



**BENEMERITA UNIVERSIDAD  
AUTONOMA DE PUEBLA**

---

---

---

**Facultad de Ciencias de la Computación  
Diplomado en Bases de Datos**

**SISTEMA DE BASES DE DATOS DE UNA  
INMOBILIARIA**

**TESIS**

**Que para obtener el título de:  
Ingeniero en Ciencias de la Computación**

**Presenta:  
Heidi Ortíz García**

**Asesor:  
M.C. Yolanda Moyao Martínez**

**Puebla, Pue.**

**Julio de 2007**

# **AGRADECIMIENTO**

---

Principalmente doy gracias a Dios ya que me ha dado vida para continuar en el camino.

A mis padres por todo el apoyo, cariño y comprensión que me dieron y que me dan día con día.

A mis hermanos que siempre me dieron ánimo para seguir adelante.

A Isra que es una persona muy importante en mi vida y que nunca me dejó caer, que siempre estuvo conmigo cuando lo necesité.

A la maestra Yolanda M. por dedicarme su tiempo, por haberme apoyado en todo y por ser un pilar muy importante para poder terminar la Tesis.

A todos gracias por darme la fuerza necesaria para seguir adelante.

Heidi Ortiz García.

# Indice

Agradecimiento	2
Indice	3
Introducción	5
<b>CAPITULO 1: MARCO TEÓRICO</b>	<b>6</b>
1.1 Ingeniería de software	6
1.1.1 Productos de Software	6
1.1.2 Modelos de Desarrollo de Software	7
1.1.3 Retos de la Ingeniería de Software	11
1.2 Bases de Datos	12
1.2.1 Sistema Gestor de Base de Datos	12
1.2.2 Sistema de Base de Datos	13
1.2.3 Modelos de Datos	13
1.2.4 Modelo Entidad-Relación	16
1.2.5 Modelo Relacional	16
1.2.6 Teoría de la Normalización	18
1.3 MySQL	19
1.3.1 La estructura de MySQL	19
1.3.2 La seguridad de MySQL	19
1.4 PHP	20
1.4.1 Seguridad y PHP	21
1.4.2 Manejo de conexiones	21
1.5 Servidor Apache	22
<b>CAPITULO 2: ANÁLISIS DEL SISTEMA</b>	<b>23</b>
2.1 Planteamiento del problema	23
2.1.1 Especificación de Requerimientos	23
2.1.2 Antecedentes	24
2.1.3 Alcances	24
2.1.4 Metodología	25
2.1.5 Definición de Requerimientos Funcionales	25
2.1.6 Definición de Requerimientos No Funcionales	26
2.1.7 Diagrama de Flujo de Datos	26
<b>CAPITULO 3: DISEÑO DEL SISTEMA.</b>	<b>44</b>
3.1 Modelo Conceptual	44
3.2. Modelo Entidad-Relación	44

3.3 Diccionario de Datos	47
3.4 Modelo Relacional	51
3.5 Normalización	52
<b>CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA</b>	54
4.1 Herramientas para el desarrollo del Sistema	54
4.2 Conexión de MySQL con PHP	55
4.3 Interfaces del Sistema de Bases de Datos	58
4.4 Pruebas	65
<b>CONCLUSIONES</b>	70
<b>PERSPECTIVAS</b>	71
<b>BIBLIOGRAFÍA</b>	72

# **INTRODUCCIÓN**

---

El almacenamiento, manipulación y recuperación de información en forma eficiente, es vital y estratégico para cualquier organización. Las bases de datos juegan un rol crítico en casi todas las áreas donde las computadoras son usadas, incluyendo negocios, ingeniería, medicina, leyes, educación, etc.

En estos días el estar a la vanguardia en la Tecnología es una parte muy importante en el mundo de los negocios ya que los procesos se automatizan, mejorando así la calidad de los servicios que otorga una empresa.

Es por eso que se ha propuesto un Sistema para una Inmobiliaria, mismo que facilitará la organización de su información ya que todos los datos se almacenan en forma integrada, y están sujetos a un control centralizado, así como la reducción de tiempo ya que cuando se tienen pocos datos no es mucha la pérdida de tiempo y espacio, pero cuando hablamos de miles de datos, nos enfrentamos a un serio problema.

El siguiente documento está dividido en 4 capítulos, en los cuales se menciona el contenido de cada uno de una forma general.

En el capítulo 1, se presenta el Marco Teórico, es decir los conceptos de las herramientas que utilizamos para el desarrollo del Sistema.

En el capítulo 2, se presenta el Análisis del Sistema, definiendo el planteamiento de los requerimientos, así como sus alcances y metodología utilizada entre otros.

En el capítulo 3, se presenta el Diseño del Sistema, con la ayuda del modelo entidad-relación, modelo relacional, diccionario de datos y de la Normalización.

En el capítulo 4, se presenta la Implementación y las Pruebas del Sistema, en el que se aporta código fuente de los módulos principales, así como las Interfaces, documentando también, los resultados obtenidos de las pruebas realizadas para la verificación de los requerimientos.

Finalmente, se presentan las conclusiones y las perspectivas, en las cuales se tocan los puntos principales del Sistema y la posible evolución del mismo.

## MARCO TEÓRICO

En éste capítulo, se describen conceptos básicos que se utilizaron a lo largo del desarrollo del Sistema, es conveniente realizar esta breve descripción para entender mejor la funcionalidad del Sistema realizado.

### 1.1 Ingeniería de Software

El **Software** es un programa de computadora y la documentación asociada a los mismos. Los productos de software pueden desarrollarse para un cliente en particular o para el mercado en general.

La **Ingeniería de software** es una disciplina de la Ingeniería que concierne a todos los aspectos de la producción de software. Esta adopta un enfoque sistemático para llevar a cabo su trabajo y utilizan las herramientas y técnicas necesarias para resolver el problema planteado, de acuerdo a las restricciones de desarrollo y recursos disponibles.

La economía de grandes empresas, dependen en gran parte del software, ya que más y más sistemas son actualmente controlados por software. La Ingeniería de Software, engloba teorías y métodos para el desarrollo profesional de software, así como el diseño, construcción y mantenimiento de sistemas grandes. Abarca tres elementos que permiten construir software de calidad:

- **Procesos.**- Son la unión que mantiene juntas las capas de métodos y herramientas que permiten un desarrollo racional y oportuno de software.
- **Métodos.**- Indican como construir el software y dependen de un conjunto de principios básicos.
- **Herramientas.**- Proporcionan un soporte automático o semi-automático para el proceso y para los métodos. [5] [10]

#### 1.1.1 Productos de Software

Son programas de cómputo y su documentación asociada. Están catalogados en:

- **Productos Genéricos:**  
Productos producidos por una organización para ser vendidos al mercado

- **Productos hechos a la medida:**

Aquellos que son desarrollados bajo pedido a un desarrollador específico

Dentro de sus características se encuentran:

- 1) **Mantenibles:** Es decir, que el software pueda evolucionar, pero sin perder sus especificaciones.
- 2) **Confiables:** Es decir, que el software no cause un perjuicio físico o económico.
- 3) **Eficientes:** Es decir, el software debe aprovechar todos los recursos posibles del sistema.
- 4) **Utilización adecuada:** Es decir, el software debe de tener un informe completo a cerca del sistema. [5]

### 1.1.2 Modelos de Desarrollo de Software

Es común que las especificaciones no están del todo completas o son erróneas porque no existe una distinción precisa entre la especificación de requerimientos y el diseño, realmente hasta que el sistema es terminado, es cuando se producen los errores. [5] [12] [13]

Existen diferentes modelos, que a continuación se numeran:

- **Modelo en Cascada**

El más conocido, esta basado en el ciclo convencional de una ingeniería, el paradigma del ciclo de vida abarca las siguientes actividades:

- ✓ **Ingeniería y Análisis del Sistema:** Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.
- ✓ **Análisis de los requisitos del software:** el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.
- ✓ **Diseño:** el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.
- ✓ **Codificación:** el diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea.

Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

- ✓ **Prueba:** una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.
- ✓ **Mantenimiento:** el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debido a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.
- ✓ **Desventajas:**
  1. Los proyectos reales raramente siguen el flujo secuencial que propone el modelo, siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
  2. Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos productos.
  3. El cliente debe tener paciencia. Hasta llegar a las etapas finales del proyecto, no estará disponible una versión operativa del programa. Un error importante no detectado hasta que el programa este funcionando puede ser desastroso.

La ventaja de este método radica en su sencillez ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.

La Figura 1.1 muestra el esquema del Modelo de Cascada.

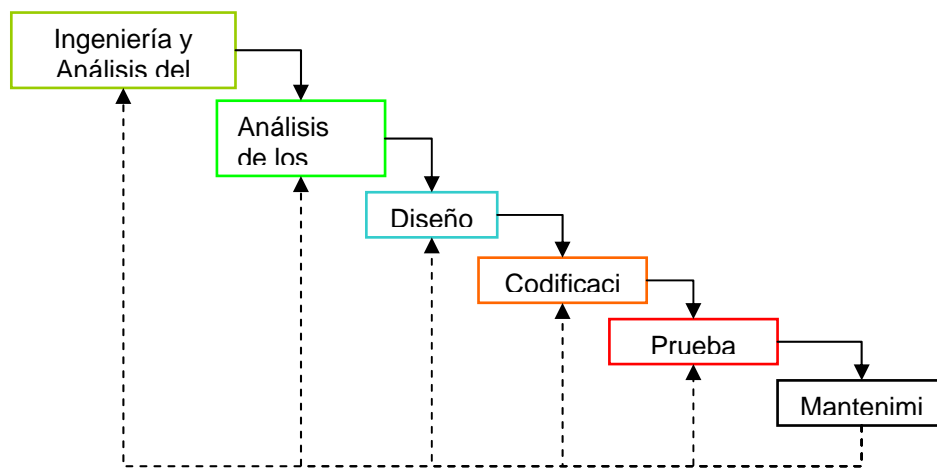


Figura 1.1 Fases del Modelo de Cascada

- **Desarrollo Evolutivo**

Este modelo construye una serie de grandes versiones sucesivas de un producto. Sin embargo, mientras que la aproximación incremental presupone que el conjunto completo de requerimientos es conocido al comenzar, el modelo evolutivo asume que los requerimientos no son completamente conocidos al inicio del proyecto.

En el modelo evolutivo, los requerimientos son cuidadosamente examinados, y sólo esos que son bien comprendidos son seleccionados para el primer incremento. Los desarrolladores construyen una implementación parcial del sistema que recibe sólo estos requerimientos.

El sistema es entonces desarrollado, los usuarios lo usan, y proveen retroalimentación a los desarrolladores. Basada en esta retroalimentación, la especificación de requerimientos es actualizada, y una segunda versión del producto es desarrollada y desplegada. El proceso se repite indefinidamente.

El desarrollo evolutivo no demanda una forma específica de observar el desarrollo de algún incremento. Todo lo que uno tiene que hacer es construir un subconjunto de requerimientos conocidos (incremental), y comprender al principio que muchos nuevos requerimientos es probable que aparezcan cuando el sistema sea desplegado o desarrollado.

