigma de programación del lenguaje Java se basa en el concepto de programación orientada a objetos, que las funciones del lenguaje soportan.
- Estructuralmente, el lenguaje Java comienza con paquetes. Un paquete es el mecanismo de espacio de nombres del lenguaje Java. Dentro de los paquetes se encuentran las clases y dentro de las clases se encuentran métodos, variables, constantes, entre otros.
- El compilador Java. Cuando se programa, se escribe el código de origen en archivos *.java y luego los compila. El compilador verifica su código con las reglas de sintaxis del lenguaje, luego escribe los códigos byte en archivos *.class.
- La **JVM** (Java Virtual Machine). Al momento de la ejecución, la JVM lee e interpreta archivos *.class y ejecuta las instrucciones del programa en la plataforma de hardware nativo para la que se escribió la JVM.
- El recolector de basura. En lugar de forzarlo a mantenerse a la par con la asignación de memoria (o usar una biblioteca de terceros para hacer esto), la plataforma Java proporciona una gestión de memoria lista para usar.

- El JRE (Java Runtime Environment) incluye las bibliotecas de códigos de la JVM y los componentes que son necesarios para programas en ejecución escritos en el lenguaje Java.

PHP (Hypertext Pre-processor)

Lenguaje de programación de uso general, de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación 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 una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

- Lenguaje totalmente libre y abierto.
- La curva de aprendizaje es muy baja.
- Los entornos de desarrollo son de rápida y fácil configuración.
- Fácil despliegue: paquetes totalmente autoinstalables que integran PHP.
- Fácil acceso a bases de datos.

Comparativo entre los tres lenguajes de Front-End. Vea tabla 3.3.7.1, tabla 3.3.7.2 y tabla 3.3.7.3.

LENGUAJE DE PROGRAMACIÓN	VENTAJA	DESVENTAJA	IDE DE DESARROLLO	SISTEMA OPERATIVO
PHP	<ul style="list-style-type: none"> • Está diseñado para la programación web. • La curva de aprendizaje es muy rápida. • Soporta en cierta medida la orientación a objetos. • Lenguaje multiplataforma. • Conexión con la mayoría de los SGBD. • Capacidad de expansión utilizando módulos. • Es de libre distribución. • Incluye gran cantidad de funciones. • No requiere definición de tipos de variables. • Existe una comunidad muy grande. • Los entornos de desarrollo son de rápida y fácil configuración. 	<ul style="list-style-type: none"> • El código fuente no se puede ocultar debido a que se interpreta en la ejecución. • La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP. • Deficiencias en la POO para aplicaciones muy grandes. • Todo el trabajo lo realiza el servidor y no delega al cliente. 	<ul style="list-style-type: none"> • Eclipse • DreamWeaver • Netbeans • Aptana • PHPEd • PHPDesigner 	<ul style="list-style-type: none"> • Unix • Windows • Linux • Mac OS X • RISC OS

Tabla 3.3.7.1 Comparativa entre PHP, JAVA y C#

LENGUAJE DE PROGRAMACIÓN	VENTAJA	DESVENTAJA	IDE DE DESARROLLO	SISTEMA OPERATIVO
<p>JAVA</p> <ul style="list-style-type: none"> • Es un lenguaje de propósito general. • Es independiente de la plataforma de desarrollo. • El JDK (Java Development Kit) es libre de licencias. • Amplia flexibilidad de la POO. • Acceso a la base a través de JDBC (Java Database Connectivity), la administración de la BD es simple. • El manejo de la memoria se hace automáticamente y utilizando el garbage collector, permite ahorro en tiempo de ejecución. 	<ul style="list-style-type: none"> • Algunas herramientas tienen un costo adicional. • Puede ser que no exista un JDBC para bases de datos poco comerciales. • Ya que los programas de Java son interpretados, nunca alcanzan la velocidad de un verdadero ejecutable. • La curva de aprendizaje no es medianamente corta. • Requiere un intérprete. • Algunas implementaciones y librerías pueden tener código rebuscado. • Una mala implementación de un 	<ul style="list-style-type: none"> • Netbeans • Eclipse • JCreator • JBuilder 	<ul style="list-style-type: none"> • Unix • Windows • Linux • Mac OS X • RISC OS 	

Tabla 3.3.7.2 Comparativa PHP, JAVA y C#

LENGUAJE DE PROGRAMACIÓN	VENTAJA	DESVENTAJA	IDE DE DESARROLLO	SISTEMA OPERATIVO
C #	<ul style="list-style-type: none"> • Es un lenguaje de propósito general. • Recolección de basura automática. • Soporta bastantes más tipos primitivos (value types), incluyendo tipos numéricos sin signo. 	<ul style="list-style-type: none"> • Es mono plataforma. • No es de libre licenciamiento. • Una mala implementación de un programa en java, puede resultar en un bajo performance de la aplicación. 	<ul style="list-style-type: none"> • Visual Studio .NET 	<ul style="list-style-type: none"> • Windows

Tabla 3.3.7.3 Comparativa entre PHP, JAVA y C#

Conclusiones para haber elegido PHP y MySQL

- Simple Instalación y Configuración

Es fácil desplegar un proyecto en PHP porque dispones de paquetes totalmente autoinstalables que integran PHP: Apache y MySQL, tanto para UNIX (LAMPP), como para Windows (XAMPP). Ambos fáciles de instalar y configurar. Se obtiene así en la PC un servidor local con acceso a base de datos, listo para que se pueda desarrollar fácilmente cualquier proyecto que se proponga, bajo impacto en tiempo por problemas de compatibilidad.

- Simplicidad de administración, manipulación y acceso a las Bases de Datos.
- Los desarrolladores se familiarizan con el lenguaje en un periodo corto ya que su sintaxis es simple; adicionalmente cumple estándares básicos de la programación orientada a objetos y de SQL.
- Lenguaje de código abierto y sin licenciamiento.
- Buena velocidad de ejecución.
- Maneja un buen control de accesos para usuario.

3.4 Aplicación y separación de los módulos según la metodología Evolutiva-Incremental

Dentro del proyecto se ejecutarán diversas actividades particulares con el fin de que se desarrolle en un proceso Evolutivo-Incremental. Para ello es necesario implementar estrategias adecuadas considerando para cada uno de los módulos que integran el proyecto. Vea figura 3.4.1.

Los módulos a desarrollar a alto nivel son:

Módulo de Back-End y Front-End

El módulo Back-End integrará la programación y lógica funcional del aplicativo a desarrollar, por medio de una base de datos relacional.

El módulo Front-End integra el entorno visual del aplicativo, el cual estará desarrollado en un ambiente web. Deberá de ser agradable y de fácil manipulación del entorno.



Figura 3.4.1 Etapas del modelo evolutivo implementadas en los módulos

Módulo de pruebas

El módulo de pruebas se desarrollará durante las etapas del Modelo Evolutivo-Incremental del proyecto.

Estas comprenden:

- Pruebas de Caja Negra
- Pruebas de Regresión
- Pruebas de Aceptación
- Pruebas Alfa
- Pruebas Beta

- Pruebas de Validación
- Pruebas de Estrés

Los módulos contemplados en el modelo Evolutivo-Incremental para el desarrollo del sistema SICE son los siguientes:

Módulo de inicio de sesión

Este módulo comprende el ingreso de los usuarios al sistema.

El usuario ingresará al sistema a través de un ID de usuario y una contraseña asignada. El ID de usuario deberá de estar registrado en la base de datos del sistema con un rol de permisos asignado.

Módulo de fin de sesión

El usuario podrá cerrar su sesión a través de un objeto mostrado en el Front-End. Como parte de la seguridad del sistema, ya que una vez que el usuario cierra la sesión deberá de ingresar su ID de usuario y contraseña.

Módulo de administración de usuarios

La administración de los usuarios deberá permitir el alta, baja, eliminación, y actualización de usuarios.

Adicionalmente se especificará el tipo de permisos por rol para el acceso a la información del sistema.

Módulo de valuación de prendas

El módulo de Valuación deberá de apoyar al usuario al manejo de las características de la prenda que el cliente lleva para empeñar y con ello generar una propuesta de servicio.

El valuador deberá de ingresar algunas características de la prenda, para que el sistema busque información relacionada y con ello estimar el valor del préstamo que se le puede ofrecer al cliente.

Módulo de alta de cliente

Este módulo se debe de ejecutar únicamente cuando ya se haya hecho una oferta de servicio al cliente y este haya aceptado. Se registrará la información personal del Cliente.

Módulo de oferta de servicio

La oferta de servicio se deberá de generar para establecer los términos de pago del préstamo ofrecido al cliente, tales como:

- Monto del préstamo
- Fechas de pago
- Características del Préstamo

Módulo de control de pagos

El módulo de control de pagos establece los pagos realizados por parte del cliente para cubrir el préstamo, así mismo actualizar el monto de la deuda.

Módulo de mantenimiento

El módulo de Mantenimiento comprende los posibles cambios que resulten de las pruebas realizadas al sistema así como de requerimientos adicionales solicitados por el cliente.

A continuación se muestran los módulos a implementar sobre el Modelo Evolutivo–Incremental así como las etapas que integran el sistema SICE. Vea figura 3.4.2.

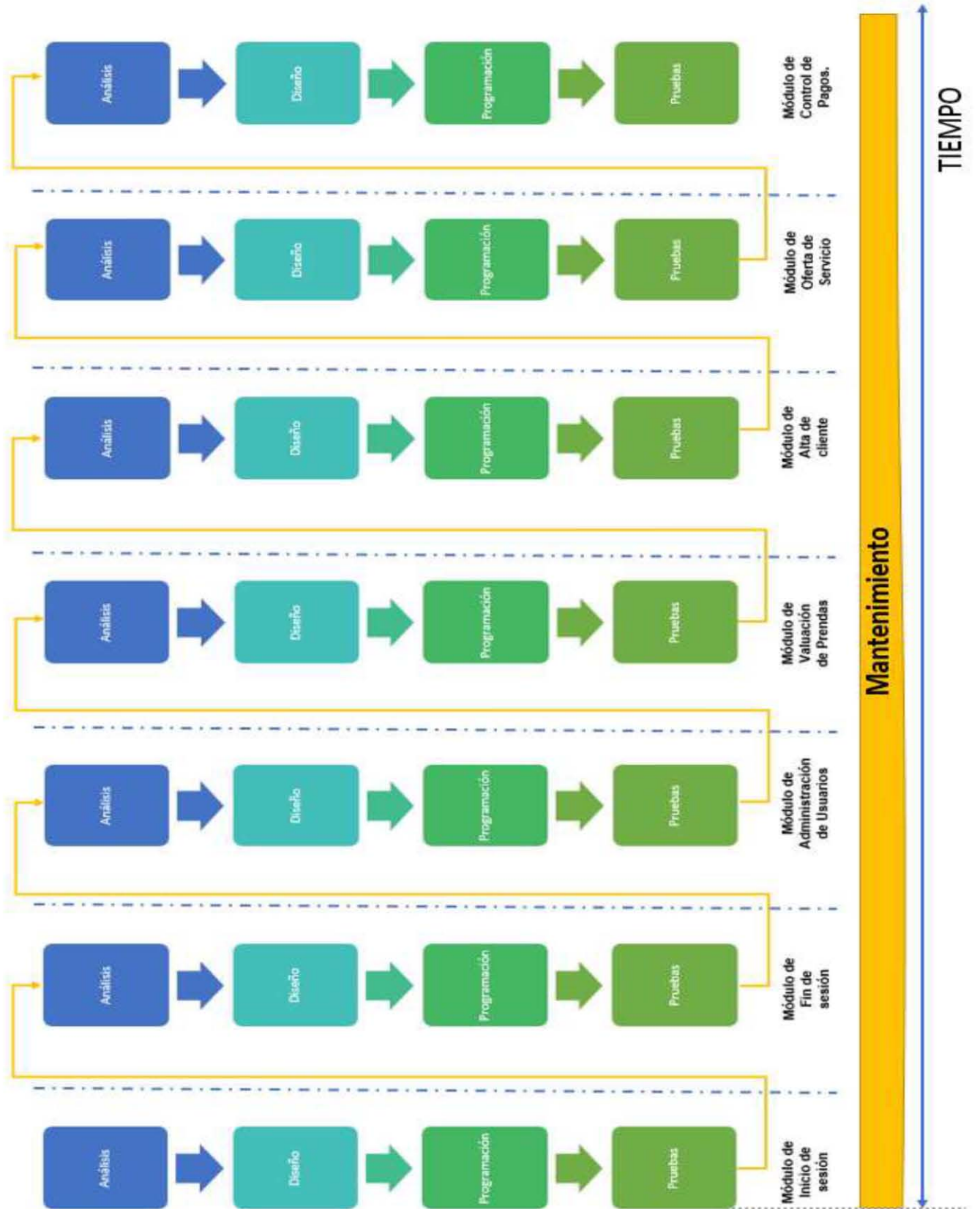


Figura 3.4.2 Modelo Evolutivo – incremental por módulo

3.5 Diseño a nivel de bloques de la interfaz del usuario

La página principal del Sistema Integral para Casas de Empeños (SICE) estará conformada por los siguientes bloques, los cuales se muestran a continuación. Vea figura 3.5.1.

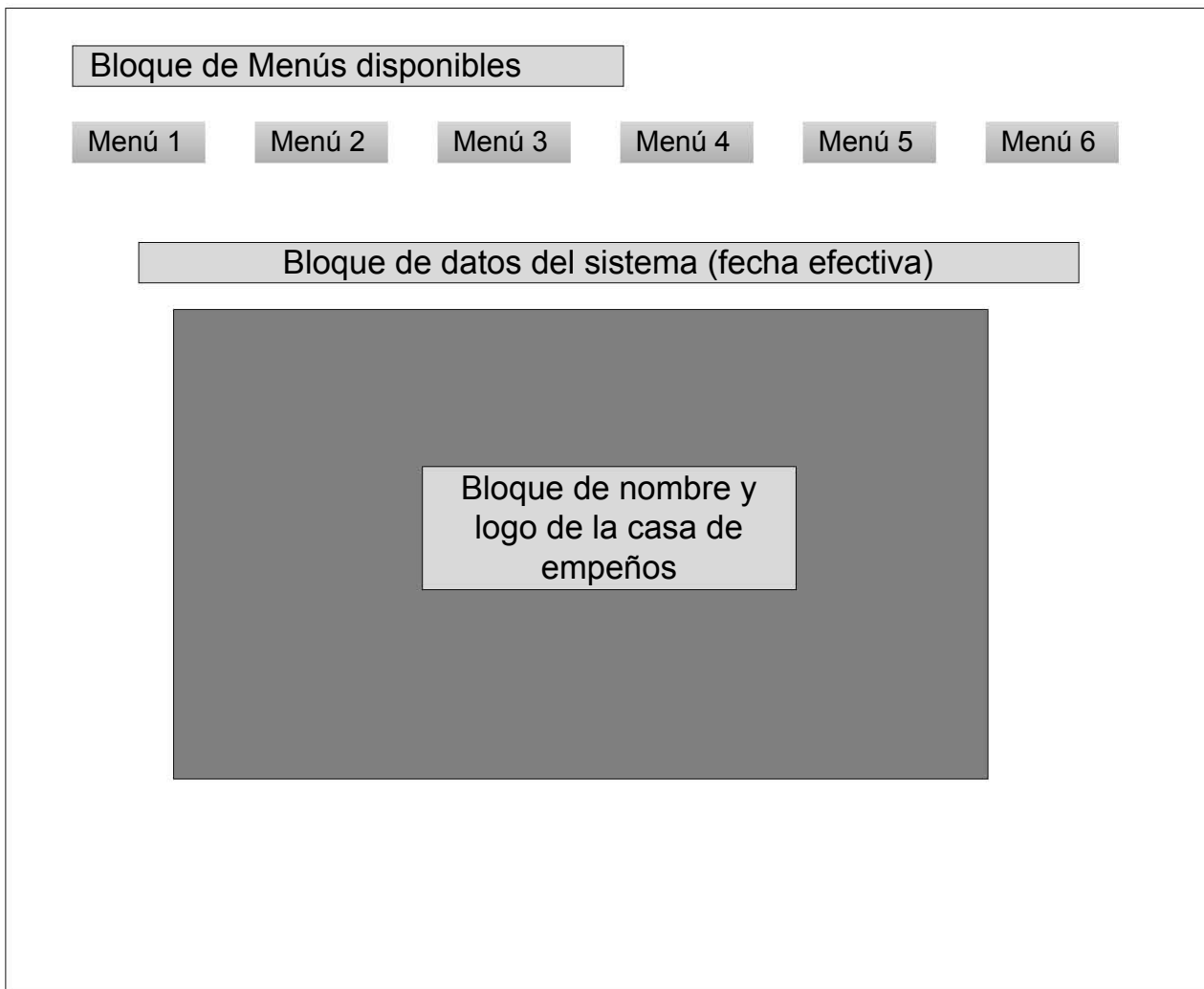


Figura 3.5.1 Página principal

Bloque de menús disponibles

Conjunto de opciones o posibilidades que se le presentarán al encargado de la casa de empeños, para que pueda elegir entre las distintas opciones y ejecutar una o más funciones, el menú tendrá el propósito de facilitar y agilizar la interacción del encargado con el sistema, el operar con un menú es práctico y la variedad de opciones se representan a través del nombre, permite dar un panorama de que se puede hacer por el sistema como es la administración de la sucursal, alta o modificación de perfiles de usuarios, efectuar pagos, empeño de prendas, gestionar los gastos propios de la sucursal.

Bloque de nombre y logo de la casa de empeño

Describe un elemento muy importante, como es el logo de la marca de la casa de empeños, el nombre de la misma, el cual es utilizado por empresas y organizaciones, para que sean reconocidas fácilmente por sus pignorantes y se familiaricen.

Bloque de fecha de sistema

Este bloque será utilizado para indicar la fecha y hora del sistema, el cual será tomado del servidor.

A continuación se muestra los bloques en los cuales se realizará la recepción y valorización de la prenda que se desea empeñar por parte del pignorante. Vea figura 3.5.2.

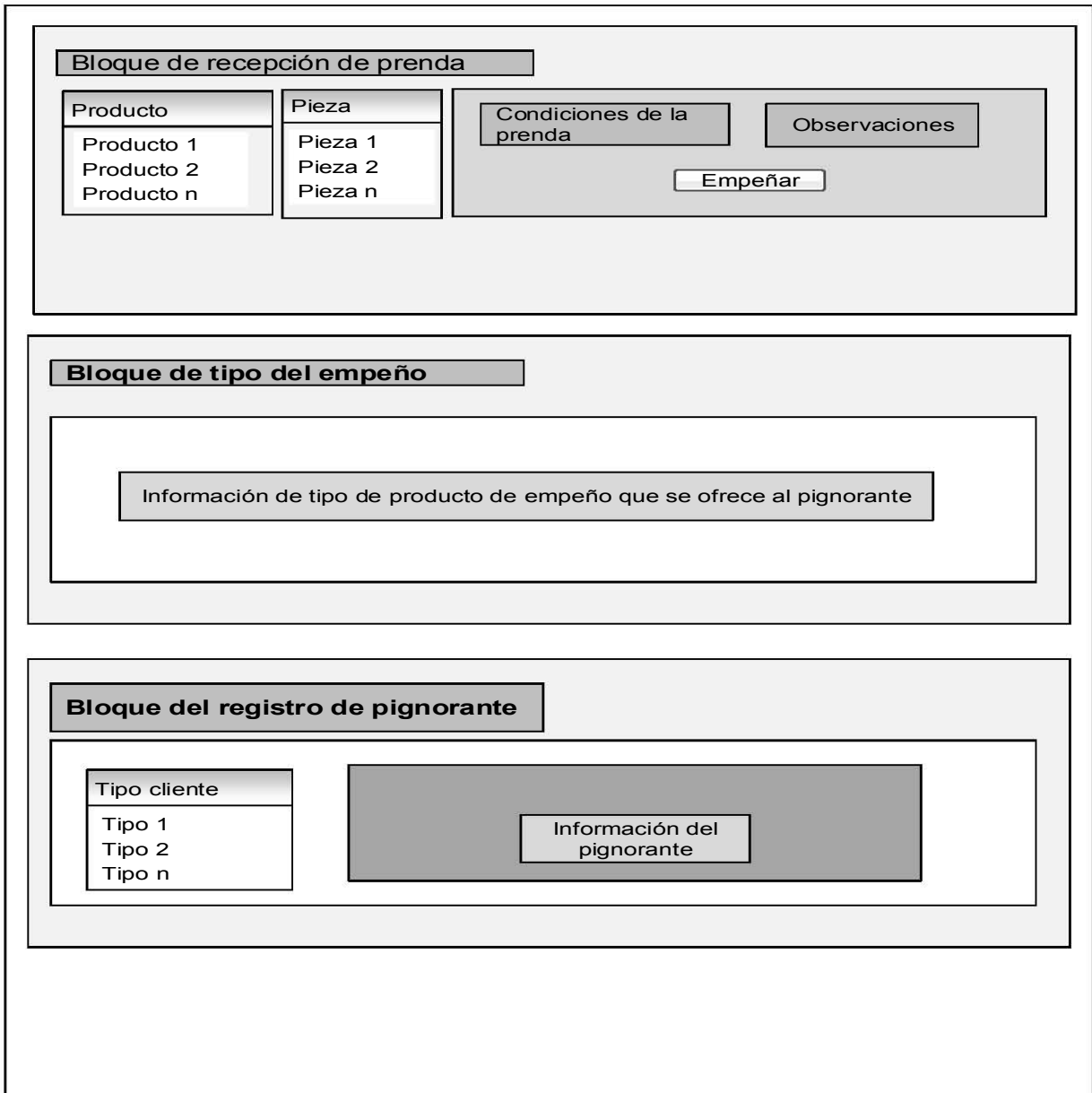


Figura 3.5.2 Recepción de prenda

Bloque de recepción de prenda

Describe la recepción de la prenda; en la cual el valuador relazará el análisis sobre el valor, la clasificación, y una descripción general de la misma; adicionalmente se podrá realizar la recepción de una o varias prendas con el mismo cliente y la misma solicitud si es necesario.

En este bloque existirá un botón llamado “Empeñar”, el cual tendrá la función de continuar el flujo hacia el bloque Tipo de Empeño que se describe a continuación.

Bloque de tipo de empeño

Este bloque permitirá ofrecer al pignorante los diversos productos disponibles en la casa de empeños, a partir del avalúo de la prenda; como son:

- Empeño Express
- Empeño Tradicional
- Empeño Semanal

Así mismo en esta sección se indicará al cliente el monto de préstamo que se le puede otorgar. Este dependerá del producto elegido por el pignorante y las características de la prenda.

Una vez que el pignorante indique al valuador qué producto es de su elección, se activará el bloque se describe a continuación.

Bloque de registro del pignorante

En este bloque se realizará el registro del cliente. En el cual el valuador podrá seleccionar si es Cliente Nuevo o un Cliente Registrado previamente en el sistema.

Cuando la opción seleccionada es: Cliente Nuevo, se solicitará información personal como:

- Nombre, dirección y teléfono.

A continuación se generará un recibo/boleta en la cual estará registrada la descripción de la prenda, el producto de empeño seleccionado, el monto del préstamo, el plan de pagos y la información del cliente.

Cuando se seleccione la opción de Cliente Registrado, el SICE validará que el cliente exista, en caso de éxito se generará el número de recibo/boleta con los datos ya mencionados en el párrafo anterior, una vez terminado este proceso se imprimirá la boleta/recibo con las condiciones del empeño. En caso de que no exista información del cliente, se deberá de registrar.

CAPÍTULO 4.- DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

4.1 Modelo esencial ambiental del Sistema Integral de Casas de Empeños

El modelo esencial deberá describir el contenido de los flujos o almacenes de datos, sin describir el medio (por ejemplo, USB o disco duro) u organización física de los datos.

El modelo esencial a su vez tiene un componente principal, también llamado Modelo ambiental.

Modelo ambiental

El modelo ambiental define las interfaces entre el sistema y la interacción con los agentes externos a este. Por lo tanto, se necesita saber qué información ingresará al sistema desde el ambiente exterior, y qué información generará como salida al ambiente externo.

Otro aspecto crítico del modelo ambiental consiste en identificar los acontecimientos que ocurren en el entorno y que deberá responder el sistema. Sólo importarán aquellos que ocurren en el ambiente exterior y los que requieren una respuesta del sistema.

Diseño de sistemas

Un aspecto fundamental del diseño de sistemas es el trabajo del analista y del diseñador, en donde el analista deberá asegurarse de entender los requerimientos del usuario, mientras que el diseñador se asegurará que dichos requerimientos sean implementados con la tecnología actual.

La actividad del diseño involucra el desarrollo de una serie de modelos. En los cuales los más importantes para el diseñador son el modelo de implementación de

sistemas y el modelo de implementación de programas, en este último se divide en un modelo de procesador, y uno de tareas.

