



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

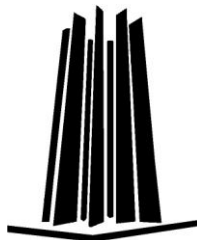
**“DISEÑO Y DESARROLLO DEL SISTEMA DE
ADMINISTRACIÓN Y CONTROL DE CLIENTES EN
LA DIVISIÓN MAQUINARIA INDUSTRIAL DE
FRESH SPLAY (DMIFS)”**

**T R A B A J O E S C R I T O
EN LA MODALIDAD DE DESARROLLO DE
UN CASO PRÁCTICO
QUE PARA OBTENER EL TÍTULO DE:**

INGENIERO EN COMPUTACIÓN

**P R E S E N T A :
J U A N J O S É M E J Í A L Ó P E Z**

ASESOR: M. en I. ARCELIA BERNAL DÍAZ



MÉXICO, 2010.



Universidad Nacional
Autónoma de México



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

A DIOS,

Por guiarme a través de la vida siendo mi padre, mi amigo, mi confidente y quien siempre me escucha. Quien ha puesto a mi disposición a las personas que han sido parte fundamental en mi camino.

A mi familia,

Por ser mi sustento, las personas más importantes en mi vida, quienes han estado conmigo en las buenas y en las malas, en mis triunfos y en mis fracasos y han sabido darme el apoyo para salir adelante cuando decaigo, así como impulsar mis logros.

A mi padre Juan,

Quien con su ejemplo y consejo me ha ayudado no sólo en la realización de mis estudios sino en todos los aspectos de mi vida. Quien con su experiencia me hace reaccionar y darme cuenta cuando estoy mal. Quien no se rinde y es gran ejemplo de trabajo y dedicación. Quien ha sido padre o amigo según lo necesite.

A mi madre Rosa,

Quien me ha sabido impulsar para lograr todo lo que he querido en la vida. No dejándome caer por ninguna razón. Quien sabe escucharme, aconsejarme y guiarme. Quien ha estado conmigo en todo momento, en mis tristezas y alegrías. Y es ejemplo de fortaleza y superación.

Agradecimientos

A mis hermanas Verónica y Angélica,

Quienes admiro por su fuerza para salir adelante y quienes han sido parte de mi inspiración para seguir avanzando. Y quienes me han apoyado siempre.

A Rogelio,

Quien ha sido un gran amigo y me ha apoyado en las buenas y en las malas, alguien con quien puedo contar siempre, quien me escucha y entiende.

A mi asesora la Mtra. En I. Arcelia Bernal,

Quien ha sido paciente, me ha ayudado y aconsejado para la conclusión de este trabajo. Es una gran persona con quien he podido contar desde que era mi maestra y agradezco que haya aceptado dirigir mi trabajo de titulación.

A mis compañeros y amigos,

*Con quienes conviví y trabajé durante mi carrera siendo un apoyo importante:
Ivan, Jorge, Luis, Ruy, etc.*

A mi tío Fidel Mejía y prima Pilar Mejía,

*Quienes permitieron que este proyecto se realizara en su empresa Fresh Splay.
Dándome el apoyo necesario.*

A la Universidad Nacional Autónoma de México,

De esta gran institución que he obtenido mucho en sus aulas y con sus maestros. Ha sido mi segundo hogar por mucho tiempo y externo mi profundo agradecimiento.

Dedicatoria

A mis padres,

Por todo lo que he mencionado en los agradecimientos. Pero además porque ha sido una meta conjunta y es de ustedes este trabajo.

A mis hermanas Vero y Angélica,

Dense cuenta que es posible lograr lo que se propongan, no solo en los estudios, en general, en su trabajo, en su vida cotidiana o en cualquier meta que se propongan.

A mi sobrina Alexia Michelle,

Tienes un apoyo en mí, puedes contar conmigo y aunque ahora eres muy pequeña algún día tendrás tus propias metas. Y si tú lo necesitas ahí estaré para apoyarte.

A mis sobrinos y primos menores,

Son muchos todos ustedes que vienen atrás de mí, puedo decirles que no es fácil, que toda meta concluida ha requerido de esfuerzo y dedicación, además de apoyo. Pero se pueden lograr los objetivos que se propongan. No solamente una carrera profesional, puede ser algún deporte, oficio o lo que a ustedes les interese, pero siempre haciendo su mayor esfuerzo. Les dedico a todos ustedes este trabajo y les digo que pueden lograr lo que se propongan.

Índice

Introducción	1
Capítulo 1 Antecedentes y planeación	3
1.1 Antecedentes	3
1.1.1 Entrevista preliminar	3
1.2 Sistema actual	4
1.3 Planeación	5
1.3.1 Reglas de negocio	5
1.3.2 Objetivo general	5
1.3.3 Alcances	5
1.3.4 Entregables	6
1.3.5 Recursos humanos	6
1.3.6 Recursos tecnológicos	6
1.3.7 Propuesta	7
1.3.8 Diagrama de Gantt	7
Capítulo 2. Herramientas para el diseño y desarrollo del proyecto	9
2.1 Modelo Entidad-relación	9
2.2 Modelo Relacional	10
2.3 MySQL	12
2.3.1 SQL	13
2.3.1.1 Creación de una base de datos	13
2.4 UML	16
2.4.1 Diagramas de casos de uso	16
2.4.2 Diagramas de clases	17
2.4.2.1 Asociación	18
2.4.2.2 Agregación	19
2.4.2.3 Composición	19
2.4.3 Diagramas de secuencia	20
2.5 Lenguaje de programación Java	21
2.5.1 Componentes swing	23
2.5.2 Patrones de ingeniería de software, interfaz DAO	23
2.5.3 Clases abstractas e interfaces	24
2.5.4 JDBC	25
2.5.5 NetBeans	26
Capítulo 3. Análisis y Diseño del sistema para la DMIFS	27
3.1 Análisis	27
3.1.1 Entrevistas a usuarios y observación	27
3.1.2 Sistema actual	27
3.1.3 Análisis de Requerimientos	28

3.1.4 Reglas de negocio	28
3.1.5 Solución de requerimientos, alcances y entregables	28
3.2 Diseño	29
3.2.1 Base de datos	
3.2.1.1 Diseño conceptual, modelo Entidad-Relación	29
3.2.1.1.1 Identificación de atributos, relaciones y llaves primarias	29
3.2.1.1.2 Diagrama E-R	31
3.2.1.2 Diseño Lógico, modelo relacional	31
3.2.1.2.1 Traducción de tipos de entidades y relaciones	31
3.2.1.2.1 Diagrama del Modelo Relacional	33
3.2.2 Aplicación	34
3.2.2.1 Casos de uso, descripción y diagramas	34
3.2.2.2 Diagramas de secuencia	42
3.2.2.3 Diagramas de clases	48
3.2.2.4 Interfaz de usuario	50
3.2.2.4 Reportes	51
Capítulo 4. Desarrollo del sistema para la DMIFS	53
4.1 Base de datos	53
4.1.1 Diccionario de datos	53
4.1.2 Código de implantación en MySQL	54
4.2 Aplicación	55
4.2.1 Clases Principales	55
4.2.2 Interfaces y conexión con la base de datos	57
4.2.3 Interfaz gráfica de usuario	57
4.3 Pruebas	69
4.4 Documentación, manuales de usuario y técnico	69
Capítulo 5. Implantación y Mantenimiento	70
5.1 Instalación	70
5.1.1 MySQL	70
5.1.2 JDK, máquina virtual de Java	70
5.1.3 Sistema	70
5.2 Capacitación	72
5.3 Pruebas con el usuario	72
5.4 Correcciones	73
5.5 Mantenimiento	73
Conclusiones	74
Bibliografía	76

Introducción

Actualmente la información es parte fundamental en las organizaciones, por tal motivo ha sido indispensable la creación de sistemas informáticos que llegan a constituir parte indispensable para su desarrollo. La tecnología actual tiene tal influencia que dichas organizaciones tienen la necesidad de utilizarla para su adaptación al entorno.

En particular las micro-empresas que están en crecimiento y aumentan sus tareas, por ejemplo de ventas, es en este punto donde tienen que buscar la forma de automatizar sus procesos para así tener un mejor desempeño.

Este es el caso de la micro-empresa conformada a través de Fresh Splay MR. en donde cada área factura con distintas personas físicas, y tienen la intención en un futuro de crear una sociedad anónima y así consolidarse como una empresa SA. De CV. Una de las áreas que ha tenido un mayor crecimiento y por lo tanto cada vez es más compleja, es la División Maquinaria Industrial. Esta división se encarga de la fabricación, adaptación, venta y mantenimiento de accesorios para maquinas engargoladoras y selladoras de válvulas, pero el manejo de información acerca de sus clientes, productos y servicios se volvió un problema, sobre todo para la realización de sus cotizaciones. Siendo parte fundamental entregar a tiempo y de forma correcta las cotizaciones, ya que de esto depende en buena medida una rápida respuesta al cliente y así una ganancia en tiempo y preferencia. Por otra parte los encargados de cotizar van obteniendo otras tareas y por lo tanto cada vez tienen menos tiempo para dedicarse a cada función que realizan.

Para solucionar este problema se requirió implementar una solución tecnológica que permita automatizar el proceso de cotizaciones y facturación, así como llevar un control sobre sus clientes, productos y servicios.

En este trabajo se lleva a acabo el proceso de desarrollo de software en donde las necesidades del usuario son analizadas y convertidas en requerimientos de software, a su vez estos requerimientos transformados en diseño y en base al diseño se genera el código que es probado antes de implementarlo, una vez implementado el sistema se procede a darle soporte y mantenimiento.

El capítulo uno contiene la planeación del proyecto que incluye el objetivo, la asignación de recursos y tiempo así como la propuesta.

En el capítulo dos se presentan las herramientas utilizadas para el diseño y desarrollo del proyecto, tales como: UML, Modelo Entidad-Relación, Modelo Relacional, MySQL, NetBeans y Java.

El análisis y diseño del sistema son manejados en el capítulo tres, que abarca desde un análisis detallado de requerimientos hasta los diseños conceptuales y lógicos de la base de datos y la aplicación.

El desarrollo es tratado en el capítulo cuatro donde con base en el diseño, el sistema gestor de base de datos MySQL, el entorno de desarrollo integrado NetBeans y el lenguaje de programación Java es creado el sistema en su conjunto, así como la realización de pruebas y la documentación.

Por último el capítulo cinco comprende la implantación y mantenimiento que va desde la instalación del sistema, la capacitación al usuario, las pruebas con el usuario y las correcciones que se deban hacer.

Capítulo 1

Antecedentes y planeación

1.1 Antecedentes

La información es parte fundamental en las organizaciones, por tal motivo ha sido indispensable la creación de sistemas informáticos que llegan a constituir parte indispensable para su desarrollo. La tecnología actual tiene tal influencia que dichas organizaciones tienen la necesidad de utilizarla para su adaptación al entorno. En especial las pequeñas empresas que están en crecimiento, ya que aumentan sus tareas y tienen que buscar la forma de automatizar sus procesos para tener un mejor desempeño.

La micro-empresa formada a través de la marca registrada Fresh Splay, que tiene su localización en el oriente de la Ciudad de México, cuenta con distintas áreas de trabajo y producción en donde cada área factura con diferentes personas físicas y a medida que va creciendo tienen la intención de crear una sociedad anónima y así consolidarse como una empresa de Sociedad Anónima de Capital Variable. Las áreas con que se cuenta son: la división maquinaria industrial, distribuidora de artículos, esencias de perfumería, tiendas de perfumería y artículos varios

Una de las áreas con un mayor crecimiento y por consiguiente cada vez más compleja, es la División Maquinaria Industrial. Esta división se encarga de la fabricación, adaptación, venta y mantenimiento de accesorios para máquinas engargoladoras y selladoras de válvulas, pero el manejo de información acerca de sus clientes, productos y servicios se volvió un problema, sobre todo para la realización de sus cotizaciones. Siendo parte fundamental entregar a tiempo y de forma correcta las cotizaciones, ya que de esto depende en buena medida una rápida respuesta al cliente y así una ganancia en tiempo y preferencia.

Por otra parte los encargados de cotizar cada día obtienen otras tareas y por lo tanto cada vez tienen menos tiempo para dedicarse a cada función que realizan debido al crecimiento de la empresa.

En consecuencia Fresh Splay en la división maquinaria industrial decidió obtener ayuda de tecnologías de la información para mejorar su sistema actual. Y así tener la oportunidad de seguir creciendo como empresa con ayuda de recursos tecnológicos, en este caso un software que contribuya a la eficiencia de los procesos de la empresa.

1.1.1 Entrevista Preliminar

El primer contacto con Fresh Splay fue la entrevista preliminar donde se realizaron preguntas para conocer las necesidades de la empresa ya que no tenían la idea clara de que es lo que necesitaban desde un principio.

En entrevista con el presidente de la empresa, delegó el proyecto al encargado de la división de maquinaria industrial con quien se entabló la conversación. Así se pudieron conocer las principales necesidades que se requerían, tales como automatización de cotizaciones y facturación, administración de clientes, productos y servicios en la división maquinaria industrial de la empresa.

1.2 Sistema actual

A través de las entrevistas siguientes con el encargado del área y de usuarios se llegó a saber el funcionamiento del sistema que en ese momento estaba funcionando.

La división de maquinaria industrial se encarga de la fabricación, adaptación, venta y mantenimiento de máquinas engargoladoras y selladoras de válvulas. En esta área se hacen los mantenimientos para tales máquinas, así como venta de maquinaria nueva y de refacciones.

Cuando un cliente solicita una cotización de maquinaria nueva, refacciones o servicios, se realiza dicha cotización de forma manual.

Otro tipo de cotización es que debido al desgaste que tienen las máquinas de los clientes tienen que mandar al taller para hacerles un servicio de mantenimiento. Para este proceso se envían las máquinas al taller y de las oficinas de Fresh Splay un formato que los técnicos tienen que llenar en el cual se especifica por cada máquina cuales son las refacciones y servicios que se necesitan para dar el mantenimiento y la máquina sea reparada. Una vez que los técnicos hacen el llenado de los formatos los mandan de regreso a las oficinas para la cotización del presupuesto y remitirlo al cliente, para que decida si autoriza que se haga el trabajo y mande el anticipo correspondiente para así comenzar con el trabajo.

Sin embargo para Fresh Splay, el manejo de información acerca de sus clientes se ha vuelto un problema, ya que se tienen archivos en procesadores de texto con sus datos organizados por carpetas, pero al ir creciendo en clientes la búsqueda de esos datos resulta muy tardada, ya que se hace de forma manual. De igual forma para obtener los datos de sus productos y servicios, que les solicitan los clientes, tienen que buscar en una carpeta donde tienen sus productos y servicios los archivos de texto y hojas de cálculo correspondientes.

Consecuentemente cuando un cliente pide una cotización se tienen que buscar los datos del cliente en un archivo, después dependiendo de lo que solicite, se escriben manualmente los productos y servicios que se requieran para la cotización y en una carpeta distinta se buscan sus correspondientes precios.

Las cotizaciones de máquinas engargoladoras y selladoras de válvulas que se mandan a reparar o darles su mantenimiento se hacen de la siguiente manera: primero se hace la cotización por máquina y finalmente se hace la suma total, se va generando la cotización solicitada por el cliente en una hoja de cálculo, primero

se introducen los datos del cliente, enseguida se escriben los productos y servicios que requiere el cliente en secciones distintas cerrando un total por sección (máquina), y al final se hace la suma total con los totales por sección.

En su totalidad este proceso lleva mucho tiempo realizarlo ya que todo se hace de forma manual, o con algún formato ya establecido, pero en el cual siempre hay que escribir de forma manual los datos.

1.3 Planeación

Una vez conociendo las necesidades del cliente lo siguiente fue realizar el análisis previo para dar una solución al problema estableciendo las reglas de negocio, el objetivo general, los alcances y asignación de recursos, para así realizar la propuesta del proyecto.

1.3.1 Reglas de negocio.

Básicamente se deberán cumplir las siguientes reglas de negocio:

1. Automatizar el proceso de cotizaciones y facturación.
2. Llevar un control de clientes, productos y servicios.

1.3.2 Objetivo general

Diseñar e implementar una herramienta tecnológica que permita automatizar el proceso de cotizaciones y facturación en la división maquinaria industrial, así como llevar un control sobre sus clientes, productos y servicios.

Para no salirse de lo establecido es importante tener bien claro cuál es el objetivo general del proyecto y que el cliente quede de acuerdo.

1.3.3 Alcances

El sistema deberá contar con:

- 1) Una base de datos la cual contendrá información referente a los clientes, productos, servicios, cotizaciones y empleados.
- 2) La interfaz de usuario contará con:

Una interfaz gráfica para la administración y control de los clientes:
(Altas, bajas, cambios, consultas).

Una interfaz para la administración y control de los productos y servicios
(Altas, bajas, cambios, consultas).

3) Una interfaz gráfica para la realización de las cotizaciones de las máquinas que llegan y cotizaciones en general:

Se podrá consultar y agregar el cliente a la cotización extrayendo sus datos de la base de datos.

Se podrán consultar de la base de datos y agregar a la cotización los productos y servicios con sus atributos.

4) Una interfaz gráfica que permita la elaboración e impresión de facturas.

1.3.4 Entregables

Los archivos, impresiones, manuales con lo contará el usuario son los siguientes:

- 1) Archivo que contenga la cotización realizada.
- 2) Facturación
- 3) Impresión de la cotización
- 4) Impresión de consultas
- 5) Manual de usuario

1.3.5 Recursos Humanos

Los recursos humanos necesarios para el desarrollo del proyecto son:

- 1) Líder de Proyecto
- 2) Analista
- 3) Usuarios
- 4) Diseñador
- 5) Programador

Para el desarrollo del proyecto tomé distintos roles tales como: líder de proyecto, analista, diseñador y programador.

1.3.6 Recursos tecnológicos

Los recursos tecnológicos con los que ya se cuenta: son la red LAN formada por tres computadoras y dos impresoras con las que ya se tienen en el área.

- 3 PC HP pavilion. Se expandió la memoria RAM a su máxima capacidad, a un 1Gb, y a 512 mb.
- 1 impresora hp deskjet 3535
- 1 impresora Epson 850

También se cuenta con las herramientas tecnológicas para el desarrollo del sistema como es:

- MySQL para la base de datos.
- Java como lenguaje de programación.
- y el entorno de trabajo NetBeans.

1.3.7 Propuesta

Implementar un sistema de administración y control de clientes, productos y servicios, que contenga una interfaz gráfica de usuario que permita realizar cotizaciones de presupuestos, así como facturación.

El proyecto es nombrado: “Diseño y desarrollo del sistema de administración y control de clientes en la División Maquinaria Industrial de Fresh Splay (DMIFS)”. En adelante la División Maquinaria Industrial de Fresh Splay será nombrada como DMIFS.

1.3.8 Diagrama de Gantt

Una vez establecido el objetivo, la propuesta y los recursos se procede a la asignación de tiempos para el desarrollo del proyecto. Los cuales se muestran en la figura 1.1. Donde se observa de forma general desde la planeación y el ciclo de vida del desarrollo del software: análisis, diseño, programación, implementación y mantenimiento.