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

La Figura 1.2 muestra el esquema del Modelo de Desarrollo Evolutivo

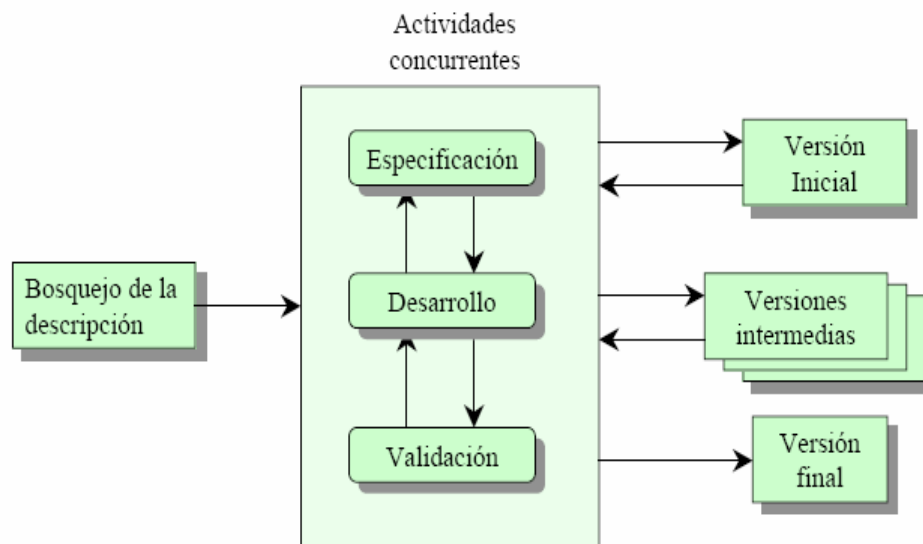


Figura 1.2 Fases del Modelo de Desarrollo Evolutivo

- **Prototipado**

Este modelo es la creación de una implementación parcial de un sistema para el propósito explícito de aprender sobre los requerimientos del sistema. Un prototipo es construido de una manera rápida tal como sea posible. Esto es dado a los usuarios, clientes o representantes de ellos, posibilitando que ellos experimenten con el prototipo.

Estos individuos luego proveen la retroalimentación sobre lo que a ellos les gustó y no les gustó acerca del prototipo proporcionado, quienes capturan en la documentación actual de la especificación de requerimientos la información entregada por los usuarios para el desarrollo del sistema real. El prototipado puede ser usado como parte de la fase de requerimientos (determinar requerimientos) o justo antes de la fase de requerimientos (como predecesor de requerimientos).

Diferente del modelo evolutivo donde los requerimientos mejor entendidos están incorporados, un prototipo generalmente se construye con los requerimientos entendidos más pobremente.

La Figura 1.3 muestra el esquema del modelo Prototipado



Figura 1.3 Fases del Modelo de Prototipado

- **Transformación Formal**

Donde un modelo matemático del sistema se transforma formalmente en la implementación.

La Figura 1.4 muestra el esquema del modelo de Transformación Formal

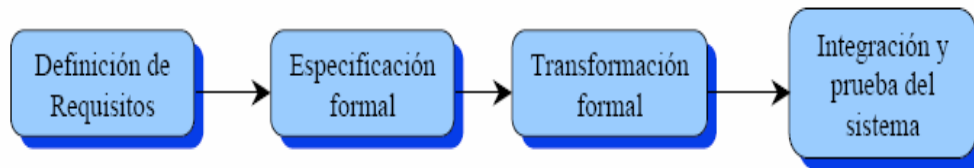


Figura 1.4 Fases del Modelo de Transformación Formal

- **Desarrollo basado en Reutilización**

Donde el sistema es ensamblado a partir de componentes existentes

La Figura 1.5 muestra el modelo de Desarrollo basado en Reutilización

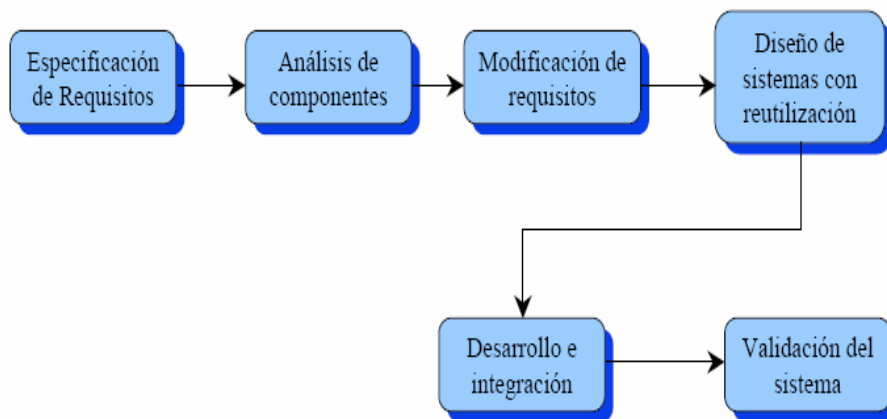


Figura 1.5 Fases del Modelo de Desarrollo basado en Reutilización

### 1.1.3 Retos de la Ingeniería de Software

- **Mantener y tratar con sistemas legados:** Tratar con una mayor diversidad de sistemas con mayores demandas de cómputo, y en menores tiempos de entrega
- **Sistemas Legados:** Sistemas antiguos que deben ser mantenidos y mejorados
- **Heterogeneidad:** Sistemas que incluyen una mezcla de software y hardware
- **Entrega:** Existe una presión incremental por una entrega a tiempo de los productos de software
- **Formalidad:** Existe una gran demanda de que exista la formalidad en el proceso de desarrollo de software [1] [13] [14]

## 1.2 Bases de Datos

Una **Base de Datos** es un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.

Surgen desde mediados de los años sesenta y sus ventajas son las siguientes:

- 1) **Independencia de datos y tratamiento.**- Existe un cambio en datos pero no implica un cambio en programas y viceversa.
- 2) **Coherencia de resultados.**- Reduce la redundancia, con esto se logran acciones lógicamente únicas y se evita la inconsistencia.
- 3) **Mejora en la disponibilidad de datos.**- Se aplica la idea de catálogos.
- 4) **Cumplimiento de ciertas normas.**- Restricciones de seguridad, es decir, el acceso a los usuarios y operaciones sobre los datos.
- 5) **Otras ventajas.**- Más eficiente la gestión de almacenamiento. [2]

### 1.2.1 Sistema Gestor de Base de Datos (SGDB)

Un **Sistema Gestor de Base de Datos** es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

#### Lenguaje de definición de datos (DDL)

Es utilizado para describir todas las estructuras de información y los programas que se usan para construir, actualizar e introducir la información que contiene una base de datos.

#### Lenguaje de manipulación de datos (DML)

Es utilizado para escribir programas que crean, actualizan y extraen información de las bases de datos.

#### Lenguaje de consulta

Es empleado por el usuario para extraer información de la base de datos, permite al usuario hacer requisiciones de datos sin tener que escribir un programa.

El SGDB, junto con la base de datos y con los usuarios, constituye el Sistema de Base de Datos. [3]

### 1.2.2 Sistema de Base de Datos

Los **Sistemas de Bases de Datos (SBD)** pueden ser estudiados desde 3 niveles distintos:

#### 1) Nivel Físico:

Es el nivel real de los datos almacenados, es decir, como se almacenan los datos. Este nivel es usado por muy pocas personas que deben estar calificadas para ello. Este nivel lleva asociada una representación de los datos, que es lo que denominamos Esquema Físico.

#### 2) Nivel Conceptual:

Es el correspondiente a una visión de la base de datos desde el punto de vista del mundo real, es decir, tratamos con la entidad u objeto representado, sin importarnos como está representado o almacenado. Este nivel lleva asociado el Esquema Conceptual.

#### 3) Nivel Visión:

Son partes del esquema conceptual, el nivel conceptual presenta toda la base de datos, mientras que los usuarios por lo general sólo tienen acceso a pequeñas parcelas de ésta. El nivel visión es el encargado de dividir estas parcelas. El esquema asociado a éste nivel es el Esquema de Visión.

Los tres niveles vistos, componen lo que conocemos como arquitectura de base de datos a tres niveles.

A menudo el nivel físico no es facilitado por muchos DBMS, esto es, no permiten al usuario elegir como se almacenan sus datos y vienen con una forma estándar de almacenamiento y manipulación de los datos. [2] [3]

### 1.2.3 Modelos de Datos

Los **modelos de datos** son una colección de herramientas conceptuales para describir los datos, las relaciones de los datos, la semántica de los datos y las ligaduras de consistencia, es decir, describe la estructura de una base de datos, a la cual denominamos esquema.

Los modelos de datos se clasifican en tres grupos diferentes: modelos lógicos basados en objetos, modelos lógicos basados en registros y modelos físicos.

#### a) Modelos lógicos basados en objetos

Se usan para describir datos a un nivel conceptual y de visualización. Su característica principal es que proporcionan capacidades estructurales muy flexibles y permiten que las ligaduras de datos sean especificadas explícitamente.

Existen varios modelos lógicos basados en objetos, entre los que se encuentran el modelo Entidad-Relación, modelo que será utilizado durante el desarrollo del Sistema de Base de Datos para la Inmobiliaria.

## b) Modelos lógicos basados en registros

Se usan para describir datos en los niveles lógico y de vistas, es decir, especifican la estructura lógica completa de la base de datos y también se usan para proporcionar una descripción de alto nivel de la implementación.

Los modelos basados en registros se llaman así, debido a que la base de datos se estructura en registros de formato fijo de diferentes tipos. Entre los modelos lógicos basados en registros se encuentra el Modelo Relacional, que es el que se utilizará para el diseño del Sistema de Base de Datos para la inmobiliaria.

Existen 3 modelos que ayudan a diseñar una base de datos:

- 1) **Modelo Jerárquico.-** Puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos.
- 2) **Modelo de Red.-** Este modelo permite la representación de muchos a muchos, de tal forma que cualquier registro dentro de la base de datos puede tener varias ocurrencias superiores a él. El modelo de red evita redundancia en la información, a través de la incorporación de un tipo de registro denominado el conector.
- 3) **Modelo Relacional.-** Este modelo se está empleando con más frecuencia en la práctica, debido a la ventajas que ofrece sobre los dos modelos anteriores, entre ellas, el rápido entendimiento por parte de usuarios que no tienen conocimientos profundos sobre Sistemas de Bases de Datos.  
La estructura básica y única del modelo relacional, es la relación llamada tabla, que sirve para representar tanto los objetos como las asociaciones entre ellos.

En el modelo relacional, una fila se denomina **tupla**, una cabecera de columna es llamado **atributo**, al número de columnas se le llama **grado**, al número de filas **cardinalidad** y la tabla es una **relación**. El tipo de datos que describen los tipos de valores que pueden aparecer en cada columna se llama **dominio**.

## c) Modelos físicos

Se usan para describir los datos al nivel más bajo. A diferencia de los modelos lógicos de los datos, son muy pocos los modelos físicos utilizados. Algunos de los más conocidos son:

- 1) El Modelo unificador
- 2) La memoria de cuadros.

Existen modelos y Sistemas de Bases de Datos, creados para proporcionar funcionalidades específicas en la problemática asociada y en las líneas de investigación abiertas, a continuación se describen algunos tipos de bases de datos:

❖ **Base de datos orientada a objetos:**

El esquema de una base de datos orientada a objetos está representado un conjunto de clases que definen las características y el comportamiento de los objetos que poblarán la base de datos. La diferencia principal respecto a los modelos examinados hasta ahora es la no positividad de los datos.

En efecto, con una base de datos tradicional (entendiendo con este término cualquier base de datos no orientada a objetos), las operaciones que se tienen que efectuar en los datos se les piden a las aplicaciones que los usan.

Con una base orientada a objetos, al contrario, los objetos memorizados en la base de datos contienen tanto los datos como las operaciones posibles con tales datos.

En cierto sentido, se podrá pensar en los objetos como en datos a los que se les ha puesto una inyección de inteligencia que les permite saber cómo comportarse, sin tener que apoyarse en aplicaciones externas.