En el nivel del modelo del procesador, el diseñador tratará principalmente de decidir cómo asignar el modelo esencial a los distintos procesadores (CPU) y cómo deberán comunicarse entre sí. Existen las siguientes opciones:

- El modelo esencial completo se le puede asignar a un solo procesador (solución de computadora principal).
- Cada burbuja del modelo esencial se puede asignar a un procesador distinto (solución distribuida).
- Se puede escoger una combinación de servidores o PC's para minimizar costos, maximizar confiabilidad o lograr algún otro objetivo.

4.1.1 Diagrama de flujo de datos

Un diagrama de flujo de datos (DFD) es una representación gráfica de un sistema de información. Los DFD son una de las perspectivas esenciales de análisis de sistemas. Los usuarios podrán visualizar la forma en que el sistema funcionará.

Modelo de datos se dividen en niveles, los cuales son:

- **Nivel 1:** Diagrama general del sistema.
- **Nivel 2:** Diagrama de detalle de procesos involucrados.
- **Nivel 3:** Diagrama de detalle o expansión.

Nivel 1

El diagrama de contexto muestra a través de flujos de datos las interacciones existentes entre los agentes externos y el sistema, sin describir en ningún momento la estructura del sistema de información.

Burbuja: Representa el sistema, y se representa con un círculo y dentro de él se pone el nombre del sistema.

Agente externo: Representa una persona, organización o sistema que interactúa con el sistema que se está modelando. Se representa por medio de un rectángulo y dentro de él se pone el nombre del agente externo o también conocido como entidad.

Almacén: Representa un conjunto de datos del mismo tipo. Se representa por medio de dos líneas paralelas donde se pone entre ellas el nombre que identifica el conjunto de datos. El nombre debe ir en mayúsculas y el almacén sólo se puede poner si es mantenido por un agente de externo

Flujo de datos: Representa la salida y entrada de datos que se direcciona al sistema o proviene de este, se representa con una línea curva que finaliza con una flecha, la cual indica la dirección. Los únicos flujos que pueden existir son: agente externo y burbuja, burbuja y almacén. Vea figura 4.1.1



Figura 4.1.1 Diagrama de contexto del sistema

Adicionalmente, es posible descomponer, a su vez, en subprocesos al diagrama

general, esto con el fin de mostrar de una manera más detallada los procesos que se llevan a cabo para realizar el registro de una solicitud de crédito hipotecario. La figura 4.1.1.2 describe de manera general el diagrama de procesos nivel 1 del sistema a través de todos los subprocessos y cómo interactúan con la base de datos.

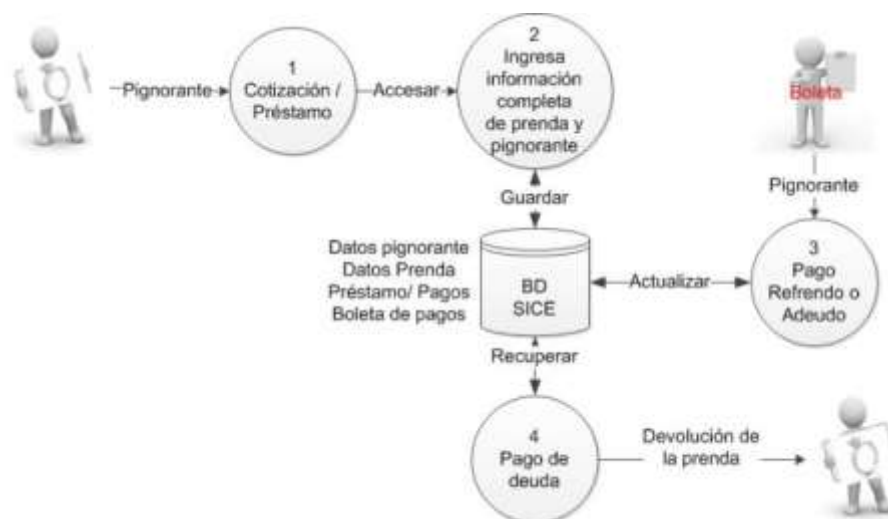


Figura. 4.1.1.2 Diagrama de procesos N1

1. **Cotización o Préstamo:** Este proceso valida el inicio de sesión al ingresar al sistema. Consiste en entregar al pignorante el monto máximo que se puede prestar en efectivo de la prenda. De ser favorable para el cliente, el valuador ingresa al sistema para crear los datos del nuevo pignorante.
2. **Ingresar información completa de prenda y pignorante:** Proceso en el que se recopila toda la información referente al pignorante (y/o actualización), los datos y características de la prenda que se empeñara, etc., para en su caso, aprobación del préstamo.
3. **Pago de refrendos o Adeudos:** En este proceso se reciben los pagos que se deben realizar (en el tiempo estipulado) del préstamo otorgado al pignorante. Así mismo se cobrarán los intereses por haberse excedido en los límites de la fecha estipulada para hacer los pagos con su debido recargo por incumplimiento.
4. **Consignación y Finiquitos:** En este proceso se cobrará el pago de la deuda. Con la cual se actualiza la base de datos finiquitando la deuda. En caso de que

la prenda sea consignada por falta de pago. **Ver apéndice 1.**

Nivel 2

En el diagrama de nivel superior se plasman todos los procesos que describen al proceso principal. En este nivel los procesos no suelen interrelacionarse directamente, sino que entre ellos debe existir algún almacenamiento o entidad externa que los una. Esta regla de construcción sirve como ayuda al analista para contemplar que en un nivel tan elevado de abstracción (DFD Nivel 1) es altamente probable que la información que se maneje sea almacenada en el sistema aunque no esté especificado por un Requisito funcional, siendo en realidad un requisito no-funcional.

Los tres procesos más importantes en el nivel anterior son:

- Ingresar información significativa del pignorante y prenda.
- Pago de Refrendo ó Adeudo.
- Consignación y Finiquitos.

A continuación se muestra la información referente al proceso: Ingresar información completa de pignorante y prenda, mediante el diagrama de la figura 4.1.1.3. Se muestra el diagrama de procesos en nivel 2: Captura de solicitud explotada en sus subprocesos correspondientes.

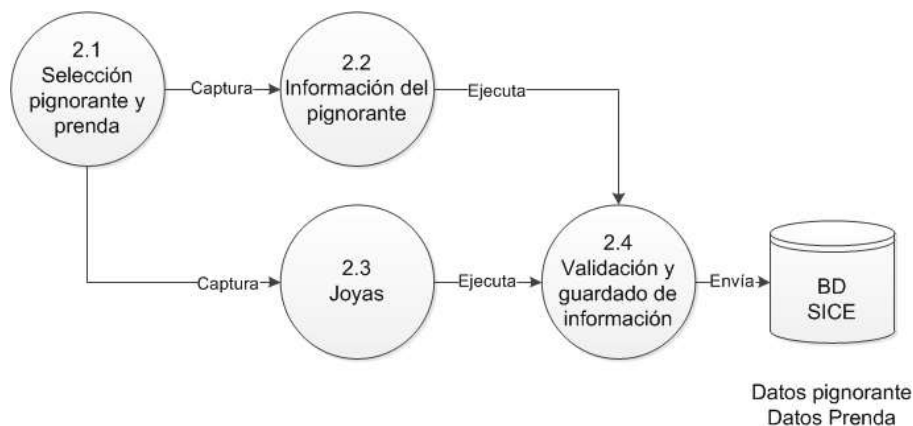


Figura 4.1.1.3 Diagrama de proceso N2: pignorante y prenda

- 2.1. **Selección pignorante y prenda:** Este subproceso se encarga de dar al usuario la opción de elegir el tipo de registro que ingresará a la base de datos.
- 2.2. **Información del pignorante:** El subproceso de Información de pignorante, permite ingresar nuevos datos de un pignorante no registrado o buscar una solicitud ya hecha para hacer modificaciones y continuar con la captura de datos de la misma.
- 2.3. **Joyas:** Este subproceso nos permite capturar los datos que permiten agregar la información específica de cada joya, como puede ser: Oro, Plata, Piedras preciosas, relojes, etc.
- 2.4. **Validación y guardado de información:** El subproceso de validación se encarga de verificar que la información ingresada corresponda con lo que se pide. Se cerciora de que los campos donde se pide texto, haya texto, donde se pidan números haya números, donde no se permitan caracteres nulos no se les encuentre, etc. Si la información es válida se almacena en la base de datos.

El proceso de Pago de Refrendo o Adeudos. También puede ser explotado en varios subprocesos que permiten entender de mejor manera la forma en que los procesos que se ejecutan y qué tarea realizarán cada uno de ellos. Vea figura 4.1.1.4.

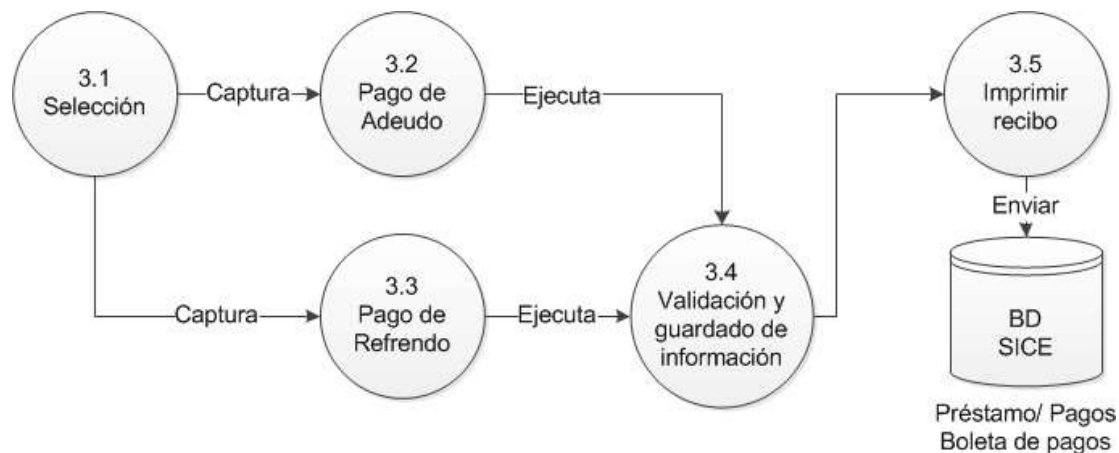


Figura 4.1.1.4 Diagrama de procesos N2: Pago de refrendo, adeudo

- 3.1. **Selección:** Este subproceso se encarga de dar al usuario la opción de elegir el tipo de registro que ingresará a la base de datos.

- 3.2. **Pago de adeudo:** Mediante este subproceso se realiza una actualización en la BD, del o los adeudos y fechas límite de liquidación.
- 3.3. **Pago de refrendo:** En este subproceso se brinda la opción al pignorante de volver a refrendar una prenda ya en resguardo.
- 3.4. **Validación y guardado de información:** En este subproceso se confirma la actualización de los datos para su almacenamiento en la BD.
- 3.5. **Imprimir recibo:** Mediante este subproceso se realiza una actualización sobre la base de datos con toda la información necesaria que nos permite generar un recibo de no adeudo o pagos realizados y adeudos por liquidar.

El cuarto proceso es Pago de deuda. Pueden ser explotados en varios subprocesos que ayudarán a comprender cómo interactúan los procesos y cómo funcionan cada uno de ellos. Vea figura 4.1.1.5.

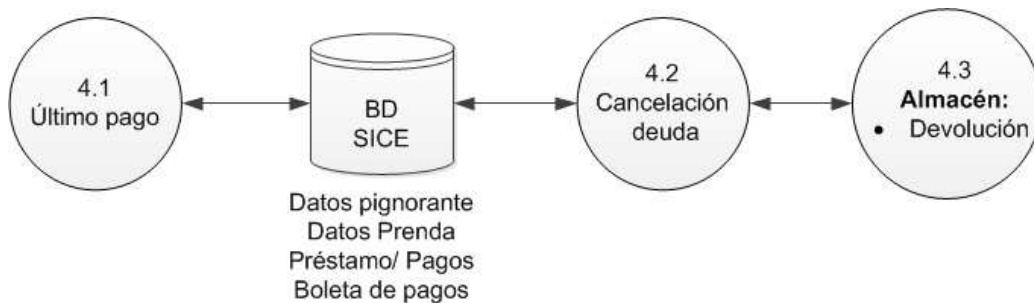


Figura 4.1.1.5 Diagrama de procesos N2: Pago de deuda

- 4.1 **Último pago:** Subproceso en el que hace la consulta a la BD para verificar si se ha liquidado el 100% de la deuda. De ser correcto se actualiza el sistema para hacer el requerimiento de la prenda a Almacén.
- 4.2 **Cancelación deuda:** En este subproceso se entrega la liberación de la boleta y cancelación de la deuda.
- 4.3 **Almacén:** Mediante este subproceso se realiza una actualización del stock en resguardo para la devolución de la prenda al pignorante.

Nivel 3

En un diagrama de nivel 3 o mayor, comienzan a explotarse las excepciones a los caminos principales de la información dado que aumenta progresivamente el nivel de detalle. De aquí en adelante se permiten los flujos entre procesos.

El DFD (Diagrama de Flujo de Datos) nivel 3 puede considerarse el máximo para ser validado en forma conjunta con el usuario dado que en los niveles posteriores el alto grado de complejidad del diagrama puede resultar de muy difícil lectura para personas ajenas al equipo de sistemas. También se recomienda el diagrama de nivel superior.

En los diagramas de flujo se consideran los siguientes pasos:

- Existe siempre un camino que permite llegar a una solución (finalización del algoritmo).
- Existe un único inicio de proceso.
- Existe un único punto de fin para el proceso de flujo excepto por el rombo que indica una comparación con dos caminos posibles.

El diagrama de flujo permite conocer cómo será construido el sistema (Vea figura 4.1.1.6), es una representación gráfica de los pasos que se siguen en toda una secuencia de actividades, dentro de un proceso o un procedimiento, identificándolos mediante símbolos de acuerdo con su naturaleza; incluye, además, toda la información que se considera necesaria para el análisis.

Los diagramas de flujo no sólo se emplean para modelar sistemas de procesos de información, sino también para modelar organizaciones enteras, esto es, como una herramienta para la planeación estratégica y de negocio.

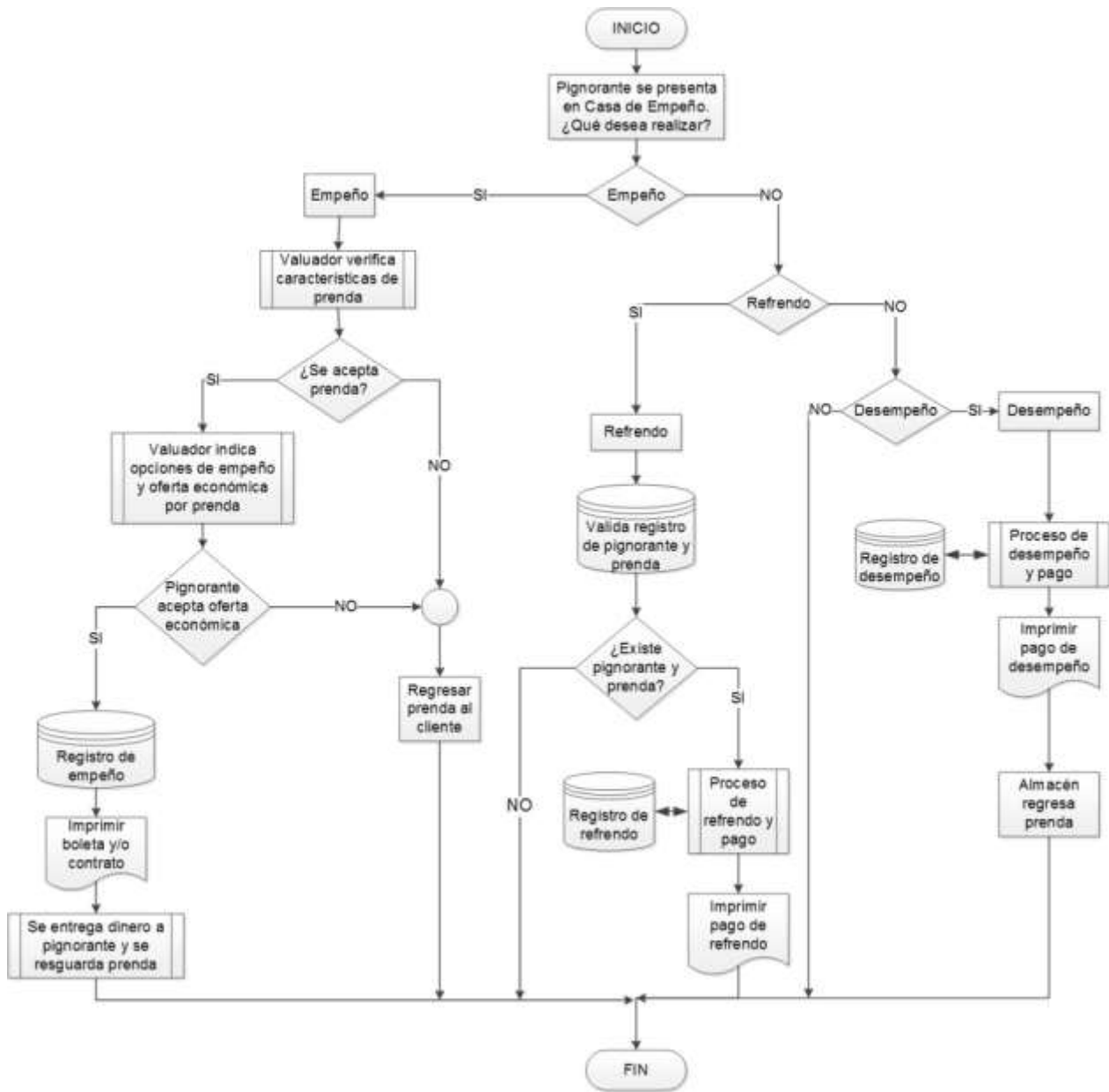


Figura 4.1.1.6 Diagrama de flujo general

Del anterior diagrama general desglosaremos los diagramas de flujo particulares para poder dar una mejor descripción, los cuales se muestran a continuación:

- Diagrama de flujo Empeños
- Diagrama de flujo Desempeños
- Diagrama de flujo Refrendos
- Diagrama de flujo de Pagos

- Diagrama de flujo Creación de Sucursal
- Diagrama de flujo creación de Usuario
- Diagrama de flujo Unir Usuario a Sucursal
- Diagrama de flujo Notificación al Pignorante sobre sus pagos

Diagrama de flujo Empeños Es el diagrama de flujo más importante debido a que sin él, sería inútil tener la información en nuestra base de datos. Vea figura 4.1.1.7.

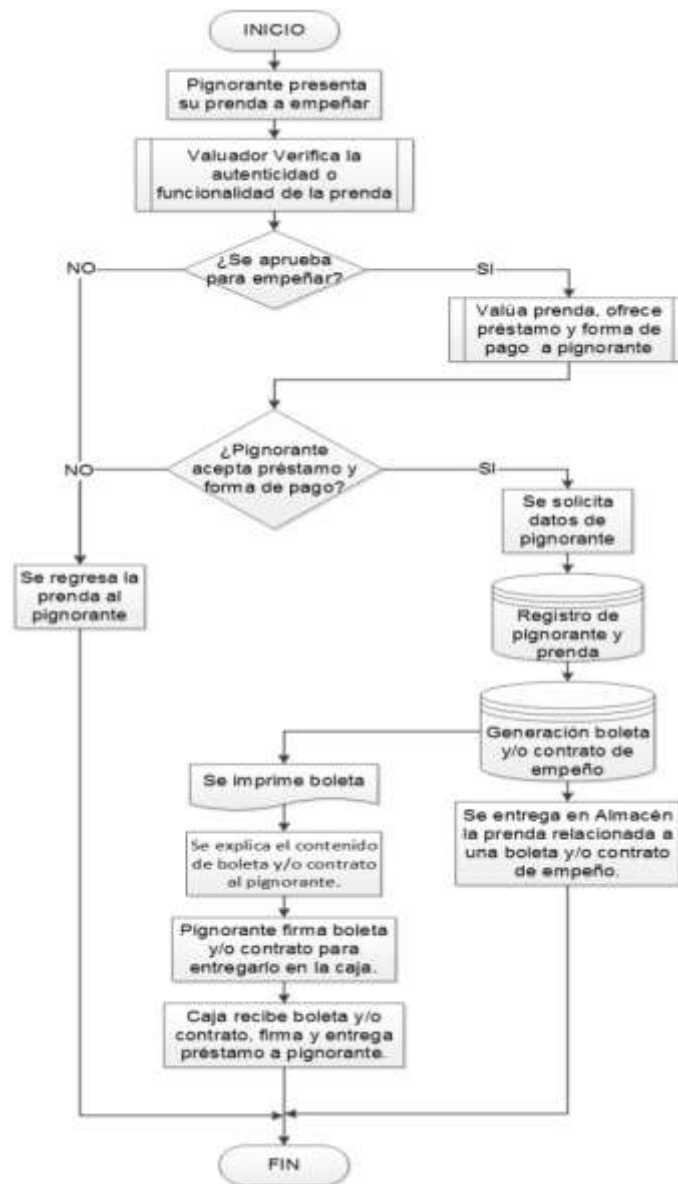


Figura 4.1.1.7 Diagrama de flujo empeños

Una vez que se cuenta con la información fidedigna del pignorante, datos de la prenda, monto de préstamo, etc. El sistema generará una boleta y/o contrato de empeño. La cual entregará el valuador al pignorante explicando las fechas de pagos, cláusulas y amonestaciones por incumplimiento de pago, cerrando el contrato con las firmas que deberá llenar el pignorante.

El sistema generará la boleta y se la asignará a la prenda una copia de la boleta y contrato de empeño para ser almacenada.

El pignorante deberá dirigirse a la caja para entregar la boleta, el contrato firmado con la cual se le entregará el préstamo previamente acordado. Vea figura 4.1.1.8.

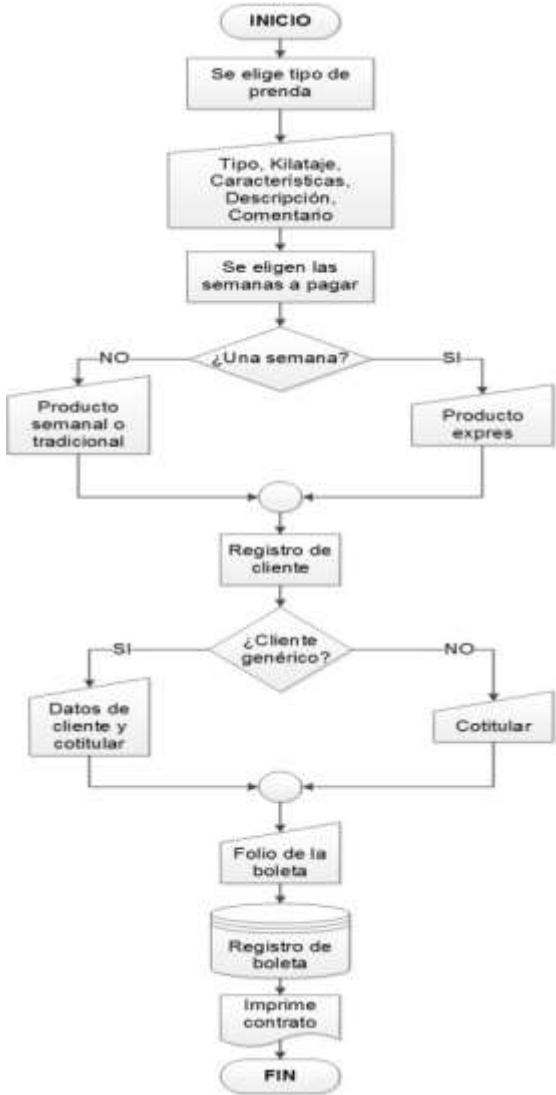


Figura 4.1.1.8 Diagrama de flujo empeño (tipo de prenda)

Se puede seleccionar un tipo de prenda a través de las opciones que mostrará el sistema.

El siguiente proceso es elegir el tiempo de pago; esto ya lo hemos descrito con anterioridad. Al elegir el tipo de préstamos ya sea semanal, express, o tradicional se registran los datos del cliente y datos de un cotitular, para el caso de que el pignorante por alguna razón no pueda ir a hacer sus pagos, pueda ir el cotitular que el pignorante elija para realizar el o los pagos.

Diagrama de flujo de pagos

El sistema se encontrará interconectado entre sucursales el pignorante tiene la posibilidad de realizar el pago de la prenda que fuera resguardada con anterioridad en cualquiera de las sucursales.

Se verificará el status de la boleta dentro del sistema. Si la boleta no está vencida se registrará el número de pagos y se entregará el monto total previamente pagado y el monto restante a pagar. Se hace el cobro y se registrará dicho pago dentro del sistema.