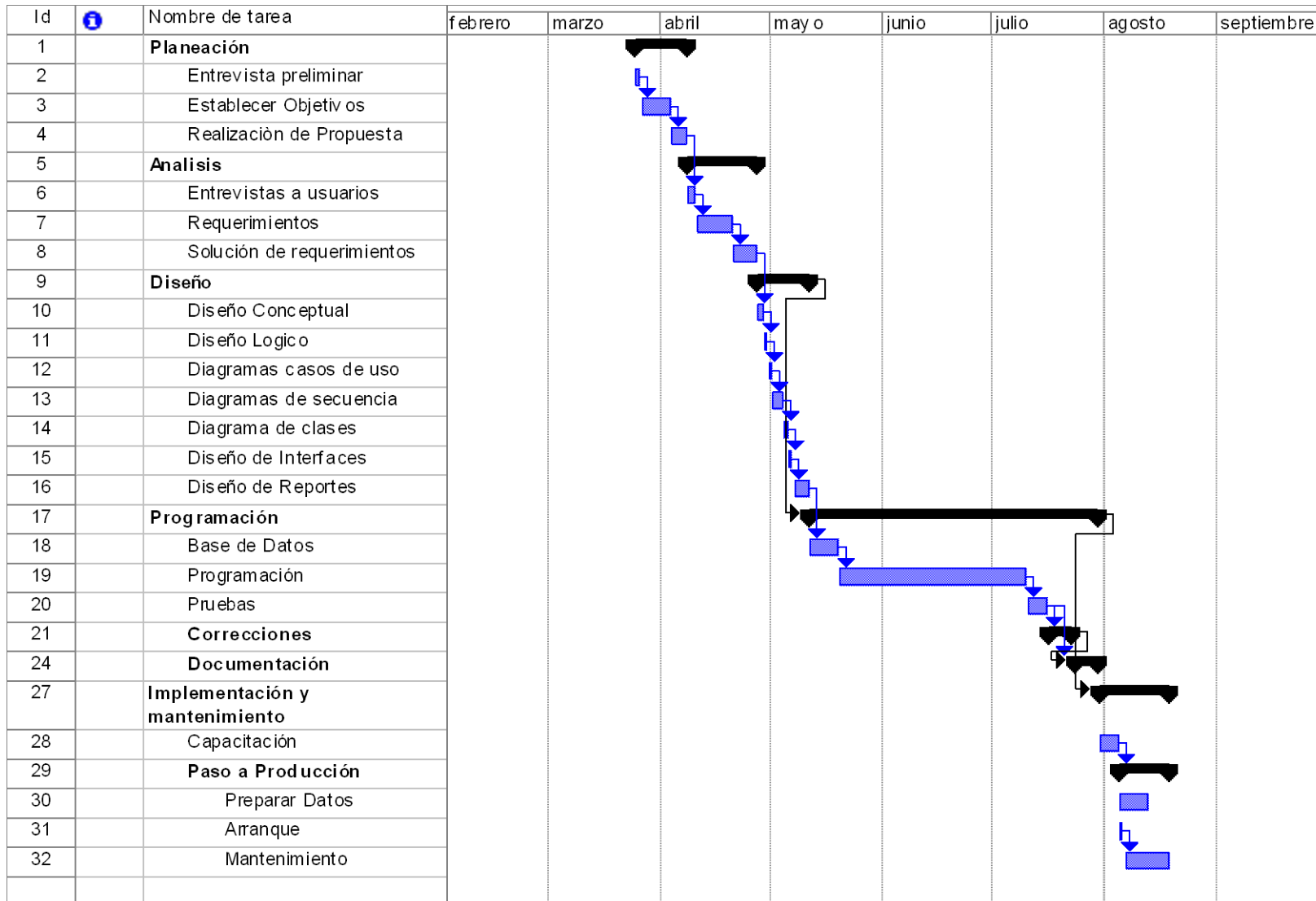


Figura 1.1. Se muestra el diagrama de Gantt con los tiempos utilizados para las tareas del proyecto

Capítulo 2

Herramientas para el diseño y desarrollo del proyecto

2.1 Modelo Entidad-Relación (E-R)

Para el diseño de la base de datos se utilizó el modelo de datos entidad-relación que es basado en una percepción del mundo real consistente en un conjunto de objetos básicos llamados entidades y en relaciones entre esos objetos y está pensado para el proceso de diseño de la base de datos ya que representa la estructura lógica general de la base de datos.

La estructura general de este modelo se puede ver con un diagrama E-R. En donde se utilizan diversos símbolos tales como:

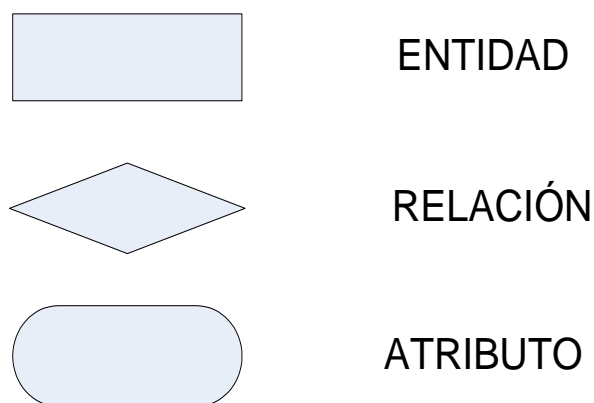


Figura 2.1. Símbolos utilizados en el modelo E-R.

Una entidad es un objeto que existe y se distingue de otros objetos. Se expresa la distinción asociando con cada entidad un conjunto de atributos que describen el objeto. Se representa por el símbolo rectángulo y por ejemplo una entidad para la base de datos en cuestión podría ser Cliente.

El atributo es cada una de las características que posee una entidad, y que agrupadas permiten distinguirla de otras entidades del mismo conjunto. Así mismo para el ejemplo de Cliente como entidad sus atributos podrían ser: nombre, teléfono, dirección, empresa, etc.

Relación es una asociación entre diferentes entidades. En este caso se puede relacionar la entidad Cliente con Facturas, por Cliente solicita Facturas.

Un conjunto de relaciones es una colección de relaciones del mismo tipo y un conjunto de entidades es una colección de entidades del mismo tipo.

Es necesario identificar la clave que es un conjunto de atributos que identifican de forma unívoca una entidad. En este caso se le puede asignar al Cliente un número de cliente el cual funcionará como llave primaria. La clave primaria se escoge de forma arbitraria de entre un conjunto de posibles llaves candidatas.

Otros conceptos que deben conocerse son la generalización y la especialización:

Generalización es el proceso según el cual se crea un conjunto de entidades a partir de otros que comparten ciertos atributos. Algunas veces hay situaciones en que sea conveniente crear una entidad como una unión de otras, en un principio diferentes, aunque con atributos comunes. Así se puede disminuir el número de conjuntos de entidades y facilita el establecimiento de relaciones.

La especialización al contrario de la generalización, es el proceso donde se crean varios tipos de entidades a partir de uno. Cada una de los conjuntos de entidades resultantes contendrá sólo algunos de los atributos del conjunto original.

De esta forma una vez obteniendo el modelo conceptual a través del diagrama E-R se procede a hacer el modelo relacional que es el diseño lógico pasando de un modelo al otro.

2.2 Modelo Relacional.

El modelo relacional se ha establecido actualmente como el principal modelo de datos para las aplicaciones de procesamiento de datos. Ha conseguido la posición principal debido a su simplicidad, que facilita el trabajo del programador en comparación con otros modelos anteriores como el de red y el jerárquico.

Este modelo se basa en relaciones y una relación es un conjunto de datos referentes a un conjunto de entidades y organizados en forma tabular, que se compone de filas y columnas, en la que cada intersección de fila y columna contiene un valor. Se muestra en la figura 2.2.

	Atributo o columna	atributo	atributo	atributo
Registro o fila registro				

Figura 2.2. Organización tabular de entidades en el modelo relacional.

Una fila o registro es cada una de las tuplas de una relación y contiene la información relativa a una única entidad y una columna o atributo es cada una de las características que posee una entidad, y que agrupadas permiten distinguirla de otras entidades del mismo conjunto.

Existen atributos, para determinadas entidades, que carecen de valor. El modelo relacional distingue entre valores vacíos y valores nulos. Un valor vacío se considera un valor tanto como cualquiera no vacío, sin embargo, un nulo NULL indica la ausencia de valor.

El nulo es muy importante en el modelo relacional, ya que permite trabajar con datos desconocidos o ausentes.

Hay un dominio para los valores de las columnas que son los posibles valores que puede tener. Esto tiene mucha importancia, ya que será un dato importante a la hora de dimensionar la relación. Por ejemplo, si definimos un atributo del tipo entero, el dominio más amplio sería, lógicamente, el de los números enteros. Pero este dominio es infinito, y las computadoras no pueden manejar números infinitos enteros. Al definir un atributo se dispone de distintas opciones para guardar datos enteros, enteros cortos, enteros largos etc. De igual forma para otro tipo de valores como caracteres, cadenas de caracteres, etc.

En el modelo relacional no deben existir dos registros iguales, cada atributo sólo puede tomar un único valor del dominio, es decir, no pueden contener listas de valores y el orden de los registros dentro de la relación y el de las columnas, dentro de cada registro, no es importante.

Otro concepto del modelo es la cardinalidad que es el número de registros que contiene una relación. Esta cambia a lo largo del tiempo, siempre se pueden añadir y eliminar registros.

De igual forma el grado es el número de atributos de cada registro. El grado de una relación es un valor constante. Esto no quiere decir que no se puedan agregar o eliminar atributos de una relación; lo que significa es que si se hace, la relación cambia. Cambiar el grado implicará modificaciones en las aplicaciones que hagan uso de la base de datos, ya que cambiarán conceptos como claves e interrelaciones.

La clave es un conjunto de atributos que identifica de forma unívoca a un registro. Puede estar compuesto por un único atributo o una combinación de varios. Aquí no existe el concepto de clave múltiple. Cada clave sólo puede hacer referencia a un registro de una tabla. Por lo tanto, todas las claves de una relación son únicas.

La clave candidata es cada una de las posibles claves de una relación, en toda relación existirá al menos una clave candidata. Esto implica que ninguna relación puede contener registros repetidos.

La clave primaria es la clave candidata elegida para identificar los registros, las claves primarias no pueden tomar valores nulos y son elegidas arbitrariamente.

También hay otras llaves como la Alternativa que es una clave candidata que no es clave primaria.

La foránea es el atributo dentro de una relación que contiene claves primarias de otra relación. No hay nada que impida que ambas relaciones sean la misma.

Se dice que dos relaciones están interrelacionadas cuando una posee una clave foránea de la otra. Cada una de las claves foráneas de una relación establece una interrelación con la relación donde esa clave es la principal.

Hay diferentes tipos de relaciones entre registros:

- Uno a uno: a cada registro de una relación le corresponde un y sólo un registro de otro.
- Uno a varios de donde a cada registro le corresponden varios en otro.
- Y varios a varios cuando varios registros de una relación se pueden corresponder con varios registros.

Para convertir los diagramas E-R a diagramas relacionales existen varias reglas:

1. Para cada conjunto de entidades fuertes se crea una relación con una columna para cada atributo.

2. Para cada conjunto de entidades débiles se crea una relación que contiene una columna para los atributos que forman la clave primaria de la entidad fuerte a la que se encuentra subordinada y una columna para cada atributo de la entidad.

3. Para cada interrelación se crea una relación que contiene una columna para cada atributo correspondiente a las claves principales de las entidades interrelacionadas.

4. Lo mismo para entidades compuestas, añadiendo las columnas necesarias para los atributos añadidos a la interrelación.

Ya construido el modelo relacional con su diagrama se pasa a la realización en MySQL.

2.3 MySQL

El sistema de gestión de bases de datos relacionales MySQL es el más popular del mundo, tiene como características ser multihilo y multiusuario. MySQL se ofrece bajo la licencia GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos

deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Bajo la licencia GNU GPL es posible recurrir a este software para el uso de este proyecto en particular, ya que es totalmente compatible con la licencia del lenguaje de programación java empleado para el desarrollo.

Otra característica es que es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.

Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. La parte SQL de MySQL se refiere a Structured Query Language. SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. El estándar SQL ha evolucionado desde 1986 y existen varias versiones.

2.3.1 SQL

Existen dos lenguajes para el manejo de bases de datos:

DDL (Data Definition Language) Lenguaje de definición de datos. Es el lenguaje que se usa para crear bases de datos y tablas, y para modificar sus estructuras, así como los permisos y privilegios. Y este lenguaje trabaja sobre unas tablas especiales llamadas diccionario de datos.

DML (Data Manipulation Language) lenguaje de manipulación de datos. Es el que se usa para modificar y obtener datos desde las bases de datos.

SQL engloba ambos lenguajes DDL y DML, los dos forman parte del conjunto de sentencias de SQL.

2.3.1.1 Creación de una base de datos

Cada conjunto de relaciones que componen un modelo completo forma una base de datos. Desde el punto de vista de SQL, una base de datos es sólo un conjunto de tablas, y para organizarlas o distinguirlas se accede a ellas mediante su nombre. A nivel de sistema operativo, cada base de datos se guarda en un directorio diferente.

A continuación se describen las sentencias más comunes utilizadas en este proyecto:

Para crear una base de datos se usa una sentencia

CREATE DATABASE “nombre de la base de datos”.

Crear una tabla:

CREATE TABLE “nombre de la tabla”

La sintaxis de esta sentencia es muy compleja, ya que existen muchas opciones y posibilidades diferentes a la hora de crear una tabla.

En su forma más simple, la sentencia CREATE TABLE creará una tabla con las columnas que se indiquen. Pero hay más opciones a la hora de definir columnas, además del tipo y el nombre, se pueden definir valores por defecto, permitir o no que contengan valores nulos, crear una clave primaria, indexar, etc.

La sintaxis para definir columnas es:

```
nombre_col tipo [NOT NULL | NULL] [DEFAULT valor_por_defecto]
                [AUTO_INCREMENT] [[PRIMARY] KEY] [COMMENT 'string']
                [definición_referencia]
```

Al definir cada columna se decide si podrá o no contener valores nulos.

Si se define una columna como clave primaria, automáticamente se impide que pueda contener valores nulos, pero este no es el único caso en que puede ser interesante impedir la asignación de valores nulos para una columna.

La opción por defecto es que se permitan valores nulos, NULL, y para que no se permitan, se usa NOT NULL.

Para cada columna también se puede definir, opcionalmente, un valor por defecto. El valor por defecto se asignará de forma automática a una columna cuando no se especifique un valor determinado al añadir filas.

Si una columna puede tener un valor nulo, y no se especifica un valor por defecto, se usará NULL como valor por defecto.

También se puede definir una clave primaria sobre una columna, usando la palabra clave KEY o PRIMARY KEY.

Sólo puede existir una clave primaria en cada tabla, y la columna sobre la que se define una clave primaria no puede tener valores NULL. Si esto no se especifica de forma explícita, MySQL lo hará de forma automática.

En MySQL existe la posibilidad de crear una columna auto-incrementada, aunque esta columna sólo puede ser de tipo entero.

Si al insertar una fila se omite el valor de la columna auto-incrementada o si se inserta un valor nulo para esa columna, su valor se calcula automáticamente, tomando el valor más alto de esa columna y sumándole una unidad. Esto permite crear, de una forma sencilla, una columna con un valor único para cada fila de la tabla.

Para la inserción de nuevos registros se utiliza la sentencia INSERT:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name [(col_name,...)]
      VALUES ({expression | DEFAULT},...),(...),...
      [ ON DUPLICATE KEY UPDATE col_name=expression, ... ]
```

Un ejemplo es:

```
INSERT INTO usuarios (nombre, apepat, apemat, telefono) VALUES( 'Juan José',
'Mejía','López', 52364556);
```

Para la actualización de registros se utiliza la sentencia UPDATE que se muestra a continuación:

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name
      SET col_name1=expr1 [, col_name2=expr2 ...]
      [WHERE where_definition]
      [ORDER BY ...]
      [LIMIT row_count]
```

Un ejemplo es:

```
UPDATE usuarios SET telefono = 50236012 WHERE nombre = 'Juan José'
```

Para la realización de consultas se utiliza la sentencia SELECT:

```
SELECT [ALL | DISTINCT | DISTINCTROW]
      expresion_select,...
      FROM referencias_de_tablas
      WHERE condiciones
      [GROUP BY {nombre_col | expresion | posicion}
      [ASC | DESC], ... [WITH ROLLUP]]
```

En este caso se puede dar un ejemplo muy sencillo en donde se seleccionaran todos los registros de una tabla a través de esta sentencia de una tabla.

```
SELECT * FROM clientes;
```

2.4 UML

El Lenguaje Unificado de Modelado es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores

Hay que tener en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación.

El éxito de los proyectos de desarrollo de aplicaciones o sistemas se debe a que sirven como enlace entre quien tiene la idea y el desarrollador. UML es una herramienta que cumple con esta función, ya que le ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien este involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos mediante el proceso de desarrollo.

Un modelo UML indica qué hará el sistema, más no cómo lo hará y esto lo hace mediante diagramas tales como los usados para la elaboración del sistema en cuestión como son: diagramas de casos de uso, diagramas de clases y diagramas de secuencia.

2.4.1 Diagramas de casos de uso

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica.

Los casos de uso son lo que hace el sistema desde el punto de vista del usuario. Describe un uso del sistema y cómo este interactúa con el usuario. Se representa mediante un ovalo.

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palitos. Esta representación sirve tanto para actores que son personas como para otro tipo de actores. Se puede observar en la figura 2.3.

Un diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Actores, casos de uso y relaciones entre casos de uso son elementos de un diagrama de casos de uso. Entre dos casos de uso puede haber las siguientes relaciones:

- Extender: cuando un caso de uso especializa a otro extendiendo su funcionalidad.
- Incluir: cuando un caso de uso utiliza a otro.

Se representan como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extiende>> o <<incluye>> según sea el tipo de relación.

En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea. La figura 2.3. muestra un ejemplo.

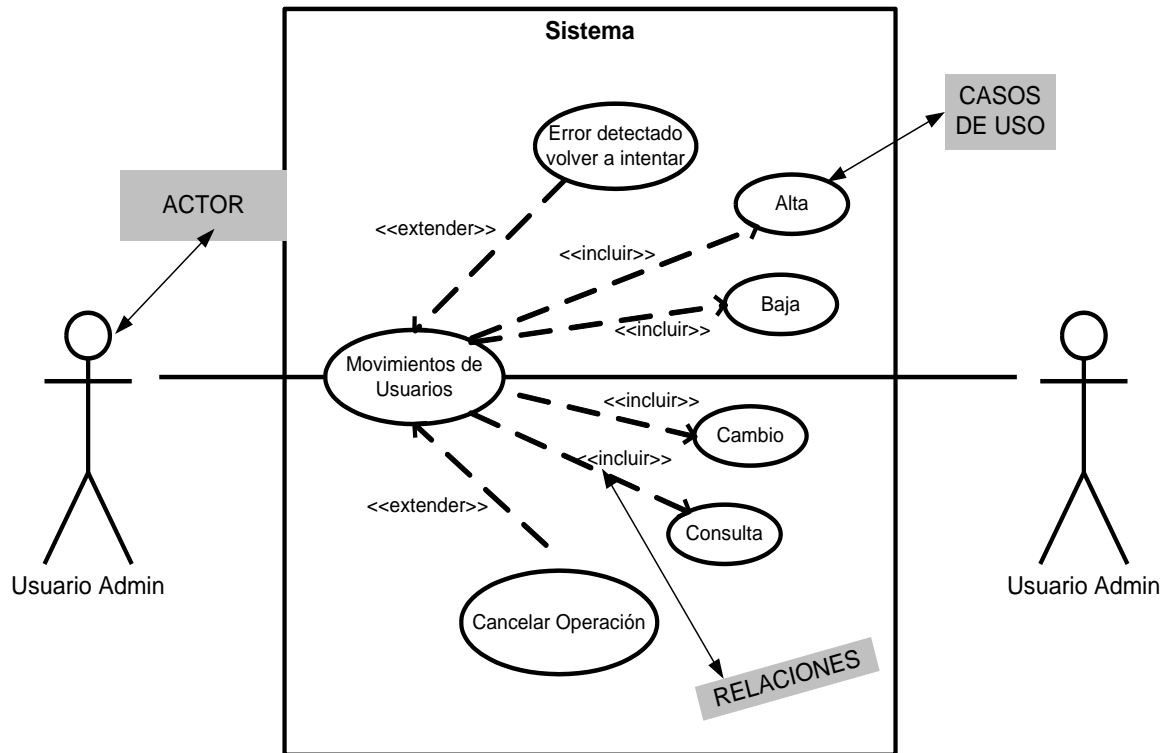


Figura 2.3. Muestra un diagrama de casos de uso con sus actores, casos de uso y relaciones entre casos de uso.

2.4.2 Diagramas de clases

Los diagramas de clases se encargan de mostrar las diferentes clases que componen un sistema y la relación que existe entre ellas. Este tipo de diagramas son estáticos porque muestran las clases, junto con sus métodos y atributos y las

relaciones estáticas entre ellas: qué clases tienen comunicación con otras clases o qué clases forman parte de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase tienen el mismo comportamiento y el mismo conjunto de atributos.

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Y normalmente muestran los siguientes elementos:

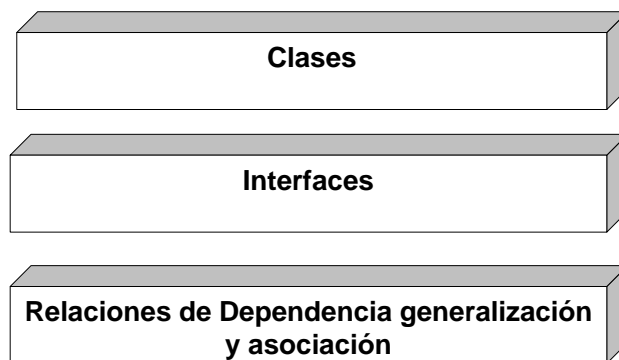


Figura 2.4. Elementos de los diagramas de clases.

2.4.2.1 Asociación

Una asociación es una relación estructural que especifica que los objetos de un elemento están conectados con los objetos de otro.

Gráficamente una asociación se representa como una línea continua que conecta la misma o diferentes clases. Lleva un nombre y una dirección representada con un triángulo como se muestra en la figura 2.5.