❖ **Base de datos distribuidas:**

Las bases de datos distribuidas se están utilizando cada vez más en la misma medida en que se usan las arquitecturas de cliente-servidor. Los principales problemas que se generan por el uso de la tecnología de bases de datos distribuidas son en lo referente a duplicidad de datos y a su integridad al momento de realizar actualizaciones a los mismos. Además, el control de la información puede constituir una desventaja, debido a que se encuentra diseminada en diferentes localidades geográficas.

❖ **Tendencias futuras:**

En el futuro la mayoría de las organizaciones cambiarán la forma convencional de manejo de la información a la arquitectura de base de datos a las ventajas derivadas de su uso. El uso de las bases de datos distribuidas se incrementará de manera considerable en la medida en que la tecnología de comunicación de datos brinde más facilidades para ello. El uso de las bases de datos facilitará y soportará en gran medida a los sistemas de información para la toma de decisiones. [2] [3] [9]

### 1.2.4 Modelo Entidad-Relación

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Este modelo distingue los siguientes elementos:

- **Entidad**

Es una cosa u objeto (real o abstracto) que existe en la realidad y acerca del cual queremos almacenar información en la base de datos.

Cada entidad tiene propiedades específicas, llamadas atributos, que la describen y una entidad particular tendrá un valor para cada uno de sus atributos.

Existen dos clases de entidades: fuertes, que son aquellas cuyos ejemplares tienen existencia por sí mismos y débiles, en las cuales la existencia de un ejemplar depende de que exista un cierto ejemplar de otro tipo de entidad.

- **Interrelación**

Es una asociación, vinculación o correspondencia entre diferentes entidades.

- **Dominio y Valor**

En un tipo de entidad o interrelación sus propiedades o características toman valores. El conjunto de posibles valores que puede tomar una cierta característica se denomina dominio.

- **Atributo**

Un atributo es cada una de las propiedades o características que tiene un tipo de entidad o un tipo de interrelación.

- **Claves**

Una **superclave** es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades.

Una **clave candidata** o **clave primaria** es una superclave tal que los subconjuntos propios de esta no son superclave. [2] [3]

### 1.2.5 Modelo Relacional

Las bases de datos relacionales son el tipo de bases de datos actualmente más difundido. Los motivos de este éxito son fundamentalmente dos:

- 1) Ofrecen sistemas simples y eficaces para representar y manipular los datos
- 2) Se basan en un modelo, el relacional, con sólidas bases teóricas

Gracias a su coherencia y facilidad de uso, el modelo se ha convertido en el más usado para la producción de DBMS.

La estructura fundamental del modelo relacional es precisamente la, **relación**, es decir una tabla bidimensional constituida por líneas (tuplas) y columnas (atributos). Las relaciones representan las entidades que se consideran interesantes en la base de datos.

Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos representarán las propiedades de la entidad.

Las tuplas en una relación son una colección no ordenada de elementos diferentes. Para distinguir una tupla de otra, se recurre al concepto de **llave primaria**, o sea a un conjunto de atributos que permiten identificar unívocamente una tupla en una relación.

Naturalmente, en una relación puede haber más combinaciones de atributos que permitan identificar unívocamente una tupla (**llaves candidatas**), pero entre éstas se elegirá una sola para utilizar como llave primaria. Los atributos de la llave primaria no pueden asumir el valor nulo, en tanto que ya no permitirían identificar una tupla concreta en una relación. Esta propiedad de las relaciones y de sus llaves primarias está bajo el nombre de integridad de las entidades.

A menudo, para obtener una llave primaria compuesta de pocos atributos fácilmente manipulables, se introducen uno o más atributos ficticios, con códigos identificativos unívocos para cada tupla de la relación.

Cada atributo de una relación se caracteriza por un nombre y por un dominio. El dominio indica qué valores pueden ser asumidos por una columna de la relación. A menudo un dominio se define a través de la declaración de un tipo para el atributo, pero también es posible definir dominios más complejos y precisos.

El motor de datos se ocupará de controlar que en los atributos de las relaciones se incluyan sólo los valores permitidos por sus dominios. Característica fundamental de los dominios de una base de datos relacional es que sean **atómicos**, es decir que los valores contenidos en las columnas no se puedan separar en valores de dominios más simples. Más formalmente se dice que no es posible tener atributos multivalorados.

Las bases de datos relacionales efectúan todas las operaciones en las tablas usando el álgebra relacional, aunque normalmente no le permiten al usuario usarla. El usuario interactúa con la base de datos a través de una interfaz diferente: el lenguaje SQL, un lenguaje declarativo que permite escribir conjuntos

de datos. Las instrucciones SQL vienen descompuestas por el motor de datos en una serie de operaciones relacionales. [2] [3]

### **1.2.6 Teoría de la Normalización**

La normalización de datos es un proceso durante el cual los esquemas de relación insatisfactorios se descomponen repartiendo sus atributos entre esquemas de relación más pequeños que poseen propiedades deseables.

El objetivo principal de la normalización es eliminar en lo posible las anomalías de actualización y las redundancias.

Inicialmente Codd propuso tres formas normales basadas en las dependencias funcionales, más tarde, Fagin introduce la cuarta y quinta formas normales basadas en las dependencias multivaluadas y las dependencias de proyección-combinación, respectivamente.

- **Primera Forma Normal (1FN)**

Se dice que una relación está en 1FN cuando cada atributo sólo toma un valor del dominio simple subyacente.

- **Segunda Forma Normal (2FN)**

Se dice que una relación está en 2FN sí:

- 1) Está en 1FN
- 2) Cada atributo no principal tiene dependencia funcional completa respecto de cada una de las claves.

- **Tercera Forma Normal (3FN)**

Un esquema de relación R está en 3FN sí, y sólo sí:

- 1) Está en 2FN
- 2) No existe ningún atributo no principal que dependa transitivamente de alguna clave R.

- **Cuarta Forma Normal (4FN)**

Una relación se encuentra en 4FN sí, y sólo sí, las únicas dependencias multivaluadas no triviales son aquellas en las cuales una clave multivaluada determina un atributo, es decir, toda dependencia multivaluada viene determinada por una clave candidata.

- **Quinta Forma Normal (5FN)**

Una relación está en 5FN si, y sólo sí, está en todas las anteriores formas normales y toda dependencia de combinación está implicada por una clave candidata. [2] [3]

## 1.3 MySQL

Se eligió **MySQL** como gestor de base de datos porque permite hacer aplicaciones en distintos sistemas operativos sin necesidad de grandes recursos de máquina.

Todo el sistema de permisos de acceso al servidor, a las bases de datos y sus tablas, **MySQL** lo almacena en una tabla llamada **mysql**, que como todas está en el directorio /data, a menos que hallamos especificado otro directorio.

En la base de datos mysql es donde se guardan todos los permisos y restricciones a los datos de nuestras bases de datos. La principal herramienta es **mysqladmin**, la cual como indica su nombre es la encargada de la administración.

MySQL, crea por defecto al usuario root, con todos los permisos posibles habilitados, podemos utilizar este usuario como administrador o crear otro. [6] [11]

### 1.3.1 La estructura de MySQL

En el directorio /benc se encuentran ejemplos de script y SQL, en el directorio /share están los mensajes de error del servidor para los distintos idiomas.

Los directorios /include y /lib contienen los ficheros \*.h y las bibliotecas necesarias, en /bin están los ficheros ejecutables y en /data se encuentran como subdirectorios cada una de las bases de datos que hayamos creado.

Como hemos dicho, para cada base de datos que nosotros creamos, MySQL crea un directorio con el nombre que le hemos asignado a la base de datos. Dentro de este directorio, por cada tabla que definamos MySQL va a crear tres archivos:

**1) mitabla.ISD:**

El archivo con extensión ISD, es el que contiene los datos de nuestra tabla.

**2) mitabla.ISM:**

Contiene información acerca de las claves y otros datos que MySQL utiliza para buscar datos en el fichero ISD.

**3) mitabla.FRM:**

El archivo FRM, contiene la estructura propia de la tabla.

Dado que las bases de datos de MySQL son simples ficheros de un directorio, para realizar copias de seguridad, podremos utilizar las herramientas de comprensión que habitualmente usamos en nuestro sistema. [6] [11]

### 1.3.2 La seguridad de MySQL

Como se dijo anteriormente, todo el sistema de permisos, MySQL lo guarda en una base de datos llamada mysql, la cual se compone de cinco tablas:

### 1) host:

La tabla host nos informa sobre que máquinas podrán acceder a nuestro sistema, así como a las bases de datos que tendrán acceso y sus diferentes permisos.

### 2) user:

La tabla user contiene información sobre los usuarios, desde que máquinas pueden acceder a nuestro servidor MySQL, su clave y sus diferentes permisos.

Finalmente las siguientes tablas

### 3) db

### 4) tables\_priv

### 5) columns\_priv

nos proveen de un control individual de las bases de datos, tablas y columnas. [6] [11]

## 1.4 PHP

Oficialmente, PHP es un preprocesador de hipertextos, es un lenguaje de programación soportado por HTML. La sintaxis está heredada de C, Java y Perl. Este lenguaje está orientado para los constructores de páginas Web, permitiéndoles crear páginas dinámicamente generadas de forma rápida. PHP, está más orientado a conexiones entre páginas Web y servidores donde se almacenan toda clase de Bases de Datos.

Soporta un único mecanismo de seguridad que permite que varios usuarios estén corriendo scripts PHP sobre el mismo servidor. Este mecanismo está basado en un esquema de permisos de ficheros, permitiendo el acceso a aquellos ficheros que son apropiados por el mismo identificador de usuario que el del script que está intentando acceder a ese fichero, o bien cuando el fichero está en el directorio que es propiedad del mismo identificador de usuario que el del script que está intentando acceder.

El código de PHP está incluido en tags especiales "<?,?>". Lo que hace diferente a PHP es que el código que se deba ejecutar se ejecuta siempre en el servidor, así, al ejecutar el script anterior, el cliente recibirá sólo los resultados de la ejecución por lo que es imposible para el cliente acceder al código que generó la página.

### El protocolo CGI:

El protocolo CGI (Common Gateway Interface) fue creado para establecer un protocolo Standard de comunicación entre el Web-Server y cualquier lenguaje de programación de forma tal que desde el lenguaje "X" puedan recibirse datos que el usuario envía usando el método "POST" o "GET" y además el resultado de la aplicación sea derivado por el Web-Server al browser.

Típicamente para recibir datos se usa alguna biblioteca o módulo del lenguaje elegido que implementa el protocolo CGI y para enviar datos simplemente se envían al Standard-Output desde el lenguaje elegido y el Web-Server se encarga de redireccionar esto al browser. [7]

### 1.4.1 Seguridad y PHP

PHP es un intérprete que puede ser incluido en un servidor Web como un módulo o como un CGI binario. Con él se pueden realizar accesos a ficheros, conexiones de red, etc. PHP está diseñado para ser más seguro que cualquier otro lenguaje de programación de CGI's, como Perl o C.

El método de CGI binario, lo que hace es instalar PHP en el directorio cgi-bin, esto permite a PHP reaccionar ante diversos tipos de ataques.

Un posible ataque, es cuando intenta acceder a los ficheros del servidor Web a los que no se debe tener acceso.

Para evitar esto, existen opciones de configuración que redirigen todas las peticiones al intérprete de PHP forzando una inspección de acceso al fichero que se pide.