En caso contrario en que la boleta esté vencido el pago, el sistema entregará al cajero la fecha límite de pago. Agregando intereses moratorios por falta de pago y el total a pagar en ese momento. Vea figura 4.1.1.9.

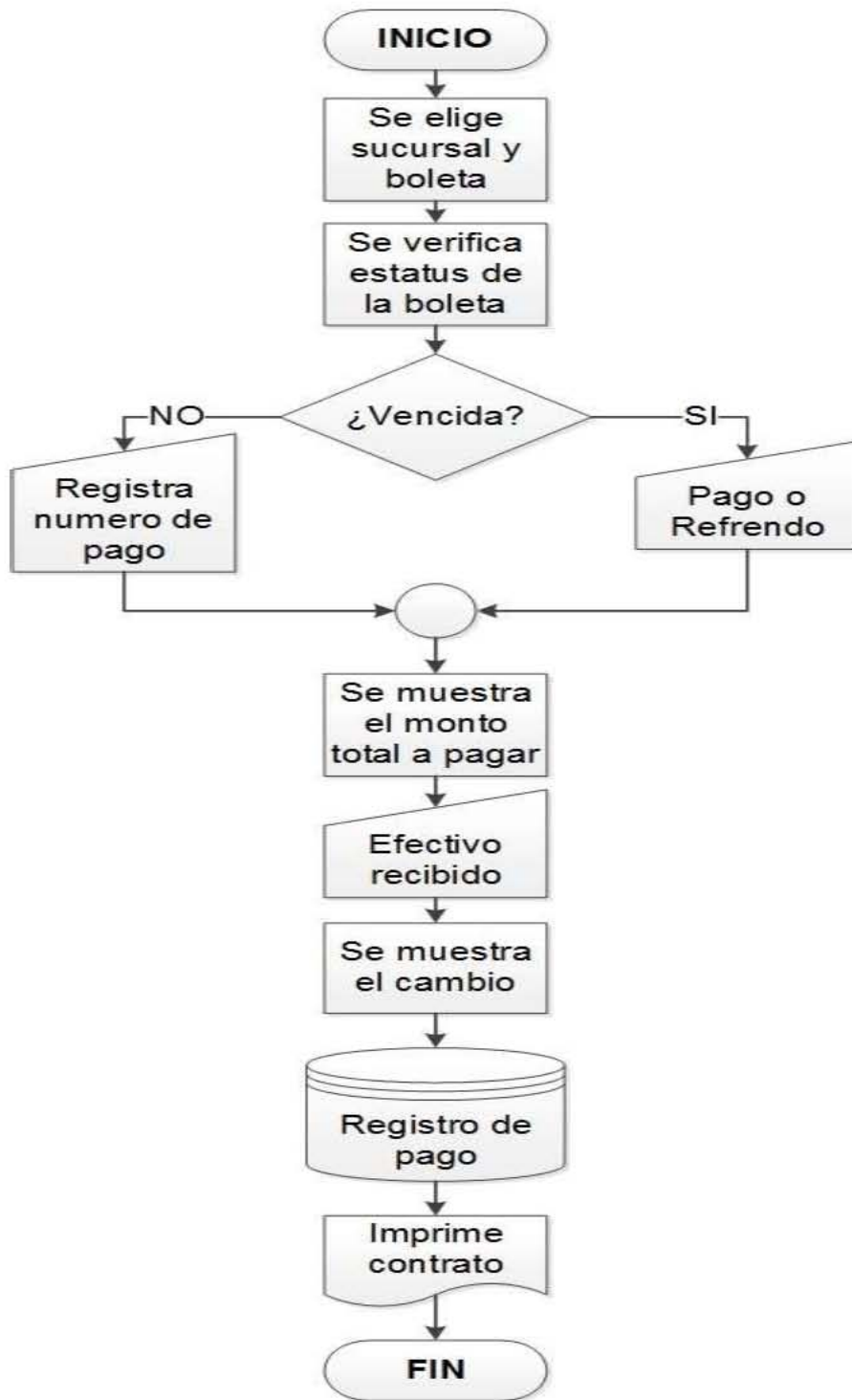


Figura 4.1.1.9 Diagrama de flujo de pagos

Diagrama de flujo desempeño

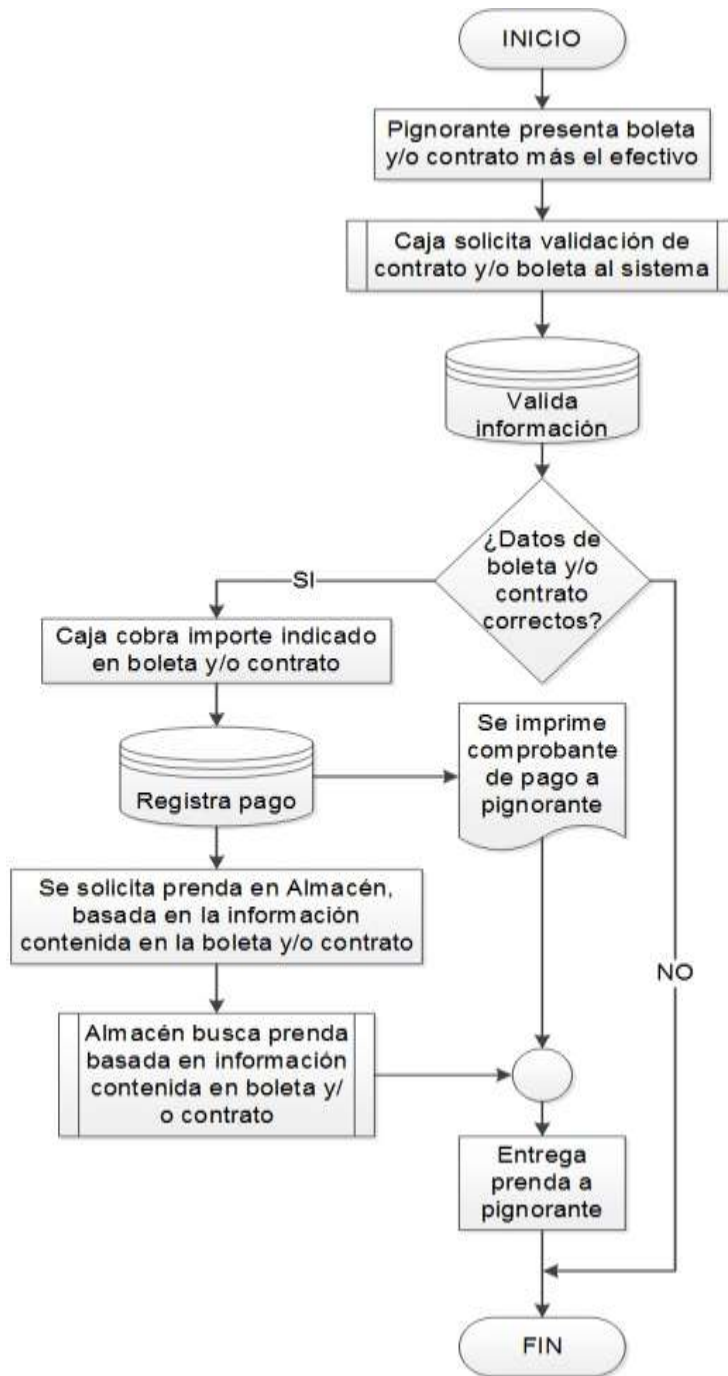


Figura 4.1.1.10 Diagrama de flujo desempeño

Este subproceso inicia cuando el pignorante llega a la Casa de Empeños y se dirige a cajas para entregar su boleta y/o contrato de empeño y el total de efectivo descrito en la boleta.

El personal de caja ingresará al sistema y llenará los datos requeridos de la boleta y/o contrato.

El sistema validará la información y corroborará si los datos de la boleta y el total a pagar es correcto. De ser incorrecta la información se verificará la causa del rechazo del sistema; esto sucederá si no se liquida el total del préstamo o por algún dato incorrecto ingresado al sistema.

Si la información es correcta, se cobrará el importe total que se indica en la boleta. Se registra el pago y se hace la solicitud para la entrega de la prenda. El sistema registrará la solicitud de pago e imprimirá un comprobante de la misma.

Diagrama de flujo creación de sucursal

El administrador registrará los datos de la nueva sucursal. Iniciando por el nombre, el cual identificará cada una de las sucursales y este es única para cada una de ellas.

Se registrará los datos de la sucursal: Nombre, dirección, teléfono, correo electrónico, etc. Y se actualiza la Base de Datos de la nueva sucursal.

El siguiente diagrama de flujo muestra el proceso de creación de sucursal. Vea figura 4.1.1.11.



Figura 4.1.1.11 Diagrama de creación de sucursal

Diagrama de flujo creación de usuario

Una vez creada la sucursal se podrá registrar nuevos usuarios que laboren en cada una de las sucursales.

Se registrará un nuevo usuario con un identificador característico dentro del sistema ingresando su información: Nombre, dirección, teléfono, fecha de nacimiento, puesto, etc. Para terminar el registro se guarda la información en la Base de Datos. Vea figura 4.1.1.12.



Figura 4.1.1.12 Diagrama de flujo creación de usuario

Diagrama de flujo usuario - sucursal

Una vez creada la sucursal y el usuario se vincularán para tener la opción de localizar a un usuario (pignorante) de manera más eficiente en cuál de las sucursales fue registrado junto con la prenda empeñada; de esta forma el pignorante podrá acudir a cualquier sucursal a realizar los pagos de la prenda empeñada. De esta forma se actualizará la Base de Datos para poder ser rastreada la información en cualquiera de las sucursales que estarán interconectadas en el servidor principal. Vea figura 4.1.1.13.



Figura. 4.1.1.13 Diagrama de flujo usuario a sucursal

Diagrama de flujo notificación al usuario sobre sus pagos

Para llevar un control estricto de todos los pignorantes así como su estatus los cuales pueden ser OK (al corriente) o Deudor. Se elige cualquiera de las sucursales y se lanza una petición de notificación de Boletas vencidas, la cual nos indica que usuarios no han cumplido con sus pagos. Dentro de esta notificación se verifica si ya se ha llamado al usuario para que realice el pago vencido o si es la primera vez que se atrasa en el pago para en este caso hacer la notificación a cobranza para que se comunique con el cliente y realice el pago, agregando en la Base de Datos la actualización de que ya se ha notificado al usuario de dicho problema de atraso. Vea figura 4.1.1.14.

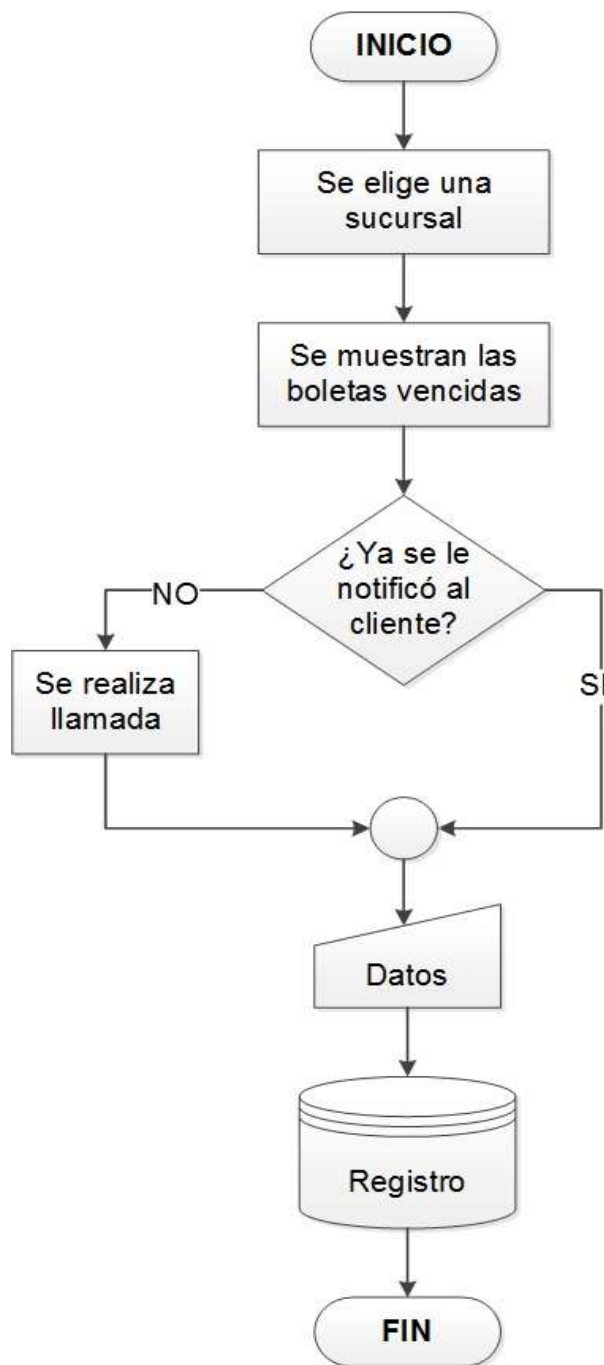


Figura 4.1.1.14 Diagrama de flujo Notificación al Usuario sobre sus pagos

4.1.2 Diagrama de Entidad-Relación del sistema (SICE)

Creado por Peter Chen en 1976, permite describir las entidades involucradas en una base de datos, así como las relaciones y restricciones de ellas.

El modelo Entidad Relación (ER) proporciona una herramienta para representar información del mundo real a nivel conceptual.

Algunas características que debe de cumplir un diagrama Entidad-Relación son:

- Notación gráfica.
- Semántica clara.
- Fácil de entender aún por no especialistas.
- Independiente de cualquier Sistema de Administración de Bases de Datos.

Entidad

Las entidades representan cosas u objetos (ya sean reales o abstractos), que se diferencian claramente entre sí.

Un conjunto de entidades o tipo entidad es un conjunto de entidades que comparten las mismas propiedades.

Atributos

Los atributos definen o identifican las características de la entidad. Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...).

Los atributos pueden ser:

- Simples o compuestos.
- Nulos
- Llaves

Relación

Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.

Relaciones de cardinalidad

Expresa el número de entidades que pueden asociarse vía un tipo de relación.

Para las relaciones binarias la cardinalidad puede ser:

- Uno a uno
- Uno a varios
- Varios a uno
- Varios a varios

Claves

Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad) o le aplica un vínculo (exactamente como comentábamos en las relaciones).

Clave primaria

Identifica inequívocamente un solo atributo no permitiendo que se repita en la misma entidad.

Clave externa o clave foránea

Este campo tiene que estar estrictamente relacionado con la clave primaria de otra entidad, para así exigir que exista previamente ese clave, vea las figuras 4.1.2.4, 4.1.2.5 y 4.1.2.6.

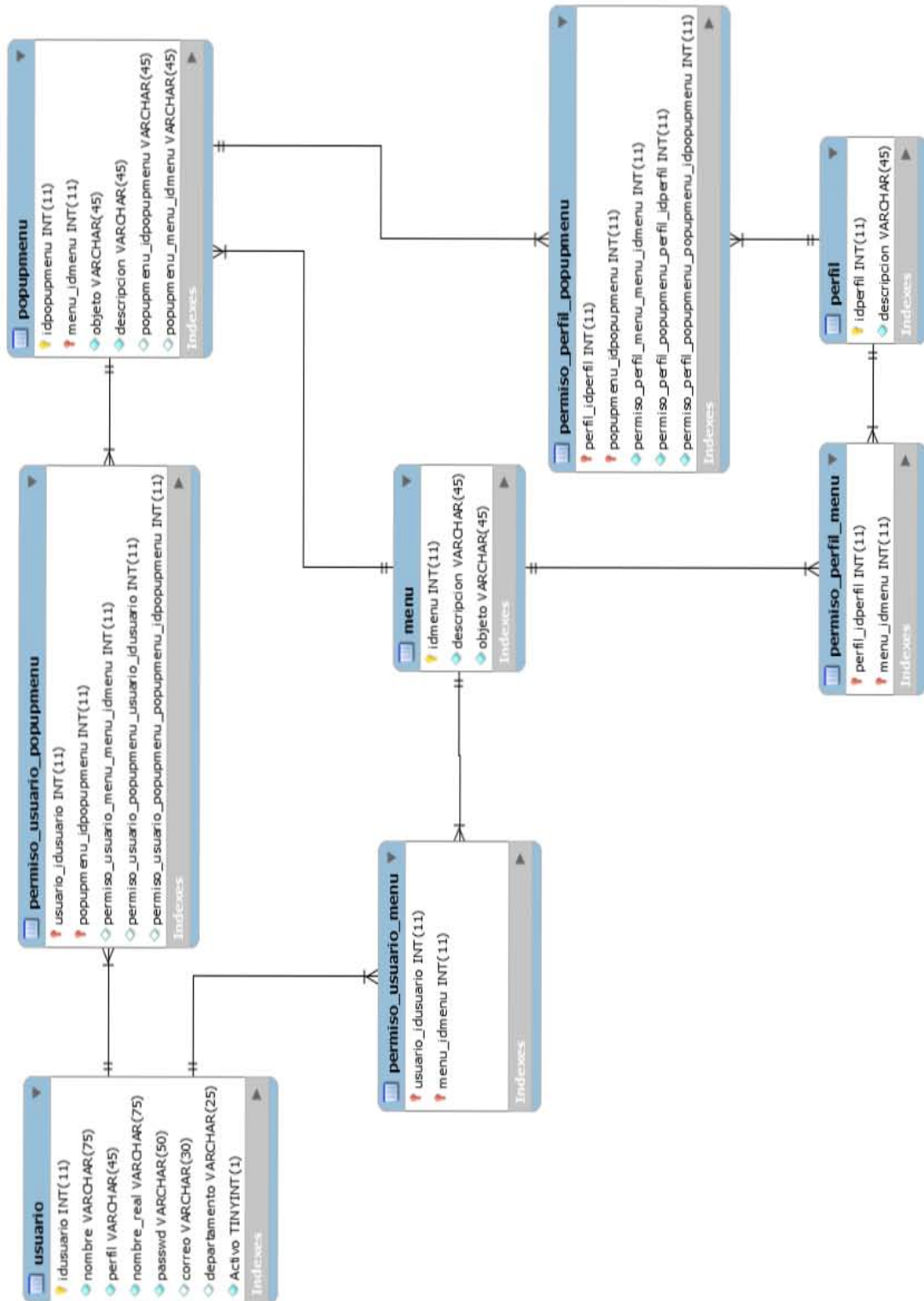


Figura 4.1.2.5 Diagrama de Entidad-Relación II

4.1.3 Diccionario de datos

En términos generales el diccionario de datos (DD) es un listado organizado de todos los datos que pertenecen a un sistema. El objetivo de un diccionario de datos es dar precisión sobre los atributos o campos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades. Son buenos complementos a los diagramas de flujo de datos, los diagramas entidad-relación por mencionar algunos.

A continuación de la tabla 4.1.3.1 a la 4.1.3.14 se muestran los campos y algunas de las características que se manejarán en la base de datos para su implementación.

Diccionario de Datos del SICE

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
id_clientes	INT	11	✓	✓		CLIENTES	Identificador del cliente.
nombre_clientes1	VARCHAR	40				CLIENTES	Nombre del cliente.
nombre_clientes2	VARCHAR	40				CLIENTES	Complemento del nombre del cliente.
codigo_postal	VARCHAR	8				CLIENTES	Código postal
direccion	VARCHAR	45				CLIENTES	Dirección postal del cliente
id_colonia	VARCHAR	10				CLIENTES	Identificador de la colonia
colonia	VARCHAR	45				CLIENTES	Descripción de la colonia de residencia
Id_Delegacion	VARCHAR	10				CLIENTES	Identificador de la delegación
ciudad	VARCHAR	15				CLIENTES	Descripción de la ciudad de residencia
id_estados	VARCHAR	10				CLIENTES	Identificador de estado de residencia del cliente.

Tabla 4.1.3.1. Clientes

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
estado	VARCHAR	25				CLIENTES	Desc. Del estado de residencia del cliente.
Id_pais	VARCHAR	10				CLIENTES	Identificador del país.
pais	VARCHAR	15				CLIENTES	Desc. Del país de residencia del cliente.
rfc	VARCHAR	15				CLIENTES	Número del Registro Federal de Contribuyentes asociado al cliente.
tipo_Identificacion	VARCHAR	20				CLIENTES	Tipo de identificación que presenta el cliente.
identificacion	VARCHAR	20				CLIENTES	Descripción de la identificación del cliente.
mail	VARCHAR	45				CLIENTES	Dirección electrónica del cliente.
cotitular	VARCHAR	100				CLIENTES	Nombre de la persona con la que el titular comparte responsabilidades en el contrato.

Tabla 4.1.3.1. Clientes

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
telefono_1	VARCHAR	15				CLIENTES	Teléfono particular del cliente.
telefono_2	VARCHAR	15				CLIENTES	Teléfono opcional del cliente.
fax	VARCHAR	13				CLIENTES	Teléfono de fax del cliente.
tipo	VARCHAR	4				CLIENTES	Tipo de cliente
id_zona	VARCHAR	10				CLIENTES	Identificador de zona en la que se realiza el contrato.
impuesto	VARCHAR	1				CLIENTES	Impuesto aplicable por contrato. Por el momento es Nulo.
observaciones	VARCHAR	30				CLIENTES	Indicaciones o comentarios especiales sobre el cliente.
activo	VARCHAR	5				CLIENTES	Indica si el registro tiene estatus activo.

Tabla 4.1.3.1. Clientes

Campo	Tipo de Dato	Long.	No Nulo	PK	FK	Relación con tabla	Descripción
id_comprobante	INT	2	✓	✓		TIPO_COMPROBANTE	Identificador de comprobante.
descripcion	VARCHAR	45				TIPO_COMPROBANTE	Descripción del comprobante.
activo	TINYINT	4				TIPO_COMPROBANTE	Indica si el registro tiene estatus activo.

Tablas 4.1.3.2. Tipo Comprobante

Campo	Tipo de Dato	Long.	No Nulo	PK	FK	Relación con tabla	Descripción
idmenu	INT	11	✓	✓		MENU	Identificador del menú.
descripcion	VARCHAR	45	✓			MENU	Se almacena el nombre de cada menú.
objeto	VARCHAR	45	✓			MENU	Nombre que tiene el id en el sistema.

Tabla 4.1.3.3. Menú

Campo	Tipo de Dato	Long.	No Nulo	PK	FK	Relación con tabla	Descripción
id_metales	INT	11	✓	✓		METALES	Identificador del metal
descripcion	VARCHAR	45				METALES	Descripción o nombre del metal.
activo	TINYINT	4				METALES	Indica si el registro tiene estatus activo.

Tabla 4.1.3.4. Metales

Campo	Tipo de Dato	Long.	No Nulo	PK	FK	Relación con tabla	Descripción
id_monedas	INT	11	✓	✓		MONEDAS	Identificador de la moneda.
descripcion	VARCHAR	20				MONEDAS	Descripción o nombre de la moneda.
activo	TINYINT	4				MONEDAS	Indica si el registro tiene estatus activo

Tabla 4.1.3.5. Monedas

Campo	Tipo de Dato	Long.	No Nulo	PK	FK	Relación con tabla	Descripción
id_piezas	INT	11	✓	✓		PIEZAS	Identificador de la pieza.
ramo_id_ramo	INT	11	✓		✓	PIEZAS	Identificador del ramo al que pertenece la pieza.
descripcion	VARCHAR	45				PIEZAS	Descripción o nombre de la pieza.
activo	TINYINT	1				PIEZAS	Indica si el registro tiene estatus activo.

Tabla 4.1.3.6. Piezas

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
id_plan_pago_detalle	INT	10	✓	✓		PLAN_PAGO_DETALLE	Identificador del detalle del plan de pago.
id_boleta_detalle	INT	11	✓		✓	PLAN_PAGO_DETALLE	Identificador del detalle de la boleta
status	INT	11	✓		✓	PLAN_PAGO_DETALLE	Estatus del pago
tipo_producto	INT	11				PLAN_PAGO_DETALLE	Tipo de producto asociado a la boleta
numero_pago	INT	2				PLAN_PAGO_DETALLE	Número del pago realizado al préstamo.
precio_pieza	DOUBLE	(9,2)				PLAN_PAGO_DETALLE	Precio en el que fue valuada la prenda
almacenaje	DOUBLE	(9,2)				PLAN_PAGO_DETALLE	Monto por almacenaje.
seguro	DOUBLE	(9,2)				PLAN_PAGO_DETALLE	Monto de seguro asociado a la boleta

Tabla 4.1.3.7. Plan Pago Detalle

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
capital	DOUBLE	(9,2)				PLAN_PAGO_DETALLE	Monto que el cliente recibe sin cargo por intereses.
interes	DOUBLE	(9,2)				PLAN_PAGO_DETALLE	Interés aplicado a la boleta
fecha	DATE	DATE				PLAN_PAGO_DETALLE	Fecha en la que se generó la boleta.
fecha_pago	DATE	DATE				PLAN_PAGO_DETALLE	Fecha en que se realizó el pago.
pago_total	DOUBLE	(9,2)				PLAN_PAGO_DETALLE	Monto total del pago del préstamo.
referendo	DOUBLE	(9,2)				PLAN_PAGO_DETALLE	Monto por refrendo.
morosidad	DOUBLE	(9,2)				PLAN_PAGO_DETALLE	Monto por morosidad
sucursal_idsucursal	INT	11				PLAN_PAGO_DETALLE	Identifica la sucursal en donde se realiza el pago.