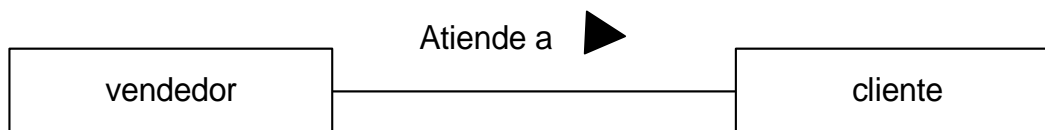


Figura 2.5. La asociación se representa con una línea y un triángulo que indica la dirección.

También pueden asociarse varias clases con una en particular, o tener multiplicidad, esto quiere decir que se pueden asociar una clase con otra o con varias dependiendo del caso.

2.4.2.2 Agregación

Una asociación normal entre dos clases representa una relación estructural entre iguales, es decir, ambas clases están conceptualmente en el mismo nivel, sin ser ninguna más importante que la otra. A veces, se desea modelar una relación “todo/parte”, en la cual una clase representa una cosa grande (el todo), que consta de elementos más pequeños (las partes).

Se especifica añadiendo a una asociación normal un rombo vacío en la parte del todo, a esto se le llama agregación y se observa en la figura 2.6.



Figura 2.6. La agregación se hace añadiendo a una asociación normal un rombo vacío en la parte del todo

2.4.2.3 Composición

Una composición es un tipo muy representativo de una agregación. Cada componente dentro de una composición puede pertenecer tan solo a un todo. El símbolo de una composición es el mismo que el de una agregación, excepto que el rombo está relleno.

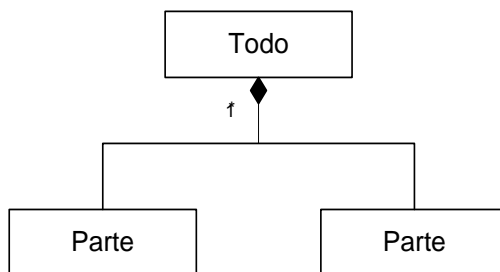


Figura 2.7. En la composición las partes en su conjunto integran un todo.

A continuación se muestra un ejemplo de un diagrama de clases completo para una interfaz.

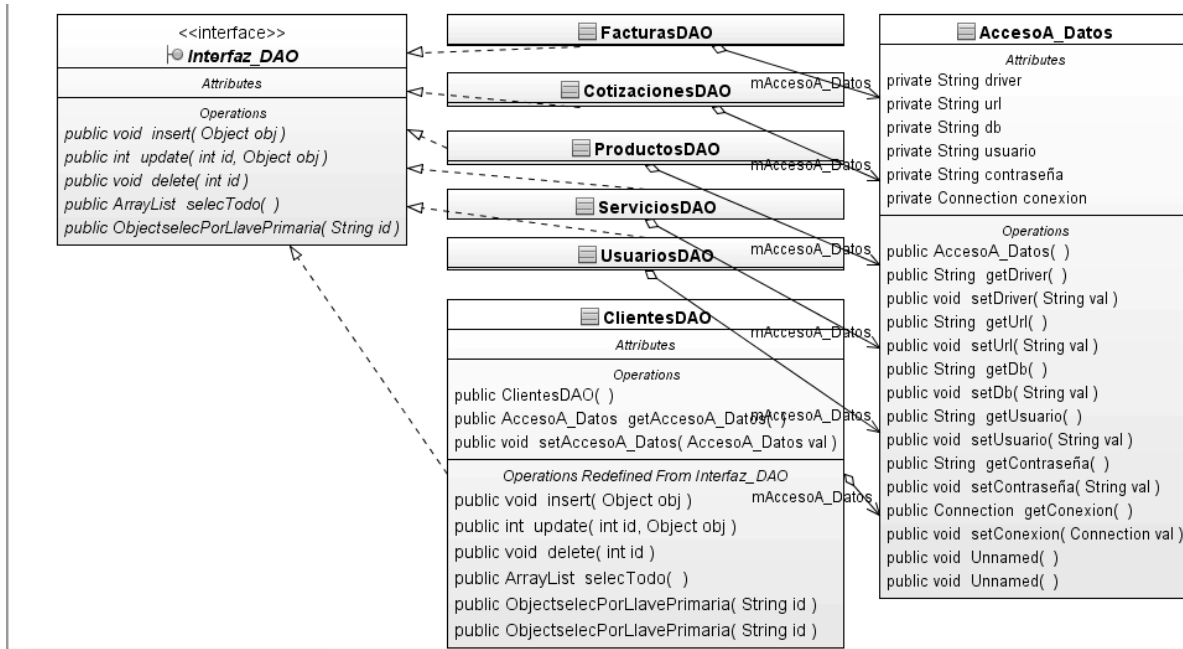


Figura 2.8. En esta imagen se puede ver un diagrama de clases, se observan clases con sus atributos y métodos, así como la asociación

2.4.3 Diagramas de secuencia

Un diagrama de secuencia muestra las interacciones entre objetos ordenadas mediante una secuencia temporal. Muestra los objetos que se encuentran en un escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad.

Los diagramas de secuencia sirven para mostrar la interacción con el usuario o con otro sistema. El propósito es capturar y documentar los requisitos de interfaz, pero no el mostrar como se va a implementar dicha interfaz.

Los diagramas de secuencia, o de interacción de objetos, se utilizan para validar los casos de uso. Documentan el diseño desde el punto de vista de los casos de uso. Observando qué mensajes se envían a los objetos, componentes o casos de uso y viendo de modo general cuanto tiempo consume el método invocado, los diagramas de secuencia ayudan a comprender los cuellos de botella potenciales, para así poder eliminarlos. A la hora de documentar un diagrama de secuencia resulta importante mantener los enlaces de los mensajes a los métodos apropiados del diagrama de clases. En la siguiente figura se muestra un diagrama de secuencia.

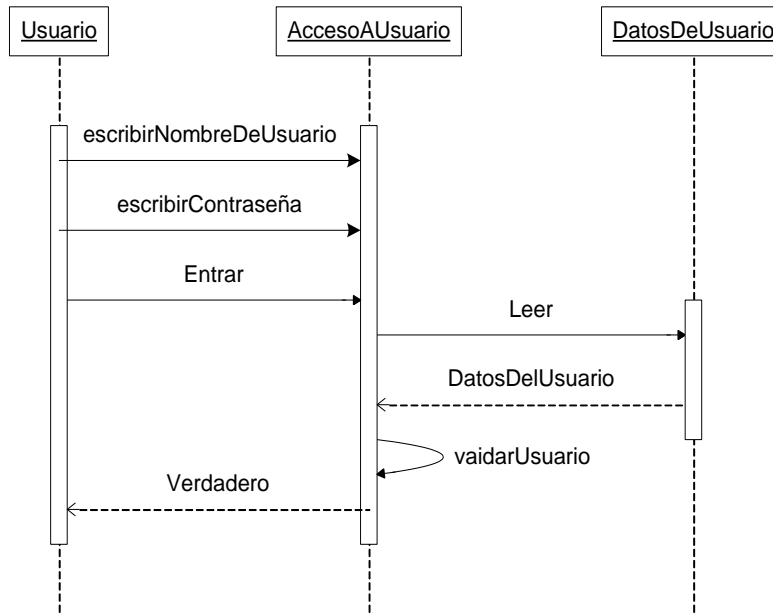


Figura 2.9. Ejemplo de un diagrama de secuencia para validar a un usuario.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

Hay dos tipos de mensajes: síncronos y asíncronos. Los síncronos se corresponden con llamadas a métodos del objeto que recibe el mensaje. El objeto que envía el mensaje queda bloqueado hasta que termina la llamada. Este tipo de mensajes se representan con flechas con la cabeza llena. Los mensajes asíncronos terminan inmediatamente, y crean un nuevo hilo de ejecución dentro de la secuencia. Se representan con flechas con la cabeza abierta.

También se representa la respuesta a un mensaje con una flecha discontinua.

Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos puede ser arbitraria. El método llamado o invocado, pertenece a la definición de la clase instanciada por el objeto en la recepción final del mensaje.

2.5 Lenguaje de programación Java

Para el desarrollo de la aplicación se decidió usar el lenguaje de programación java, ya que cuenta con una licencia GNU GPL de distribución libre y tiene gran soporte gracias a su robustez.

Java fue desarrollado por Sun Microsystems en 1995. Es un lenguaje de programación orientado a objetos, diseñado para distribuir contenidos a través de

la red. Una de sus principales características es que permite operar en forma independiente de la plataforma y del sistema operativo que se está utilizando.

Java pone una capa sobre cualquier plataforma de hardware y sobre cualquier sistema operativo, que permite que cualquier aplicación desarrollada en Java quede ligada únicamente a Java. Esto se refiere al concepto de JVM (Java Virtual Machine), el cual es un software que funciona como intérprete. Este software interpreta instrucciones para cualquier máquina sobre la que está corriendo, de manera que una misma aplicación pueda ejecutarse en una Mac, o en una PC con Linux o con Windows.

Al instalar Java, este actuará como una capa de abstracción entre un programa y el sistema operativo. Así brinda la ventaja de no tener que desarrollar varias versiones de la misma aplicación. Se puede decir que Java es lo más cercano que existe hoy a un lenguaje de computación universal, siempre exista una máquina virtual para esa plataforma.

Actualmente existen dos especificaciones en torno a Java:

J2EE (Java 2 Enterprise Edition): Está orientada al desarrollo de aplicaciones de propósito general.

J2ME (Java 2 Micro Edition): Nuevo estándar para dispositivos inalámbricos o handhelds y celulares:

Para el desarrollo de la aplicación en cuestión se utiliza la versión más actual en el momento Java2 con SDK 1.6.

Los programas de Java se pueden incluir embebidos en páginas HTML, y descargarse con navegadores Web, sin embargo, como lenguaje de propósito general, puede utilizarse para la programación de aplicaciones independientes, ya que se encuentra orientado a objetos desde su origen este es el caso de la aplicación en cuestión.

Una aplicación es un programa que se ejecuta sobre un sistema operativo, como los creados con cualquier otro lenguaje de programación.

Java es un lenguaje de programación orientado a objetos en donde se modela el mundo real, así cualquier cosa de este mundo real puede ser modelada como un objeto de una clase.

Los objetos tienen propiedades y atributos, así como comportamientos. Las propiedades y atributos se definen con datos, y los comportamientos con métodos. La definición genérica del objeto se define utilizando clases. Se puede decir que una clase es un molde para construir objetos.

Un objeto es la materialización de una descripción de una clase, y ese proceso recibe el nombre de instanciación de una clase.

Así mismo java es un lenguaje muy robusto y fiable, por ello Java contempla la verificación de posibles errores, y por ello a eliminado las propensiones a errores como los apuntadores, así como el manejo de la excepción en tiempo de ejecución. También utiliza el mecanismo de recolección de basura “garbage” en tiempo de ejecución, en vez de liberación de memoria una vez que el programa ha terminado.

2.5.1 Componentes swing

Swing es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, despletables y tablas. La cual facilita la realización de la interfaz de usuario y así se pueda dar mayor prioridad a la realización de la lógica de programación en vez de a la parte gráfica.

2.5.2 Patrones de ingeniería de software, interfaz DAO

Los patrones de ingeniería de software son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.

A lo largo de multitud de diseños de aplicaciones hay problemas que se repiten o que son análogos, que responden a un cierto patrón. Es bueno tener el conocimiento de algunos patrones y así hacer un uso de ellos.

Se dice que los patrones de diseño no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables. Han revolucionado el diseño orientado a objetos y todo buen arquitecto de software debería conocerlos.

Para este proyecto en particular se decidió utilizar el patrón de diseño DAO.

El patrón DAO (Data Access Object) es una solución para el acceso a datos desde una aplicación orientado a objetos y una base de datos relacional, empleando únicamente la interfaz de programación (API) nativa del manejador de base de datos, en este caso JDBC.

El Objetivo de este Patrón es encapsular la fuente de datos, ocultar los métodos para acceso a datos. También la “complejidad” del uso de JDBC a la capa de presentación o de negocio. El siguiente código muestra como se utiliza

```
package DAO;  
  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;
```

```

import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Date;
import sdmifs.Cotizacion;

public class CotizacionDAO implements InterfazDAO {

    private AccesoADatos db;

    public CotizacionDAO () {
        db = new AccesoADatos();
    }

    public void insert (Object obj) throws SQLException {
        String sql = "insert into cotizaciones(idclave,idlogin," +
            " monto_cot,fecha_cot) values(?,?,?,?);";
        PreparedStatement pstmt = db.getConexion().prepareStatement(sql);
        pstmt.setInt(1, ((Cotizacion) obj).getIdclave());
        pstmt.setString(2, ((Cotizacion) obj).getIdLogin());
        pstmt.setDouble(3, ((Cotizacion) obj).getMontoCot());
        pstmt.setString(4, ((Cotizacion) obj).getFechaCot());

        pstmt.executeUpdate();
    }

    ...

}

```

2.5.3 Clases abstractas e interfaces

Una clase abstracta es una clase que no se puede instanciar y únicamente se usa para definir subclases y se puede reconocer cuando alguno de sus métodos no tiene implementación, se utiliza cuando se desea definir una abstracción que englobe objetos de distintos tipos y se quiera hacer uso de polimorfismo.

Una interfaz es una clase completamente abstracta. En java las interfaces se declaran con la palabra reservada interface de manera similar en las clases abstractas con abstract.

Para la declaración de una interfaz solo pueden aparecer las declaraciones de sus métodos y las definiciones de las constantes simbólicas.

2.5.4 JDBC

JDBC (Java Database Connectivity), es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utiliza SQL del modelo de base de datos que se esté utilizando.

Es una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar.

Para usar una base de datos en particular, el usuario ejecuta su programa junto con la biblioteca de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para esto se deben de proveer el localizador a la base de datos y los parámetros de conexión específicos. Una vez accediendo se puede realizar cualquier tipo de tareas con la base de datos, tareas a las que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

Características de JDBC:

- Desarrollado por SUN
- Es una especificación de cómo hacer conexión a bases de datos
- Los fabricantes de SMBD escribieron controladores que se adecuaron a la especificación JDBC
- Es un traductor que convierte los mensajes propietarios de bajo nivel del SMBD a mensajes de bajo nivel comprensibles a la API JDBC y viceversa.
- Es una especificación para acceder a diferentes SMBD.
- Basada 100% en java.
- Compuesto por dos partes: API y Driver

Los pasos para establecer una conexión a la base de datos según JDBC son:

- Cargar el Driver
- Definir la URL de conexión.
- Establecer la conexión.
- Ejecutar manipulaciones y consultas.
- Procesar resultados.
- Cerrar la Conexión

2.5.5 NetBeans

Para el desarrollo del proyecto se decidió utilizar un entorno integrado de desarrollo gratuito, de código abierto para desarrolladores de software como lo es NetBeans.

NetBeans permite que las aplicaciones sean desarrolladas a partir de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos.

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

La plataforma de NetBeans ofrece servicios para las aplicaciones de escritorio, permitiendo desarrollar y enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están: la administración de las interfaces de usuario como menús y barras de herramientas, administración de las configuraciones del usuario, administración del almacenamiento, administración de ventanas, Framework basado en asistentes paso a paso.

Para este proyecto se utiliza la versión NetBeans 6.5 que se puede descargar libremente desde su página en Internet.

Una vez contando con las herramientas necesarias para desarrollar el proyecto el siguiente paso es el análisis para resolver los requerimientos a detalle y proceder con el diseño del sistema, temas a tratar en el capítulo siguiente.

Capítulo 3

Análisis y diseño del sistema para la DMIFS

Parte fundamental en el ciclo de desarrollo del software es el análisis en el cual se descompone el problema en partes para facilitar su solución, y así tener las bases para un buen diseño.

3.1 Análisis

Es necesario determinar que es lo que en realidad se pretende construir y para esto se debe realizar un análisis profundo de los requerimientos que se necesitan para el sistema. Qué elementos intervienen en el sistema que se va a desarrollar, así como su estructura, buscando a detalle las funcionalidades que debe de cumplir y así dar una descripción clara del sistema a construir.

3.1.1 Entrevistas a usuarios y observación

Una parte muy importante son las entrevistas con los usuarios que se hicieron en donde se plantaron preguntas tales como: ¿qué es lo que necesita?, ¿cuál es el proceso que le lleva más tiempo realizar?, ¿cuál cree que son las prioridades para un mejor desempeño en la empresa?, ¿cómo manejan su información actualmente?, entre otras. Y de estas preguntas surgen otras preguntas más específicas para conocer las necesidades más importantes que se deban resolver.

Otro aspecto es la observación la cual se realizó en distintas visitas al lugar de trabajo para conocer el funcionamiento que se tiene en la actualidad y así poder detectar las necesidades del área DMIFS.

3.1.2 Sistema actual

Se conoció la forma de trabajo de la DMIFS que se encarga de la fabricación, adaptación, venta y mantenimiento de máquinas engargoladoras y selladoras de válvulas. En esta área se hacen los mantenimientos para tales máquinas, así como venta de maquinaria nueva y de refacciones.

Existen diferentes tipos de cotizaciones que realizan, pero se pueden englobar en dos tipos distintos, ya que una es de propósito general, en donde se cotiza maquinaria nueva así como otros productos y un tipo de cotización para mantenimientos de maquinaria.

Cuando mandan máquinas para mantenimiento a las oficinas, se envían al taller junto con un formato que los técnicos tienen que llenar en el cual se especifica por cada máquina cuáles son las refacciones y servicios que se necesitan para dar el mantenimiento y la máquina sea reparada. Cuando se completa el llenado de los formatos se mandan de regreso a las oficinas para la cotización del presupuesto

El manejo de información acerca de sus clientes se hace en procesadores de texto con sus datos organizados por carpetas.

De igual forma los datos de los productos y servicios, están guardados de esta forma en archivos de texto y hojas de cálculo

Para hacer una cotización se buscan los datos del cliente en un archivo, después dependiendo de lo que solicite, se escriben manualmente los productos y servicios que se requieran para la cotización y en una carpeta distinta se buscan sus correspondientes precios.

De la misma forma capturan los datos manualmente para realizar las facturas, esto lo hacen en una hoja de cálculo donde ya tienen un formato preestablecido.

Para consultar de forma rápida las facturas que ya han sido otorgadas a clientes en una libreta se lleva la información de: el cliente a quien fue efectuada la factura, la fecha y el monto.

3.1.3 Análisis de requerimientos

En el análisis de requerimientos se establecieron las reglas de negocio, así como la solución de requerimientos los alcances del sistema.

3.1.4 Reglas de negocio.

Las reglas de negocio a seguir para el proyecto son las siguientes:

1. Automatizar el proceso de cotizaciones, obteniendo un archivo por cotización.
2. Automatizar el proceso de facturación para poder imprimir las facturas de una forma más eficiente.
3. Llevar un control de clientes, productos y servicios en donde se pueda tener acceso a los datos correspondientes.

3.1.5 Solución de requerimientos, alcances y entregables

Requerimientos y alcances para el sistema:

- 1) Una base de datos para guardar la información referente a los clientes, productos, servicios, cotizaciones, facturas y usuarios.
- 2) La interfaz de usuario contará con:

Una interfaz grafica para la administración y control de los clientes:
(Altas, bajas, cambios, consultas).

Una interfaz para la administración y control de los productos y servicios (Altas, bajas, cambios, consultas).

3) Una interfaz gráfica para la realización de las cotizaciones en general y de las máquinas que llegan para mantenimiento:

Se podrá consultar y agregar el cliente a la cotización extrayendo sus datos de la base de datos.

Se podrán consultar de la base de datos y agregar a la cotización los productos y servicios con sus atributos.

4) Una interfaz gráfica que permita la elaboración e impresión de facturas.

3.2 Diseño

Con el análisis concluido el diseño fue el paso subsecuente comenzando por la base de datos y después la aplicación.

3.2.1 Base de datos

En esta parte se muestra el diseño de la base de datos, tanto su diseño conceptual como diseño lógico, el diseño físico se describe en el capítulo cuatro ya que forma parte del desarrollo.

3.2.1.1 Diseño conceptual, modelo Entidad-Relación

Con ayuda del análisis previo es importante identificar las entidades que serán la base fundamental para el diseño de la base de datos.

3.2.1.1.1 Identificación de entidades, sus atributos, relaciones y llaves primarias.

Lo primero es identificar las entidades, esto se logró conforme a: entrevistas con el cliente, cuestionarios, observación y requerimientos que se necesitaran, así se identificaron las siguientes entidades:

Cliente, Refacciones, Servicios, Usuarios, Facturas, Cotización como entidades fuertes y Teléfonos como una entidad débil dependiente de Cliente.

Identificación de atributos.