Algunas opciones de seguridad son:

- 1) Si se activa la opción **disable-force-cgi-redirect** se obliga a que las peticiones del tipo http sean analizadas por el intérprete PHP.
- 2) Combinar la directiva **Action y AddHandler**, mediante estas opciones se configura la redirección de las llamadas para que sean interpretadas. Esta opción ha sido probada en Apache y a éste servidor se refiere.
- 3) La tercera opción es utilizar las directivas **doc\_root y user dir**. Estas directivas se utilizan en servidores Web que no disponen de la facilidad del redireccionamiento, supongamos que un script no se ejecuta correctamente, en este caso, el código se muestra en pantalla y esto puede violar la propiedad intelectual de ese script.
- 4) Otra práctica segura es mantener la instalación del intérprete fuera del árbol Web, si esto es así, se deberán hacer los ficheros PHP ejecutables, modificando los atributos del fichero y además se deberá incluir en la primera línea del script la dirección del intérprete.

En el caso de tener PHP instalado como un módulo del servidor Apache, este hereda todas las características del servidor. [7]

### 1.4.2 Manejo de conexiones

En PHP las conexiones que se mantienen pueden tener tres estados: Normal (0), Aborted (1) y Timeout (2).

En un script normal, el estado es **normal**, cuando el cliente desconecta, el estado pasa a ser **aborted** y si el límite impuesto por PHP ha transcurrido, el estado es **timeout**.

Una función muy útil para estos casos, es **connection\_status()** que devuelve el estado de la conexión. [7]

## **Conexiones a bases de datos**

Las conexiones persistentes son enlaces SQL que no se cierran cuando la ejecución del script termina. El comportamiento de estas conexiones es el siguiente:

- Cuando se invoca una conexión de este tipo, PHP comprueba si existe una conexión de este mismo tipo o por el contrario, se trata de una nueva conexión. En el caso de que exista, se procede a su uso, y en el caso de que no exista, la conexión se crea. Dos conexiones se consideran iguales cuando están realizadas sobre el mismo servidor, con el mismo usuario y la misma contraseña.

Pero en realidad, estas conexiones permanentes, no proporcionan ningún tipo de funcionalidad adicional frente a conexiones temporales, debido a la forma en que los servidores Web funcionan. [4] [7]

## **1.5 Servidor Apache**

Para poder utilizar PHP y MySQL es necesaria la configuración de un Servidor, en particular se utilizó el Servidor Apache. En el sitio oficial del servidor, se debe descargar el fichero `apache_2.2.44-win32-x86-no_ss.msi`, este fichero se debe ejecutar desde el explorador de archivos.

Una vez instalado el servidor, es necesario comprobar su funcionalidad, si la instalación fue correcta, se puede ver en la página del Servidor Apache.

Para tener el acceso a la base de datos se va a realizar lo siguiente:

- Instalación del Servidor Apache
- Instalación del lenguaje de programación PHP
- Instalación del Manejador de bases de datos MySQL
- Creación de la base de datos en MySQL
- Creación de un programa en PHP, el cual va a establecer la conexión entre la base de datos y la página HTML. [8]

## **ANÁLISIS DEL SISTEMA**

En éste capítulo, se describe el análisis de los requerimientos para tener una idea más clara acerca del funcionamiento del Sistema. Esto nos permite especificar la función y el rendimiento del software, así como establecer las necesidades que debe cumplir.

### **2.1 Planteamiento del problema**

Dentro de las necesidades de hoy en día está el tener procesos automatizados que faciliten el trabajo, es por esto que se propuso un Sistema que permita la organización de una inmobiliaria, dentro de este Sistema, se consideró conveniente llevar el control mediante una base de datos.

El Sistema es manejado por dos tipos de usuarios: el administrador y los empleados, dentro del menú principal se tienen las opciones de Venta, Renta, Compra, Clientes, Inmuebles y Empleados, y dependiendo del tipo de usuario que sea podrá realizar altas, bajas, modificaciones y consultas en los diferentes submenús.

Este Sistema pretende, principalmente ser una aplicación confiable, segura, económica y de fácil acceso.

Se pretende tener el Sistema como una aplicación Web, ya que brinda un mejor servicio para hacer más fácil la compra-venta del inmueble.

#### **2.1.1. Especificación de Requerimientos**

Dentro de un Sistema de Bases de Datos, se proporciona un servicio a través del cual se puede manipular la información por medio del personal autorizado. Existen dos tipos de usuario:

- Usuario “Empleado”, su menú consta de: Ventas, Rentas, Clientes y Salir del Sistema donde consulta e introduce información
- Usuario “Administrador”, su menú consta de: Ventas, Rentas, Compras, Clientes, Inmuebles, Usuarios y Salir del Sistema, el cual tiene acceso a todo el Sistema.

- Para el acceso de los usuarios se necesita de login y password, para un mejor manejo del Sistema
- Se manejan tres tipos de inmuebles: Casas-Habitación, locales y terrenos
- Dentro del menú de inmuebles, se tiene una opción para realizar el avalúo de alguna propiedad comenzando con una cantidad inicial, misma que incrementará de acuerdo a los servicios y construcción que tenga el inmueble
- Para realizar el avalúo, los servicios que se toman en cuenta son: alumbrado, drenaje y pavimentación, en cuanto a construcción se toman en cuenta los metros totales del inmueble, metros construidos y el número de plantas extras
- Al realizar la consulta de un inmueble, aparecerán todos los datos acerca de éste, incluyendo la foto de la propiedad.

### **2.1.2 Antecedentes**

Actualmente, el control de las inmobiliarias es lento, ya que para obtener información de los diferentes inmuebles, de los distintos clientes, y las valuaciones lleva tiempo. Esto nos ha motivado a desarrollar un Sistema que controla y administra cada una de las actividades que se realizan en una inmobiliaria.

Dentro del mercado existen Sistemas que cumplen con estas funciones, pero este Sistema en particular, realiza cualquier función dentro del mismo Sistema como son: Altas, Bajas, Modificaciones y Consultas sin necesidad de entrar a la base de datos para efectuar el cambio.

El Sistema se realizó para tener y dar información a cerca del inmueble que se esté tratando. Las operaciones tomadas en cuenta para los inmuebles son: la compra, venta y renta.

### **2.1.3 Alcances**

- ❖ Un Sistema de Bases de Datos que mantenga el control de una inmobiliaria en la que incluye:
  - Datos generales de las Ventas.
  - Datos generales de las Rentas.
  - Datos generales de las Compras.
  - Información a cerca de los Clientes.
  - Información a cerca de los Inmuebles.
  - Información a cerca de los Usuarios.
- ❖ Una tesina en la cual se presentan las etapas, análisis, diseño, implementación y pruebas del Sistema.

#### **2.1.4 Metodología**

Se utilizó el ciclo de vida del software con la metodología de cascada. La mecánica de aplicación de cada una de las fases del modelo de cascada se presenta conforme el progreso que se tiene del Sistema.

En la primera fase se realizó la especificación de los requerimientos para poder tener más detalladas las necesidades del Sistema.

En la segunda fase, se realizó el análisis de los requerimientos, estableciendo los requerimientos funcionales y los no funcionales, así como el diseño del Sistema.

Para la tercera fase, se realizó la implementación del Sistema, una vez implementado, se realizaron las pruebas necesarias para verificar que el Sistema satisfacía los requerimientos.

En la cuarta fase, se realizó la integración del Sistema y sus pruebas, dando paso a la quinta fase y última para la operación y manejo del Sistema.

#### **2.1.5 Definición de Requerimientos Funcionales**

- ❖ El usuario puede acceder al Sistema sólo introduciendo su nombre de usuario y contraseña.
  
- ❖ El Sistema verifica que tipo de usuario accesa para poder entrar al menú correspondiente.
  
- ❖ El Sistema permite realizar las siguientes funciones de acuerdo al tipo de usuario que haya accedido:
  - Ventas:  
Dentro de Ventas se tienen las opciones de: Altas, Bajas, Consultas y Modificaciones.
  
  - Rentas:  
Dentro de Rentas se tienen las opciones de: Altas, Bajas, Consultas y Modificaciones.
  
  - Compras:  
Dentro de Compras se tienen las opciones de: Altas, Bajas, Consultas y Modificaciones.
  
  - Clientes:  
Dentro de Clientes se tienen las opciones de: Altas, Bajas, Consultas y Modificaciones.

- Inmuebles:  
Dentro de Inmuebles se tienen las opciones de: Altas, Bajas, Consultas y Modificaciones.
- Usuarios:  
Dentro de Usuarios se tienen las opciones de: Altas, Bajas, Consultas y Modificaciones.

### **2.1.6 Definición de Requerimientos No Funcionales**

- El equipo necesario es una computadora con procesador PIV a 2.4 GHZ con 512 MB de memoria RAM y un HD de 80 GB.
- La velocidad de carga y respuesta del sistema, va a depender de los componentes de Hardware.
- Si no se cuenta con Hardware necesario, no funcionará adecuadamente el Sistema de Bases de Datos.

### **2.1.7 Diagrama de Flujo de Datos**

Con el modelo funcional, se fueron desglosando cada una de las funciones, a través de niveles que van mostrando cada una de las acciones del Sistema.

El Sistema tiene las siguientes funciones:

1. Ventas
2. Rentas
3. Compras
4. Clientes
5. Inmuebles
6. Empleados

Cada una de estas funciones, dependiendo del usuario, se podrán ejecutar altas, bajas, modificaciones y consultas.

Los diagramas resultantes para estas funciones son los siguientes:

#### **Nivel 0.- Sistema General**

La Figura 2.1 muestra el Sistema General, donde el empleado o el administrador del Sistema, envían una orden al Sistema de Bases de Datos, éste la procesa, la ejecuta y la muestra en la pantalla dependiendo del tipo de usuario.

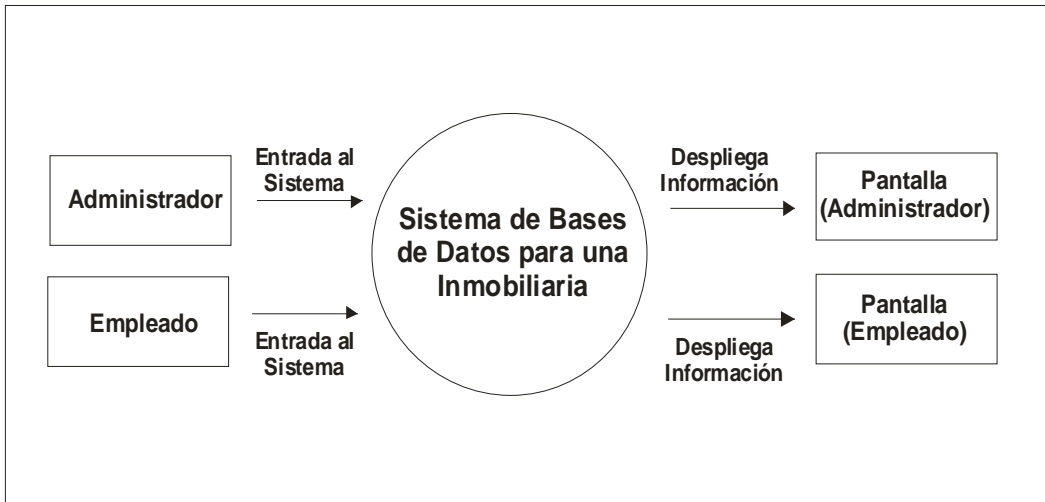


Figura 2.1. Nivel 0.- Sistema General

### Nivel 1.- Acceso al Sistema

La Figura 2.2 muestra el acceso al Sistema, tanto el administrador como el empleado introducen login y password, los cuales son validados para poder tener el acceso.