Tabla 4.1.3.7. Plan Pago Detalle

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
id_productos	INT	11	✓	✓		PRODUCTOS	Identificador del producto.
descripcion	VARCHAR	45				PRODUCTOS	Descripción del producto.
numero_periodos	INT	11				PRODUCTOS	Número de semanas en las que el producto se
de_semana	INT	11				PRODUCTOS	Indica el número de semana inicial.
a_semana	INT	11				PRODUCTOS	Indica el número de semana final.
almacenaje	DOUBLE	(9,2)				PRODUCTOS	Monto establecido por concepto de almacenamiento de acuerdo a las características del producto.
seguro	DOUBLE	(9,2)				PRODUCTOS	Monto del seguro de acuerdo a las características del producto.
interes	DOUBLE	(9,2)				PRODUCTOS	Monto de Interés que se cobrará de acuerdo a las características del producto.

Tabla 4.1.3.8. Productos

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
compra	DOUBLE	(9,2)				PRODUCTOS	Monto al que se compra el producto.
aplica_incremento	TINYINT	1				PRODUCTOS	Bandera que indica si se le aplica o no un incremento al producto.
apartir_semana	INT	11				PRODUCTOS	Indica el número de la semana a partir de la cual se aplica un incremento.
retiro_anticipado	TINYINT	1				PRODUCTOS	Indica el número de la semana final en la que se aplicó un incremento.
aplica_interes_moratorio	TINYINT	1				PRODUCTOS	Bandera que indica si al producto se le aplica un interés moratorio.
aplica_interes_refrendo	TINYINT	1				PRODUCTOS	Bandera que indica si al producto se le aplica un interés por refrendo.
pago_unico	TINYINT	1				PRODUCTOS	Indica si hay pago único para el producto.

Tabla 4.1.3.8. Productos

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
intereses_moratorio	DOUBLE	(9,2)				PRODUCTOS	Monto de interés moratorio.
intereses_referendo	DOUBLE	(9,2)				PRODUCTOS	Monto de interés de referendo.
semana_fundicion	INT	11				PRODUCTOS	Semana en la que se fundió la prenda, solo aplica para metales.
activo	TINYINT	1				PRODUCTOS	Indica si el registro tiene estatus activo.

Tabla 4.1.3.8. Productos

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
id_ramo	INT	11	✓	✓		RAMO	Identificador del ramo.
descripcion	VARCHAR	45				RAMO	Descripción del ramo.
activo	TINYINT	1				RAMO	Indica si el registro tiene estatus activo.

Tabla 4.1.3.9. Ramo

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
idsucursal	INT	11	✓	✓		REPORTESALDOS	Identificador de la sucursal.
descripcion	VARCHAR	255				REPORTESALDOS	Descripción de reporte de saldos.
saldoanterior	DOUBLE	(9,2)				REPORTESALDOS	Monto de saldo anterior.
empenos	DOUBLE	(9,2)				REPORTESALDOS	Monto de empeños
pagos	DOUBLE	(9,2)				REPORTESALDOS	Monto de pagos.
ventas	DOUBLE	(9,2)				REPORTESALDOS	Monto por ventas.
dotacion	DOUBLE	(9,2)				REPORTESALDOS	Monto de dotación para operaciones de sucursal.
gastos	DOUBLE	(9,2)				REPORTESALDOS	Monto de gastos de la sucursal.
saldofinal	DOUBLE	(9,2)				REPORTESALDOS	Monto de Saldo Final

Tabla 4.1.3.10. Reporte Saldos

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
id_sucursales_tipos	INT	11	✓	✓		SUCURSALES_TIPOS	Identificador del tipo de sucursal.
descripcion	VARCHAR	45				SUCURSALES_TIPOS	Descripción de la sucursal.
activo	TINYINT	1				SUCURSALES_TIPOS	Indica si el registro tiene estatus activo.

Tablas 4.1.3.12. Sucursales Tipos

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
id_sucursal_usuarios	INT	11	✓	✓		SUCURSAL_USUARIOS	Identificador de usuario por sucursal.
id_usuario	INT	11	✓		✓	SUCURSALES_TIPOS	Identificador de usuario.
id_sucursal	INT	11	✓		✓	SUCURSAL_USUARIOS	Identificador de sucursal.
activo	TINYINT	1				SUCURSALES_TIPOS	Indica si el registro tiene estatus activo.

Tablas 4.1.3.13. Sucursal Usuarios

Campo	Tipo de Dato	Longitud	No Nulo	PK	FK	Relación con tabla	Descripción
idusuario	INT	11				USUARIO	Identificador del usuario.
nombre	VARCHAR	75				USUARIO	Nombre del usuario.
perfil	VARCHAR	45				USUARIO	Nombre del perfil al que pertenece el usuario
nombre_real	VARCHAR	75				USUARIO	Nombre completo del usuario.
passwd	VARCHAR	50				USUARIO	Contraseña del usuario.
correo	VARCHAR	30				USUARIO	Dirección electrónica del usuario.
d departamento	VARCHAR	25				USUARIO	Área donde labora el usuario.
activo	TINYINT	1				USUARIO	Indica si el registro tiene estatus activo.

Tablas 4.1.3.14. Usuario

4.1.4 Normalización de la Base de Datos

La normalización es un proceso para evaluar y corregir estructuras de tablas a fin de minimizar redundancias de datos, con lo cual se reduce la probabilidad de anomalías de datos.

La normalización funciona por medio de una serie de etapas llamadas formas normales. Las primeras tres etapas se describen como primera forma normal (1FN), segunda forma normal (2FN) y tercera forma normal (3FN).

- **Primera Forma Normal 1FN:** Incluye la eliminación de datos duplicados
- **Segunda Forma Normal 2FN:** No incluye dependencias parciales; esto es, ningún atributo es dependiente de sólo una parte de la llave primaria.
- **Tercera Forma Normal 3FN:** Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llaves.

La información que se muestra en la tabla 4.1.4.1 se aplicará la 1FN , 2FN y se separará en tres diferentes tablas, por medio del campo común Id_Sucursal.

IdSucursal	Sucursal_Tipo	Id_Sucursal_Tipo	descripcion	direccion	telefono	siglas	sucursal_reporte	Activo	Id_Sucursal_usuario	Id_usuario	Id_Sucursal	Activo
20	S		SUCURSAL TOLUCA	ESTADO DE MEXICO	59875421	TOX	20	1	24	82	20	1
20	S		SUCURSAL TOLUCA	ESTADO DE MEXICO	59875421	TOX	20	1	25	80	20	1
20	S		SUCURSAL TOLUCA	ESTADO DE MEXICO	59875421	TOX	20	1	26	77	20	1
20	S		SUCURSAL TOLUCA	ESTADO DE MEXICO	59875421	TOX	20	1	28	1	20	1
20	S		SUCURSAL TOLUCA	ESTADO DE MEXICO	59875421	TOX	20	1	41	81	20	1
20	S		SUCURSAL TOLUCA	ESTADO DE MEXICO	59875421	TOX	20	1	49	66	20	1
18	S		SUCURSAL TLAHUAC	H	59939925	TH	18	1	20	70	18	1
18	S		SUCURSAL TLAHUAC	H	59939925	TH	18	1	21	83	18	1
18	S		SUCURSAL TLAHUAC	H	59939925	TH	18	1	22	74	18	1
18	S		SUCURSAL TLAHUAC	H	59939925	TH	18	1	24	1	18	1
18	S		SUCURSAL TLAHUAC	H	59939925	TH	18	1	29	66	18	1
18	S		SUCURSAL TLAHUAC	H	59939925	TH	18	1	40	65	18	1
21	S		SUCURSAL NALC...	ESTADO DE MEXICO	512385...	NAU	21	1	27	79	21	1
21	S		SUCURSAL NALC...	ESTADO DE MEXICO	512385...	NAU	21	1	28	84	21	1
21	S		SUCURSAL NALC...	ESTADO DE MEXICO	512385...	NAU	21	1	29	85	21	1
21	S		SUCURSAL NALC...	ESTADO DE MEXICO	512385...	NAU	21	1	30	1	21	1
21	S		SUCURSAL NALC...	ESTADO DE MEXICO	512385...	NAU	21	1	40	81	21	1
21	S		SUCURSAL NALC...	ESTADO DE MEXICO	512385...	NAU	21	1	47	65	21	1
17	S		SUCURSAL IZAPA...	H	51767463	IZT	17	1	17	68	17	1
17	S		SUCURSAL IZAPA...	H	51767463	IZT	17	1	18	75	17	1
17	S		SUCURSAL IZAPA...	H	51767463	IZT	17	1	19	76	17	1
17	S		SUCURSAL IZAPA...	H	51767463	IZT	17	1	22	1	17	1
17	S		SUCURSAL IZAPA...	H	51767463	IZT	17	1	38	66	17	1
17	S		SUCURSAL IZAPA...	H	51767463	IZT	17	1	46	65	17	1
14	S		SUCURSAL CHIT...	H	54514714	CHI	14	1	14	68	14	1

Figura 4.1.4.1 Información a normalizar 1FN y 2FN

Se muestra la tabla sucursal

idsucursal	Sucursales_Tipos_Id_Sucursales_Tipos	descripcion	direccion	telefono	siglas	sucursal_reporta	Activo
10	5	SUCURSAL URI MAGEN	BOSQUES	55961963	UMTEST1	0	0
20	5	SUCURSAL TOLUCA	ESTADO DE MÉXICO	59875421	TOL	19	1
18	5	SUCURSAL TLAHUAC	M	59868685	TH	14	1
21	5	SUCURSAL NAUCALPAN	ESTADO DE MÉXICO	5922685948	NAU	19	1
17	5	SUCURSAL IZTAPALAPA	M	59787493	IZT	14	1
16	5	SUCURSAL CUAJIMALPA	M	59834216	CUAJ	14	1
15	5	SUCURSAL AZCAPOTZALCO	MEXICO, DISTRITO FEDERAL	59868407	AZC	14	1
8	5	SUC CENTRO	AV CENTRO # 120	123456789	SC	0	0
9	5	NAUCALPAN	PARQUE DE LA NARANJA NO. 17	5565656532	NAU0001	0	0

Figura 4.1.4.2 Tabla sucursal

Se muestra la tabla sucursal_usuarios

	Id_Sucursal_Usuarios	Id_Usuario	Id_Sucursal	Activo
▶	1	1	12	1
	2	1	11	1
	3	53	11	1
	4	54	12	1
	5	55	11	1
	6	57	12	1
	7	56	11	0
*	NULL	NULL	NULL	NULL

Figura 4.1.4.3 Tabla sucursal_usuarios

Se muestra la tabla sucursales_tipos

	Id_Sucursales_Tipos	Descripcion	Activo
▶	5	TIENDA	1
	6	CONCENTRADORA	1
*	NULL	NULL	NULL

Figura 4.1.4.4 Tabla sucursales_tipos

Para la 3FN se dividió la información que se muestra a continuación.

id_Articulos_Metales	metales_id_Metales	Descripcion	Activo	id_Articulos_Metales	id_Metales	Factor
5	1	5 K. - ORO FINO	1	5	1	0.1666
6	1	6 K. - ORO FINO	1	6	1	0.2083
7	1	7 K. - ORO FINO	1	7	1	0.2500
8	1	8 K. - ORO FINO	1	8	1	0.2916
9	1	9 K. - ORO FINO	1	9	1	0.3333
10	1	10 K. - ORO FINO	1	10	1	0.3750
11	1	11 K. - ORO FINO	1	11	1	0.4166
12	1	12 K. - ORO FINO	1	12	1	0.4583
13	1	13 K. - ORO FINO	1	13	1	0.5000
14	1	14 K. - ORO FINO	1	14	1	0.5416
15	1	15 K. - ORO FINO	1	15	1	0.5833
16	1	16 K. - ORO FINO	1	16	1	0.6250
17	1	17 K. - ORO FINO	1	17	1	0.6666
18	1	18 K. - ORO FINO	1	18	1	0.7083
19	1	19 K. - ORO FINO	1	19	1	0.7500
20	1	20 K. - ORO FINO	1	20	1	0.7916
21	1	21 K. - ORO FINO	1	21	1	0.8333
22	1	22 K. - ORO FINO	1	22	1	0.8750
23	1	23 K. - ORO FINO	1	23	1	0.9166
24	1	24 K. - ORO FINO	1	24	1	0.9583
25	2	PLATA 0.999	1	25	2	1.0000
26	2	PLATA 0.720	1	26	2	0.9250
27	2	PLATA 0.925	1	27	2	0.7200

Figura 4.1.4.5 Información a normalizar 3FN

Derivado de la aplicación 3FN se obtuvieron dos tablas:

- articulos_metales_factor
- articulos_metales

id_Articulos_Metales	id_Metales	Factor
6	1	0.2083
7	1	0.2500
8	1	0.2916
9	1	0.3333
10	1	0.3750
11	1	0.4166
12	1	0.4583
13	1	0.5000
14	1	0.5416
15	1	0.5833
16	1	0.6250
17	1	0.6666
18	1	0.7083
19	1	0.7500
20	1	0.7916
21	1	0.8333
22	1	0.8750
23	1	0.9166
24	1	0.9583
25	2	1.0000
26	2	0.9250
27	2	0.7200

Figura 4.1.4.6 Tabla articulos_metales_factor

id_Articulos_Metales	metales_id_Metales	Descripcion	Activo
8	1	8 K. - ORO FINO	1
9	1	9 K. - ORO FINO	1
10	1	10 K. - ORO FINO	1
11	1	11 K. - ORO FINO	1
12	1	12 K. - ORO FINO	1
13	1	13 K. - ORO FINO	1
14	1	14 K. - ORO FINO	1
15	1	15 K. - ORO FINO	1
16	1	16 K. - ORO FINO	1
17	1	17 K. - ORO FINO	1
18	1	18 K. - ORO FINO	1
19	1	19 K. - ORO FINO	1
20	1	20 K. - ORO FINO	1
21	1	21 K. - ORO FINO	1
22	1	22 K. - ORO FINO	1
23	1	23 K. - ORO FINO	1
24	1	24 K. - ORO FINO	1
25	2	PLATA 0.999	1
26	2	PLATA 0.720	1
27	2	PLATA 0.925	1

Figura 4.1.4.7 Tabla articulos_metales

4.2 Creación de la base de datos

Implementación de Base de Datos

En esta sección se mostrará la descarga e instalación de la herramienta XAMPP la cual es una distribución de Apache libre y sencilla de instalar.

El XAMPP contiene las siguientes herramientas:

- Apache
- MySQL
- PHP
- PHPMyadmin.

Adicionalmente se instalará la siguiente herramienta MySQL WorkBench la cual tiene la ventaja de gestionar bases de datos de manera local o remota y la obtención del diagrama Entidad - Relación.

Las herramientas ya mencionadas serán instaladas en un Sistema Operativo Windows 8.1 a 64 bits.

Descarga e instalación de herramientas

La herramienta XAMPP se puede descargar de la página web oficial.⁷

Una vez finalizada la descarga se ejecuta el instalador, y se recomienda realizar la instalación en C:\. Vea figura 4.2.1.

⁷<https://www.apachefriends.org/es/download.html>

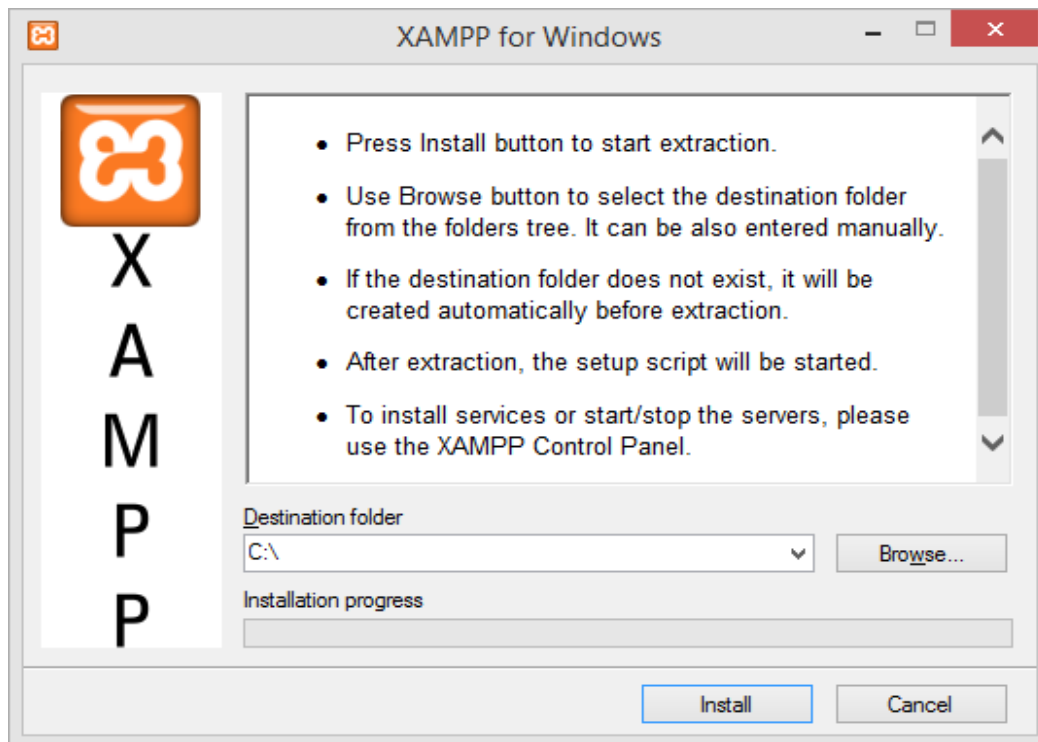


Figura 4.2.1 Instalación de XAMPP

Una vez realizada la instalación se deberá ejecutar la herramienta, la cual mostrará los servicios instalados y ejecutándose. Vea figura 4.2.2.

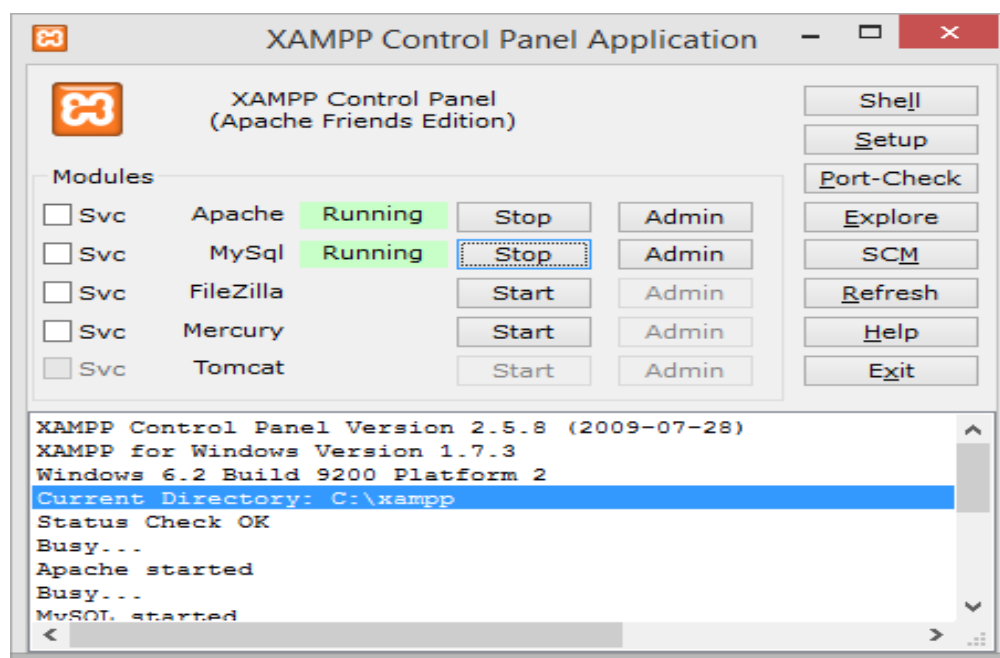


Figura 4.2.2 Servicios de XAMPP

El siguiente paso es desplegar un navegador web, e ingresar la siguiente dirección http. A continuación se deberá seleccionar el idioma de preferencia y con ello se instalará la versión de PHP, MySQL y Apache. Vea figura 4.2.3.

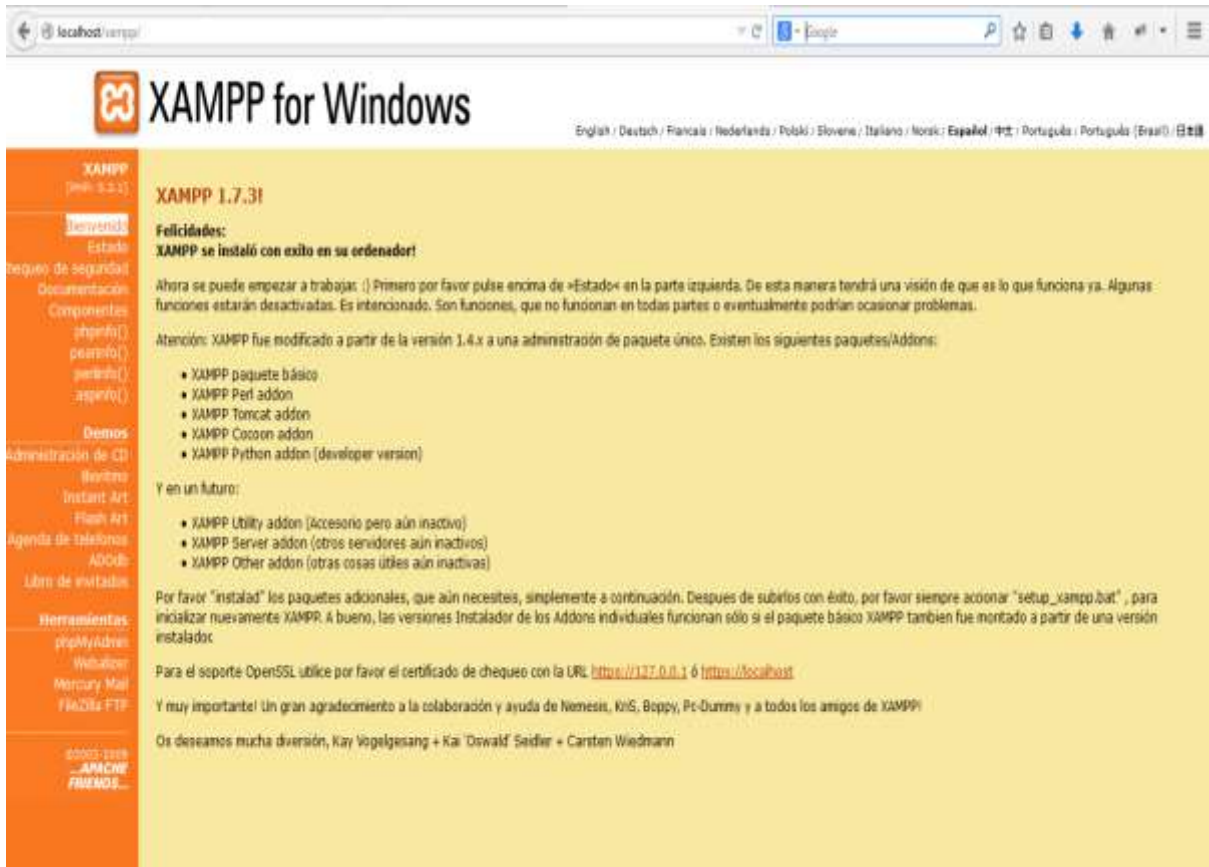


Figura 4.2.3 Página de administración de XAMP

La herramienta de WorkBench se puede descargar de su página web oficial.⁸

Una vez finalizada la descarga se ejecutará el instalador, el cual solicitará el directorio de instalación de la herramienta. Vea figuras 4.2.4 y 4.2.5