Una vez localizadas las identidades, el siguiente paso es la identificación de los atributos correspondientes:

Cliente

- Clave
- Razon_Social
- Responsable
- RFC
- Calle
- Numero
- Piso
- Departamento
- Colonia
- CP
- Deleg_Muni
- Entidad_F
- E_mail

Productos

- Codigo_P
- Producto
- Descripcion_P
- Precio

Facturas

- Folio
- idClave
- Total_Fac
- Fecha_Fac

Telef_Fax

- IdClave
- Tel1
- Ext1
- Tel2
- Ext2
- Fax

Usuarios

- Nombre
- Apepat
- Apemat
- Puesto
- login
- contraseña

Servicios

- Codigo_S
- Servicio
- Descripcion_S
- Precio

Cotización

- Num_Cot
- idClave
- Total_Cot
- Fecha_Cot

Identificación de relaciones:

A continuación se identificaron las relaciones existentes entre las entidades las cuales son:

Solicita: Clientes x Cotizaciones (Clientes solicitan cotizaciones).

Solicita: Clientes x Facturas

Tiene: Clientes x Telef_Fax

Realizadas por: Cotizaciones x Empleado

Realizadas por: Facturas x Empleados

Consultan: Empleados x Servicios

Consultan: Empleados x Productos

Consultan datos: Empleados x Clientes

Identificación de las llaves primarias:

Para identificar las llaves primarias se hizo conforme a conveniencia para el buen manejo de la información

- Clave para Clientes
- Codigo P para Productos
- Codigo S para Servicios
- Login para Usuarios
- Folio para Facturas
- Num_Cot para Cotización

3.2.1.1.2 Diagrama E - R

Con lo anterior se llega a la realización del diagrama Entidad – Relación, donde solamente se muestran los atributos que son llaves primarias en la figura 3.1.

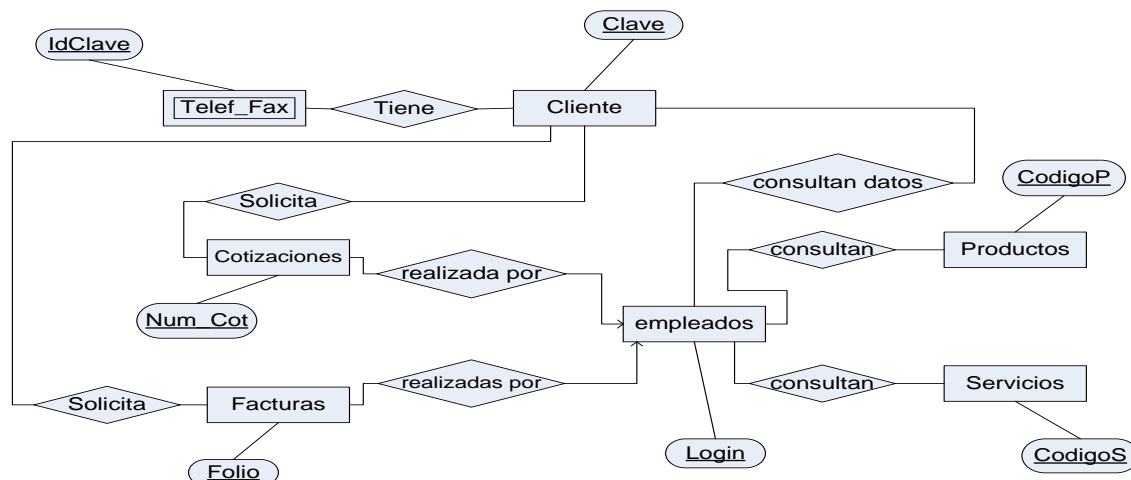


Figura 3.1. Diagrama E – R. Se muestran únicamente las llaves primarias de las entidades.

3.2.1.2 Diseño Lógico, modelo relacional.

Para pasar del modelo conceptual al lógico se hace la traducción de las entidades y relaciones a su equivalente en tablas de modelo relacional que se está usando.

3.2.1.2.1 Traducción de tipos de entidades y relaciones

A continuación se traducen las entidades al modelo relacional se muestra la descripción así como su representación en tablas.

Cliente (Clave, Razon_Social, RFC, Responsable, Calle, Número, Piso, Colonia, CP, Deleg_Mun, Ent_Federativa E_mail).

Cliente												
Clave	Razon_S	Resp	RFC	Calle	Numero	Piso	Dep	Col	CP	Deleg	Ent_Fed	E_mail

Figura 3.2. Tabla Cliente, para el registro de los clientes de la DMIFS

Telefonos (IdClave, Tel1, Ext1, Tel2, Ext2, Tel3, Ext3)

Telef_Fax					
IdClave	Tel1	Ext1	Tel2	Ext2	Fax

Figura 3.3. Tabla Telef_Fax, en la cuál se guardan los telefonos y fax de los clientes.

Productos(Codigo_P, Servicio, Descripcion_P, Precio_P).

Productos			
Codigo_P	Producto	Descripción_P	Precio_P

Figura 3.4. Tabla Productos, para el almacenamiento de productos.

Servicios(Codigo_S, Servicio, Descripcion_S, Precio_S).

Servicios			
Codigo_S	Servicio	Descripción_S	Precio_S

Figura 3.5. Tabla Servicios, para el almacenamiento de servicios.

Empleados(Nombre, Apellido_pat, Apellido_mat, Puesto, login, contraseña)

Empleados				
Nombre	Apepat	Apemat	login	contraseña

Figura 3.6. Los empleados serán registrados en la tabla empleados.

Facturas(Folio, idClave, Total_Fac, Fecha_Fac)

Facturas			
Folio	idClave	Total_Fac	Fecha_Fac

Figura 3.7. Tabla Facturas, servirá para almacenar datos que podrán ser consultados.

Cotizaciones(Num_Cot, idClave, Total_Cot, Fecha_Cot)

Cotizaciones			
Num_Cot	idClave	Total_Cot	Fecha_Cot

Figura 3.8. Tabla Cotizaciones, en la cual se guardarán datos que podrán ser consultados.

3.2.1.2.2 Diagrama del modelo relacional

Se presenta el diagrama del modelo relacional en el cual se puede observar que la tabla Telef_Fax está ligada a la tabla Clientes ya que depende de su llave primaria al ser una entidad débil. También se muestra a las tablas cotizaciones y facturas las cuales tienen como llave foránea a la llave primaria de la tabla clientes, así por ejemplo, si se desea consultar las cotizaciones y facturas de algún cliente en específico bastará con buscar por la llave primaria.

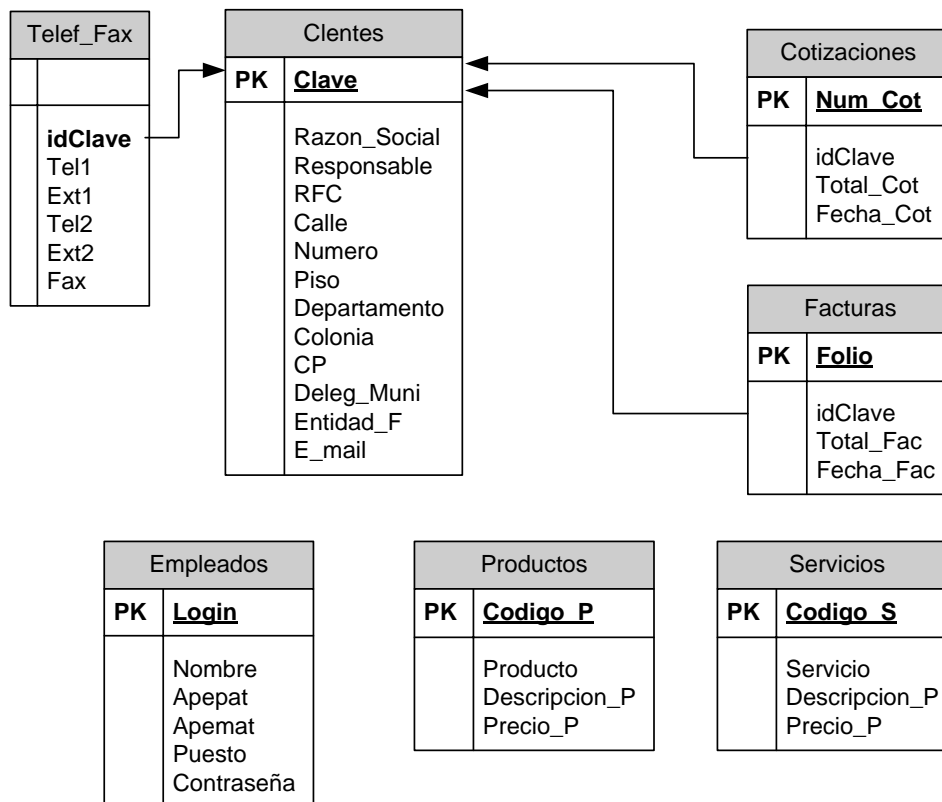


Figura 3.9. Diagrama del modelo relacional se muestra que las tablas cotizaciones y facturas tienen como llave foránea a la llave primaria de la tabla clientes y la tabla Telef_fax depende directamente de la tabla clientes. Así mismo las tablas: Empleados, Productos y Servicios son independientes.

3.2.2 Aplicación

El diseño de la aplicación se hizo con la metodología UML, principalmente con tres tipos de diagramas: de casos de uso, de secuencia y de clases. También se muestra como se diseñó la interfaz gráfica así como se diseñaron los reportes.

3.2.2.1 Casos de uso, descripción y diagramas

Para comenzar el diseño de la aplicación se describen los casos de uso utilizados para el sistema. Se establece: el nombre del caso de uso, sus objetivos asociados, las precondiciones que se deban de cumplir para que el caso de uso siga adelante, los pasos o acciones a seguir y verificar si existe alguna poscondición o excepciones que se puedan presentar. A continuación se muestran las tablas de descripciones de los casos de uso.

01	Acceder al sistema	
Objetivos Asociados	Dar seguridad a la información	
Requisitos Asociados	El sistema deberá comportarse tal como se describe en el siguiente caso de uso para el acceso al usuario	
Precondición	Ser usuario del sistema	
Secuencia Normal	Paso	Acción
	1	El usuario procede a introducir su nombre de usuario y contraseña para el acceso al sistema
	2	El sistema da acceso al usuario
Poscondición	Tener un nombre de usuario y contraseña para el sistema correcto	
Excepciones	Paso	Acción
	2	Si el nombre de usuario o contraseña son incorrectos en tres ocasiones, no se le da acceso a usuario
	2	Si se desea cancelar la operación, el sistema cancela la operación y el caso de uso termina

Tabla 3.1. Descripción del caso de uso 01 (Acceder al sistema).

02	Movimientos de clientes	
Objetivos Asociados	Gestionar los clientes de la división	
Requisitos Asociados	El sistema deberá comportarse tal como se describe en el siguiente caso de uso para clientes	
Precondición	Ninguna	
Secuencia Normal	Paso	Acción
	1	El usuario solicita al sistema comenzar el proceso de alta, baja, cambio, consulta del cliente
	2	Se realiza el movimiento

	3	El sistema avisa que ya está hecho el movimiento deseado
Poscondición	El movimiento debe ser válido	
Excepciones	Paso	Acción
	2	Si el movimiento es ilegal el sistema avisa la causa del error, en caso de que en una Alta ya exista el sistema avisa, en caso de que en una Baja no exista el sistema avisa, en caso de datos no permitidos se avisa al usuario.
	2	Si se desea cancelar la operación, el sistema cancela la operación y el caso de uso termina

Tabla 3.2. Descripción del caso de uso 02 (Movimiento de Clientes).

03	Movimientos de Productos	
Objetivos Asociados	Gestionar los Productos de la empresa	
Requisitos Asociados	El sistema deberá comportarse tal como se describe en el siguiente caso de uso para movimientos de productos	
Precondición	Ninguna	
Secuencia Normal	Paso	Acción
	1	El usuario solicita al sistema comenzar el proceso de alta, baja, cambio o consulta del Producto
	2	Se realiza el movimiento
	3	El sistema avisa que ya está hecho el movimiento deseado
Poscondición	El movimiento debe ser Valido	
Excepciones	Paso	Acción
	2	Si el movimiento es ilegal el sistema avisa la causa del error, en caso de que en una Alta ya exista o una Baja no exista se manda un mensaje, en caso de datos no permitidos el sistema avisa el error.
	2	Si se desea cancelar la operación, el sistema cancela la operación y el caso de uso termina

Tabla 3.3. Descripción del caso de uso 03 (Movimientos de Productos).

04	Movimientos de servicios	
Objetivos Asociados	Gestionar los Servicios de la empresa	
Requisitos Asociados	El sistema deberá comportarse tal como se describe en el siguiente caso de uso para movimientos de servicios	
Precondición	Ninguna	
Secuencia Normal	Paso	Acción

	1	El usuario solicita al sistema comenzar el proceso de alta, baja, cambio o consulta del Servicio
	2	Se realiza el movimiento
	3	El sistema avisa que ya está hecho el movimiento deseado
Poscondición	El área está registrada en el sistema	
Excepciones	Paso	Acción
	2	Si el movimiento es ilegal el sistema avisa la causa del error, en caso de que en una Alta ya exista o en caso de que en baja no exista el sistema avisa, en caso de datos no permitidos el sistema avisa el error.
	2	Si se desea cancelar la operación, el sistema cancela la operación y el caso de uso termina

Tabla 3.4. Descripción del caso de uso 04 (Movimientos de Servicios).

05	Hacer cotizaciones	
Objetivos Asociados	Si el cliente acepta la cotización se procede con la facturación	
Requisitos Asociados	El sistema deberá comportarse tal como se describe en el siguiente caso de uso para las cotizaciones	
Precondición	Se ha solicitado una cotización	
Secuencia Normal	Paso	Acción
	1	Obtener los datos del cliente que pidió la cotización
	2	Si el cliente no existe se procede a dar de alta. (caso de uso 02)
	3	Obtener los datos de los Productos y Servicios solicitados
	4	Si los productos o servicios no existen se procede a dar de alta. (caso de uso 03 o 04 según sea el caso)
	5	El sistema genera la cotización conforme a los datos agregados.
	6	Se puede mandar a imprimir la cotización, y se guarda como archivo.
Poscondición	Los Productos y Servicios deben de ser válidos	
Excepciones	Paso	Acción
	4	Si no se cuenta con los productos o servicios, no se incluyen o se cancela la cotización.
	6	Si se desea se puede imprimir o solamente se guarda el archivo para después imprimirlo o hacer uso de él.

Tabla 3.5. Descripción del caso de uso 05 (Hacer cotizaciones).

06	Hacer facturas	
Objetivos Asociados	Guardar datos referentes a las facturas para su consulta	
Requisitos Asociados	El sistema deberá comportarse tal como se describe en el siguiente caso de uso para las facturas	
Precondición	Ninguna	
Secuencia Normal	Paso	Acción
	1	Si hay una cotización previa, se consulta para generar más fácil la factura y se procede al paso 3 en caso contrario al paso 2
	2	Se hace la factura llenando los campos correspondientes al Cliente, Productos, Servicios, con sus precios para hacer un total parcial
	3	Una vez teniendo la factura completa se manda a imprimir
Poscondición	El área está registrada en el sistema	
Excepciones	Paso	Acción
	2	Si se desea cancelar la operación, el sistema cancela la operación y el caso de uso termina

Tabla 3.6. Descripción del caso de uso 06 (Hacer Facturas).

07	Generar un reporte sobre las cotizaciones realizadas en un periodo determinado	
Objetivos Asociados	Presentar los informes requeridos	
Requisitos Asociados	El sistema deberá comportarse tal como se describe en el siguiente caso de uso para las cotizaciones	
Precondición	Ninguna	
Secuencia Normal	Paso	Acción
	1	Proporcionar de que fecha a que fecha se requiere el reporte.
	2	Pedir se procese la información obtenida para dar paso al reporte.
	4	El sistema presenta el reporte.
Poscondición	El área está registrada en el sistema	
Excepciones	Paso	Acción
	2	Si se desea cancelar la operación, el sistema cancela la operación y el caso de uso termina
	1	Si las fechas son erróneas o no aparecen en el sistema se indica que hay un error.

Tabla 3.7. Descripción del caso de uso 07 (Generar un reporte sobre las cotizaciones realizadas en un periodo determinado).

08	Consultar Datos Sobre las Facturas	
Objetivos Asociados	Obtener información rápida referente a las facturas expedidas	
Requisitos Asociados	El sistema deberá comportarse tal como se describe en el siguiente caso de uso para las facturas	
Precondición	Se debe tener algún dato referente a la factura para hacer la consulta	
Secuencia Normal	Paso	Acción
	1	Proporcionar alguno de estos datos al sistema para la búsqueda de la factura: Folio, Fecha, Clave del Cliente,
	2	Pedir se procese la información obtenida para dar paso a la consulta.
	4	El sistema presenta el reporte.
Poscondición	El área está registrada en el sistema	
Excepciones	Paso	Acción
	2	Si se desea cancelar la operación, el sistema cancela la operación y el caso de uso termina
	1	Si los datos proporcionados son erróneos el sistema se indica que hay un error.

Tabla 3.8. Descripción del caso de uso 08 (Consultar datos sobre las facturas).

09	Movimientos de Usuarios	
Objetivos Asociados	Administrar Usuarios	
Requisitos Asociados	El sistema deberá comportarse tal como se describe en el siguiente caso de uso de movimientos de usuarios	
Precondición	El Administrador Usuario tiene el usuario y contraseña por defecto	
Secuencia Normal	Paso	Acción
	1	El usuario Administrador accede al sistema con su cuenta y contraseña.
	2	El usuario Administrador del sistema procede a hacer un movimiento de alta, baja, cambio o consulta de los usuarios del sistema.
	3	El sistema avisa que ya está hecho el movimiento deseado
Poscondición	Gestionar los Clientes de la empresa	
Excepciones	Paso	Acción
	2	Si el movimiento es ilegal el sistema avisa la causa del error, en caso de que en una Alta ya exista el sistema avisa o en caso de que en una baja no exista, si los datos no son permitidos el sistema avisa el error.

	2	Si se desea cancelar la operación, el sistema cancela la operación y el caso de uso termina
--	---	---

Tabla 3.9. Descripción del caso de uso 09 (Movimientos de Usuarios).

Una vez realizada la descripción de los casos de uso lo siguiente fue el diseño de los diagramas de casos de uso en donde se muestra de forma gráfica lo descrito con anterioridad.

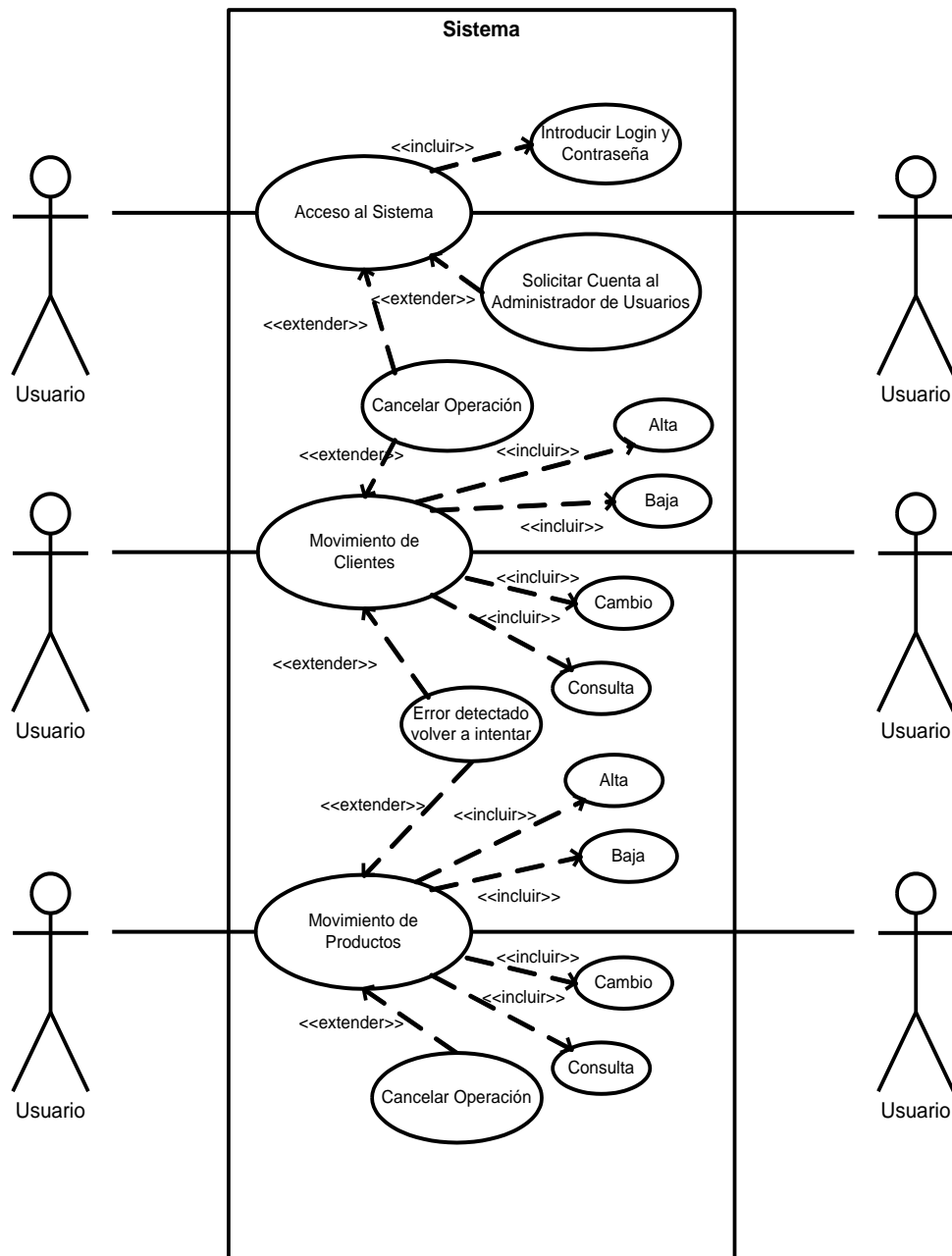


Figura 3.10. Se observan los casos de uso: 01 para acceder al sistema es necesario contar con contraseña asignada por el administrador, 02 en el cual se pueden realizar movimientos de clientes y 03 para movimientos de productos.