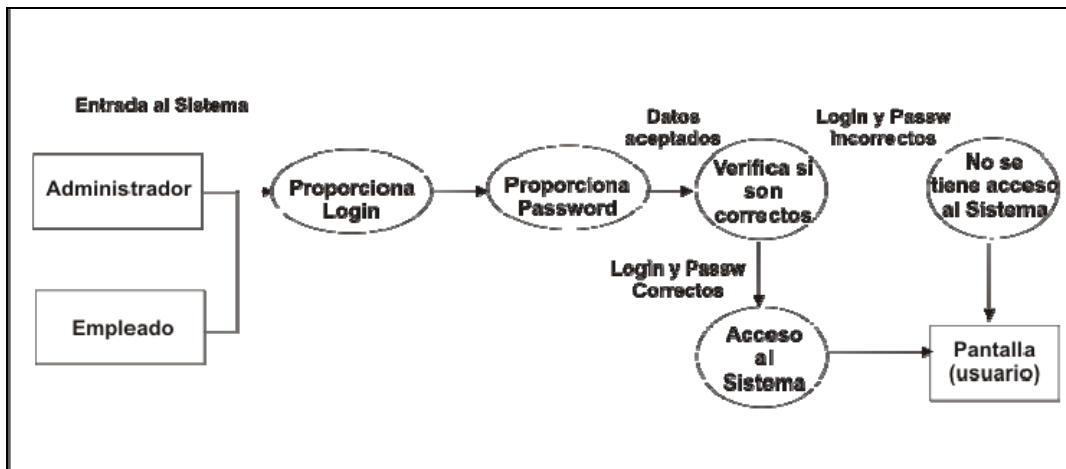


Figura 2.2. Nivel 1.- Acceso al Sistema

### Nivel 2.- Funciones generales del Sistema

La Figura 2.3 muestra las funciones generales del Sistema de Base de datos, dependiendo del tipo de usuario se desplegará el menú correspondiente.

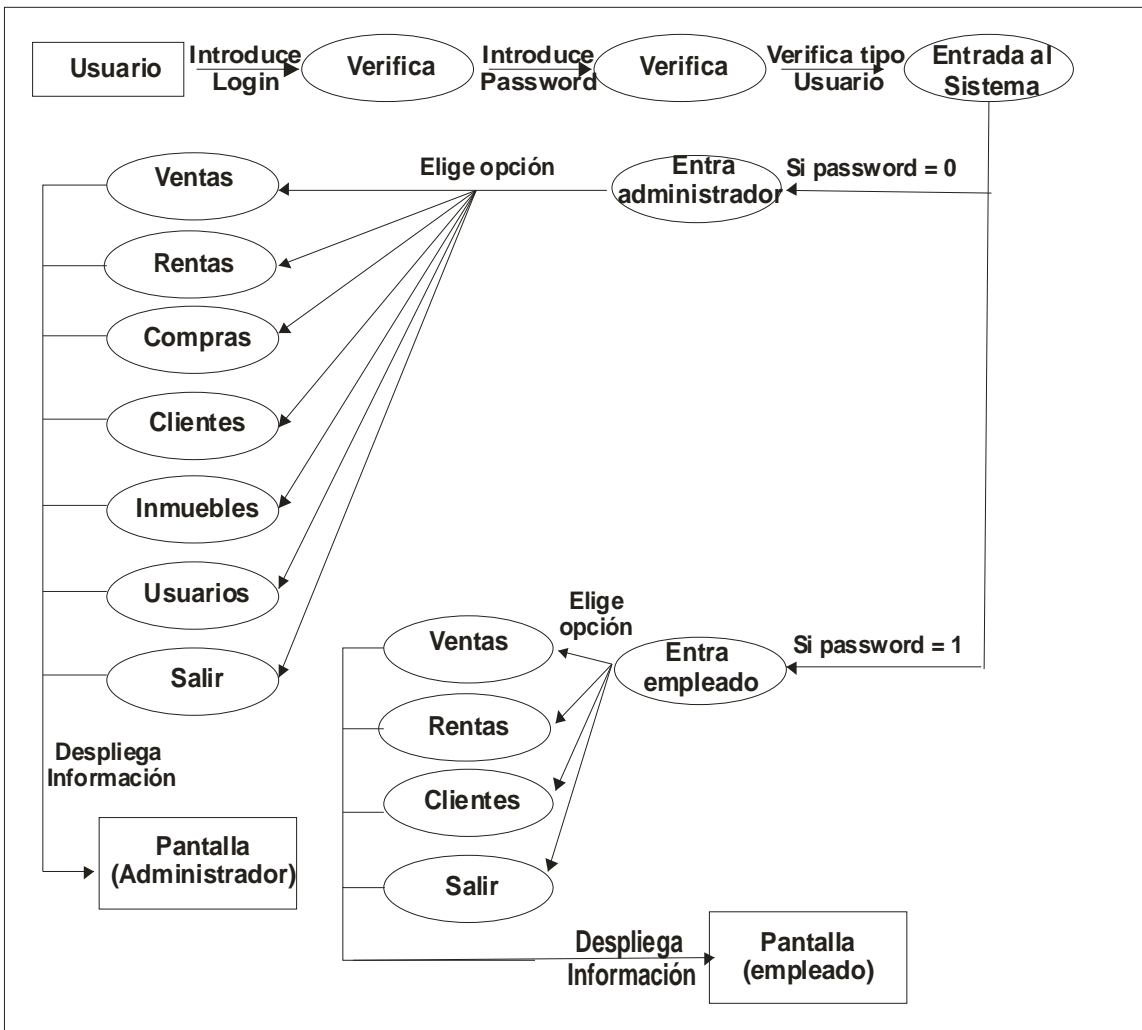


Figura 2.3. Nivel 2.- Funciones generales del Sistema

### Nivel 3.- Ventas

En la Figura 2.4 se muestran las funciones que se pueden realizar en el módulo de ventas.

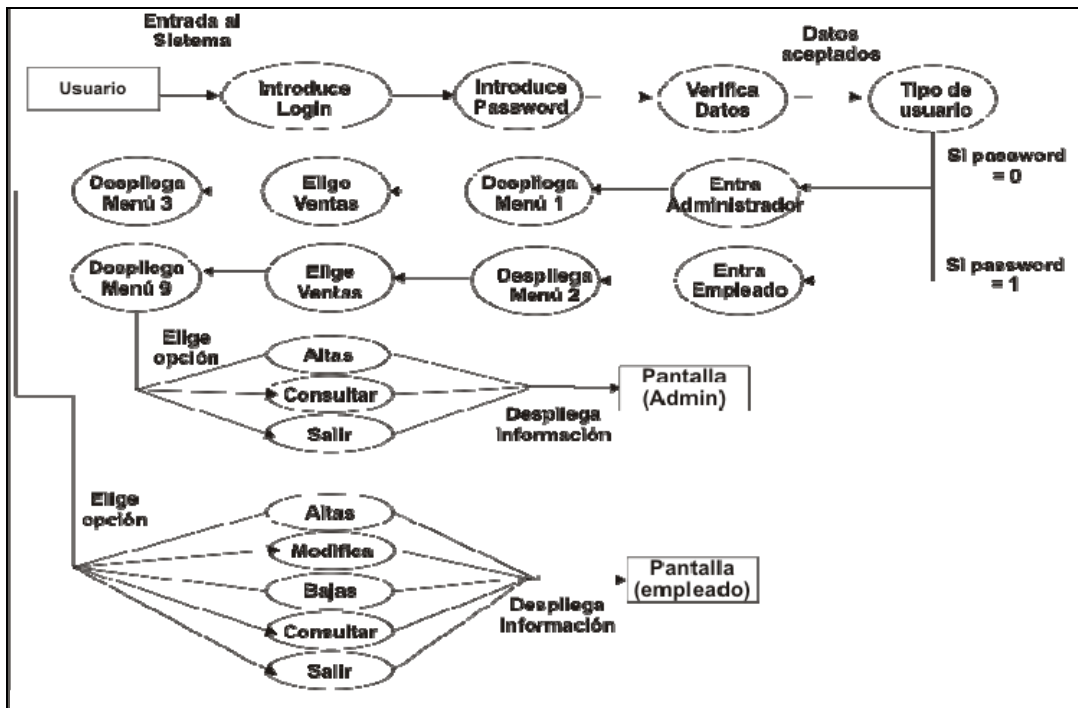


Figura 2.4. Nivel 3.- Ventas

### Nivel 3.1.- Alta de Ventas

Para dar de alta una venta, el administrador o el empleado, deben introducir los datos correspondientes, posteriormente se validan los datos, y en caso de ser correctos se da de alta la venta en la base de datos como se muestra en la Figura 2.5.

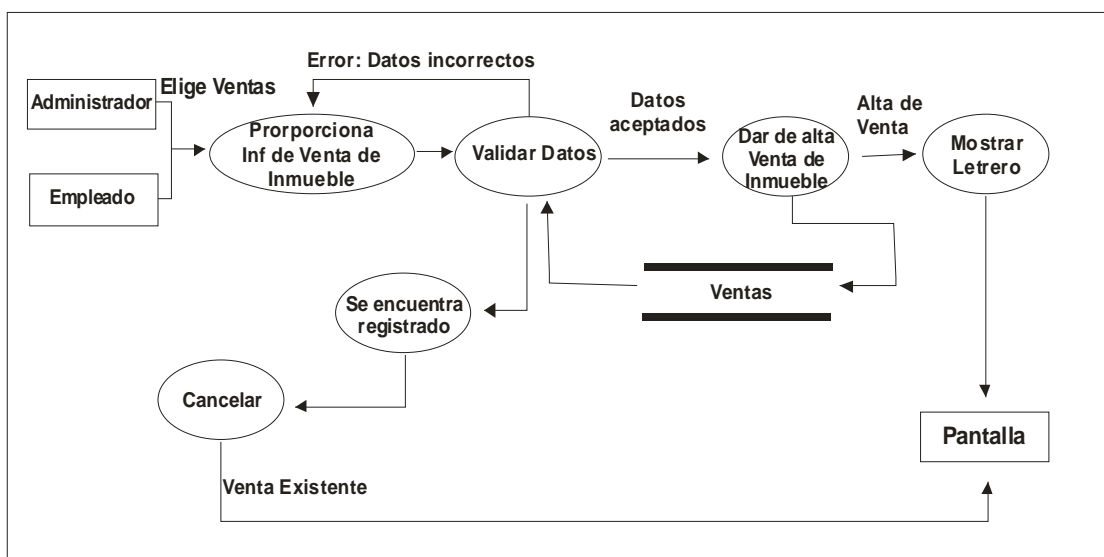


Figura 2.5. Nivel 3.1.- Alta de Ventas

### Nivel 3.2.- Modificar Ventas

La Figura 2.6 muestra la modificación de una venta, para ello, el administrador debe proporcionar el dato a modificar, posteriormente se valida y si es encontrado, se realizan los cambios y se guarda la nueva información en el almacén.