⁸ <http://www.mysql.com/products/workbench/>

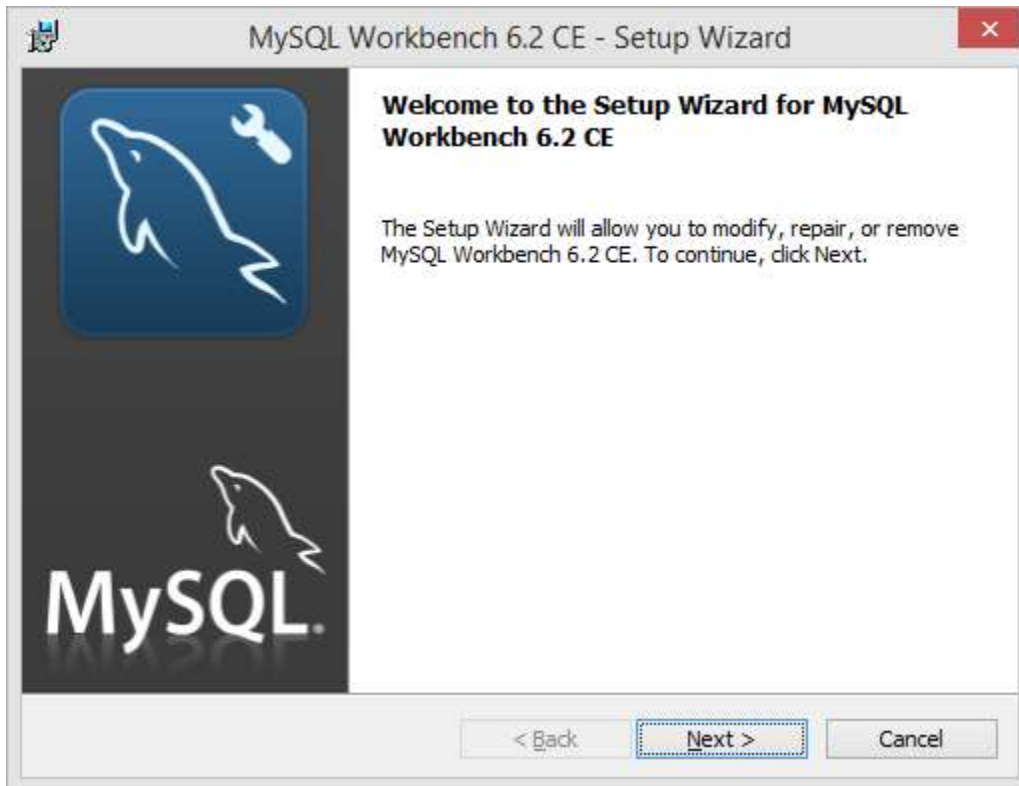


Figura 4.2.4 Instalación de Workbench

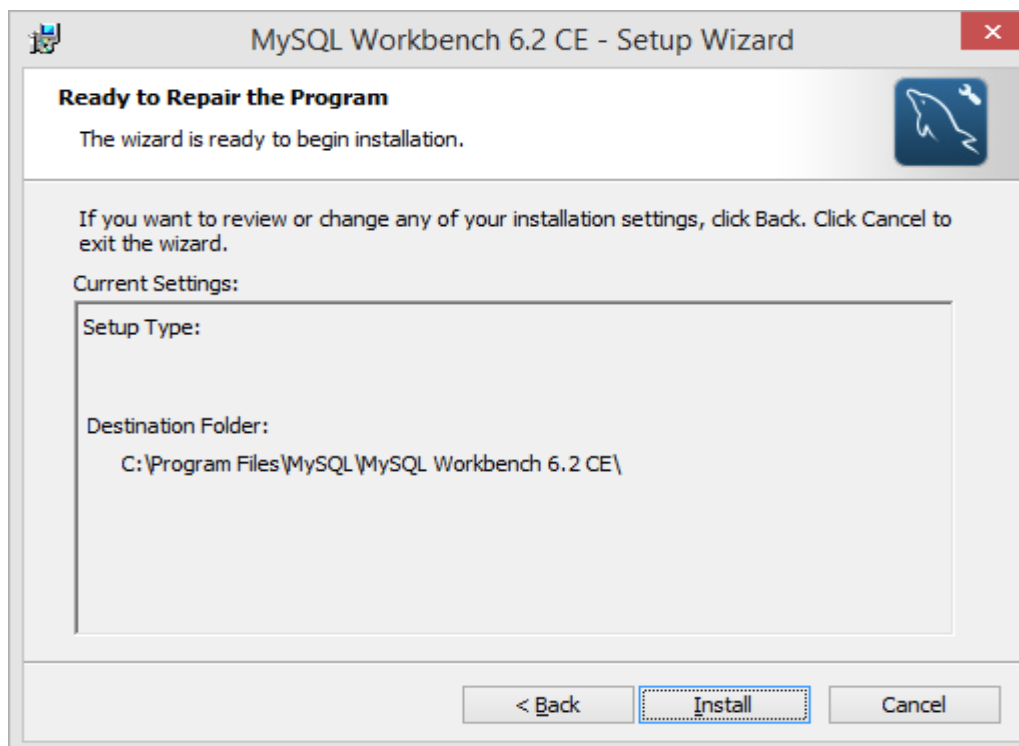


Figura 4.2.5 Directorio de instalación de Workbench

Una vez concluida la instalación, se procederá a su ejecución. Vea figura 4.2.6.

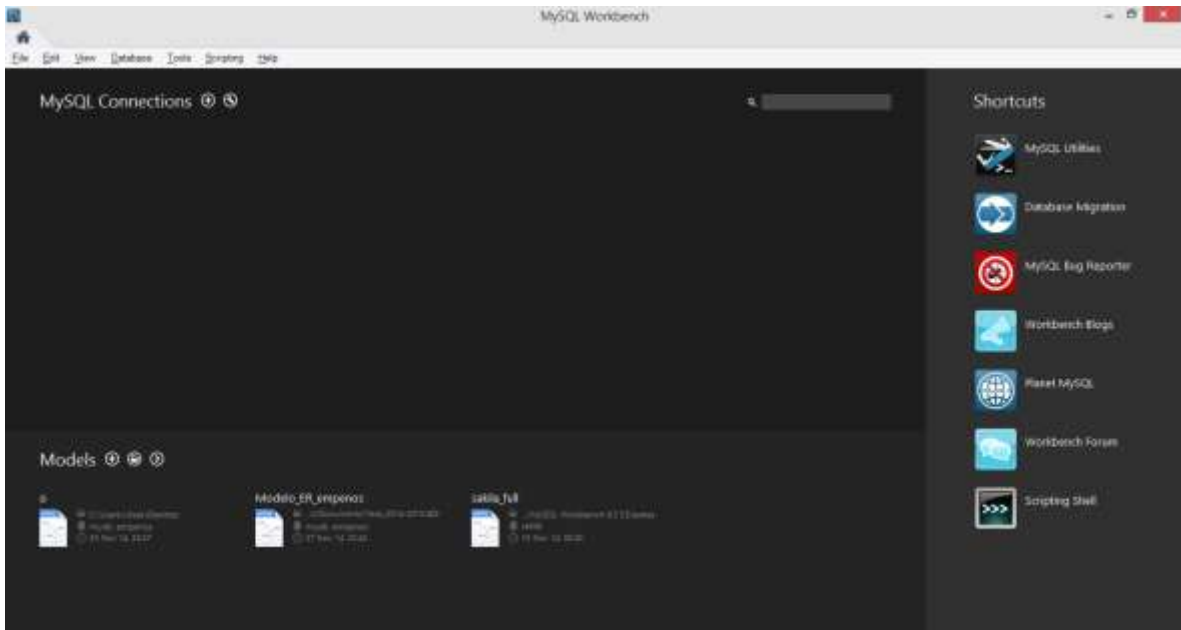


Figura 4.2.6 Ejecución de Workbench

Se establece la conexión con el servidor de base de datos, con la opción de configuración de la herramienta. Vea figura 4.2.7.

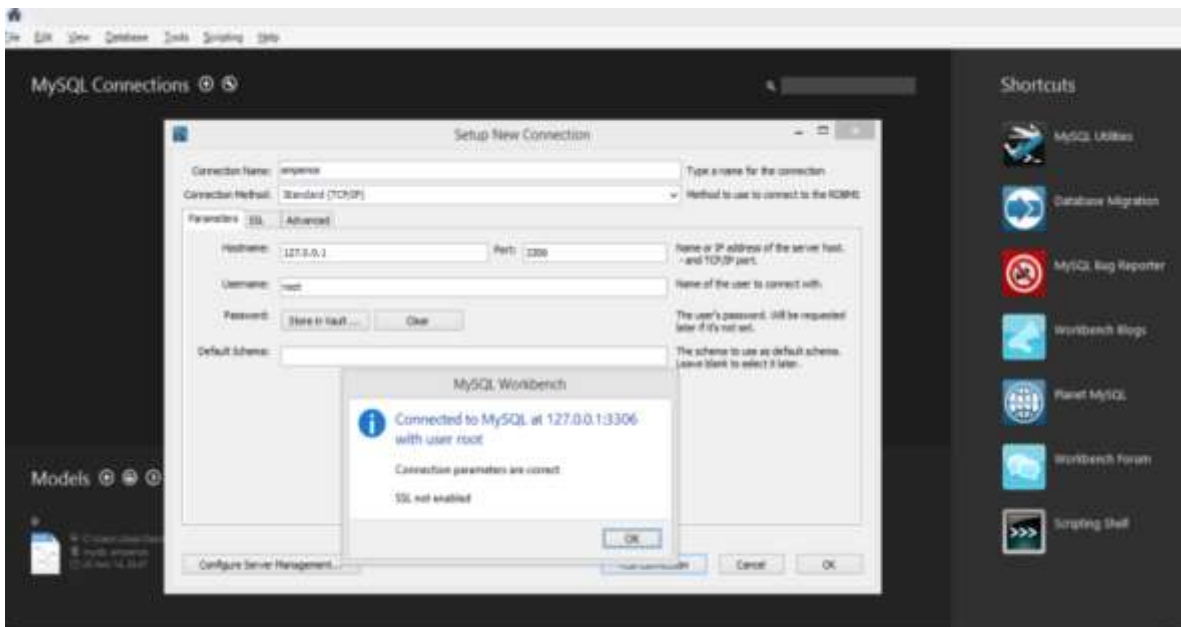


Figura 4.2.7 Conexión a Base de Datos con Workbench

Creación de Base de Datos

Una vez instaladas las herramientas, se realizará la validación de los servicios que estén en funcionamiento.

Acceder a la herramienta WorkBench y realizar la creación de la nueva base de datos “empenos”. Vea figuras de 4.2.8 a 4.2.10.

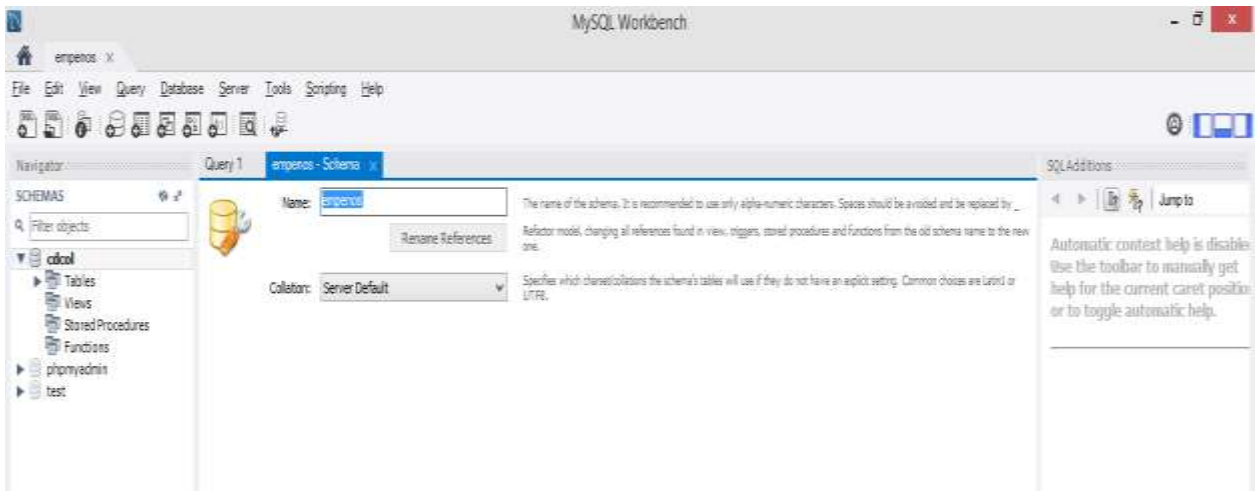


Figura 4.2.8 Creación de base de datos: empenos

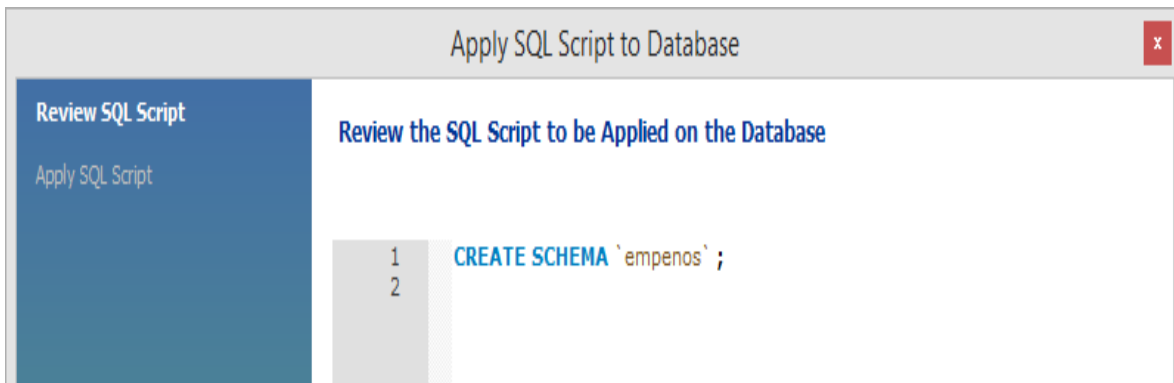


Figura 4.2.9 Script de creación de Base de Datos “empeños”

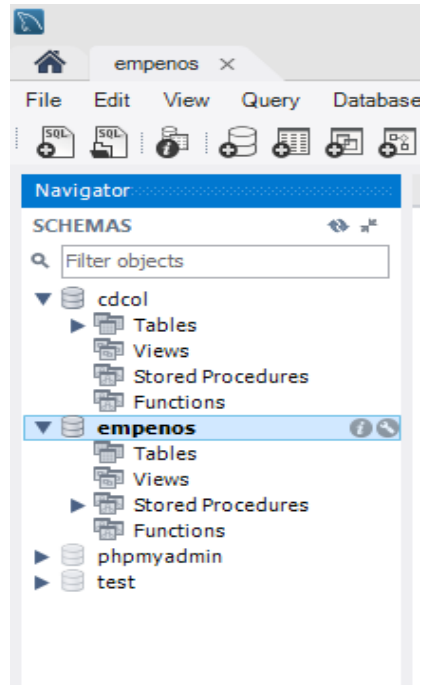


Figura 4.2.10 Base de datos empenos

A continuación se realizará la creación de la tabla “cliente” a partir del diagrama entidad relación planteado con anterioridad. Vea figuras 4.2.11 a la 4.2.13.

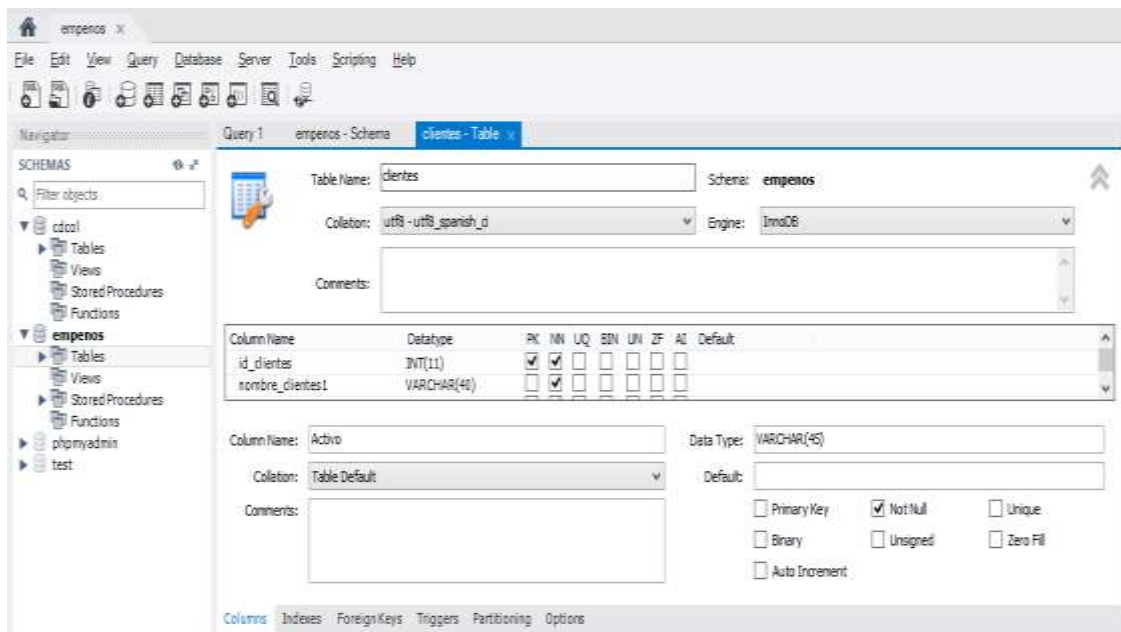


Figura 4.2.11 Creación de la tabla clientes

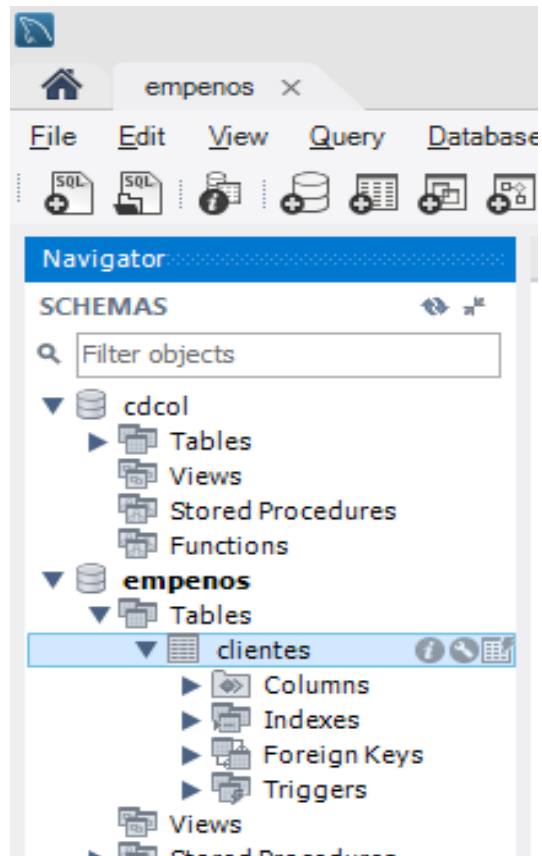


Figura 4.2.12 Creación de la tabla: clientes

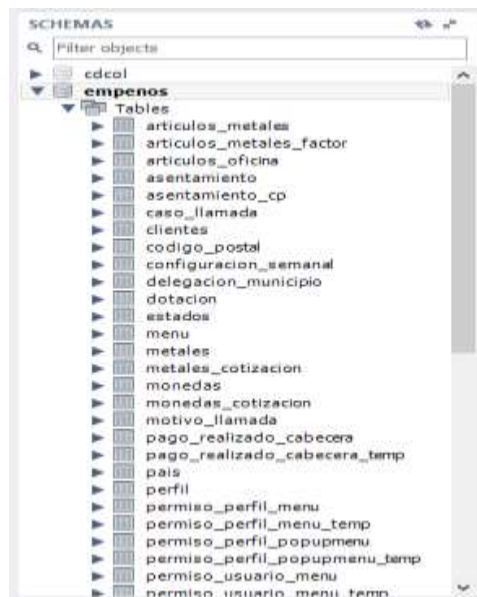


Figura 4.2.13 Árbol de tablas

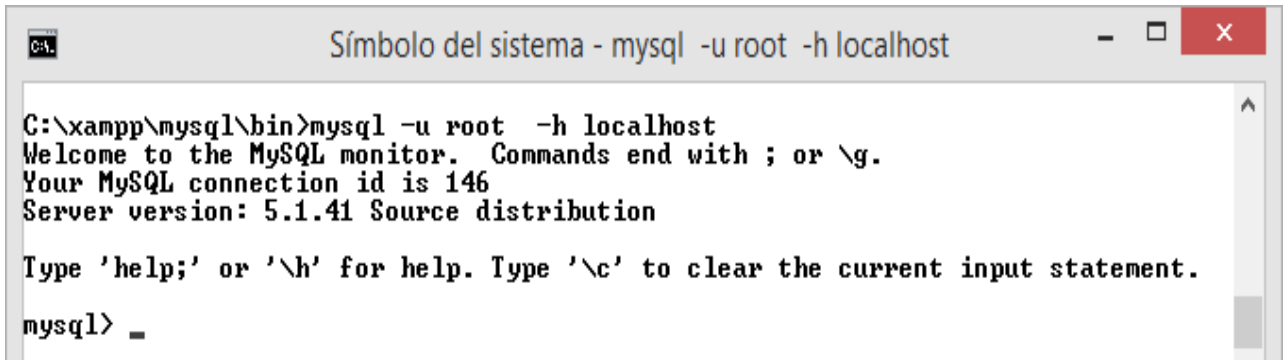
Script para la creación de una tabla

La tabla “menu” será creada a partir de un script por medio de la consola de MySQL.

El proceso de conexión se realizará utilizando la siguiente línea de comandos “mysql -u root -h localhost” donde:

- mysql: Base de datos
- -u: Nombre del usuario
- -h: Nombre de servidor

Una vez creada la conexión a la base de datos se mostrara la consola de MySQL. Vea figura 4.2.14.



```
C:\xampp\mysql\bin>mysql -u root -h localhost
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 146
Server version: 5.1.41 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

Figura 4.2.14 Comando para conectarse a la consola de MySQL

Por medio del siguiente comando “show databases;” se mostrará las bases de datos existentes en MySQL.

Se mostrará la BD “empenos”. Vea figura 4.2.15


```

C:\> Símbolo del sistema - mysql -u root -h localhost
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| empenos |
| mysql |
| phpmyadmin |
| test |
+-----+
6 rows in set (0.00 sec)

mysql> _

```

Figura 4.2.15 Comando para listar base de datos

Se establecerá la conexión a la base de datos “empenos” mediante el siguiente comando “use empenos” y se deberá ejecutar el script en la consola de comandos. Vea figura 4.2.16.