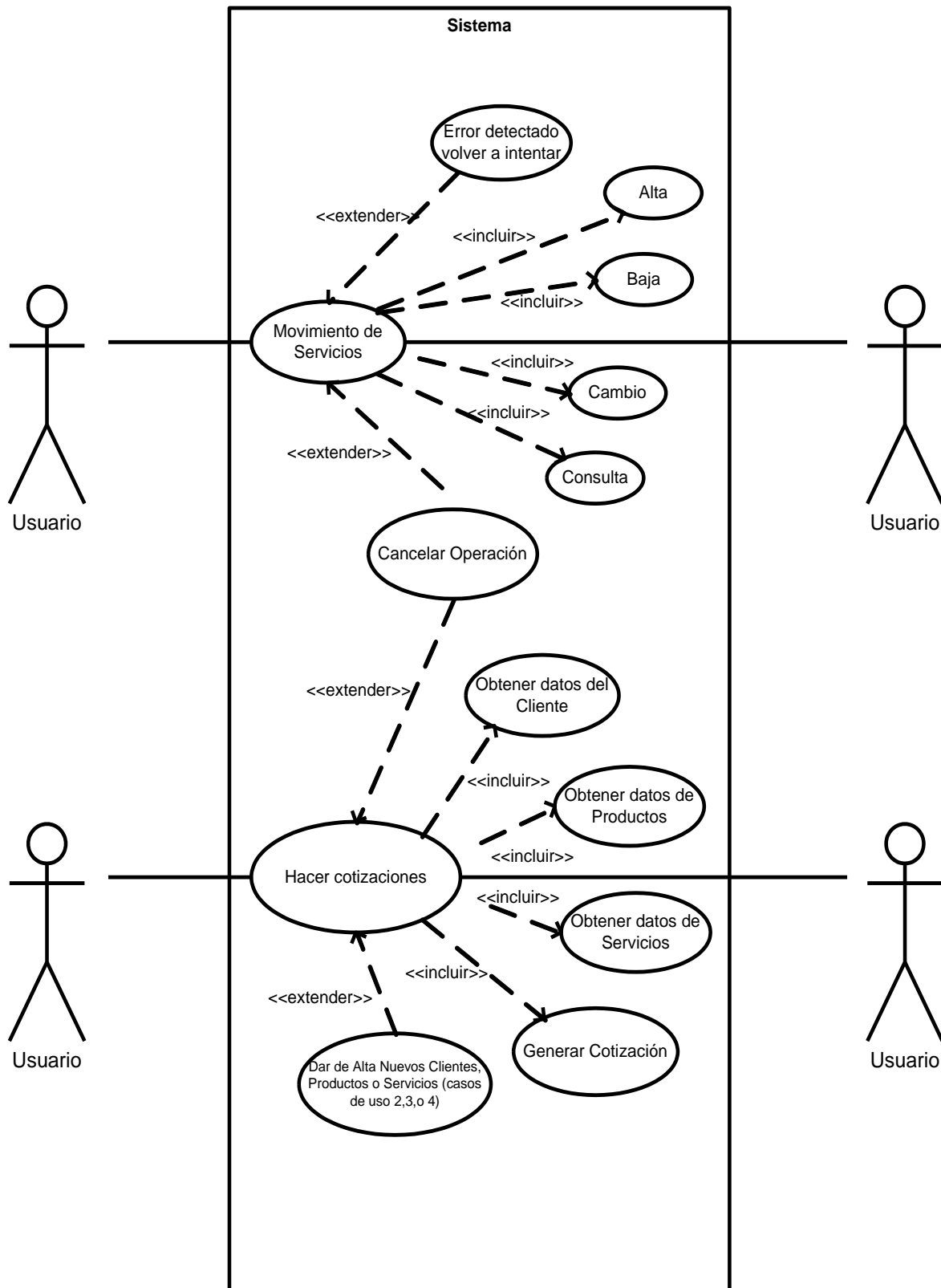


Figura 3.11. En esta imagen se muestran los casos de uso 04 para movimientos de servicios y 05 para generar cotizaciones.

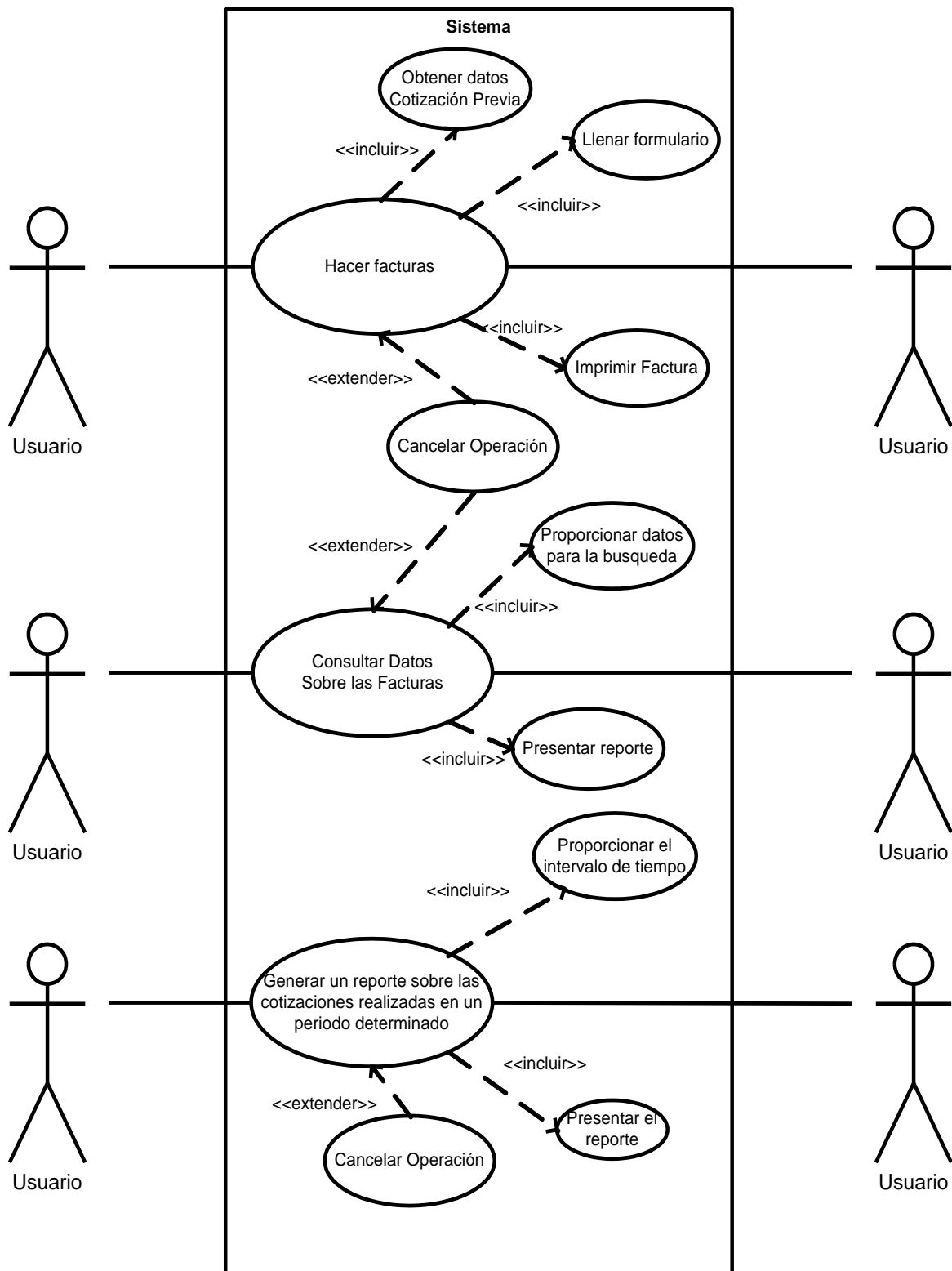


Figura 3.12. En la imagen se muestran los casos de uso 06 para la realización de facturas, 07 para consultar datos sobre las facturas y 08 en donde se generan reportes.

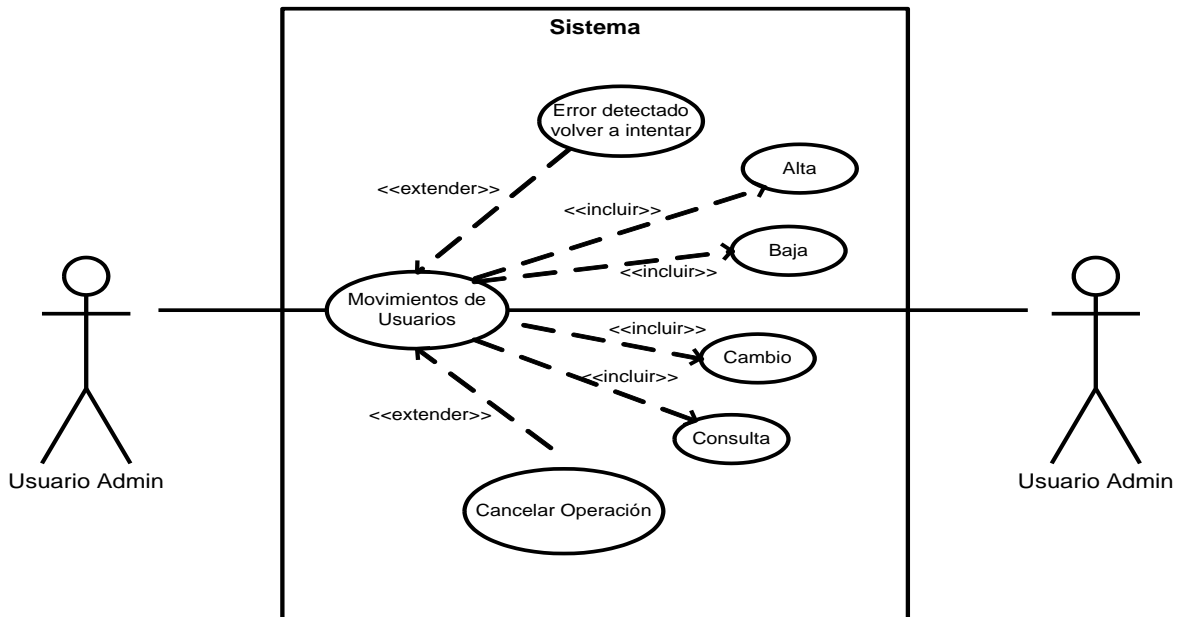


Figura 3.13. El caso de uso 09 trata de los movimientos que se pueden realizar de usuarios.

3.2.2.2 Diagramas de secuencia

Por cada caso de uso se generó un diagrama de secuencia, como se describió en el capítulo dos, los diagramas de secuencia se utilizan para validar los casos de uso y documentan el diseño desde el punto de vista de los casos de uso. Observando qué mensajes se envían a los objetos, componentes o casos de uso y viendo de modo general cuánto tiempo consume el método invocado. Los diagramas de secuencia ayudan a comprender los cuellos de botella potenciales.

A continuación por cada diagrama de caso de uso se efectuó su diagrama de secuencia correspondiente y se observan en las figuras subsiguientes.

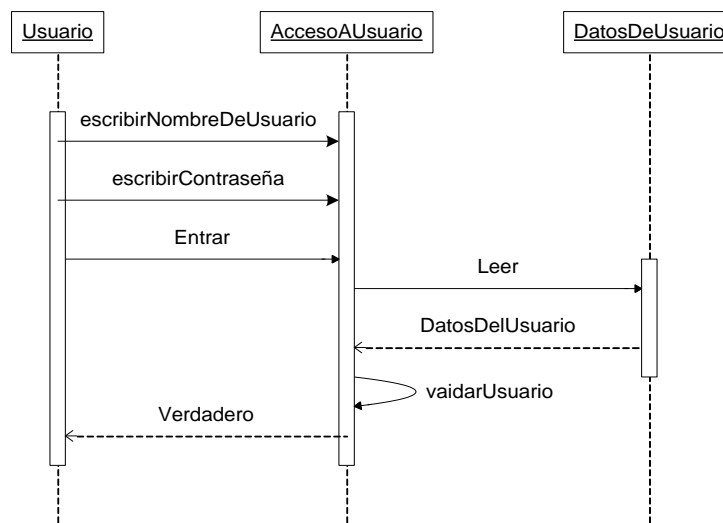


Figura 3.14. Diagrama de secuencia Acceso Usuario.

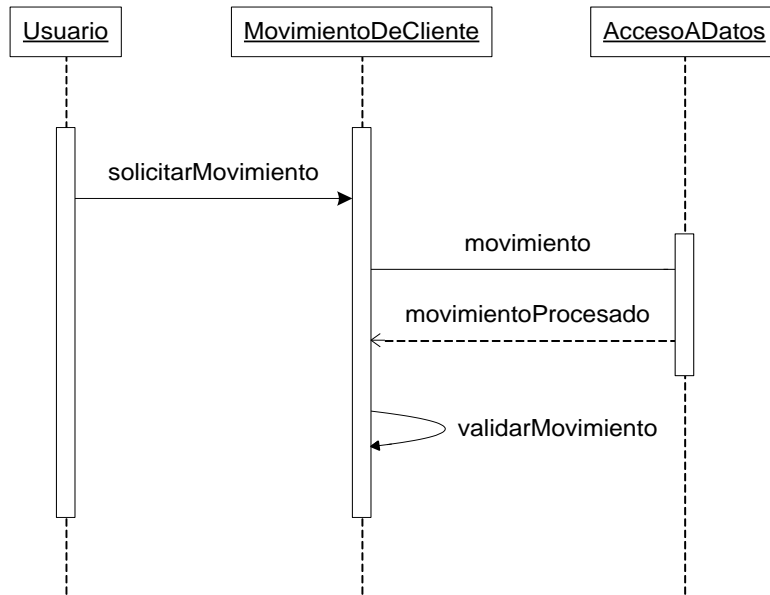


Figura 3.15. Diagrama de secuencia movimiento de cliente.

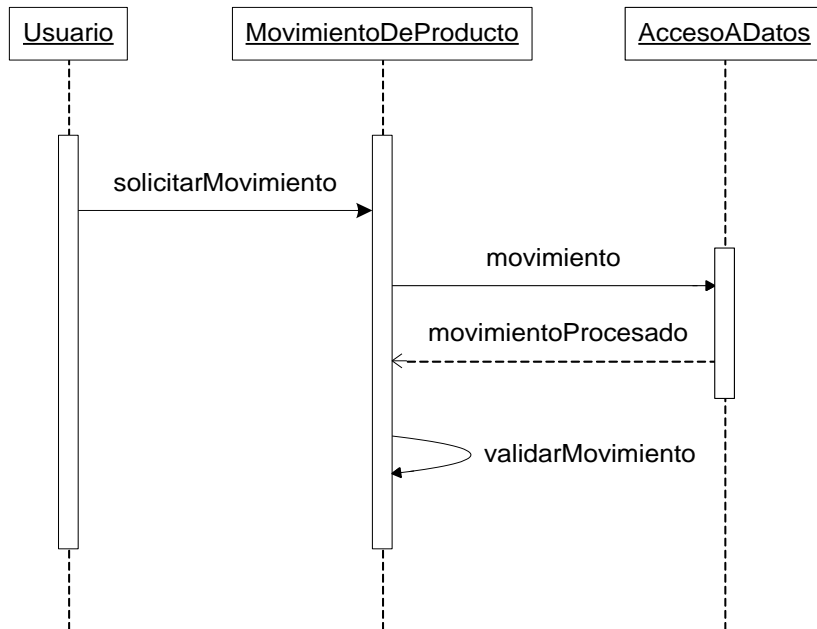


Figura 3.16. Diagrama de secuencia movimiento de producto.

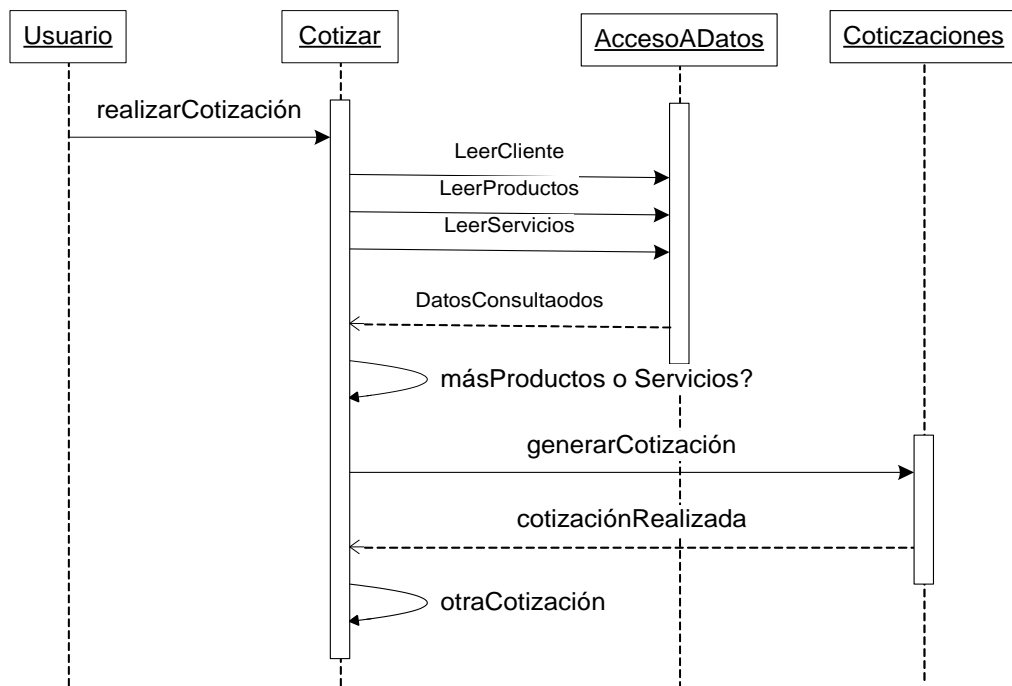


Figura 3.17. Diagrama de secuencia de hacer cotizaciones.

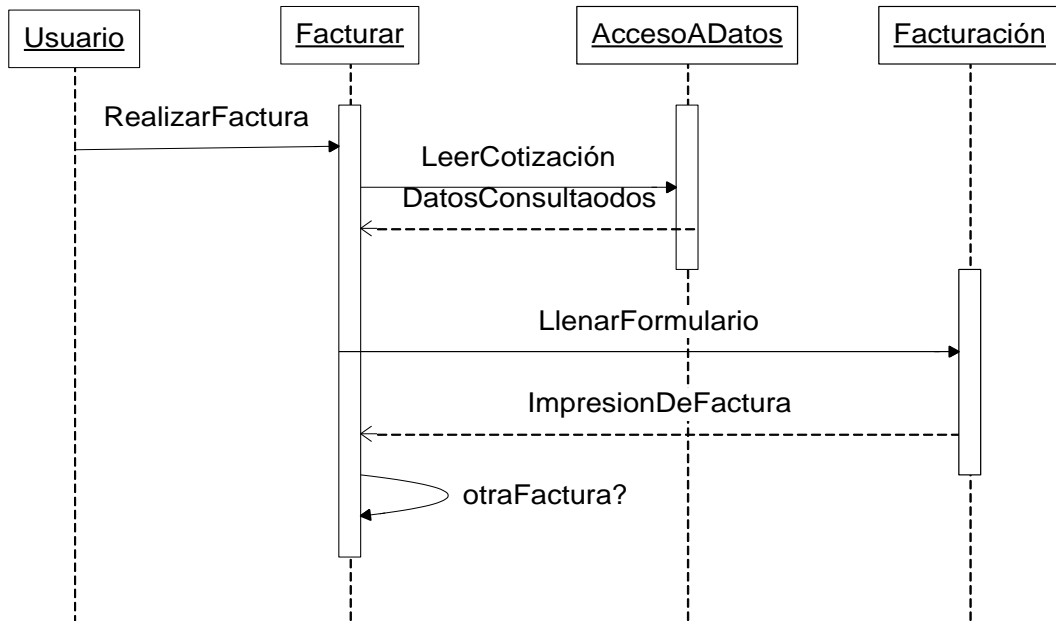


Figura 3.18. Diagrama de secuencia de hacer facturas.