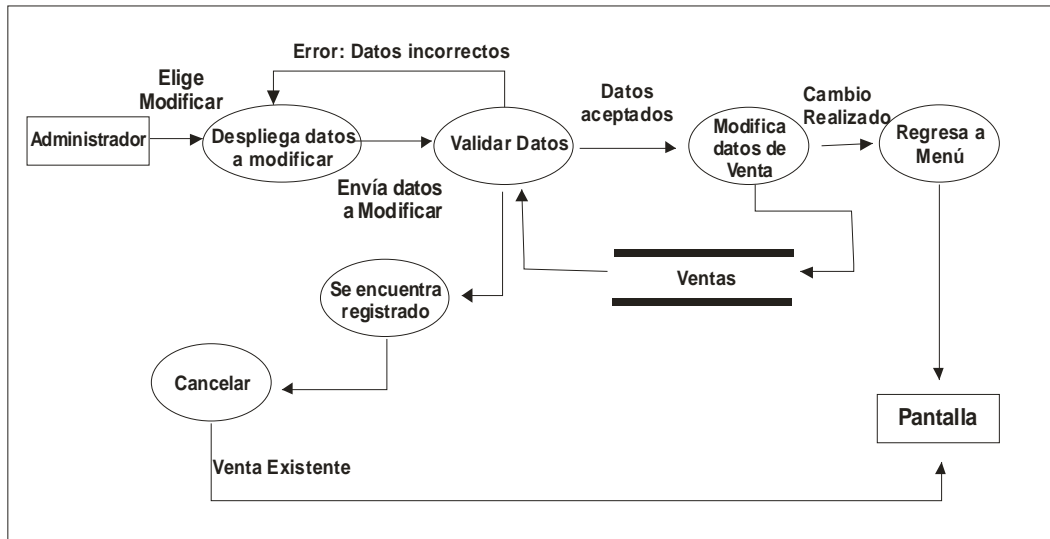


Figura 2.6. Nivel 3.2.- Modificar Ventas

### Nivel 3.3.- Baja de Ventas

En la Figura 2.7 se muestra el proceso para dar de baja una venta, para ello, el administrador selecciona la venta que desea eliminar, la cual se valida y se busca en el almacén de datos. Cuando el elemento haya sido encontrado, el administrador determinará la eliminación o la cancelación de la misma.

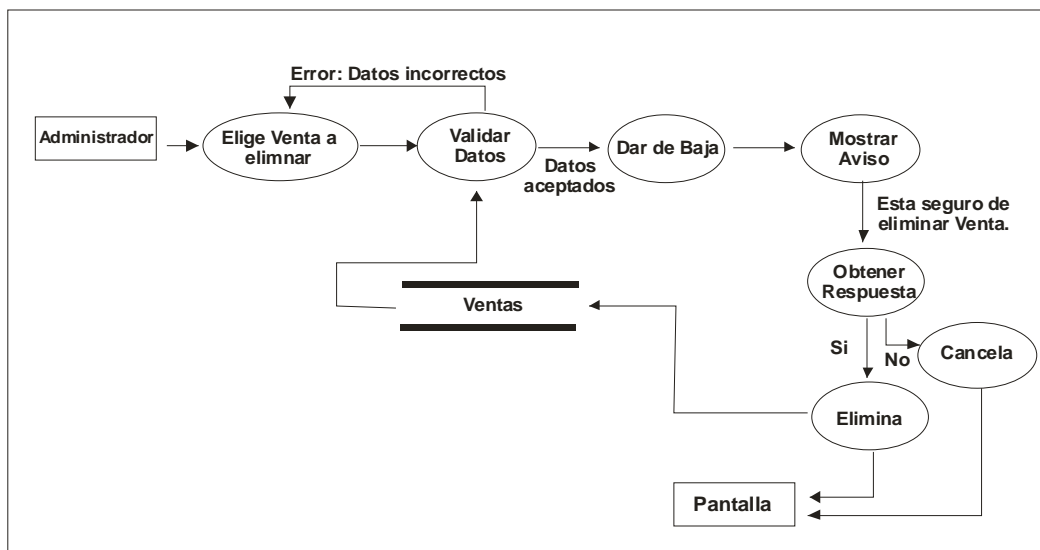


Figura 2.7. Nivel 3.3.- Baja de Ventas

### Nivel 3.4.- Consulta de Ventas

La Figura 2.8 muestra como el administrador o el empleado puede realizar una consulta de venta, en la que se selecciona el inmueble a consultar y se despliega la información correspondiente.

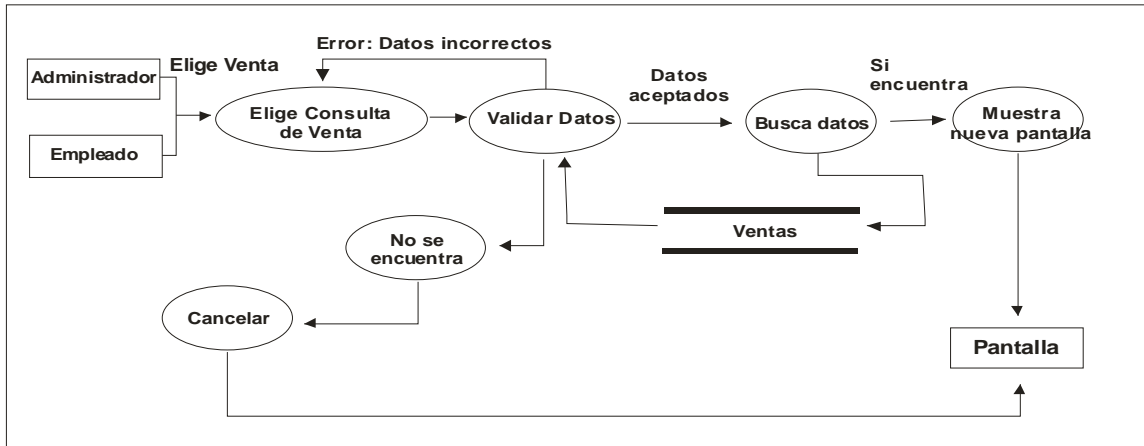


Figura 2.8. Nivel 3.4.- Consulta de Ventas

### Nivel 4.- Rentas

La Figura 2.9 muestra el módulo de Rentas, en el cual se podrán ejecutar los submenús: Altas, Bajas, Modificaciones y Consultas, que dependiendo del tipo de usuario podrá realizar en el Sistema de base de datos.

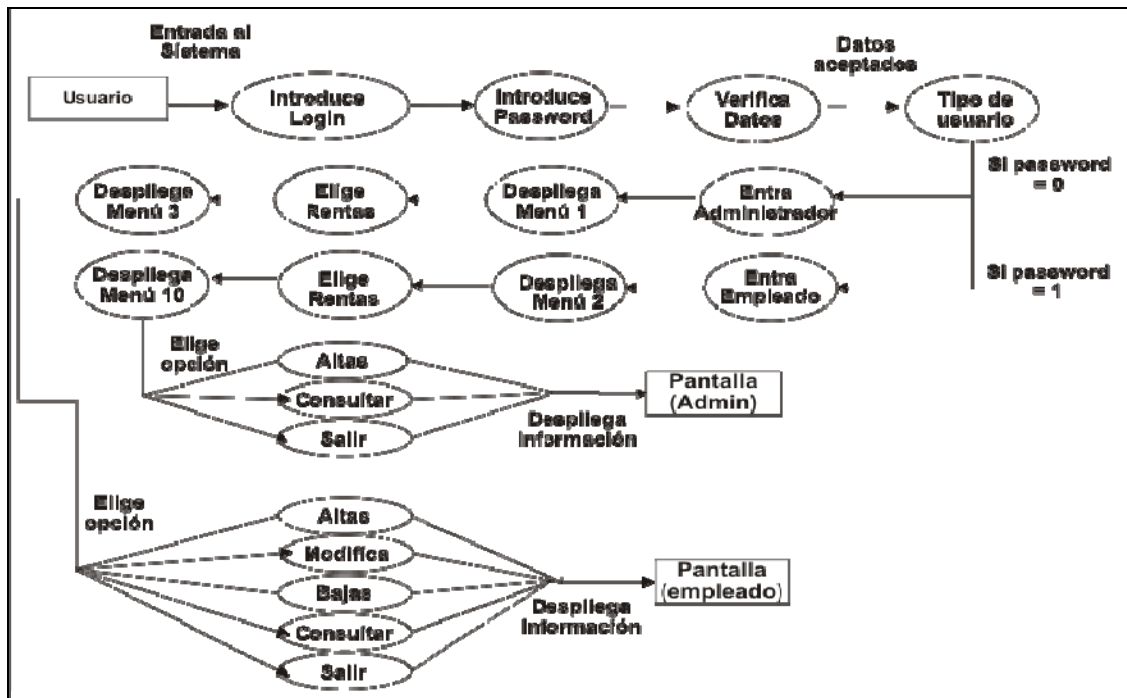


Figura 2.9. Nivel 4.- Rentas

## Nivel 4.1.- Alta de Rentas

En la Figura 2.10 se muestra como el administrador o el empleado deben introducir los datos correspondientes a la Renta que se dará de alta, posteriormente se validan los datos, en caso de ser correctos se da de alta la renta en la base de datos.

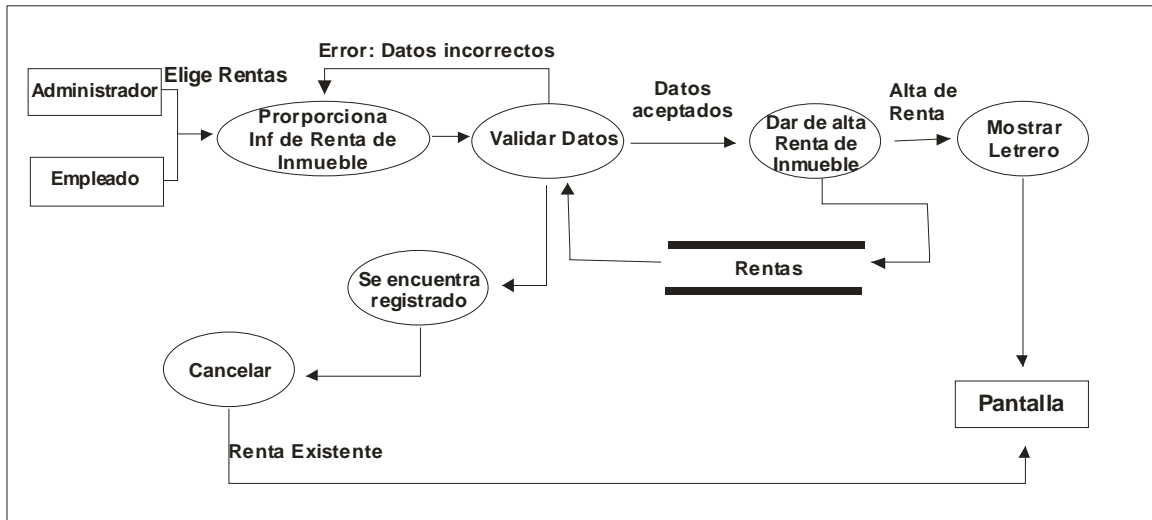


Figura 2.10. Nivel 4.1.- Alta de Rentas

## Nivel 4.2.- Modificar Rentas

En la Figura 2.11 se muestra el proceso para modificar una renta, en este módulo, el administrador debe seleccionar el icono correspondiente, posteriormente se despliegan los datos a modificar, se valida y si es encontrado el dato, se realizan los cambios y se guarda la nueva información en el almacén.