```

C:\> Símbolo del sistema - mysql -u root -h localhost
54 rows in set (0.03 sec)

mysql> CREATE TABLE caso_llamada (
->   Id_Caso_Llamada int(5) NOT NULL auto_increment,
->   Tipo_Caso varchar(50) NOT NULL,
->   Id_Tipo_Llamada int(5) NOT NULL,
->   PRIMARY KEY (Id_Caso_Llamada)
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci AUTO_INCREMENT=7 ;
Query OK, 0 rows affected (0.09 sec)

mysql> select * from caso_llamada
-> ;
Empty set (0.00 sec)

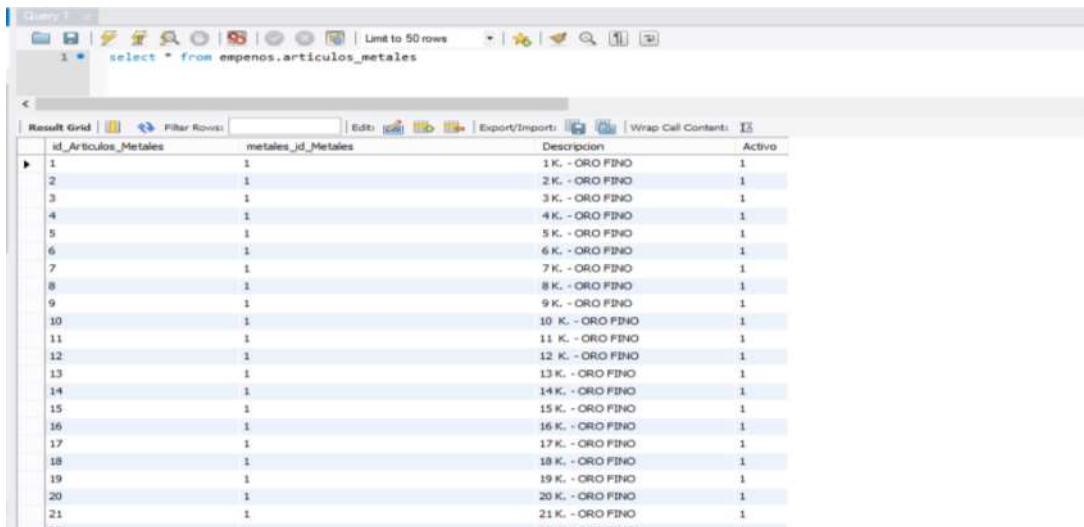
mysql> describe caso_llamada;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Id_Caso_Llamada | int(5)        | NO   | PRI | NULL    | auto_increment |
| Tipo_Caso       | varchar(50)   | NO   |     | NULL    |                |
| Id_Tipo_Llamada | int(5)        | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

```

Figura 4.2.16 Creación de tabla por medio de consola de MySQL

Consulta de información a una tabla

Por medio de la siguiente consulta en SQL “select * from empenos.articulos_metales” se podrá verificar la información de la tabla creada anteriormente. Vea figuras 4.2.17 y 4.2.18.



id_Articulos_Metales	metales_id_Metales	Descripción	Activo
1	1	1 K. - ORO FINO	1
2	1	2 K. - ORO FINO	1
3	1	3 K. - ORO FINO	1
4	1	4 K. - ORO FINO	1
5	1	5 K. - ORO FINO	1
6	1	6 K. - ORO FINO	1
7	1	7 K. - ORO FINO	1
8	1	8 K. - ORO FINO	1
9	1	9 K. - ORO FINO	1
10	1	10 K. - ORO FINO	1
11	1	11 K. - ORO FINO	1
12	1	12 K. - ORO FINO	1
13	1	13 K. - ORO FINO	1
14	1	14 K. - ORO FINO	1
15	1	15 K. - ORO FINO	1
16	1	16 K. - ORO FINO	1
17	1	17 K. - ORO FINO	1
18	1	18 K. - ORO FINO	1
19	1	19 K. - ORO FINO	1
20	1	20 K. - ORO FINO	1
21	1	21 K. - ORO FINO	1

Figura 4.2.17 Consulta de información por medio de por medio de Workbench

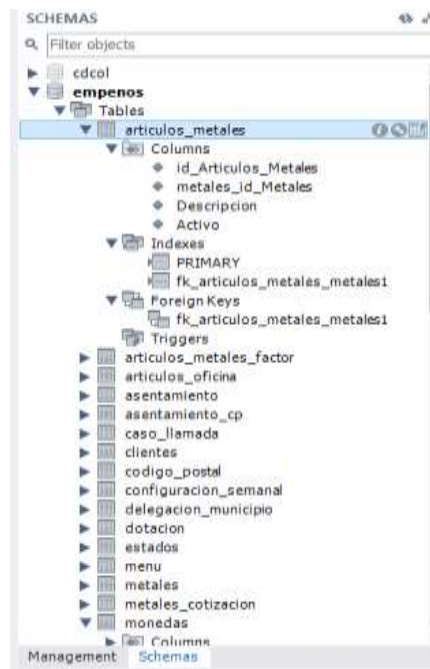
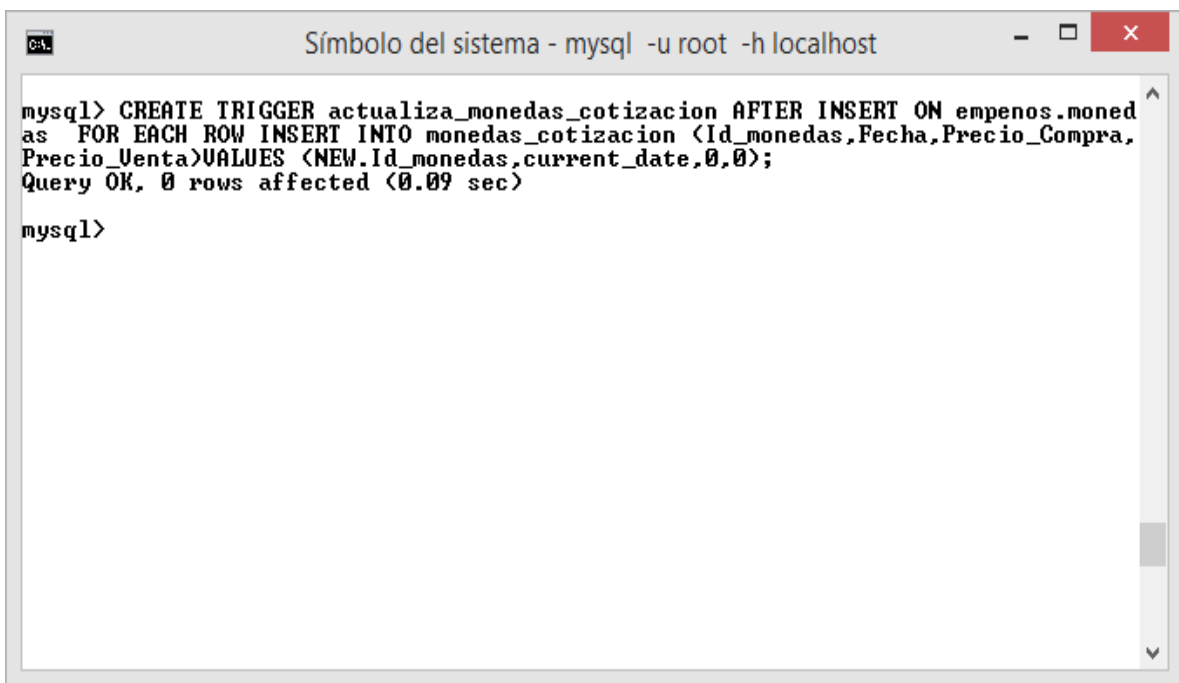


Figura 4.2.18 Árbol de consultas por medio de Workbench

Creación de un trigger

Un **trigger** (disparador) en una base de datos, es un procedimiento que es ejecutado **Before**(antes) y/o **After**(después)de cumplirse alguno(s) de los siguientes eventos como son: actualización, borrado, e inserción a la tabla asociada al trigger.

La siguiente sentencia creará un disparador para sentencias de inserción dentro la tabla “monedas”, y también realizará una inserción en la tabla “monedas_cotizacion”. Vea figura 4.2.19.

A screenshot of a terminal window titled "Símbolo del sistema - mysql -u root -h localhost". The terminal shows a MySQL prompt where a CREATE TRIGGER statement is entered and executed. The statement is: `CREATE TRIGGER actualiza_monedas_cotizacion AFTER INSERT ON empenos.monedas FOR EACH ROW INSERT INTO monedas_cotizacion (Id_monedas, Fecha, Precio_Compra, Precio_Venta) VALUES (NEW.Id_monedas, current_date, 0, 0);`. The output of the query is "Query OK, 0 rows affected (0.09 sec)". The prompt returns to "mysql>".

```
mysql> CREATE TRIGGER actualiza_monedas_cotizacion AFTER INSERT ON empenos.monedas
FOR EACH ROW INSERT INTO monedas_cotizacion (Id_monedas, Fecha, Precio_Compra,
Precio_Venta)VALUES (NEW.Id_monedas, current_date, 0, 0);
Query OK, 0 rows affected (0.09 sec)
mysql>
```

Figura 4.2.19 Creación de disparador por medio de consola de MySQL

Por medio de la herramienta Workbench se podrá visualizar la existencia del disparador previamente creado. Vea figuras 4.2.20 y 4.2.21.

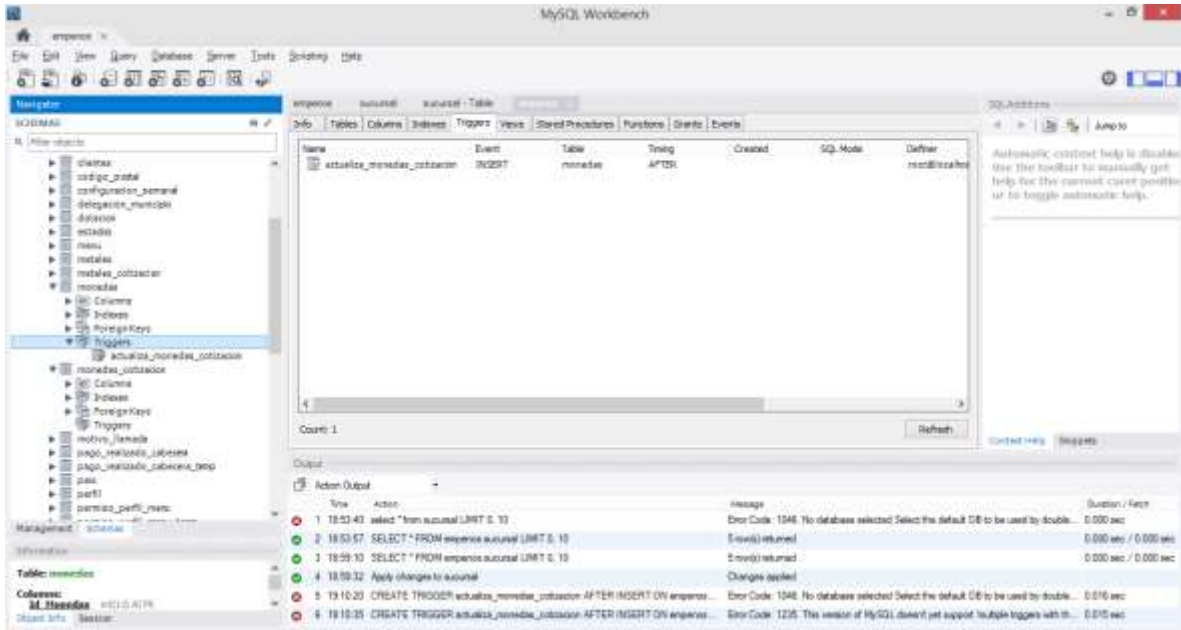


Figura 4.2.20 Visualización de disparador por medio de Workbench

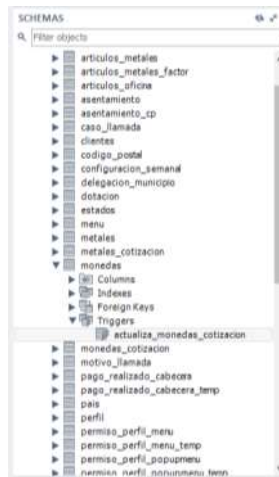


Figura 4.2.21 Árbol de disparador por medio de Workbench

Creación de un procedimiento almacenado

Un procedimiento almacenado es un conjunto de comandos SQL, adicionalmente contiene una serie de estructuras de control que implementan lógica a este, una vez creado y guardado en el servidor de base de datos, podrá ser ejecutado de manera directa en el motor de base de datos, o ser accedido a través de llamadas por aplicaciones u otros procedimientos almacenados.

La creación del procedimiento almacenado se podrá realizar desplegando el árbol de la base de datos “empenos” →Stored Procedures, dar clic con botón izquierdo Create Store Procedure. Vea figura 4.2.22.

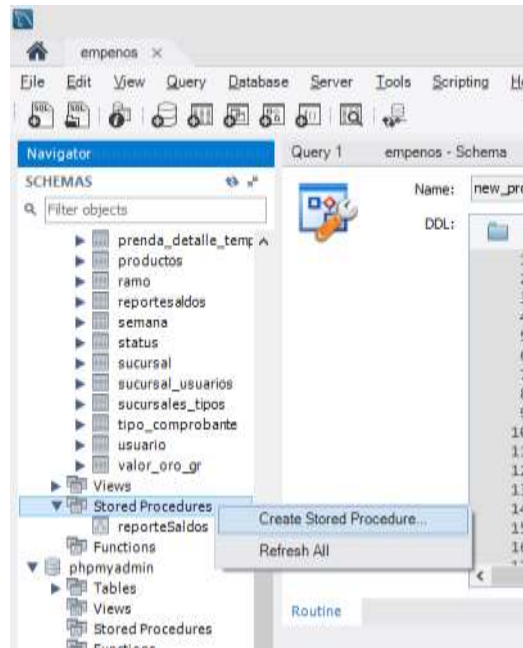


Figura 4.2.22 Creación de procedimiento almacenado en Workbench

Se desplegará una nueva ventana, en la cual se deberá ingresar el script del procedimiento almacenado. Vea figura 4.2.23.