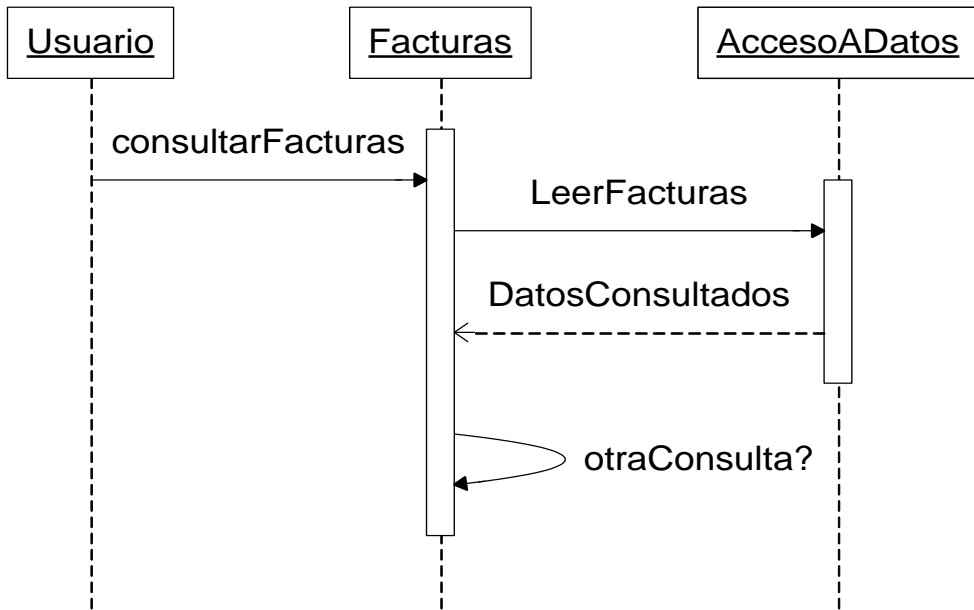


Figura 3.19. Diagrama de secuencia de consultar facturas.

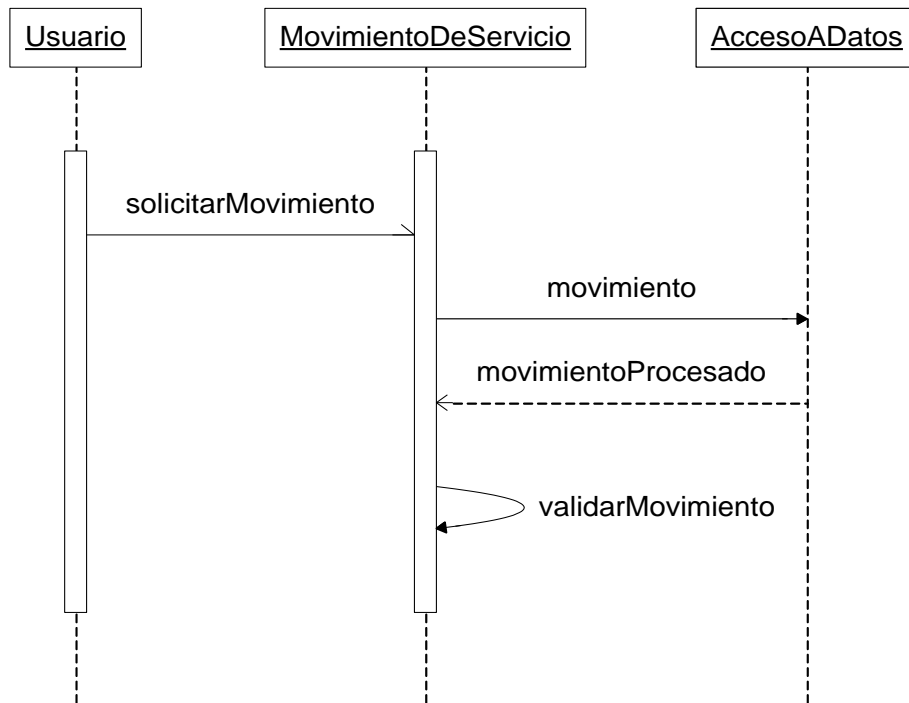


Figura 3.20. Diagrama de secuencia de movimiento de servicio.

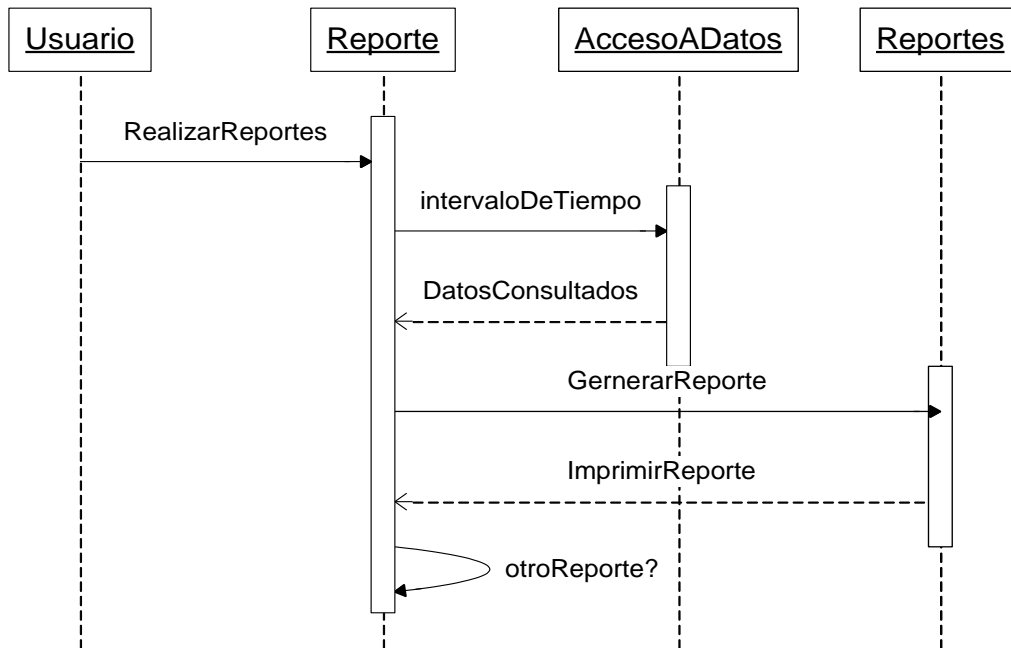


Figura 3.21. Diagrama de secuencia de generar un reporte sobre las cotizaciones realizadas en un periodo determinado.

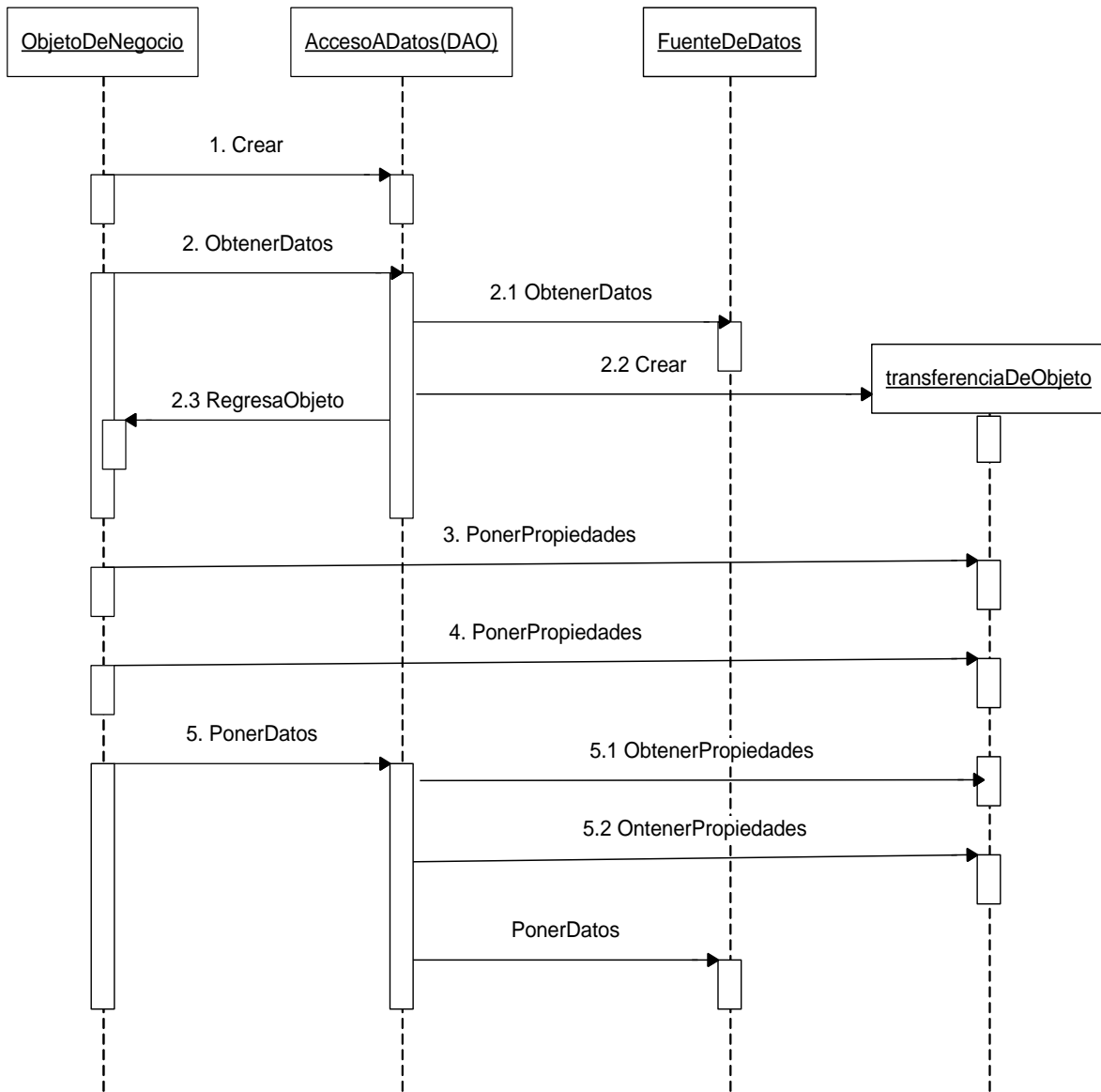


Figura 3.22. Diagrama de secuencia de la interfaz DAO (patrón de ingeniería de software usado para el sistema).

3.2.2.3 Diagramas de clases

A continuación se presentan los diagramas de clases, de las clases más importantes utilizadas para el desarrollo del proyecto

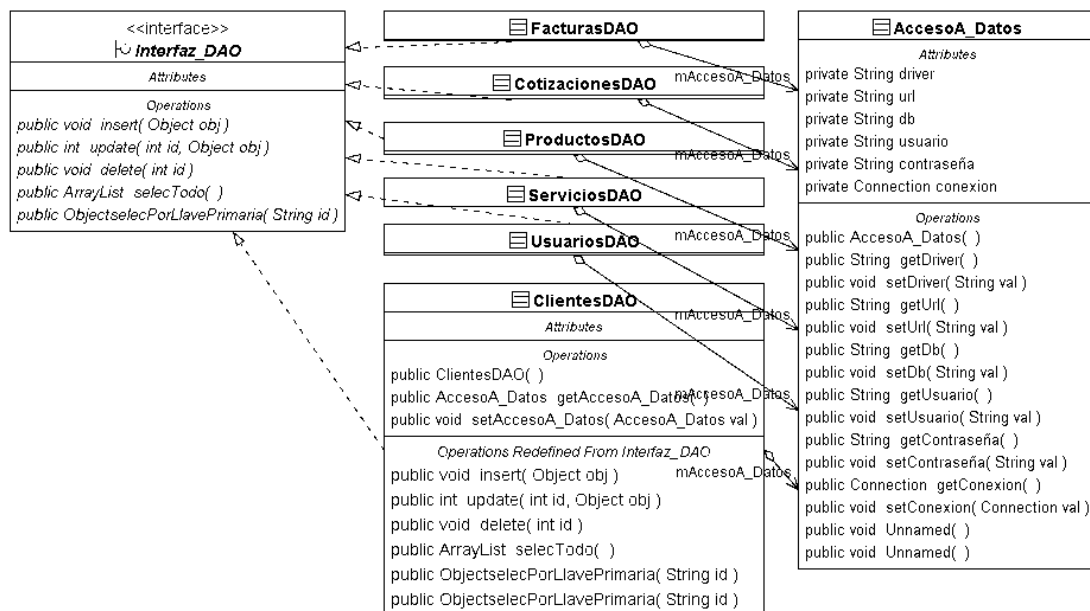


Figura 3.23. Diagrama de clase de la interfaz DAO. Se observan clases con sus atributos y métodos, así como la asociación existente.

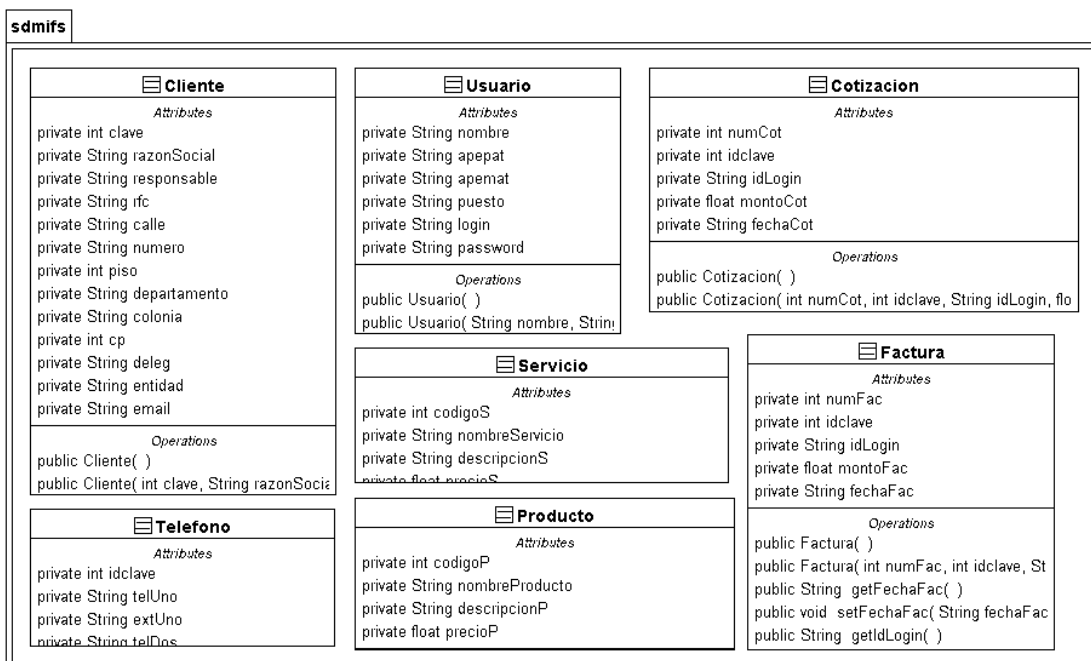


Figura 3.24. Diagrama de clase del paquete sdmifs que incluye las clases Cliente, Usuario, Cotización, Servicio, Teléfono, Producto y Factura.

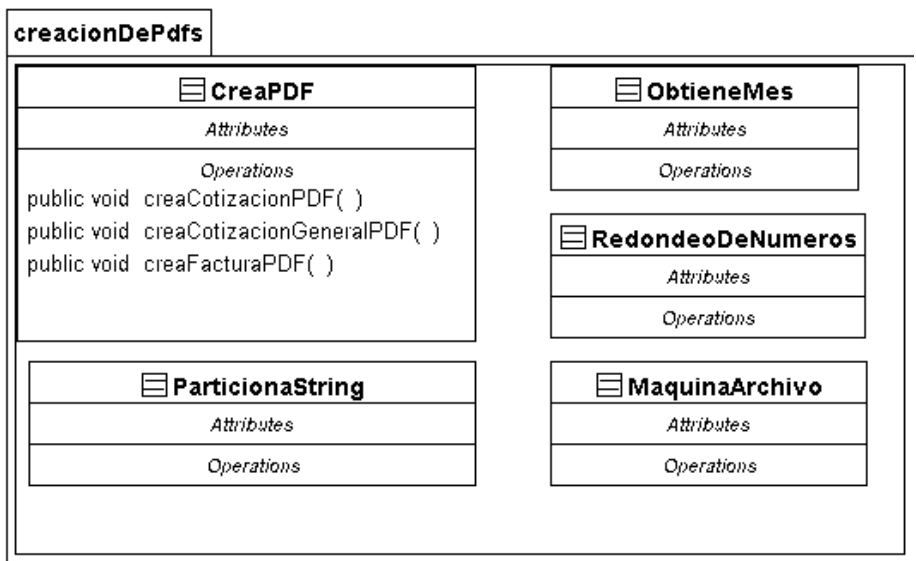


Figura 3.25. Diagrama de clases del paquete creacionDePdfs: se pueden notar la clase CreaPDF así como sus métodos creaCotización, creaCotizaciónGeneralPDF y creaFacturaPDF. Así como algunas clases auxiliares.

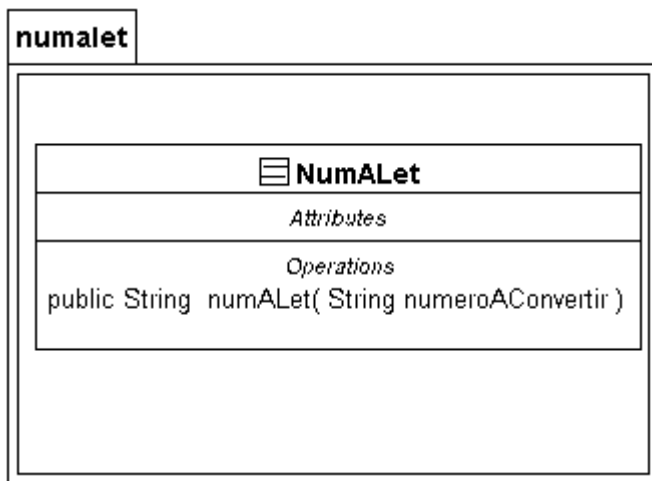


Figura 3.26. Esta clase permite convertir una cantidad numérica a su equivalente en escrito con letras.

3.2.2.4 Interfaz de usuario

Para la interfaz de usuario se utilizó la ayuda del entorno de desarrollo integrado NetBeans y java swing los cuáles facilitan el diseño gráfico de la aplicación para poder enfocarse en la parte lógica de los requerimientos. La interfaz de usuario se diseño de una forma muy amigable e intuitiva, de buena interacción con el usuario ya que provee de advertencias y diálogos en caso de omitir algún paso de importancia. A continuación se muestra la ventana de facturación en la figura 3.27. Como un ejemplo del diseño de la interfaz.