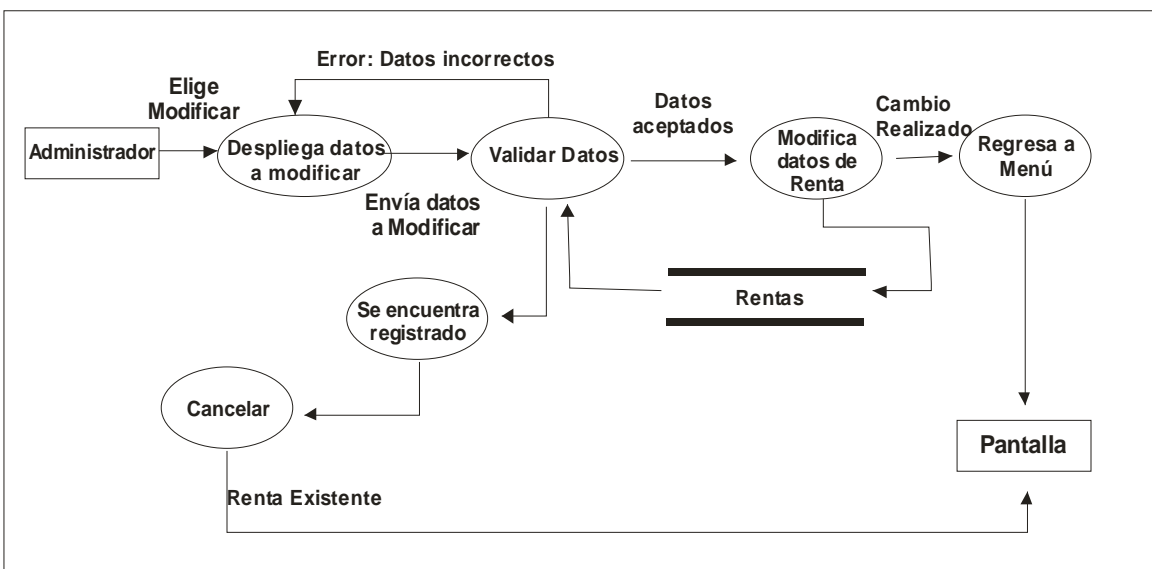


Figura 2.11. Nivel 4.2.- Modificar Rentas

### Nivel 4.3.- Baja de Rentas

En la Figura 2.12 se muestra como dar de baja una renta, en este módulo, el administrador selecciona la renta que desea eliminar, la cual se valida y se busca en el almacén de datos. Cuando el elemento haya sido encontrado, el administrador determinará la eliminación o la cancelación de la misma.

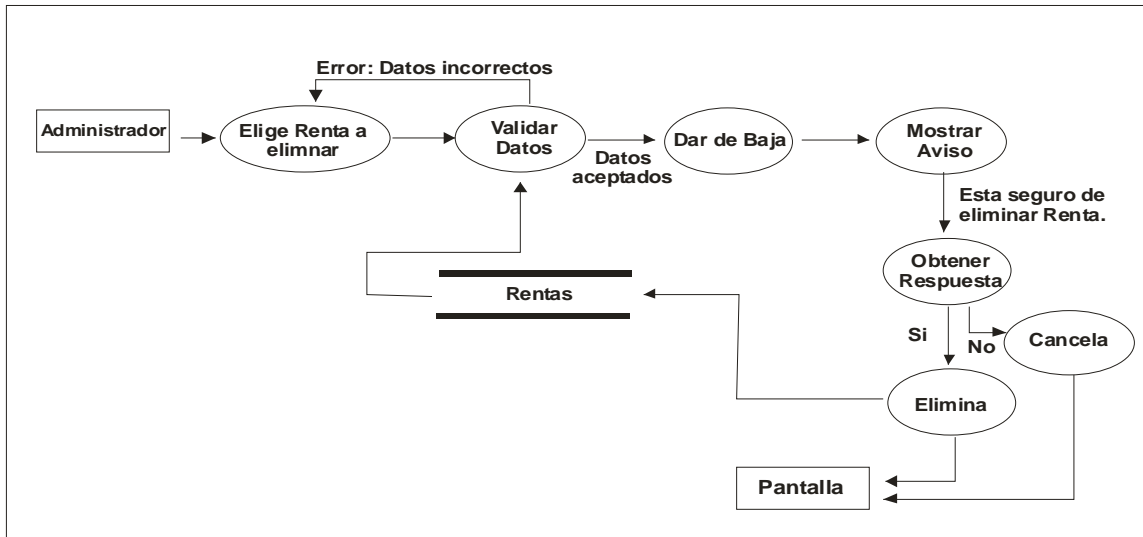


Figura 2.12. Nivel 4.3.- Baja de Rentas

### Nivel 4.4.- Consulta de Rentas

En la Figura 2.13 se describe como el administrador o el empleado pueden realizar la consulta de una renta, esta se selecciona y se despliega la información correspondiente.

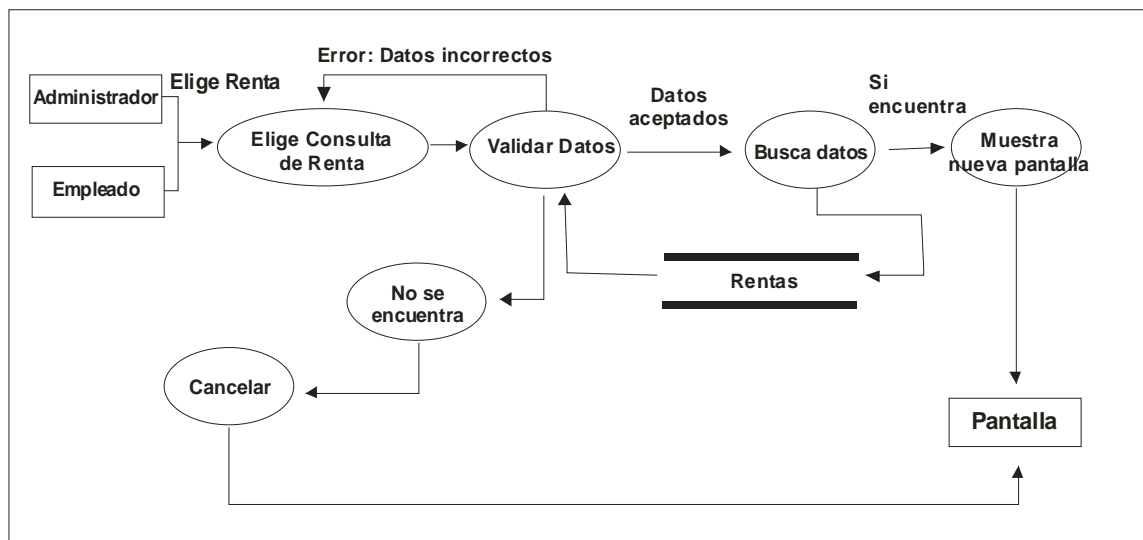


Figura 2.13. Nivel 4.4.- Consulta de Rentas

## Nivel 5.- Compras

Dentro de Compras el administrador podrá ejecutar los submenús en el Sistema de Base de Datos, como son: Altas, Bajas, Modificaciones y Consultas como lo muestra la Figura 2.14.

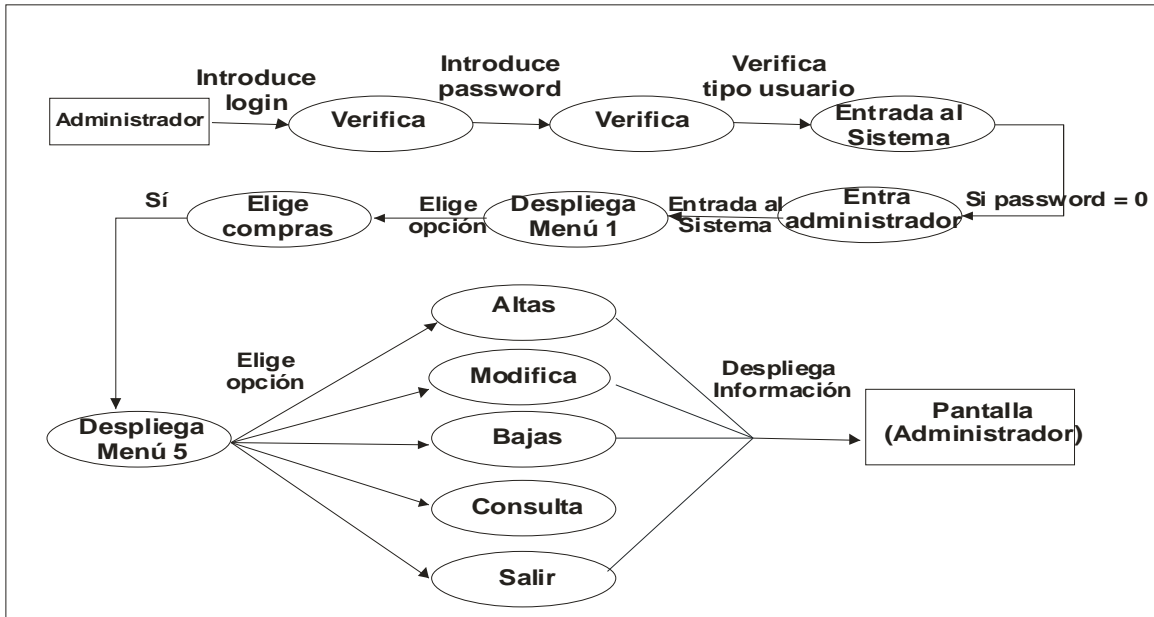


Figura 2.14. Nivel 5.- Compras

## Nivel 5.1.- Alta de Compras

En la Figura 2.15 se muestra como el administrador debe introducir los datos correspondientes a la Compra que se dará de alta, posteriormente se validan los datos, y en caso de ser correctos se da de alta la compra en la base de datos.

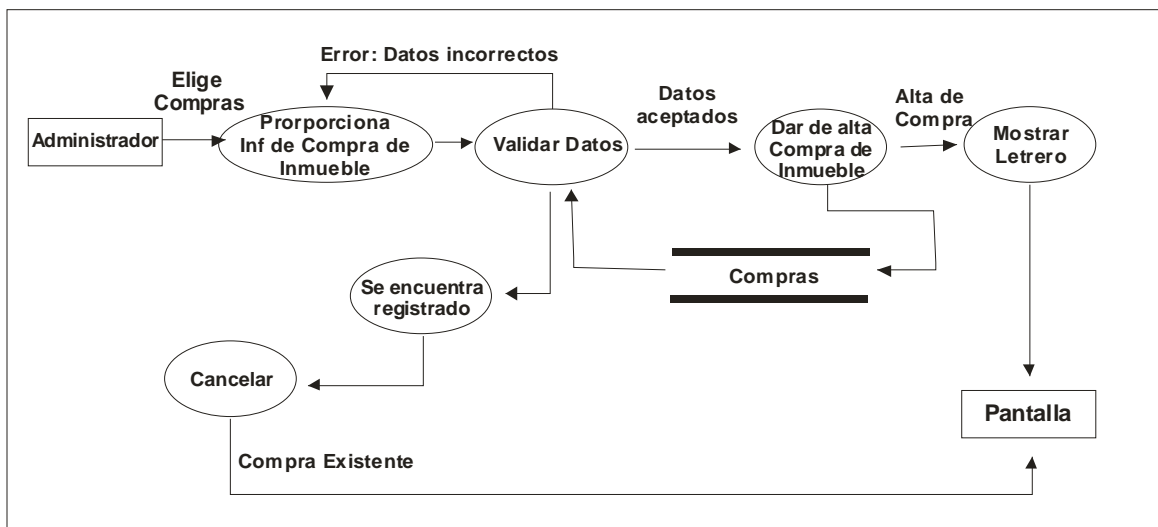


Figura 2.15. Nivel 5.1.- Alta de Compras

## Nivel 5.2.- Modificar Compras

En la Figura 2.16 se describe el proceso para modificar una compra, para ello, el administrador debe seleccionarla, posteriormente se mostrarán los datos a modificar, después se valida y si es encontrado el dato, se realizan los cambios y se guarda la nueva información en el almacén.