```

1 CREATE DEFINER='root'@'localhost' PROCEDURE `pagos` (IdBoleta INT,IdSucursal INT,IPagos INT,Desempeno INT,SoloInteres INT,Cobrar INT)
2 BEGIN
3     DECLARE Fecha_Fundición DATE;
4     DECLARE Subtotal,Total,VInteres,WVorosidad DOUBLE(10,2) DEFAULT 0;
5     DECLARE control,Contador,Punto,IdVencidos,IdPendientes,IdPenven,IdPagados,PagoUnico,Semana INT DEFAULT 0;
6     DECLARE CursorVencidos CURSOR FOR SELECT Id_Plan_Pago_Detalle FROM plan_pago_detalle WHERE Status=1 AND Fecha<CURRENT_DATE AND Id_Boleta_Detalle=IdBoleta 0
7     DECLARE CursorPendientes CURSOR FOR SELECT Id_Plan_Pago_Detalle FROM plan_pago_detalle WHERE Status=1 AND Id_Boleta_Detalle=IdBoleta ORDER BY Numero_Pago;
8     DECLARE CursorPenVen CURSOR FOR SELECT Id_Plan_Pago_Detalle FROM plan_pago_detalle WHERE CURRENT_DATE<=Fecha AND Status=1 AND Id_Boleta_Detalle=IdBoleta OR
9     DECLARE CursorPagados CURSOR FOR SELECT Id_Plan_Pago_Detalle FROM plan_pago_detalle WHERE (Status=1 OR Status=0) AND Id_Boleta_Detalle=IdBoleta ORDER BY Nu
10    DECLARE CONTINUE HANDLER FOR NOT FOUND SET control = 1;
11    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN ROLLBACK;SELECT 'Ocurrió un error en la operacion';END;
12    SELECT pago_unico INTO PagoUnico FROM pago_realizado_cabecera,productos WHERE Tipo_Producto=Id_Productos AND Id_Boleta_Cabecera=IdBoleta;
13    SELECT (Cobrar*Aplica_Interes_Moratorio*Interes_Moratorio/100) INTO WVorosidad FROM pago_realizado_cabecera,productos WHERE Tipo_Producto=Id_Productos AND
14    SELECT (Cobrar*Aplica_Interes_Referendo*Interes_Referendo/100) INTO VInteres FROM pago_realizado_cabecera,productos WHERE Tipo_Producto=Id_Productos AND Id
15    SELECT a_semana INTO Semana FROM pago_realizado_cabecera,productos WHERE Tipo_Producto=Id_Productos AND Id_Boleta_Cabecera=IdBoleta;
16    START TRANSACTION;
17    IF(SoloInteres=1)THEN
18        IF(PagoUnico=1)THEN
19            SELECT Interes_Total INTO Subtotal FROM pago_realizado_cabecera WHERE Id_Boleta_Cabecera=IdBoleta;
20            IF(Desempeno=1)THEN
21                SELECT COUNT(Id_Plan_Pago_Detalle) INTO Contador FROM plan_pago_detalle WHERE Status=1 AND Fecha<CURRENT_DATE AND Id_Boleta_Detalle=IdBoleta;
22                SET IPagos=IPagos+Contador;
23            ELSE
24                SELECT COUNT(Id_Plan_Pago_Detalle) INTO Contador FROM plan_pago_detalle WHERE Status=1 AND Id_Boleta_Detalle=IdBoleta;
25                SET IPagos=Contador;
26            END IF;
27            IF(IPagos=1)THEN
28                SET control=1;
29            END IF;

```

Figura 4.2.23 Script de procedimiento almacenado en Workbench

Se deberá seleccionar la opción aplicar para crear el procedimiento almacenado “pagos” en la base de datos “empenos”. Vea figuras 4.2.24 y 4.2.25.

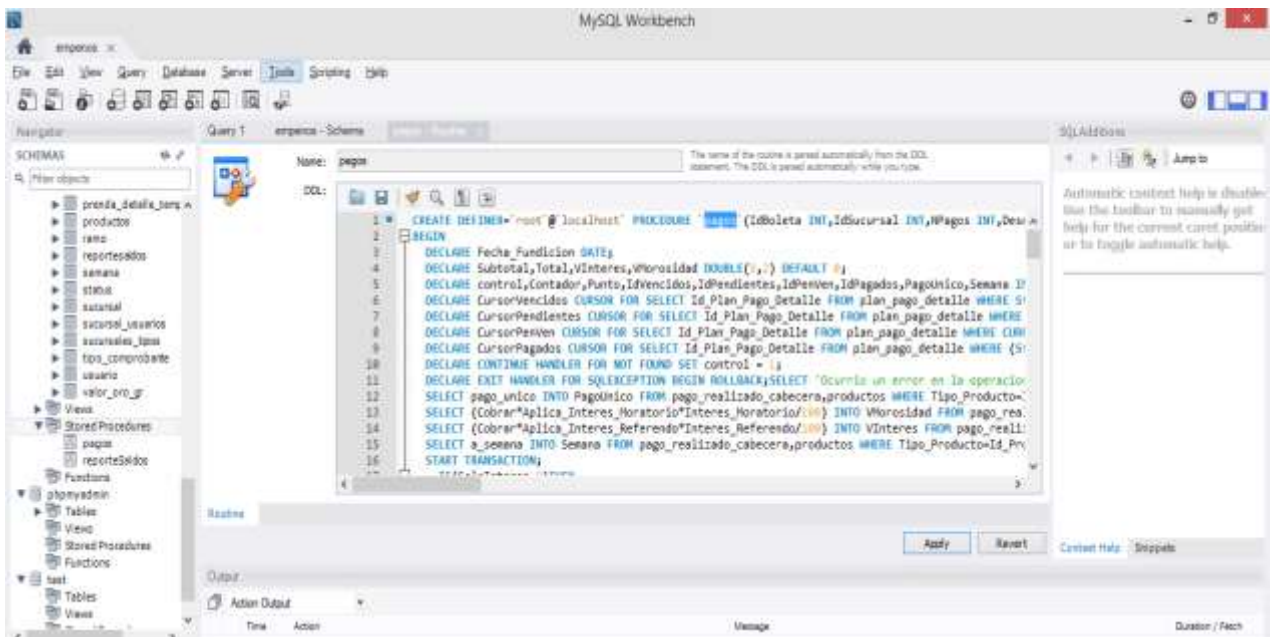


Figura 4.2.24 Procedimiento almacenado creado

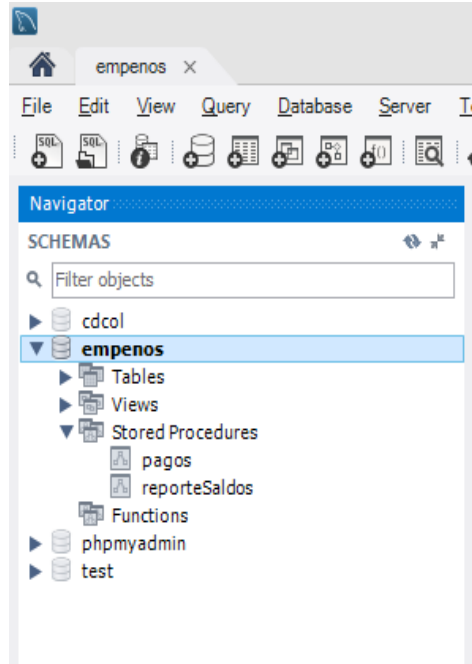


Figura 4.2.25 Árbol de procedimientos almacenados por medio de Workbench

4.3 Desarrollo del front-end del SICE

Para acceder al sistema SICE, es necesario tener una URL (Uniform Resource Locator) de acceso, y una cuenta con contraseña. Los usuarios tendrán perfiles dentro del sistema de acuerdo a sus labores.

El sistema validará que el usuario y contraseña estén registrados dentro de la base de datos, de esta manera se controla la información que se le mostrará para cada usuario de acuerdo al tipo de perfil que le fue asignado.

La pantalla de acceso al sistema estará conformada por dos campos: usuario y contraseña, los cuales tienen que ser llenados para poder ingresar al sistema. Vea figura 4.3.1.

A dark blue rectangular box containing a login form. It has two white input fields: the first is labeled 'Usuario:' and the second is labeled 'Contraseña:'. Below these fields is a white button with the text 'INGRESAR' in blue capital letters.

Figura 4.3.1 Pantalla de acceso al SICE

Una vez que se haya logrado acceder al sistema se mostrará el menú de los módulos que componen el sistema. Estos dependerán del perfil que se le haya asignó al usuario. Vea figura 4.3.2.



Martes, 10 de Marzo de 2015 17:14:12



Figura 4.3.2 Módulos que conforman el SICE

Módulo de Catálogos

Este módulo permitirá realizar las siguientes acciones: crear y modificar las sucursales, los tipos de sucursales y la asignación de usuarios a las sucursales, como puede verse en la figura 4.3.3. Para acceder a las diferentes tareas que contiene este módulo basta con dar clic en un vínculo correspondiente.



Figura 4.3.3. Módulo catálogos

Crear sucursal

Para crear una nueva sucursal en el sistema, se deberá de llenar los campos que se muestran dentro de la figura 4.3.4. Esta pantalla contará con un menú llamado "Tipo", vea figura 4.3.5. Este catálogo "Tipo" permitirá elegir el tipo de sucursal. Si se selecciona como tipo Concentradora, se deberá elegir que tiendas pertenecerán a esta concentradora, vea figura 4.3.6.



Figura 4.3.4 Crear sucursal

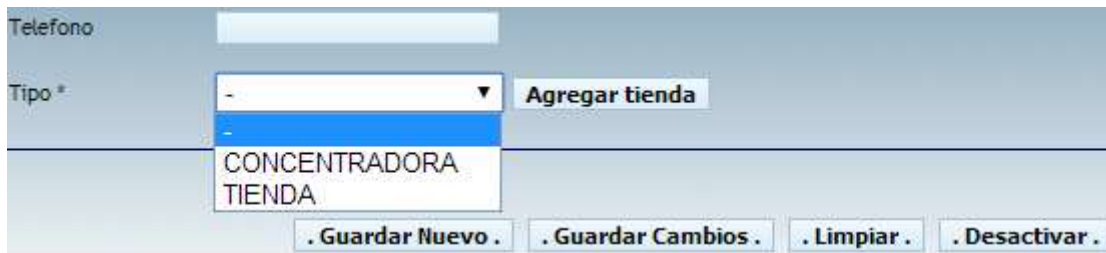


Figura 4.3.5 Catálogo tipo de sucursal



Figura 4.3.6 Lupa búsqueda de sucursal

Alta de usuario

Para crear un nuevo usuario dentro del SICE, se deberá seleccionar el catalogo mantenimiento y seleccionar la opción usuarios, vea figura 4.3.7. Una vez dentro de la pantalla de “Usuarios” se tendrá que ingresar todos los campos que son requeridos en esta pantalla (los cuales se denotan con un ‘*’). Vea figuras 4.3.8., 4.3.9.



Figura 4.3.7 Catalogo mantenimiento

The image shows a form titled 'Nuevo Usuario' with a 'Salir' link on the right. The form contains the following fields:

- Nombre Completo del Usuario*: Arlene Acevedo Zermeno
- Nombre de Usuario (login)*: arleneacevedo
- Contraseña*: [masked with asterisks]
- Confirma la Contraseña*: [masked with asterisks]
- Correo Electrónico: arlene_@hotmail.com
- Departamento: Sucursal
- Este usuario se creará: [dropdown menu with options: 'Configurar todos los permisos' (selected), 'Configurar todos los permisos Basandose en un Perfil existente']

Figura 4.3.8 Alta de usuario

Contraseña*:
 Confirma la Contraseña*:
 Correo Electrónico:
 Departamento:
 Este usuario se creará:

Selecciona el Perfil:

-
- ADMINISTRADOR
- CAJERO
- GERENTE
- VALUADOR

SICE

4.3.9 Catálogo seleccionar perfil ya existente

Modificar perfil

Para modificar un perfil existente dentro del SICE, se deberá seleccionar el catálogo mantenimiento y elegir la opción perfiles, vea figura 4.3.7. Dentro de la pantalla de perfiles, se deberá seleccionar la opción editar perfil, y elegir el perfil a modificar, el sistema mostrará los módulos a los que tiene acceso el perfil seleccionado, como puede verse en la figura 4.3.10. Para agregar el acceso a un módulo se deberá de elegir dentro del menú de módulos y accionar el botón aceptar. Vea figura 4.3.11. Para quitar el acceso a un módulo se debe de dar clic en la acción borrar del módulo seleccionado. Vea figura 4.3.12

Permisos para el Perfil ADMINISTRADOR en Menú

Mapa de Navegacion	Elige el menu y despues da click en agregar:	-	Agregar
ADMINISTRADOR	Menús	REPORTES ATENCION	Modificar
	CATALOGOS		[Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono]
	OPERACIONES		[Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono]
	ADMIN		[Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono]
	MANTENIMIENTO		[Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono]
	SALIR		[Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono] [Icono]

Figura 4.3.10 Módulo de un perfil seleccionado



Figura 4.3.11 Agregar módulo a un perfil



Figura 4.3.12 Quitar módulo a un perfil

Empeños

Para realizar el empeño de una prenda dentro del sistema se deberá de elegir el catálogo "Operaciones", después seleccionar la opción empeños y elegir la sucursal se estará realizando el empeño. El sistema mostrará la pantalla para capturar las características de la prenda, como se puede ver en la figura 4.3.12.

En esta pantalla cuenta con tres catálogos:

- Metal.- En la cual se puede seleccionar el metal de la prenda.
- Pieza.- En la cual se puede especificar qué tipo de pieza es.
- Artículo metal.- En la que se selecciona la calidad del metal, quilates o porcentaje de pureza.

Vea figuras 4.3.13, 4.3.14 y 4.3.15 respectivamente.

Operaciones							
Reportes							
Atención							
Salir							
Empeños							Salir
Sistema Integral para Casas de Empeños (SICE)							
Atiende : Ricardo Gómez Ramírez				Oro fino \$	588.71	/gr	
				Plata fina\$	7.77	/gr	
				Fecha:	11-03-2015		
Metal	Pieza	Artículo Metal	Gramos	Piedras gr	Observaciones	Precio	Agregar
Seleccione ▼	Seleccione ▼	Seleccione artículo ▼	<input type="text"/>	<input type="text"/>	<input type="text"/>	\$0.00	<input style="color: green;" type="button" value="+"/>

Figura 4.3.12 Pantalla de empeño de una prenda

Metal

ORO ▼

Seleccione

ORO

PLATA

Figura 4.3.13 Catálogo quilates

Pieza

Seleccione ▼

Seleccione

ANILLO

ARETES

ARRACADA

BRAZALETES

CADENA

COLLAR

DIJE

ESCLAVA

GARGANTILLA

MEDALLA

MONEDA

PULSERA

RELOJ

SEMANARIO

Figura 4.3.14 Catálogo pieza

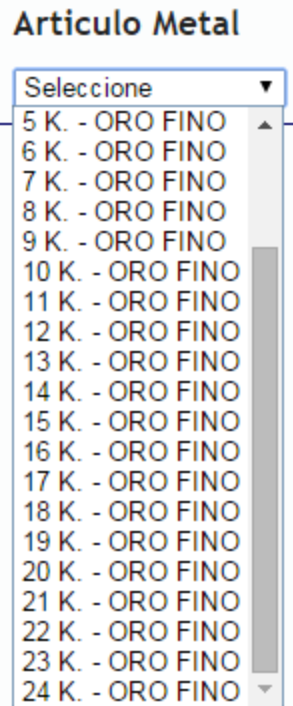


Figura 4.3.15 Catálogo artículo metal

Para poder empeñar una prenda se deberá de capturar los datos del pignorante, el sistema permite por medio de un catálogo seleccionar:

- Un cliente ya existente
- Un cliente nuevo
- Un cliente genérico

Sí, el pignorante es nuevo se deberá de capturar en el sistema sus datos, para después almacenarlos en la base datos. Vea figura 4.3.16.

Captura del Cliente ✕

Nombre(s) *

Apellidos *

Calle y Número *

C.P. *

Colonia

Nueva Colonia

Delegación / Municipio *

Estado *

Tipo de Identificación

Figura 4.3.16 Alta de nuevo cliente

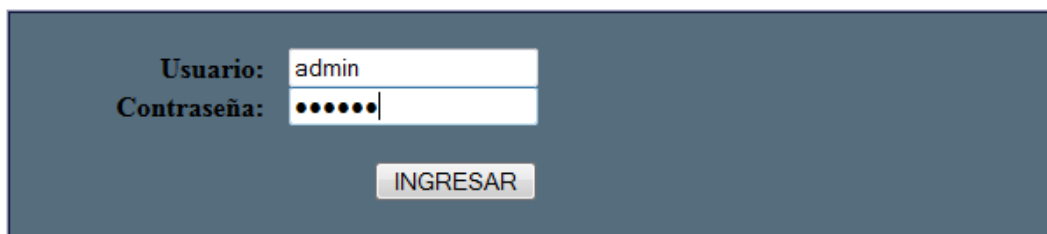
4.4 Generación de pruebas y mantenimiento

Una parte esencial dentro del desarrollo del sistema SICE consiste en el ciclo de pruebas, en la cual se incluyen actividades enfocadas a llevar al sistema a producir fallas con el propósito de eliminar errores antes de tener una versión final. Existe una gran cantidad de escenarios para un sistema y para cada una de los escenarios se pueden realizar pruebas que ayuden a determinar y mejorar la calidad final del sistema. Con el fin de garantizar la funcionalidad del sistema SICE, se realizaron las siguientes pruebas:

Pruebas de caja negra

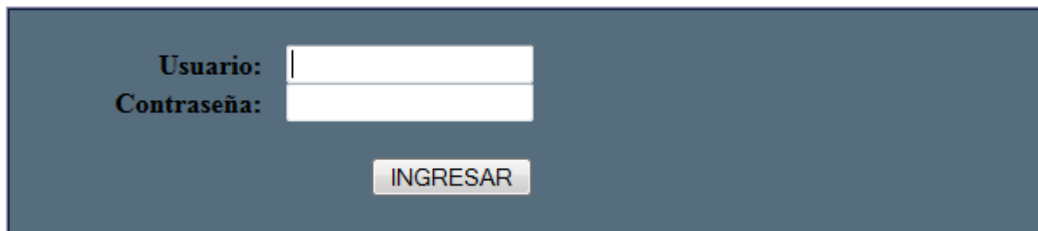
Este tipo de pruebas son aplicadas sobre el Front-End del sistema y se realizan sin conocer la estructura o el funcionamiento de cada componente interno del sistema. Las pruebas consistieron en comprobar el funcionamiento del proceso de autenticación inicial para acceder al sistema.

A continuación se muestra la pantalla de validación de usuario, en la cual se ingresa un usuario no valido para el sistema, y el mensaje del sistema indica que hay un error en la autenticación. Vea figuras 4.4.1 y 4.4



The image shows a login interface on a dark blue background. It features two input fields: the first is labeled 'Usuario:' and contains the text 'admin'; the second is labeled 'Contraseña:' and contains six black dots. Below these fields is a button labeled 'INGRESAR'.

Figura 4.4.1 Accediendo al sistema con usuario no válido



Usuario:

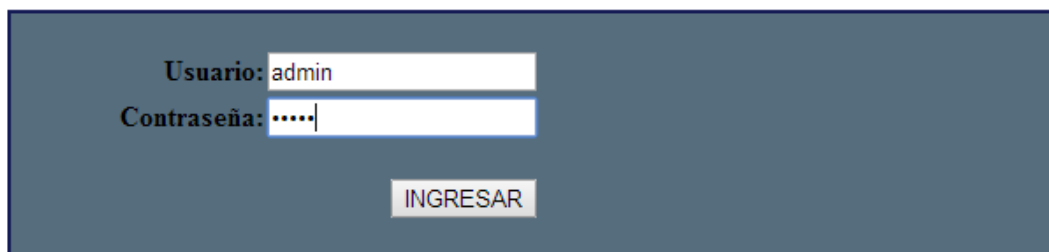
Contraseña:

INGRESAR

**Error de Autenticación.
Intente de nuevo.**

Figura 4.4.2 Mensaje de usuario no válido

Cuando en la pantalla de autenticación se ingresa un usuario válido, después de dar clic en aceptar, se muestra la pantalla con los módulos a los que tiene permiso de acceder el usuario en el SICE. Vea figuras 4.4.3 y 4.4.4.



Usuario:

Contraseña:

INGRESAR

Figura 4.4.3 Accediendo al sistema con usuario válido

Martes, 10 de Marzo de 2015 17:12:03



Figura 4.4.4 Pantalla de inicio del sistema SICE

Una vez dentro del sistema se podrá seleccionar catálogos, sucursal y búsqueda por código de sucursal. El sistema mostrará un mensaje indicando que no se encontró ninguna sucursal, como puede verse en la figura 4.4.5. En cambio sí, se busca un código de sucursal válido el sistema muestra información de la sucursal. Vea figura 4.4.6.

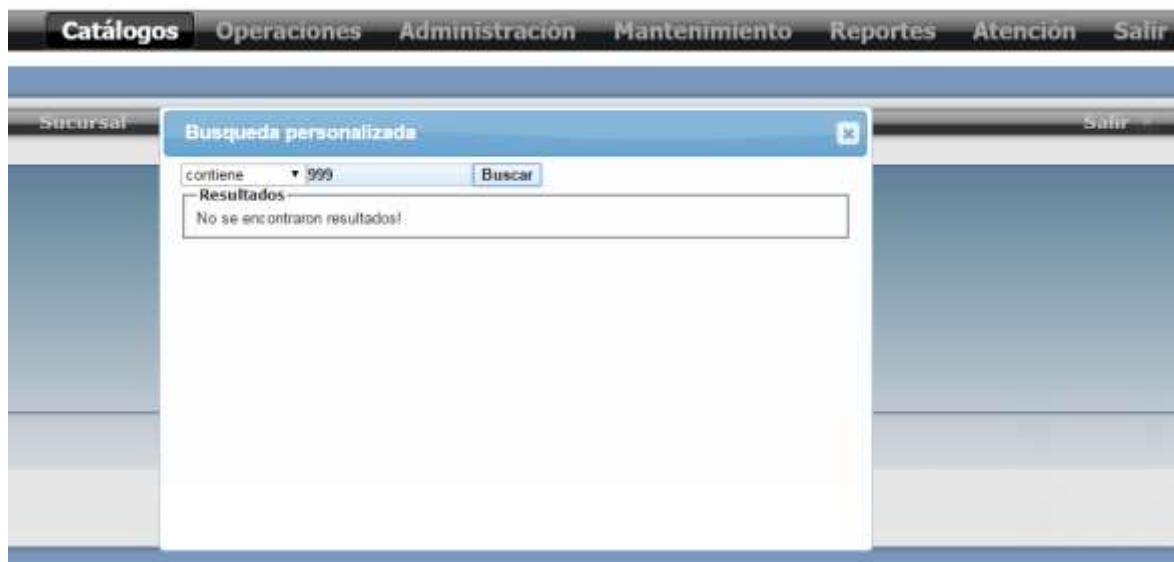


Figura 4.4.5 Búsqueda por código de sucursal inválido



Figura 4.4.6 Búsqueda por código de sucursal válido

Pruebas de caja blanca

Este tipo de pruebas se basan en los datos del software, por lo cual dependen directamente del código fuente ya que se conoce este. Para el SICE se aplicó esta prueba, en el módulo de empeños al momento de generar la nota con la cual el pignorante podrá refrendar su prenda, para asegurar que la nota de la prenda a empeñar cumple con los requerimientos dados por el usuario. Vea figuras 4.4.7 y 4.4.8.



PRESTAMO TRADICIONAL

NÚMERO DE BOLETA: 78
 NÚMERO DE ENVASE: 000000074
 FECHA:
 11/03/2015

SUCURSAL Y DIRECCIÓN:
 SUCURSAL TOLUCA ESTADO DE MÉXICO

NOMBRE: Gabriel David Corona # DE CLIENTE
 Zamora 3
 DOMICILIO: Norte 16, No. 7 MARIA ISABEL VALLE DE CHALCO MEXICO
 COTITULAR: Arlene Acevedo Zermeño
 Num identificación C.P. TELÉFONO: CELULAR: EMAIL:
 123456789abc 56615 59868407 3529689425 gabriel@gmail.com

MATERIAL/KILATAJE	PESO (gr)	AVALUO	PRESTAMO	OBSERVACIÓN
14 K. - ORO FINO	5.00	\$ 1,578.28	\$ 1,515.15	
TOTAL >>>>>	5.00	\$ 1,578.28	\$ 1,515.15	

LA PRENDA DESCRITA GARANTIZA EL CUMPLIMIENTO DEL CONTRATO PRENDARIO, EL CLIENTE RECIBIRÁ UN TOTAL DE: >>>>>

\$ 1,487.53

DETALLES DE PRESTAMO

AVALUO TOTAL	PLAZO	FECHA DE VENCIMIENTO	TOTAL A PAGAR
1,578.28	4 semanas	08/04/2015	\$ 1,657.59
ALMACENAJE y SEGURO	CAT	TAZA INTERES SEMANAL	FECHA COMERCIALIZACIÓN
21.8751 %	148.6220 % tasa anual	2.3500000 %	15/04/2015

DETALLES DE PAGOS

PERIODO/SEMANA	VENCIMIENTO	FECHA	PRESTAMO	REFRENDO	PAGO TOTAL
0		11/03/2015	\$ 0.00	\$ 0.00	\$ 0.00
1		18/03/2015	\$ 1,515.15	\$ 35.61	\$ 1,550.76
2		25/03/2015	\$ 1,515.15	\$ 71.22	\$ 1,586.37
3		01/04/2015	\$ 1,515.15	\$ 106.83	\$ 1,621.98
4	08/04/2015	08/04/2015	\$ 1,515.15	\$ 142.44	\$ 1,657.59

VENCIMIENTOS EN DOMINGOS Y DÍAS FESTIVOS SE RECORREN AL DÍA HÁBIL INMEDIATO ANTERIOR.

AVISO IMPORTANTE

SI EL CLIENTE NO PAGA EN LA FECHA DE VENCIMIENTO, EL CLIENTE VA A TENER 3 (TRES) DÍAS HÁBILES PARA PAGAR LO QUE DEBE MÁS UNA TAZA DE PENALIDAD DE 1% SEMANAL SOBRE EL VALOR DE EL PRESTAMO, SI EL CLIENTE NO PAGA 3 (TRES) DÍAS HÁBILES DESPUÉS DE LA FECHA DE VENCIMIENTO, SE APLICARA LA CLÁUSULA DE DACIÓN EN PAGO.

PERSONA AUTORIZADA POR EL DEUDOR:

EN CASO DE NO PODER PRESENTARME A RECUPERAR LA PRENDA OTORGADA EN GARANTÍA, AUTORIZO AL SEÑALADO COMO COTITULAR, PARA QUE A MÁS TARDAR EN LA FECHA DE VENCIMIENTO, LO HAGAN MI NOMBRE Y POR MI CUENTA, PREVIO EL PAGO DE MI ADEUDO Y CONTRA LA ENTREGA DEL PRESENTE CONTRATO Y PRESENTACIÓN DE SU IDENTIFICACIÓN OFICIAL. EL DEUDOR PRENDARIO EXPRESA QUE CONOCE Y ESTÁ CONFORME CON LAS DECLARACIONES Y CLÁUSULAS DEL PRESENTE CONTRATO QUE SE CONTIENEN EN EL REVERSO DE ESTE DOCUMENTO, Y QUE SE LE HA INFORMADO EL COSTO ANUAL TOTAL. EL DEUDOR PRENDARIO RECIBE EN EL ACTO Y A SU ENTERA SATISFACCIÓN LA PRENDA ARRIBA DESCRITA.

FIRMA DEL CLIENTE	FIRMA DE NOTARIO PÚBLICO	FIRMA DEL CLIENTE
Gabriel David Corona Zamora	Ricardo Gómez Ramírez	
Acepto	Valido	Acepto



Figura 4.4.7 Generación de la nota de un empeño