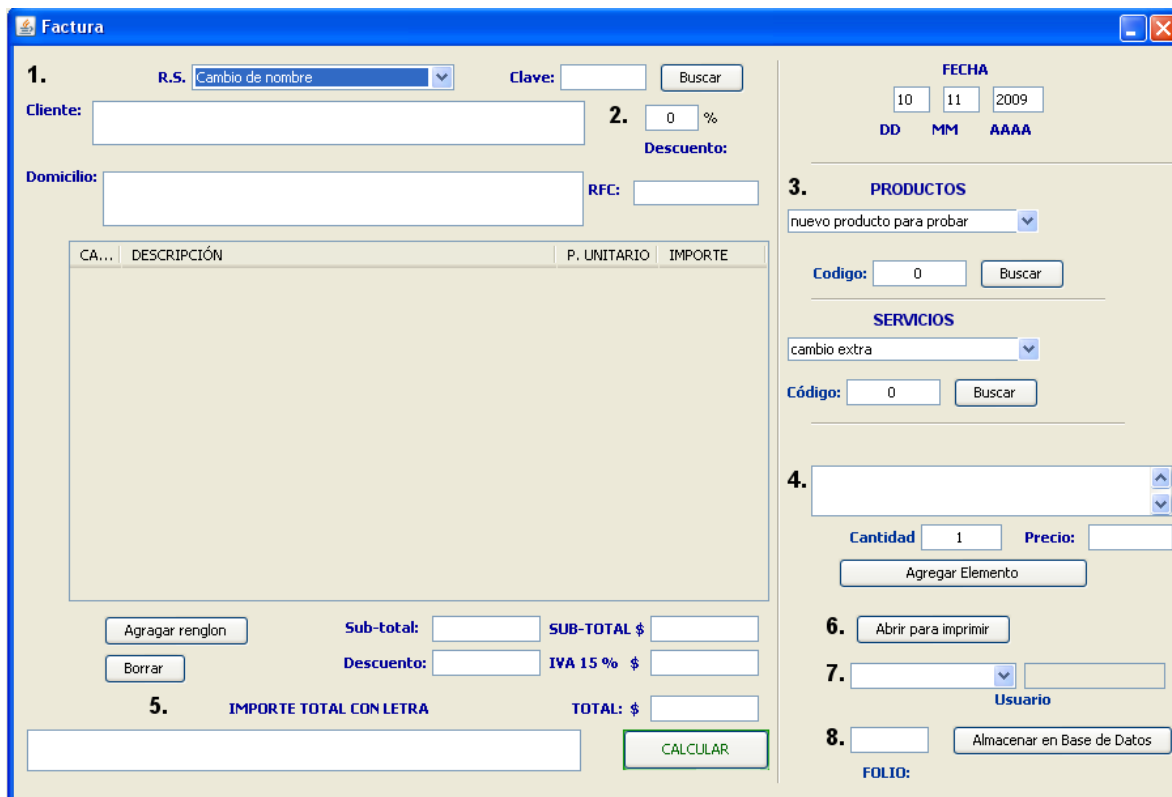


Figura 3.27. Ventana de facturación de la aplicación diseñada con la ayuda de Netbeans y swing de java.

3.2.2.5 Reportes

Los reportes más importantes según los requerimientos del sistema son: las cotizaciones en general, cotizaciones de mantenimientos de maquinaria, así como facturación.

La cotización en general se diseñó de la siguiente manera:

Logo	DIVISION MAQUINARIA INDUSTRIAL <small>FABRICACION Y ADAPTACION DE ACCESORIOS PARA MAQUINAS ENGARGOLADORAS Y SELLADORAS DE VALVULAS</small>	Logo	
		Fecha No. De Cotización Por quien fue cotizado	
Razón Social			
Dirección			
At'n de:			
Descripción de la cotización			
Cantidad	Descripción	Precio Unitario	Importe
			Total: \$0.00
ATENTAMENTE Nombre de quien cotiza Cargo Dirección de las oficinas			

Figura 3.28. Diseño de la cotización en general.

Para las cotizaciones de mantenimientos de maquinaria el formato se estableció de la siguiente manera:

Logo	DIVISION MAQUINARIA INDUSTRIAL <small>FABRICACION Y ADAPTACION DE ACCESORIOS PARA MAQUINAS ENGARGOLADORAS Y SELLADORAS DE VALVULAS</small>				Logo
					Fecha No. De Cotización Por quien fue cotizado
Razón Social					
Dirección					
At'n de:					
Descripción de la cotización					
Maquina #	No, Piezas	Descripción	Precio	Maquina #	No, Piezas
Total:\$0.00			Total:\$0.00		
Maquina #	No, Piezas	Descripción	Precio	Maquina #	No, Piezas
Total:\$0.00			Total:\$0.00		
Total: \$0.00					
ATENTAMENTE					
Nombre de quien cotiza Cargo					
Dirección de las oficinas					

Figura 3.29. Diseño de la cotización de mantenimientos de maquinaria.

Por último ya que las facturas están impresas lo que se hizo fue calcular las áreas de impresión.

Con el diseño concluido lo siguiente es el desarrollo del sistema, en el siguiente capítulo se tratan las partes más importantes y el código tanto de la base de datos como la aplicación.

Capítulo 4

Desarrollo del sistema para la DMIFS.

4.1 Base de datos

Para comenzar el desarrollo de la base de datos es importante el diccionario de datos para la integridad de la información. A su vez se procedió a pasar del diseño al código en SQL.

4.1.1 Diccionario de datos

Aquí se muestra el diccionario de datos, para saber que tipos de datos serán aceptados por las tablas y su longitud.

Tabla Cliente	
Dato	Tipo
Clave	INT
Razon_Social	VARCHAR(60)
Responsable	VARCHAR(40)
RFC	VARCHAR(13)
Calle	VARCHAR(40)
Numero	INT
Piso	SMALLINT
Departamento	SMALLINT
Colonia	VARCHAR(40)
CP	MEDIUMINT
Deleg_Muni	VARCHAR(40)
Entidad_F	VARCHAR(40)
E_mail	VARCHAR(40)

Tabla Telefonos	
Dato	Tipo
IdClave	INT
Tel1	VARCHAR(15)
Ext1	VARCHAR(6)
Tel2	VARCHAR(15)
Ext2	VARCHAR(6)
Fax	VARCHAR(15)

Tabla Usuarios	
Dato	Tipo
Nombre	VARCHAR(20)
Apepat	VARCHAR(15)
Apemat	VARCHAR(15)
Puesto	VARCHAR(20)
login	VARCHAR(10)
contraseña	VARCHAR(10)

Tabla Productos	
Dato	Tipo
Codigo_P	INT
Servicio	VARCHAR(60)
Descripcion_P	VARCHAR(300)
Precio	FLOAT

Tabla Servicios	
Dato	Tipo
Codigo_S	INT
Servicio	VARCHAR(60)
Descripcion_P	VARCHAR(300)
Precio	FLOAT

Facturas	
Dato	Tipo
Folio	INT
idClave	INT
Total_Fac	INT
Fecha_Fac	DATE

Cotizaciones	
Dato	Tipo
<u>Num_Cot</u>	INT
idClave	INT
Total_Cto	INT
Fecha_Cot	DATE

Figura 4.1. Diccionario de datos.

4.1.2 Código de implantación en MySQL

Conforme al diseño se procede a realizar la programación de la base de datos en MySQL. Aquí se muestra parte del código utilizado.

```
CREATE TABLE clientes(
clave          int AUTO_INCREMENT NOT NULL PRIMARY KEY,
razon_social   varchar(60) NOT NULL,
responsable    varchar(40) NOT NULL,
rfc            varchar(13),
calle          varchar(40),
numero        varchar(11),
piso          smallint,
departamento  varchar(11),
colonia        varchar(40),
cp            mediumint,
deleg_muni     varchar(40),
entidad_f     varchar(40),
e_mail        varchar(40)
);
```

Se puede observar la creación de la tabla clientes en dónde se utiliza como llave primaria el atributo cliente, además es auto-incrementable, esto quiere decir que cada vez que se inserte un nuevo cliente automáticamente se genera su clave.

```
CREATE TABLE telefonos(
idclave        int NOT NULL,
tel1           varchar(15),
ext1           varchar(6),
tel2           varchar(15),
ext2           varchar(6),
fax            varchar(15),
FOREIGN KEY (idclave) REFERENCES clientes (clave)
);
```

En el código anterior se genera la tabla telefonos en dónde se hace referencia a la llave primaria de la tabla clientes ya que es dependiente de esta.

```
CREATE TABLE usuarios(
nombre        varchar(20) NOT NULL,
apepat        varchar(15) NOT NULL,
apemat        varchar(15),
puesto        varchar(20),
login         varchar(10) NOT NULL,
password      varchar(10) NOT NULL,
PRIMARY KEY (login)
);
```

```
CREATE TABLE productos(  
codigo_p      int not null,  
producto      varchar(40),  
descripcion_p varchar(300),  
precio_p      float,  
PRIMARY KEY (codigo_p)  
);
```

4.2 Aplicación

Las clases principales, conexión con la base de datos y la interfaz gráfica de usuario son tratadas a continuación, cabe mencionar que sólo se muestran pequeñas partes del código, ya que es muy extenso.

4.2.1 Clases principales

Algunas de las clases principales que se utilizan en todo momento son las contenidas en el paquete sdmifs: Cliente, Cotización, Factura, Producto, Servicio, Teléfono y Usuario.

Aquí se observa el código de la clase cotización, que contiene sus atributos: numCot, idClave, idLogin, montoCot y fechaCot, así como un constructor vacío y uno con todos sus atributos. También se observan los métodos get y set de todos los atributos.

```
package sdmifs;  
/**  
 * @author Juan Jose Mejía López  
 */  
public class Cotizacion {  
  
    private int numCot;  
    private int idclave;  
    private String idLogin;  
    private double montoCot;  
    private String fechaCot;  
  
    public Cotizacion() {  
    }  
  
    public Cotizacion(int numCot, int idclave, String idLogin, double montoCot,  
String fechaCot) {  
        this.numCot = numCot;  
        this.idclave = idclave;  
        this.idLogin = idLogin;  
        this.montoCot = montoCot;
```

```
        this.fechaCot = fechaCot;
    }

    public String getFechaCot() {
        return fechaCot;
    }

    public void setFechaCot(String fechaCot) {
        this.fechaCot = fechaCot;
    }
    public String getIdLogin() {
        return idLogin;
    }

    public void setIdLogin(String idLogin) {
        this.idLogin = idLogin;
    }

    public int getIdclave() {
        return idclave;
    }

    public void setIdclave(int idclave) {
        this.idclave = idclave;
    }

    public double getMontoCot() {
        return montoCot;
    }

    public void setMontoCot(float montoCot) {
        this.montoCot = montoCot;
    }

    public int getNumCot() {
        return numCot;
    }

    public void setNumCot(int numCot) {
        this.numCot = numCot;
    }
}
```

Otra de las clases importantes es CreaPDF que son las que generan los archivos de cotizaciones, así como el de facturación. Esta clase contiene tres métodos muy importantes creaCotizacionPDF, creaCotizacionGeneralPDF y creaFacturaPDF. Como su nombre lo indica crean los archivos PDF con la ayuda de la librería iText

de licencia libre y otras clases auxiliares como: ObtieneMes que a partir de la fecha dada en número obtiene el mes con letra o viceversa, la clase RedondeoDeNumeros, etc.

La clase NumerosALetras se encarga de validar y convertir cantidades con números a su equivalente en letras.

También son importantes las clases que contienen los modelos de los combobox o listas desplegables que se utilizan en la aplicación las cuales son: ModeloLista, ModeloListaCliente, ModeloListaProducto, ModeloListaUsuario y ModeloListaServicio.

4.2.2 Interfaces y conexión con la base de datos

La interfaz DAO es sumamente importante ya que a través de esta es como se crea la conexión a la base de datos.

En el paquete DAO se encuentran las clases: AccesoADatos, ClienteDAO, CotizacionDAO, FacturaDAO, InterfazDAO, ProductoDAO, ServicioDAO, TelefonoDAO, UsuarioDAO y RespaldaBD.

La clase AccesoADatos es de gran relevancia ya que se encarga de establecer la conexión con la base de datos: carga el driver, en este caso el driver de MySQL, define la URL de conexión, establece la conexión, crea un objeto statement, a continuación se ejecutan las manipulaciones y consultas, se procesan los resultados y por último cierra la conexión.

El patrón de diseño DAO tiene como objetivo encapsular la fuente de datos, ocultar los métodos para acceso a datos. También la “complejidad” del uso de JDBC a la capa de presentación o de negocio.

4.2.3 Interfaz gráfica de usuario

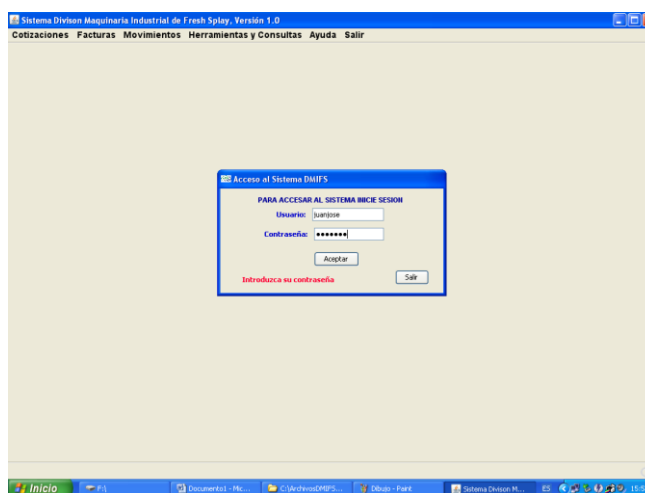


Figura 4.2. En esta figura se puede observar la ventana principal de la aplicación.

La interfaz de usuario fue creada como una aplicación independiente de java, con una ventana principal la cual se muestra en la figura 4.2. También se puede observar la ventana de acceso al sistema la cual permite tres intentos de acceso al sistema, en caso contrario se cerrará. Se podrá acceder al Menú hasta que el usuario se autentifique. Observar la figura 4.2.

Una vez entrando al sistema se podrá empezar a trabajar dependiendo de lo que se quiera, para esto se tiene el menú con las opciones con que cuenta la aplicación.



Figura 4.3. Menú de las opciones de la aplicación.

El menú en su totalidad es:

- **Cotizaciones**
 - **Nueva cotización de máquinas.**

En la figura 4.4 se muestra la ventana de cotización de máquinas que conforme a una serie de pasos obtiene el archivo correspondiente de una cotización de mantenimiento de maquinaria.

Paso 1. Se busca el cliente, ya sea por su clave o por su razón social en la lista desplegable. (Nótese que la fecha es colocada automáticamente por el sistema).

Paso 2. Si en el trato con el cliente se llegó a un acuerdo de descuento se coloca en el paso 2, en caso contrario se deja el 0 % de descuento por defecto.

Paso 3. Se seleccionan los productos y servicios que requiere la máquina en cuestión y se inserta en la tabla del punto 4, cuando se tenga llena la tabla se pasa a la tabla de cotización total.

Paso 4. A cada máquina agregada como nueva se le debe asignar su número, se pueden agregar productos y servicios, al igual que borrar por alguna equivocación, cuando se haya completado una máquina se pasa al punto 5 a la tabla de la cotización total, de está se toman los datos para el archivo de cotización que se generará.

Paso 5. Aquí se puede observar el monto total de la cotización en cantidad monetaria, así como con letra y se puede revisar en caso de algún error.

Paso 6. En este paso se selecciona el usuario que está realizando la cotización con el fin de saber quien la hizo para futuras consultas y aclaraciones.

Paso 7. Al terminar la cotización en este paso se genera el número de cotización automáticamente y el archivo en formato PDF para ser impreso o mandado por correo electrónico al cliente. Si se desea modificar la cotización se hacen los cambios necesarios y se oprime el botón modificar.

Paso 8. Por último la cotización ya está realizada y es posible abrir el archivo, en caso de que se quiera corregir algo es necesario cerrar el archivo para así poder modificarlo.

1. CLIENTE: Clave Fecha:

2. **Descuento:** %

3. PRODUCTOS

Código:

SERVICIOS

Código:

Producto

Descripción:

Cantidad **Precio:** **Clave:**

4. Maquina

número

CANT.	DESCRIPCIÓN	P. UNITARIO	IMPORTE
1	Tornillo marca patito...	6.0	6.0

Total: \$

5. Tabla de la cotización total

CANT.	DESCRIPCI...	P. UNITARIO	IMPORTE	No. MQUINA
1	Tornillo marc...	6.0	6.0	1

Total: \$

6.
Realizada por: meja123 **Número de Cotización**

7. **8.**

No. 1 **Total: \$**

Figura 4.4. Ventana Cotización de Máquinas.

➤ **Nueva cotización general**

La diferencia existente entre una cotización general y una de máquinas de acuerdo a la interfaz de usuario es mínima ya que los pasos se siguen de una manera similar, sin embargo se diferencia en el Paso 4 y 5.

Paso 4. En la tabla se agregan los productos y servicios que se están cotizando, a diferencia de la cotización de máquinas aquí solamente existe una tabla muy general y con un total.

Paso 5. En este pasó se escribe la razón de la cotización que se está realizando ya que puede tener diferentes aspectos específicos.

Por demás los siguientes puntos son similares a la cotización de máquinas, no obstante la generación de archivos es distinta ya que se tiene un método específico para cada tipo de presupuesto.

Figura 4.5. Ventana Cotización General.

- **Facturas**
 - **Nueva factura**

En el caso de la creación de facturas también se hace con una serie de pasos establecidos.

Paso 1. Se busca el cliente, ya sea por su clave o por su razón social en la lista desplegable. (Nótese que la fecha es colocada automáticamente por el sistema).

Paso 2. Si en el trato con el cliente se llegó a un acuerdo de descuento se coloca en el paso 2, en caso contrario se deja el 0 % de descuento por defecto.

Paso 3. Se seleccionan los productos y servicios que se están facturando para poder observarlos en el paso 4.

Paso 4. Se escribe la cantidad de unidades que se requiere y se agrega el elemento a la tabla del paso 5.

Paso 5. Aquí se puede observar el monto total de la factura en cantidad monetaria, así como con letra y se puede revisar en caso de algún error. Si se aplica una modificación del descuento fuera de orden es necesario oprimir calcular para que se actualicen los datos. Se pueden hacer las modificaciones que se crean necesarias en este punto.

Paso 6. En este punto se puede visualizar la forma en la que ha quedado la factura e imprimir en caso de que no contenga algún error, o también se puede modificar.

Paso 7. En este paso se selecciona el usuario que está realizando la factura con el fin de saber quien la hizo para futuras consultas y aclaraciones.

Paso 8. Por último en este paso se agrega el número de folio y se guarda en la base de datos para futuras consultas.

Figura 4.6. Ventana de factura, se pueden observar los pasos a seguir para facturar.

- **Movimientos**

- **Movimientos de clientes**

Movimientos de Clientes es donde se podrán dar de alta y baja a clientes, igualmente es para realizar actualizaciones de los datos del cliente, el botón checar dirección es importante, ya que para las facturas la dirección que aparecerá será la revisada en este formulario.

The screenshot shows a software window titled "Movimientos de Clientes". At the top, there is a search section with a dropdown menu labeled "Clientes:" containing the text "Cambio de nombre", a text input field labeled "Clave:" with the value "0", and a "Buscar" button. Below this is a horizontal line. The form contains several fields: "Razon Social:" and "RFC:" (text inputs), "Responsable:" and "Clave:" (text inputs), a "Dirección:" section with "Calle:", "Número:", "Piso:" (value "0"), and "Departamento:" (text inputs), "Colonia:", "C.P." (value "0"), and "Delegación:" (text inputs), "Entidad Federativa:" (text input), and a telephone section with "Tel:", "Ext.:" (two pairs of text inputs), and "Fax:" (text input). Below the telephone section is a "Correo Electrónico:" (text input). At the bottom of the form, there are five buttons: "Limpiar Formulario", "Checar Dirección", "ALTA", "ACTUALIZAR", and "BAJA".

Figura 4.7. Ventana de movimientos de clientes.

- **Movimientos de productos y servicios**

De la misma forma que con clientes, los productos y servicios serán dados de alta, de baja y/o actualizados con el siguiente formulario en donde el código es escrito arbitrariamente por el capturista de los datos.

Figura 4.8. Ventana de movimientos de productos y servicios.

➤ **Movimientos y consultas de usuarios**

Para que un usuario pueda hacer uso del sistema es necesario darlo de alta, es aquí donde se usa este formulario que requiere contraseña del administrador para hacer algún movimiento de usuarios o una consulta según sea el caso. Las consultas se pueden afectar por usuario, nombre o puesto.

Nombre	ApePat	ApeMat	Puesto	Usuario	Contraseña
juan	mejia	lopez	programador	mejia123	beatles

Figura 4.9. Ventana de movimientos y consulta de usuarios.

- **Herramientas y Consultas**
 - **Consultar cotizaciones**

Para consultar los datos de las cotizaciones ya realizadas se cuenta con la ventana Consultas de Cotizaciones en la cual se puede buscar por: número de cotización, clave del cliente y fecha.

Numero de Cotización	Clave Cliente	login de usuario	Monto	Fecha
1	1	mejia	500.5	2009-04-13

Figura 4.10. Consulta de cotizaciones.

- **Consultar facturas**

De la misma forma para consultar los datos de facturas se cuenta con la ventana Consultas de Facturas en la cual se puede buscar por: número de folio, clave del cliente y fecha.

Folio	Clave Cliente	login de usuario	Monto	Fecha
23	11	paprica	394981.0	2009-07-23

Figura 4.11. Consulta de facturas.

- **Consultar clientes**

Para consultar los datos de un cliente se puede buscar por: razón social, clave, RFC, delegación o municipio y teléfono.

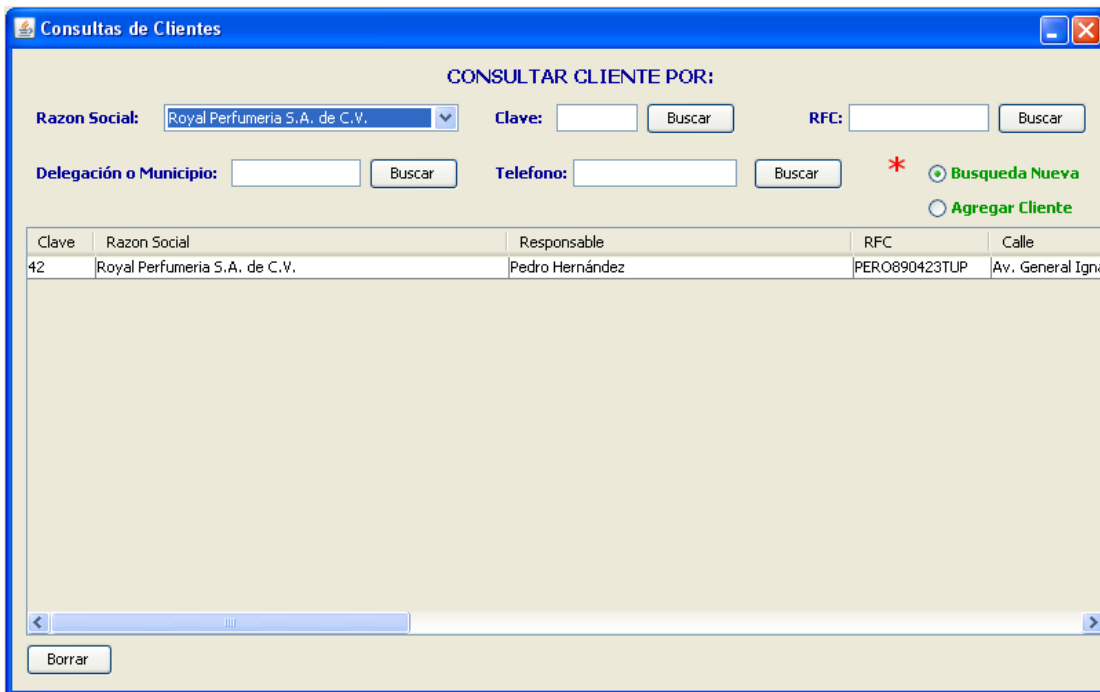


Figura 4.12. Consulta de clientes

➤ **Consultar productos y/o servicios**

Si se desea hacer una consulta de productos o servicios se puede realizar la búsqueda ya sea por el código o por una lista desplegable que se tiene en la ventana consulta de productos y servicios.

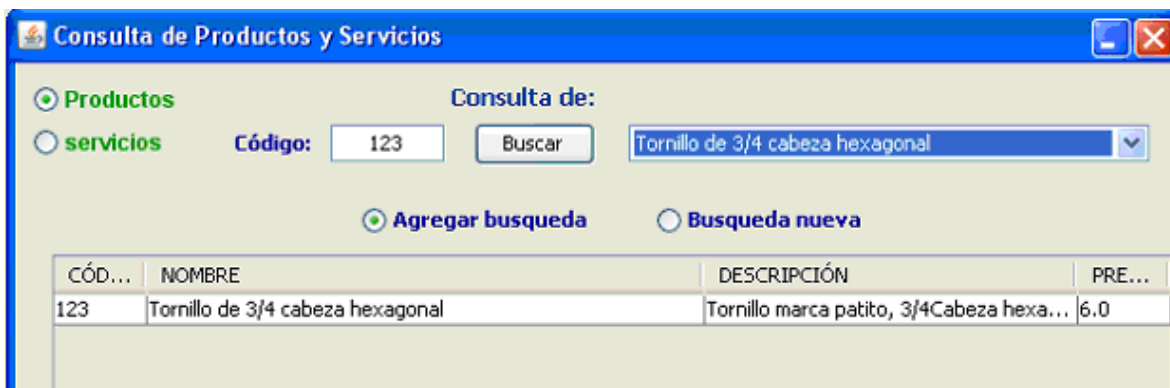


Figura 4.13. Consulta de productos y servicios.

➤ **Eliminar cotizaciones y/o facturas**

Para dar de baja cotizaciones y facturas se utiliza la siguiente ventana la cual requiere de la contraseña del administrador para mayor seguridad e integridad de la información. Se pueden dar de baja ya sea por número de cotización o folio respectivamente o también todas las facturas o cotizaciones de un cliente.

Baja de Cotizaciones y Facturas

Contraseña del Administrador: 2. **Seleccionar**

1. **Habilitado** **Cotización**
 Factura

3. **Dar de baja por lave primaria**

No. Cotización:

No. Cotización:

Clave del Cliente:

Login de usuario:

Monto:

Fecha:

4. **Dar de baja por clave de cliente**

Elimina todas las Cotizaciones o Facturas de un Cliente

Clave del Cliente:

Clave:

Razón S.

Figura 4.14. Baja de cotizaciones y Facturas

➤ **Convertir números a letras**

Se decidió añadir este módulo de conversión de números a su forma escrita en letras ya que es muy utilizado para algunos procesos en la División Maquinaria Industrial.

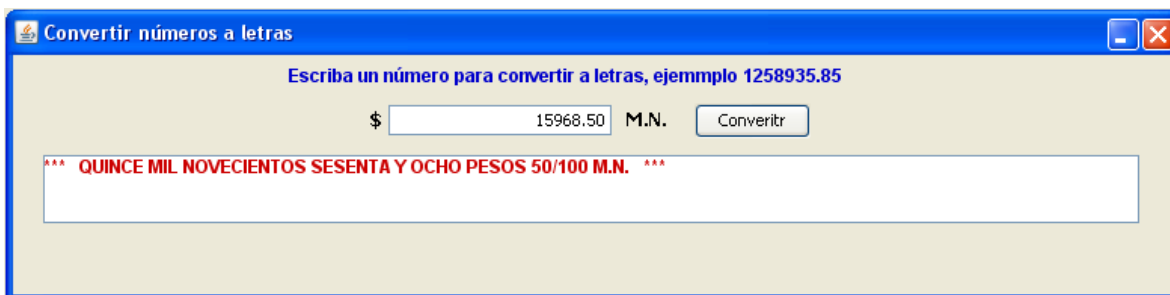


Figura 4.15. Convertidor de cantidades numéricas a su equivalente en letras.

➤ **Respaldo base de datos DMIFS**

Es importante respaldar la base de datos de forma periódica y aunque esto se hace automáticamente por el sistema operativo. Se añade un módulo en el cual se podrá respaldar la información de la base de datos así como asignar el último respaldo que se hizo, esto requiere de la contraseña del administrador.



Figura 4.16. Ventana que sirve para respaldar la base de datos, así como asignar el último respaldo.

- **Ayuda**
 - **Acerca del sistema**

El sistema es la versión 1.0 que se puede seguir mejorando con el tiempo, se muestra el Copyright con la licencia utilizada.



Figura 4.17. Acerca del Sistema División Maquinaria Industrial de Fresh Splay, Versión 1.0.

- **Manual**

En todo momento el usuario puede acceder al manual de usuario del sistema para aclarar sus dudas.

- **Salir**
 - **Salir**

Menú para salir del sistema, cabe mencionar que en todo momento al cerrar una ventana se abre un diálogo preguntando si esta de acuerdo con que se cierre el sistema, en caso de haber olvidado algún paso él sistema informa al usuario.

4.3 Pruebas

Las pruebas realizadas fueron con datos ficticios comprobando el funcionamiento de cada módulo usado. Se pudieron detectar errores de programación en donde aparecían excepciones entendibles sólo para el programador las cuales fueron retiradas y traducidas a un lenguaje apropiado para un usuario final.

También se detectaron errores sobre todo de mala escritura en sentencias SQL en la parte encapsulada por el patrón de diseño DAO las cuales fueron detectadas y corregidas.

Otro problema era el desfase de la impresión de las facturas que no concordaba con el formato real, se hicieron los cálculos y modificaciones correspondientes para solucionar este problema.

De acuerdo a la funcionalidad que se planteo en los requerimientos el sistema y se diseño con los diagramas de casos de uso y de secuencia no se presentaron problemas y el sistema funcionó correctamente.

4.4 Documentación, manuales de usuario y técnico

Se realizó el manual de usuario indicando paso por paso el uso del sistema y se agregó en forma de ayuda, también se instaló como parte del programa para cualquier consulta que el usuario desee hacer.

El manual técnico incluye el objetivo del proyecto, los requerimientos, todos los diagramas referentes al diseño y el código generado.

Con el desarrollo terminado lo sucesivo es la implantación del sistema que es tratado en el capítulo cinco y último.

Capítulo 5

Implantación y mantenimiento

5.1 Instalación

Para hacer la instalación del sistema se seleccionó un equipo para utilizarlo como servidor de la base de datos. Se configuró con una dirección IP fija para que los demás equipos puedan acceder al servidor mediante la red LAN existente en las oficinas de la División Maquinaria Industrial.

5.1.1 MySQL

Primeramente se instaló MySQL 5.0 obtenida de la página de Internet, se configuró como servidor para que los otros equipos tengan acceso a la base de datos. A continuación se asignó el usuario y contraseña de administrador.

Una vez instalado se procedió a crear la base de datos con el nombre de DMIFS y enseguida se cargó el archivo dmifs.sql con el código necesario para generar las tablas de la base de datos.

Se hicieron pruebas de funcionamiento desde la consola de MySQL y una vez terminado este proceso se continuó con la aplicación.

5.1.2 JDK, máquina virtual de Java

Las PC ya contaban con la máquina virtual de java, ya que es usada en otras aplicaciones, sin embargo fue actualizada a la versión más actual ya que tenían una versión atrasada.

5.1.3 Sistema

Para la instalación se creó el manual de instalación el cual contiene los siguientes pasos:

Insertar el CD de instalación, si no arranca el programa de instalación desde inicio ir a explorador de Windows, luego al CD insertado y buscar el archivo "Instalar Sistema DMIFS 1.0.exe". Y oprimir doble click sobre él.



Figura 5.1. Icono del instalador del sistema.

Aceptar el acuerdo de licencia

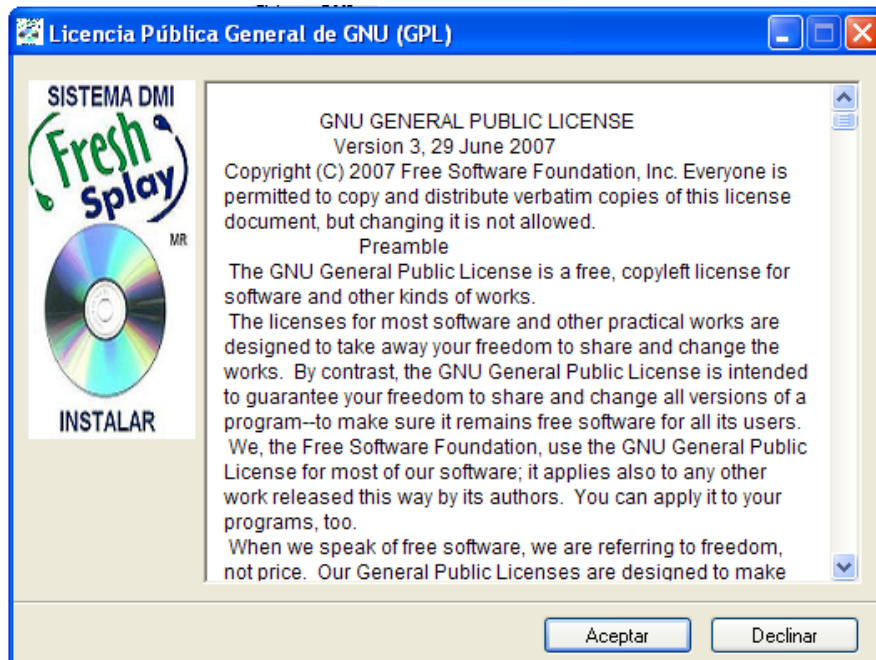


Figura 5.2. Aceptar acuerdo de Licencia.

Oprimir Instalar



Figura 5.3. Instalar Sistema DMIFS.

Verificar que la instalación se haya completado, buscando los iconos en menú inicio, todos los programas, Sistema DMIFS.

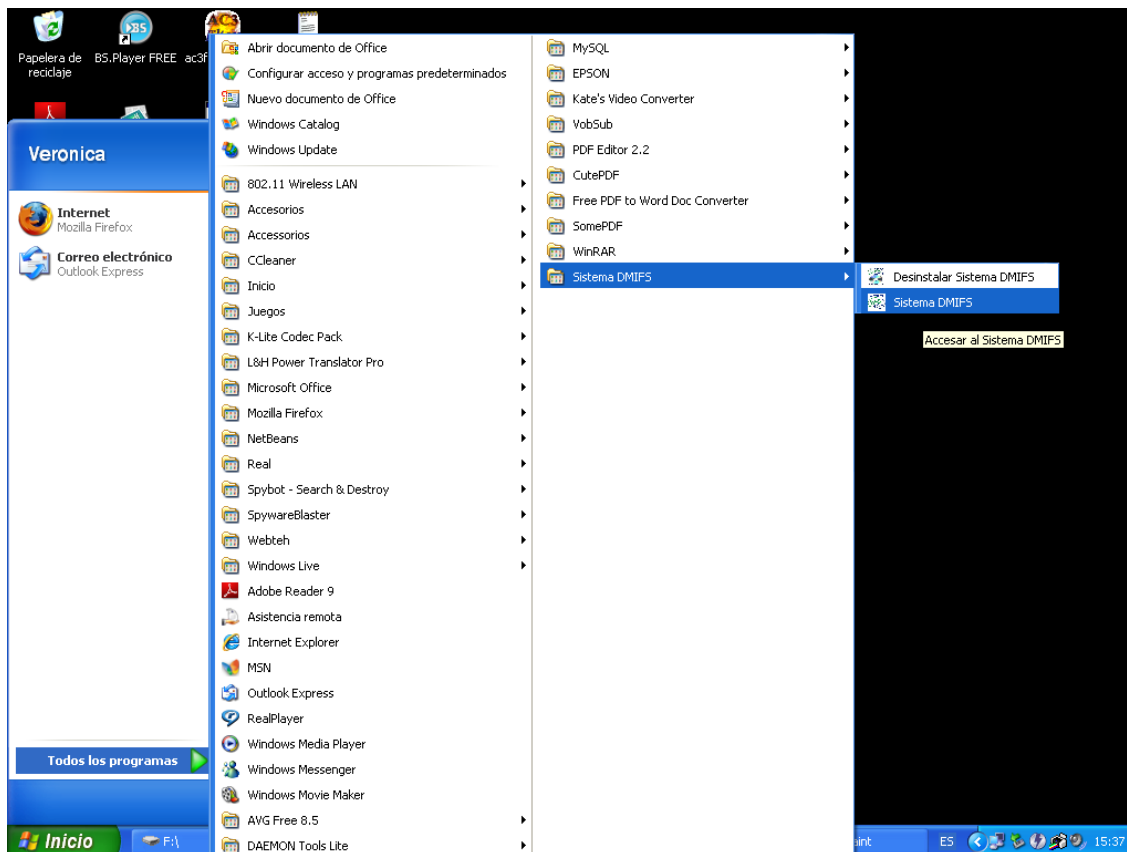


Figura 5.4. Ingresar al Sistema DMIFS

Una vez completada la instalación se puede comenzar a utilizar el sistema.

5.2 Capacitación

Ya que el sistema cuenta con una interfaz de usuario amigable e intuitiva el proceso de capacitación se hizo de manera fácil ya que los usuarios están capacitados en los procesos que se llevan a cabo. Además de contar con el manual de usuario para resolver las dudas que se puedan presentar.

5.3 Pruebas con el usuario

Una vez iniciadas las pruebas con el usuario se detectaron algunas fallas, una de ellas fue el descuento aplicado en las facturas, en las cuales se hacia de forma automática por cada producto o servicio, una vez seleccionado el porcentaje de descuento, pero en realidad se tenia que hacer un descuento del total de la suma de los productos y servicios para aparecer en la factura.

Otro problema fue el cambio de formato de la factura que fue modificada por la imprenta media pulgada.

5.4 Correcciones

Se realizaron las correcciones necesarias al sistema para su buen funcionamiento antes de comenzar a utilizarlo.

Al ya estar capacitados los usuarios se dispuso a realizar la captura de los clientes, productos y servicios. De antemano el encargado del área que funge como administrador, dio de alta a los usuarios para poder acceder al sistema.

5.5 Mantenimiento

Con el uso del sistema van surgiendo nuevos requerimientos en los cuales es necesario intervenir para el buen funcionamiento del sistema, es posible realizar adaptaciones y modificaciones para una mejora y así reforzar y ampliar el sistema.

Las correcciones que se han realizado son sobre todo de archivos de cotizaciones y la impresión de la factura. También se amplió con algunos módulos como el convertidor de números a letras muy usado por los usuarios.

Actualmente el sistema se encuentra en su versión 1.0 terminada y funcionando.

Conclusiones

Se logró el objetivo principal del proyecto y ahora la División Maquinaria Industrial de Fresh Splay cuenta con una herramienta tecnológica que permite automatizar los procesos de cotizaciones y facturación, así como llevar un control sobre sus clientes, productos y servicios.

Se cuenta con la base de datos que contiene información importante sobre los clientes, productos, servicios, cotizaciones y usuarios. Además de una interfaz gráfica amigable con la que cuentan los usuarios dónde fácilmente pueden realizar movimientos.

Las cotizaciones ahora se realizan de forma más eficiente, haciendo las consultas a la base de datos de forma sencilla, y obteniendo las cotizaciones más rápido y en un archivo en formato estándar PDF. Dejando atrás la búsqueda y escritura manual que se hacía.

Igualmente las facturas son rápidas de elaborar en la interfaz gráfica, donde se muestra la vista preliminar antes de ser impresa.

El software fue diseñado a la medida de las necesidades y requerimientos de Fresh Splay, ya que en el mercado existe software comercial que cumple con parte del propósito de este proyecto, sin embargo no dispone de puntos importantes como lo son: la elaboración de las cotizaciones que son muy específicas de esta empresa, además de la impresión de las facturas que fueron calculadas a la medida.

Se trabajo con los recursos tecnológicos que contaba la empresa. Las computadoras fueron actualizadas a su máxima capacidad de memoria, que es suficiente para el funcionamiento del sistema por un tiempo, sin embargo los equipos ya son viejos y tendrán que ser remplazados en un plazo no mayor a un año para que el sistema y otros programas instalados puedan funcionar de forma eficiente.

La aplicación inicialmente sería instalada solamente en un equipo, por tal motivo se diseñó como aplicación de escritorio, sin embargo podría mejorarse para que trabaje como aplicación web y así manejarlo como una intranet en la red de Fresh Splay, esto tendría otro tipo de consecuencias como mejorar la seguridad de la información.

El sistema fue documentado con el manual técnico que incluye el objetivo del proyecto, los requerimientos, todos los diagramas referentes al diseño: de casos de uso, de secuencia y de clases, el diseño de la base de datos y el código generado.

El software utilizado para la elaboración del sistema en su totalidad contiene la compatibilidad con la licencia GNU, por lo que cumpliendo con los lineamientos no se tuvo que pagar por utilizarlo.

En particular en este proyecto aprendí bastante, sobre todo con la práctica, he obtenido una buena experiencia, así como conocimientos reforzados y adquiridos como lo es MySQL, el lenguaje java y Netbeans, así como la metodología UML.

Bibliografía

- **Libros**

1. Deitel Harvey M., Deitel Paul J., Java cómo programar, quinta edición, Pearson Education, México, 2004, pp. 1329.
2. Larman, Craig, UML y patrones: introduccion al analisis y diseño orientado a objetos, México, Prentice-Hall Hispanoamericana, 1999, pp. 507.
3. Lowagie Bruno, iText in Action, Creating and Manipulating PDF, New York, NY, Manning, 2007, pp. 657.
4. Pérez López Cesar, MySQL para Windows y Linux, México, D. F., Alfaomega, Ra-Ma, 2004, pp. 454.
5. Pressman Roger S., Ingeniería de Software, Un Enfoque Práctico, Madrid, España, McGrawHill/Interamericana, 2002, pp. 601.
6. Silberschatz A., Korth Henry F., Sudarshan. S, Fundamentos de Bases de Datos, , Madrid, España, McGrawHill/Interamericana, 2002, pp. 787.

- **Referencia electrónica**

7. Lowagie Bruno, iText Homepage (2009), recuperado (mayo 2009), disponible en: (<http://www.lowagie.com/iText/>
8. Pozo Salvador, MySQL con Clase (2004), recuperado (mayo 2009), disponible en: <http://www.conclase.net/mysql/curso/index.php>
9. Sun Microsystems, Core J2EE Patterns - Data Access Object (2002), recuperado (mayo 2009), disponible en: <http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>
10. Sun Microsystems, MySQL (2008), recuperado (mayo 2009), disponible en: <http://www.mysql.com/>
11. Sun Microsystems, Sun Downloads (2009), recuperado (mayo 2009), disponible en: (<http://mx.sun.com/tryandbuy/index.jsp?cid=080526mx02>