```

case 'generaNota':
$objAJAP = New AdminAJAP(New Configuracion());
$fecha= $_POST['fecha'];
$fecha=date("Y-m-d",mktime(0, 0, 0, $fecha[3].$fecha[4],$fecha[0].$fecha[1], $fecha[6].$fecha[7].$fecha[8].$fecha[9]));
$f2=date("Y-m-d",mktime(0, 0, 0, $fecha[5].$fecha[6],($fecha[8].$fecha[9])+8, $fecha[0].$fecha[1].$fecha[2].$fecha[3]));
$maximo=$_POST['maximo'];
$minimo=$_POST['minimo'];
$horafecha=date("Y-m-d H:i:s");
$porciento=$_POST['porciento'];
$sql2="SELECT Importe,Precio_Compra FROM compra_temp ";
$suma = $objAJAP->ejecutar($sql2);
if($suma) $tot=$suma->fetch_array(MYSQLI_NUM);
$importetotal=$tot[0];
$total=($importetotal*100)/$porciento;
$preciocompra=$tot[1];
$gramos=0;
$totcorrida=0;
$cantidad=$total/$preciocompra;
$sql="UPDATE compra_temp SET Cantidad = '$cantidad', Precio_Compra = '$preciocompra', Importe = '$total';";
$objAJAP->ejecutar($sql);
$falta=$total;
$boletasminimas=$total/$maximo;
$numcli=comboCuenta();
$numcli=count($numcli);
$user=$_SESSION['user'];
$suc=$_POST['suc'];
$sql="SELECT idsucursal FROM sucursal WHERE descripcion = '$suc';";
$result = $objAJAP->ejecutar($sql);
$s=$result->fetch_array(MYSQLI_NUM);
$suc=$s[0];
if($maximo>$total){
    $sql="TRUNCATE TABLE compra_temp";
    $objAJAP->ejecutar($sql);
    $sql="TRUNCATE TABLE clientes_temp";
    $objAJAP->ejecutar($sql);
    echo "alert(\"El monto maximo no puede ser mayor que el importe total al 100%\");";
}elseif($numcli<$boletasminimas){

```

Figura 4.4.8 Código para nota de un empeño

Pruebas unitarias

Una prueba unitaria consiste en comprobar el correcto funcionamiento de cada módulo del sistema. En el sistema SICE se probó de manera unitaria el combo semanas a pagar, en el cual se selecciona las semanas a pagar del producto

empeñado y el sistema calcula para los dos tipos de empeños que ofrece los valores de; Tipo de pago, Préstamo, Almacenaje, Seguro, Cliente recibe, Interés y Pago total. Vea figuras 4.4.9 y 4.4.10.

Figuras 4.4.9 Selección del combo de semanas a pagar

Tipo de pago	Préstamo	Almacenaje	Seguro	Cliente recibe	Interés	Pago total
SEMANTAL	\$ 7,784.92	\$ 60.82	\$ 81.09	\$ 7,643.01	\$ 467.08	\$ 8,252.00
TRADICIONAL	\$ 7,784.92	\$ 60.82	\$ 81.09	\$ 7,643.01	\$ 731.80	\$ 8,516.72

Figuras 4.4.10 Generación de los valores de empeño del producto

Pruebas de integración

La prueba de integración es la etapa en la cual los módulos que conforman el sistema son combinados y probados en grupo. Este tipo de pruebas se realizan después de las pruebas unitarias. A continuación se muestran los resultados obtenidos para la prueba de integración del alta de un usuario, se muestran la pantalla de datos del usuario en la cual se informan nombre y contraseña con que se identificara en la pantalla acceso al SICE, departamento y perfil del usuario ver figura 4.4.11. La pantalla que sigue son los catálogos a los que tendrá acceso el usuario dentro del SICE, ver figura 4.4.12 una vez que se han guardado la información de estas pantallas, se procede a acceder al SICE con usuario creado. Vea figuras 4.4.13 y 4.4.14.



The image shows a web form titled "Nuevo Usuario". It contains the following fields and elements:

- Nombre de Usuario (login):
- Nombre Completo del Usuario:
- Contraseña:
- Confirma la Contraseña:
- Correo Electrónico:
- Departamento:
- Este usuario se creará: Configurando todos los permisos (dropdown menu)
- Buttons: Siguiente, Regresar

Figura 4.4.11 Datos básicos del usuario



Figura 4.4.12 Selección de catálogos que visualizara el usuario

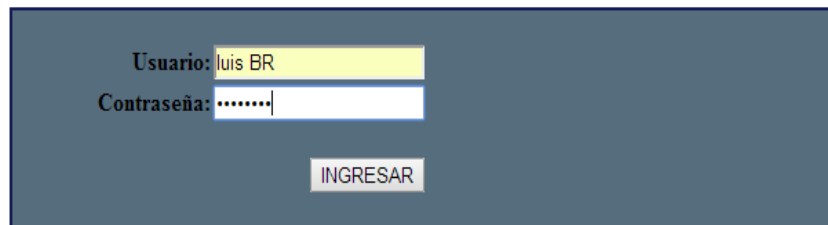


Figura 4.4.13 Inicio de sesión con el usuario creado



Figura 4.4.14 Sesión iniciada con el usuario creado

Pruebas de regresión

Este tipo de pruebas intenta descubrir errores, carencias de funcionalidad con respecto al comportamiento esperado del SICE. A continuación se muestran los resultados obtenidos para este tipo de prueba en la validación del código de metales. En la figura 4.4.15 se muestra la pantalla del catálogo metales, en el código se informará un valor alfabético, siendo que este campo solo acepta valores numéricos, la respuesta del SICE a este tipo de prueba fue un mensaje informando que el campo solo acepta valores numéricos. Vea figura 4.4.16



Figura 4.4.15 Pantalla de metales



Figura 4.4.16 Mensaje informando que el campo solo acepta números

Prueba alfa

En este tipo de pruebas el cliente o el usuario es el encargado de realizarlas, el desarrollador supervisará las pruebas y se utilizará un entorno controlado. Esta prueba es una pre-liberación del sistema, en donde se busca distinguir las fallas que son provocadas por errores de programación, de las que son originadas por el uso inadecuado por parte del usuario.

Prueba beta

La prueba beta es efectuada después de la prueba alfa, el usuario o cliente es el encargado de realizarla y se desarrolla fuera de un entorno controlado. La intención de este tipo de prueba es que el usuario o cliente pueda utilizar el sistema sin ninguna restricción o ayuda de los desarrolladores.

Prueba de stress

Este tipo de prueba está enfocado en poner a trabajar el sistema de tal modo que demande una gran cantidad de recursos, incluyendo casos en los que saturen la memoria o que causen problemas en el sistema operativo.

Prueba de recuperación

Este tipo de prueba está enfocada en determinar si el sistema es capaz de restablecer y continuar operando después de una falla de software o hardware. El método más común para llevar a cabo este tipo de pruebas es la simulación del problema que se está evaluando.

Mantenimiento

En todo sistema es necesario el mantenimiento y es una de las actividades más comunes en la ingeniería del software. El mantenimiento siempre se realizará posterior a la puesta en producción del sistema y en muchos casos se basa en mejoras que el cliente puede percibir como errores del sistema o defectos. Existen

diferentes tipos de mantenimientos a los que puede ser sometido un sistema como son; preventivo, correctivo, perfectivo y adaptativo.

Mantenimiento preventivo

Este tipo de mantenimiento radica en la optimización del software para mejorar sus propiedades sin alterar la funcionalidad del sistema.

Mantenimiento correctivo

El objetivo es identificar y corregir posibles áreas dentro del sistema. Un área se presenta cuando el comportamiento del sistema no cumple con los requerimientos dados por el usuario.

Mantenimiento perfectivo

Tiene por objetivo agregar funcionalidades al sistema, por lo regular debido a nuevos requerimientos por parte del usuario.

Mantenimiento adaptativo

Radica en la optimización del sistema debido a cambios en el entorno ya sean de hardware o software.

4.5 Generación de Comprobantes

Una vez que el pignorante realice el empeño de una prenda, el sistema le generará su comprobante. Vea figura 4.5.1.

Los puntos importantes del comprobante de empeño se listan a continuación:

- Nombre de la sucursal
- Nombre del pignorante
- Datos del pignorante como son nombre, dirección, teléfono, etc.
- Material de la pieza que empeño.
- Detalles del préstamo, como son:

- Avalúo de la prenda
- Plazo del préstamo
- Fecha de vencimiento
- Total del préstamo
- Detalle del plan de pagos, como son:
 - Fecha de semana de pago.
 - Monto a pagar por fecha de pago.

Este comprobante será entregado al pignorante y la casa de empeño se queda con una copia para el resguardo de la prenda.

Una vez que el pignorante acude a la sucursal a realizar su pago, en la sucursal donde realizó su empeño, está le generará un comprobante de pago. Vea figura 4.5.2.

Las características del comprobante de pago se describen a continuación

- Nombre de la sucursal
- Número de boleta.
- Fecha de pago
- Producto de empeño.
- Datos del pignorante como son nombre, dirección, teléfono, etc.
- Detalle de plan de pagos, como son:
 - Fecha
 - Fecha de pago
 - Capital
 - Refrendo
 - Morosidad
 - Total
 - Estatus del pago que puede ser pendiente o pagado.



SUCURSAL EMPEÑO : SUCURSAL CUAJIMALPA BOLETA :51
 SUCURSAL PAGO : SUCURSAL CUAJIMALPA FECHA : 23/03/2015

DETALLES DE CLIENTE VIGENTE PRODUCTO: SEMANAL

NOMBRE: Christian Raya Hernández

DOMICILIO:

Calle y Número Colonia Delegación Estado C.P.
 Av. Del Mazo MARCO ANTONIO CHALCO MEXICO 56600
 SOSA

IDENTIFICACIÓN # TELEFONO: CELULAR: EMAIL:
 191827364 59868483 5591876421 raya@gmail.com

HISTORIAL Y PLAN DE PAGOS

Fecha	Fecha de pago	Capital	Refrendo	Morosidad	Total	Sucursal	STATUS
09/02/2015		\$ 0.00	\$ 0.00	\$ 0.00	\$ 0.00		PAGADO
16/02/2015	23/03/2015	\$ 66.24	\$ 7.95	\$ 44.52	\$ 118.71	CUAJ	PAGADO
23/02/2015	23/03/2015	\$ 66.24	\$ 7.95	\$ 35.62	\$ 109.81	CUAJ	PAGADO
02/03/2015	23/03/2015	\$ 66.24	\$ 7.95	\$ 26.71	\$ 100.90	CUAJ	PAGADO
09/03/2015	23/03/2015	\$ 66.24	\$ 7.95	\$ 17.81	\$ 92.00	CUAJ	PAGADO
16/03/2015	23/03/2015	\$ 66.24	\$ 7.95	\$ 8.90	\$ 83.09	CUAJ	PAGADO
23/03/2015	23/03/2015	\$ 66.24	\$ 7.95	\$ 0.00	\$ 74.19	CUAJ	PAGADO
30/03/2015		\$ 66.24	\$ 7.95	\$ 0.00	\$ 74.19		PENDIENTE
06/04/2015		\$ 66.24	\$ 7.95	\$ 0.00	\$ 74.19		PENDIENTE

Figura 4.5.2 Comprobante de pago

Otro comprobante que se generará es el de gastos de oficina que se utiliza para comprobar los costos de operación de la sucursal. Vea figura 4.5.3.



SUCURSAL : SUCURSAL TOLUCA
 FECHA : 11/03/2015

HISTORIAL GASTOS DE OFICINA

Artículo	Comprobante	Fecha	Costo	Usuario
Paquete de Hojas Blancas de 1000	NOTA	09-03-2015	\$ 150.00	RICARDOGOMEZ
		Total	\$ 150.00	

Figura 4.5.3 Comprobante de gastos de oficina

El sistema SICE generará reportes que ayudará a analizar la información, estos reportes son tablas donde se mostrará la información de forma organizada. Vea figura 4.5.4.

Sucursal **SUCURSAL AZCAPOTZALCO** Buscar de **01-01-2015** a **08-04-2015** **Buscar**

Reportes								
Boleta	Fecha	Producto	Piezas	Status	Avalúo	Prestamo	Ciente Recibe	Pagar
92	2015-03-23	TRADICIONAL	1	VIGENTE	\$1,578.28	\$1,515.15	\$1,487.53	\$1,657.59
91	2015-03-23	TRADICIONAL	1	VIGENTE	\$1,578.28	\$1,515.15	\$1,487.53	\$1,657.59
90	2015-03-23	TRADICIONAL	1	VIGENTE	\$1,578.28	\$1,515.15	\$1,487.53	\$1,657.59
89	2015-03-23	TRADICIONAL	1	VIGENTE	\$1,262.63	\$1,212.12	\$1,190.02	\$1,326.04
88	2015-03-23	TRADICIONAL	1	VIGENTE	\$946.97	\$909.09	\$892.52	\$994.53
87	2015-03-23	TRADICIONAL	1	VIGENTE	\$631.31	\$606.06	\$595.02	\$663.02
86	2015-03-23	TRADICIONAL	1	VIGENTE	\$631.31	\$606.06	\$595.02	\$663.02
85	2015-03-23	TRADICIONAL	1	VIGENTE	\$631.31	\$606.06	\$595.02	\$663.02
48	2015-02-09	TRADICIONAL	1	VIGENTE	\$326.15	\$313.10	\$307.39	\$401.42
47	2015-02-09	SEMANAL	1	VIGENTE	\$124.74	\$119.75	\$104.42	\$132.35

Mostrando 1 - 10 de 15

Figura 4.5.4Reporte de operaciones por sucursal

Se podrá consultar la información por sucursal o concentradora. Vea figura 4.5.5.

Sucursal **DISTRITO FEDERAL** Buscar de **01-01-2015** a **08-04-2015** **Buscar**

Reportes									
Sucursal	Boletas	Almacenaje	Seguro	Capital	Interes	Pagos	Refrendo	Morosidad	
SUCURSAL TLAHUAC	10	\$2.45	\$3.26	\$313.10	\$7.36	\$320.46	\$0.00	\$0.00	
SUCURSAL IZTAPALAPA	3	\$2.45	\$3.26	\$313.10	\$4.70	\$317.80	\$0.00	\$0.00	
SUCURSAL CUAJIMALPA	21	\$37.62	\$50.13	\$908.29	\$72.21	\$1,114.06	\$0.00	\$133.56	
SUCURSAL AZCAPOTZALCO	17	\$3.35	\$4.46	\$428.48	\$10.07	\$438.55	\$0.00	\$0.00	

Mostrando 1 - 4 de 4

Figura 4.5.5Reporte de operaciones por concentradora

Se podrá consultar la información sobre el capital en cada sucursal. Vea figura 4.5.6.

Totales en la caja Actualizar						
SalDOS en caja de Sucursales						
Id	Sucursal	Dotacion	Pagos	Empeños	Gastos	Total
15	SUCURSAL AZCAPOTZALCO	\$ 10,000.00	\$ 438.55	\$ 1,537.07	\$ 0.00	\$ 8,901.48
16	SUCURSAL CUAJIMALPA	\$ 10,000.00	\$ 535.36	\$ 1,665.13	\$ 6,587.00	\$ 2,283.23
17	SUCURSAL IZTAPALAPA	\$ 10,000.00	\$ 317.80	\$ 307.39	\$ 0.00	\$ 10,010.41
18	SUCURSAL TLAHUAC	\$ 10,000.00	\$ 320.46	\$ 2,471.22	\$ 0.00	\$ 7,849.24
20	SUCURSAL TOLUCA	\$ 10,000.00	\$ 4,241.60	\$ 7,930.38	\$ 200.00	\$ 6,111.22
21	SUCURSAL NAULCALPAN	\$ 10,000.00	\$ 153.94	\$ 2,272.75	\$ 0.00	\$ 7,881.19

Mostrando 1 - 6 de 6

Figura 4.5.6 Reporte de operaciones por concentradora

Así mismo tendrá la facilidad de generar gráficas para analizar información de dotación, pagos, empeños y gastos por sucursal. Vea figuras 4.5.7 y 4.5.8.



Figura 4.5.7 Gráfica de cajas

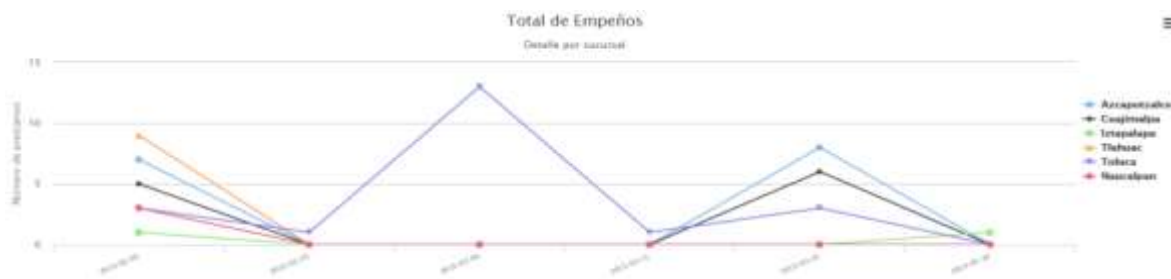


Figura 4.5.8 Gráfica de cantidad de empeños

CONCLUSIONES

- Se cubrieron satisfactoriamente los requerimientos planteados al inicio ya que se logró consolidar un sistema completo, sencillo, práctico y funcional, listo para ser usado.
- Se cumplieron los objetivos ya que el sistema es capaz de administrar una o más sucursales, gestiona diversos productos y nos entrega la información clara del detalle del préstamo prendario.
- La interfaz Web del Sistema desarrollado permite una fácil interacción del usuario con el sistema.
- El sistema logra agilizar el análisis y manejo de información permitiendo simplicidad y rapidez al trabajar con la información.
- El sistema ofrece integridad, disponibilidad y confiabilidad en la información para administrar casas de empeño.
- Con la creación del sistema se reduce el trabajo del personal al llevar el control del proceso de empeño en el registro de la información, evitando errores y obteniendo datos precisos y actualizados en todo momento. Dando como resultado la obtención de información precisa y actualizada en el momento que se requiere.
- Se cubren las necesidades de seguridad y respaldo de la información importante dentro de la base de datos.
- La implementación del sistema causará buen impacto en los usuarios ya que uno de los objetivos del sistema es el de registrar, almacenar y organizar de una manera rápida y eficiente toda la información que manejan las casas de empeño.
- La metodología evolutiva e incremental nos permite la retroalimentación durante el desarrollo y así poder actualizar o incrementar el funcionamiento del sistema.
- La utilización de herramientas visuales reduce el tiempo de desarrollo agilizando el proceso de entrega de la solución informática, además de que crea una interfaz amigable para el usuario que hace más fácil el manejo del sistema generando un menor tiempo de capacitación y por lo tanto, reducción en los gastos de la empresa.

- Este trabajo de tesis demuestra que las tecnologías de informática y las comunicaciones son herramientas esenciales para todas las actividades del ser humano, incluyendo las casas de empeño.
- Al brindar transparencia en las operaciones aumenta la satisfacción del cliente y aumenta la probabilidad de atraer más clientes.
- El éxito de muchas empresas depende del uso eficiente de su información, en la capacidad que tenga para recibir y transmitir información y su manejo adecuado, con la implantación de este sistema se mejora la organización de la información empleada para los procesos, a su vez se maneja con mayor rapidez y confiabilidad.
- Gracias a la formación integral que nos brinda la Facultad de Ingeniería en la carrera de Ingeniería en Computación, y a la experiencia laboral adquirida, se tiene la capacidad para resolver las necesidades de operación que tiene una casa de empeños, mediante la realización de un sistema.
- El Programa de Apoyo a la Titulación es una excelente opción para realizar la tesis, al formar grupos de trabajo nos crea un compromiso que nos motiva a continuar y apoyarnos entre nosotros para desarrollar un trabajo de calidad.

APÉNDICE 1

SICE establecerá en las cláusulas del contrato 13 y 14, donde el pignorante no cumpliera oportunamente con la obligación de pagar el monto del préstamo, los intereses, almacenaje y demás comisiones pactadas en el contrato antes de la fecha de vencimiento, la prenda pasará a ser propiedad del proveedor y este podrá aplicar venta y remate de la misma. Tres días después de la fecha de vencimiento se le notificará al pignorate que su prenda está en las listas de venta.

El sistema gestionará la notificación de aquellos pignorantes sobre pagos atrasados, vencimiento de contrato y lista de ventas, pero el remate de prendas no recuperadas no está contemplada en la versión actual sistema.

APÉNDICE 2

La propuesta del SICE para resguardar prendas “Electrónicos” no es factible debido a que el préstamo máximo que puede otorgar la Casa de Empeño a los pignorantes apenas rebasa el 7% del valor de la prenda a contra factura de compra. Esto se debe al avance tecnológico que se tiene día a día en los productos electrónicos y estos se devalúan en un corto lapso de tiempo. Por tal motivo cuando se le ofrece la cotización al pignorante por el electrónico que se desea empeñar, es poco probable que acepte el monto del préstamo y la casa de empeño tendrá productos poco factibles para su venta si estos no son recuperados; caso contrario con la joyería.

En la primera versión del sistema solo contempla la parte de joyería ya que son prendas que no pierden su valor en el mercado, y son de fácil almacenaje, y con esta versión se pueden aplicar los conocimientos y técnicas en el desarrollo de un sistema de una casa de empeño.

APÉNDICE 3

El pignorante deberá elegir un producto, el cual definirá el tipo de préstamo a realizar. El sistema SICE manejará 3 productos que son Express, Semanal y Tradicional, los cuales tendrán las siguientes características:

Producto Express

Este producto solo será de una semana y no habrá pago del almacenaje, seguro o interés.

Producto Semanal

Los pagos se dividirán entre las semanas de plazo que se elija. El pago total se dividirá entre las semanas, por lo que a mayor plazo será menor el pago semanal.

Producto Tradicional

Será de un solo pago y los intereses se incrementan conforme el tiempo avanza, y cada semana que pasa el interés será incrementado. Vea Tabla apéndice 3.1

Características	Productos		
	Express	Semanal	Tradicional
Plazo mínimo	1 semana	1 semana	1 semana
Plazo máximo	1 semana	12 semanas	12 semanas
Tipo de Pago	Único	Semanal	Único
% Compra	94	96	96
% Almacenaje	0	0.75	0.75
% Seguro	0	1	1
% Interés	0	1.5	2.35
Vencimiento	15 días	15 días	15 días

Tabla 3.1 Resumen de productos de empeño.

APÉNDICE 4

En el sistema manejará diversos valores, los cuales son explicados en este apéndice, se utilizará la moneda nacional (pesos) y se calcularán de la siguiente manera:

Metal fino

Para determinar este valor se tomará como referencia una onza troy (usada únicamente en joyería, orfebrería y numismática para pesar metales preciosos) cuyo valor es de 31.1035 gramos. También se tomará su precio internacional en dólares. Con estos datos se obtendrá el valor por gramo de oro en dólares, esto se hará dividiendo el valor de la onza entre los gramos de esta onza.

Para finalizar se hará la conversión de dólares a pesos tomando el precio del dólar a la venta.

$$\text{Metal fino} = (\text{precio onza troy} / \text{peso onza troy}) * \text{valor dólar}$$

Avalúo

Para obtener el valor del avalúo se necesitará el valor del metal en estado fino, el factor del metal (porcentaje de pureza), y el porcentaje de compra en México (99%) así como los gramos que tiene la prenda.

$$\text{Avalúo} = \text{valor oro fino} * \text{factor del metal} * \text{gramos} * \% \text{ compra}$$

Préstamo

Para obtener este valor de préstamo se deberá contar con el avalúo y el porcentaje de compra el cual está definido para cada producto.

$$\text{Préstamo} = \text{Avalúo} * \% \text{ Compra}$$

Almacenaje y seguro

Para obtener estos valores, los productos nos indicaran qué porcentaje del avalúo se tomara como se muestra en formula.

$$\text{Almacenaje} = \text{Avalúo} * \% \text{ Almacenaje}$$

$$\text{Seguro} = \text{Avalúo} * \% \text{ Seguro}$$

Ciente Recibe

Este valor se determinará restando el valor del almacenaje y del seguro al valor del préstamo.

$$\text{Ciente Recibe} = \text{Préstamo} - \text{Almacenaje} - \text{Seguro}$$

Interés

El interés se calculará con el valor del interés predefinido en cada producto.

$$\text{Interés} = \text{Préstamo} * \% \text{ Interés}$$

Pago total

El pago total que deberá cubrir el pignorante para desempeñar su prenda será finiquitando el valor del préstamo más el valor del interés.

$$\text{Pago total} = \text{Préstamo} + \text{Interés}$$

BIBLIOGRAFÍA

- Creación de un portal con PHP y MySQL, Pavón Puertas Jacobo; 4a Edición, México, D. F.; Alfaomega, 2011.
- Aprende PHP/MySQL Fácilmente; Barredo Cordon Adrián, 2009.
- MySQL administrator's bible; Cabral Sheeri, Murphy Keith; Wiley Pub., Indianapolis, 2009.
- PHP y MySQL: Tecnología para el desarrollo de aplicaciones web; Cobo Ángel, Gómez Patricia, Pérez Daniel; Ediciones Díaz de Santos, 2005.
- Database Modeling & Design, Teorey Toby J.; Ed. Morgan Kauffman Publishers; Third Edition 1999.
- Desarrollo web con PHP, Apache y MySQL; Scouarnec Yann Le, Naramore Elizabeth; Editor Anaya Multimedia, 2004.
- Bases de Datos, Diseño, Implementación y Administración; Coronel Carlos, Morris Steven, Rob Peter; Cengage Learning Editores, 2011
- Análisis y diseño de sistemas; Kenneth E. Kendall, Julie E. Kendall; Pearson Educación, 2005.
- Planificación y Evaluación de Proyectos Informáticos; Rodríguez Nuria, Martínez William; Editorial Universidad Estatal a Distancia, San José, Costa Rica, 2006.
- Diseño de sistemas interactivos centrados en el usuario; Granollers Toni, Lorés Vidal Jesús, Cañas Delgado José Juan; Editorial UOC, 2011.
- Técnicas cuantitativas para la gestión en la ingeniería del software; Tuya Javier, Ramos Román Isabel, Dolado Cosín José Javier; Netbiblo, 2007.
- Lessons Learned in Software Testing; Kaner Cem, Bach James, Pettichord Bret; John Wiley & Sons, Inc., NY, 2011.

Colección de URLs

- **Normatividad de casas de empeño**
 1. NORMA Oficial Mexicana NOM-179-SCFI-2007, Servicios de mutuo con interés y garantía prendaria [consulta noviembre 17, 2014]. Disponible en:

<http://www.dof.gob.mx/normasOficiales/2846/SEECO3/SEECO3.htm>

2. ACUERDO por el que se establecen las disposiciones de carácter general para la operación, organización y funcionamiento del Registro Público de Casas de Empeño [consulta noviembre 17, 2014].

Disponible en:

www.profeco.gob.mx/juridico/word/a_cde_2013.doc

3. Ley que Regula las Casas de Empeño en el Estado de México [consulta noviembre 21, 2014]. Disponible en:

<http://www.edomex.gob.mx/legistelfon/doc/pdf/ley/vig/leyvig194.pdf>

4. PROFECO planea regularizar 7 mil casas de empeño [consulta enero 7, 2015]. Disponible en:

<http://www.jornada.unam.mx/ultimas/2015/01/07/planea-la-profeco-regularizar-7-mil-casas-de-empeno-en-2015-2022.html>

- **Funcionamiento y términos comúnmente usados en Casas de Empeño**

1. ¿Qué son las casas de empeño? [consulta noviembre 10, 2014].

Disponible en:

<http://www.condusef.gob.mx/index.php/instituciones-financieras/otros-sectores/casas-de-empeno/516-ique-son-las-casas-de-empeno>

2. Glosario de Términos en Casa de Empeño [consulta noviembre 10, 2014]. Disponible en:

http://solucionesprovidencia.com/es/34_glosario.html

3. Productos y Servicios de una casa de empeño [consulta noviembre 14, 2014]. Disponible en:

http://www.montepiedad.com.mx/portal/?page=Document/doc_view_section.asp&id_document=234

4. Proceso de Empeño [consulta noviembre 17, 2014]. Disponible en:

https://www.montepio.org.mx/cms_mls/empenar_proceso

5. Proceso de Empeño [consulta noviembre 17, 2014]. Disponible en:

<http://www.prendamex.com.mx/index.php?pag=infografia&id=14>

6. Preguntas Frecuentes [consulta noviembre 28, 2014]. Disponible en:
<http://www.prendalana.com/preguntasFrecuentes.php?idm=0http://www.cashapoyoefectivo.com/>
7. Precio del oro y plata al día [consulta enero 5, 2015]. Disponible en:
<http://www.goldpricedata.com/es/gold-prices-in-mexico.php>

- **MySQL y Bases de Datos**

1. Principios Sobre Bases de Datos Relacionales [consulta diciembre 2, 2014]. Disponible en:
<http://www.jorgesanchez.net/bd/bdrelacional.pdf>
2. MySQL Workbench [consulta diciembre 4, 2014]. Disponible en:
<http://www.mysql.com/products/workbench/>
3. MySQL: Introducción, Características, Beneficios [consulta diciembre 6, 2014]. Disponible en:
<http://www.gedlc.ulpgc.es/docencia/abd/Recursos/MySQL-Intro-features-benefits-SPANISH.pdf>
4. Overview of the MySQL Database Management System [consulta enero 6, 2015]. Disponible en:
<http://dev.mysql.com/doc/refman/5.6/en/what-is-mysql.html>
5. MySQL Documentation: MySQL Reference Manuals [consulta enero 6, 2015]. Disponible en:
<http://dev.mysql.com/doc/>
6. MySQL [consulta enero 8, 2015]. Disponible en:
<http://karlitho.files.wordpress.com/2010/09/carlos-gomez.pdf>
7. Todo sobre MySQL [consulta enero 10, 2015]. Disponible en:
<https://books.google.com.mx/books?id=GS3kAgAAQBAJ&pg=PT11&dq=ventajas+de+usar+mysql&hl=es-419&sa=X&ei=TDrhVMq4MIq1ggSajYC4CQ&ved=0CCYQ6AEwAg#v=onepage&q&f=false>
8. El modelo Entidad – Relación [consulta enero 11, 2015]. Disponible en:

<https://books.google.com.mx/books?id=hbOTo05ddxAC&printsec=frontcover&dq=modelado+de+datos&hl=es&sa=X&ei=umbhVObyKMqZNu22gsAD&ved=0CDAQ6AEwAw#v=onepage&q=modelado%20de%20datos&f=false>

9. MySQL Ventajas y Desventajas [consulta enero 14, 2015]. Disponible en:

<https://mysquidaniel.wordpress.com/ventajas-y-desventajas/>

10. MySQL Ventajas [consulta enero 14, 2015]. Disponible en:
<http://sistemaspyt.blogspot.mx/2008/09/ventajas.html>

- **Desarrollo PHP y MySQL**

1. Desarrollo Web con PHP: Aprende PHP paso a paso [consulta diciembre 8, 2014]. Disponible en:

https://books.google.com.mx/books?id=1GQUAgAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

2. Desarrollo Web con PHP 6 y MySQL 5.1 [consulta diciembre 10, 2014]. Disponible en:

<https://books.google.com.mx/books?id=NjaKQgAACAAJ&dq=desarrollo+de+php&hl=es&sa=X&ei=LWHhVPSaMceagwSM1oOIDg&ved=0CB8Q6AEwATgK>

3. Guía práctica de PHP & MySQL: programación y desarrollo de webs [consulta diciembre 13, 2014]. Disponible en:

<https://books.google.com.mx/books?id=Qa7nXwAACAAJ&dq=desarrollo+de+php&hl=es&sa=X&ei=xmHhVPX3MsebNoy8g9gE&ved=0CCIQ6AEwAigU>

4. Desarrollo Web con PHP y MySQL 2015 / Web Development with PHP and MySQL [consulta enero 6, 2015]. Disponible en:

<https://books.google.com.mx/books?id=qECroAEACAAJ&dq=desarrollo+de+php&hl=es&sa=X&ei=LWHhVPSaMceagwSM1oOIDg&ved=0CCgQ6AEwAzgK>

5. Desarrollo PHP y MySQL [consulta enero 9, 2015]. Disponible en:
<https://books.google.com.mx/books?id=IMk02EiOrcAC&printsec=frontcover&dq=desarrollo+de+PHP+y+MySQL&hl=es&sa=X&ei=GmHhVNYBBJPjgwSzp4GgCA&ved=0CDAQ6AEwAg#v=onepage&q=desarrollo%20de%20PHP%20y%20MySQL&f=false>

- **Metodología evolutiva-incremental**

1. METODOLOGIAS DE DESARROLLO DE SOFTWARE [consulta enero 11, 2015]. Disponible en:
http://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CDEQFjAD&url=http%3A%2F%2Fjmpovedar.files.wordpress.com%2F2011%2F08%2Fmodelos-de-desarrollo-del-software.pptx&ei=W-WqVOruBZGtyQS90YGgDw&usg=AFQjCNFiJI98Seo1_N3yP6qvvj6inLjtDA&sig2=7B2zIUovNTKYDq6i4X0CIQ&bvm=bv.82001339,d.aWw
2. Proceso de desarrollo de Software [consulta enero 13, 2015]. Disponible en:
http://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=13&ved=0CCcQFjACOAo&url=http%3A%2F%2Fwww.dsic.upv.es%2Fasignaturas%2Ffacultad%2Fisi%2Fdoc%2FIntroduccionProcesoSW.doc&ei=peWqVKb_MlmvyATW_4GoDw&usg=AFQjCNGAyOzvxjChM6izslal_eLoQhtk2Q&sig2=UaZA-jYvLtZinzAlpnpBAg&bvm=bv.82001339,d.aWw
3. Metodologías para el desarrollo del sistemas de información [consulta enero 15, 2015]. Disponible en:
<https://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0CCMQFjAB&url=https%3A%2F%2Fprofvanessaborjas.files.wordpress.com%2F2012%2F06%2Fmetodolog>

c3adas-para-el-desarrollo-de-sistemas-de-
informac3b3n.pptx&ei=hmfhVOHDFYWigwSJsISoAw&usg=AFQjC
NGfqL3AX40dfiLLsFjtQvZAelym8A

4. Modelo evolutivo [consulta marzo3, 2015]. Disponible en:
<http://jorgetrejos.blogspot.mx/2010/08/modelo-evolutivo.html>