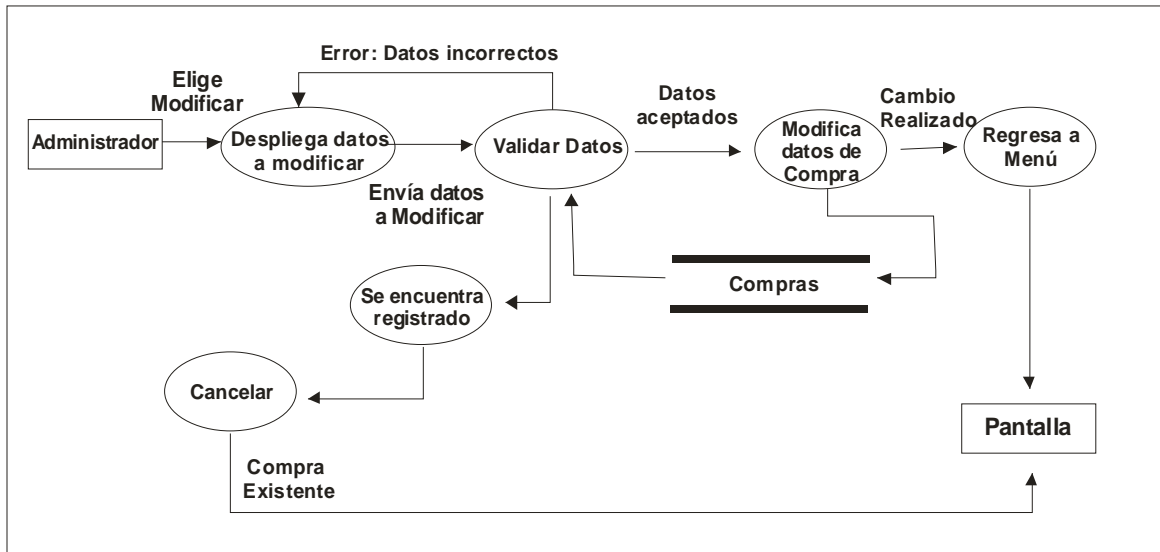


Figura 2.16. Nivel 5.2.- Modificar Compras

## Nivel 5.3.- Baja de Compras

Para dar de baja una compra, el administrador debe seleccionar la compra que quiere eliminar, la cual será validada y buscada en el almacén de datos. Cuando el elemento haya sido encontrado, el administrador determinará la eliminación o la cancelación de la misma como lo describe la Figura 2.17.

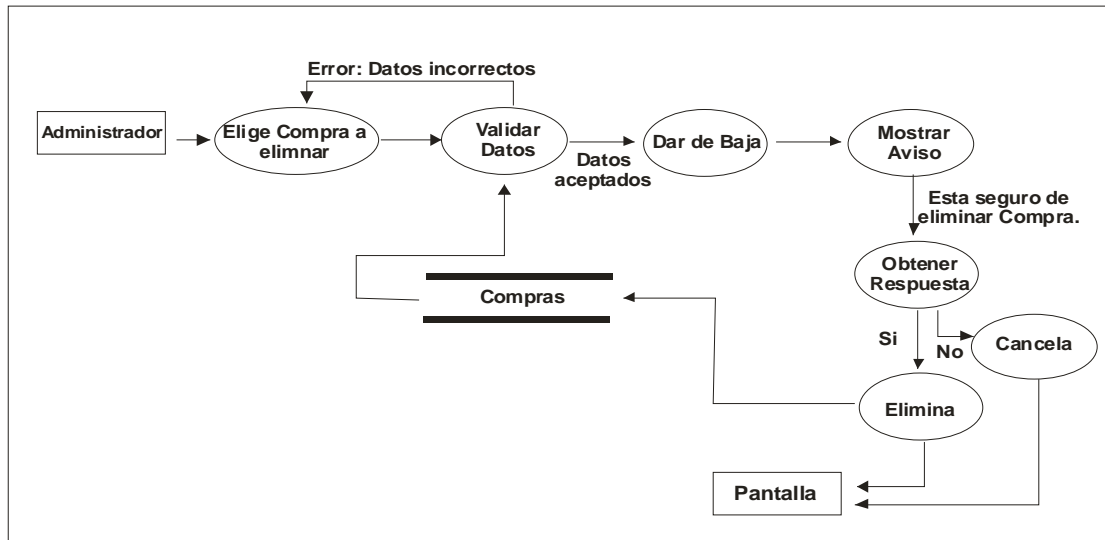


Figura 2.17. Nivel 5.3.- Baja de Compras

### Nivel 5.4.- Consulta de Compras

La Figura 2.18 muestra como el administrador podrá realizar la consulta de una compra, dentro de este módulo, se selecciona el icono correspondiente y posteriormente, se despliega la información.

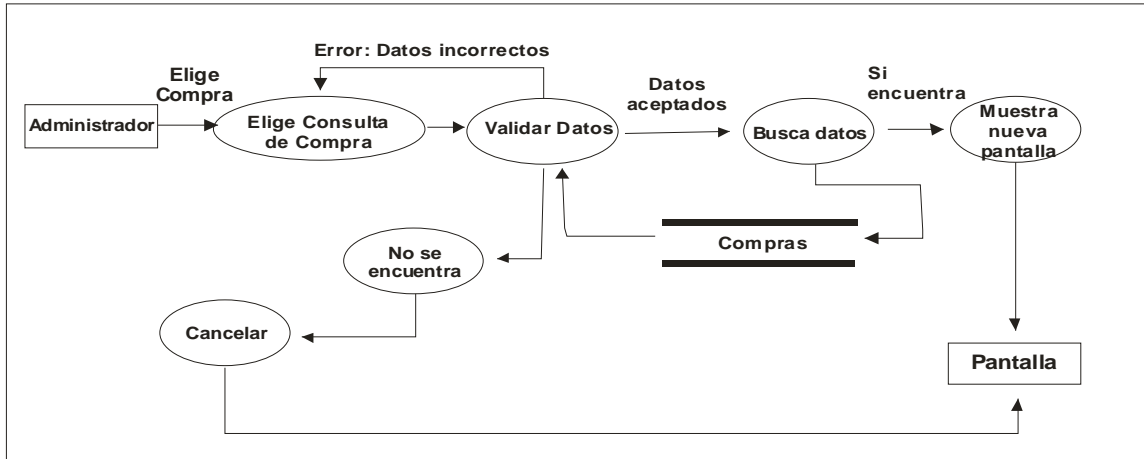


Figura 2.18. Nivel 5.4.- Consulta de Compras

### Nivel 6.- Clientes

Dentro de Clientes el administrador o el empleado pueden ejecutar los submenús en el Sistema de Base de Datos, como son: Altas, Bajas, Modificaciones y Consultas como lo describe la Figura 2.19.

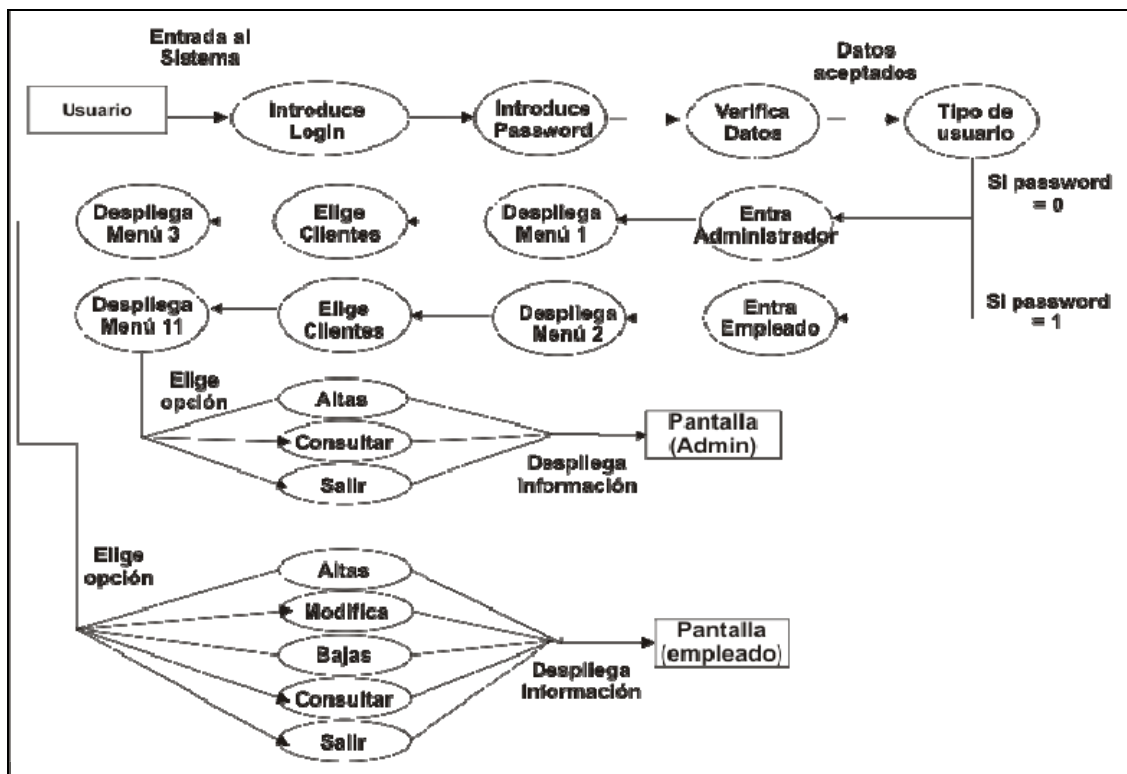


Figura 2.19. Nivel 6.- Clientes

## Nivel 6.1.- Alta de Clientes

La Figura 2.20 muestra los pasos para dar de alta a un cliente, en este módulo, el administrador o el empleado introducen los datos correspondientes, una vez que se proporcionan, se verifica que no esté registrado y se da de alta en la base de datos.

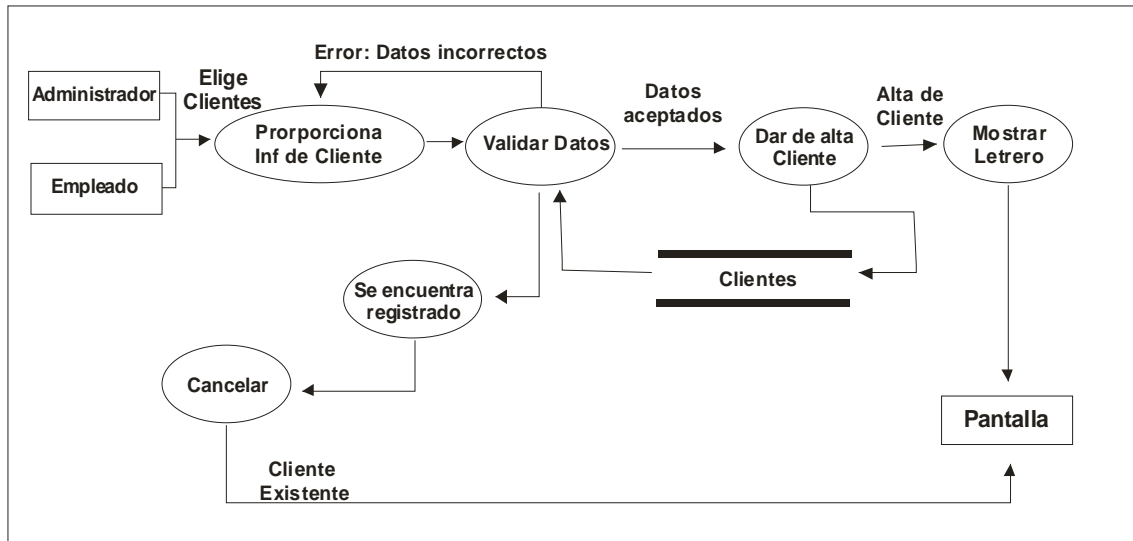


Figura 2.20. Nivel 6.1.- Alta de Clientes

## Nivel 6.2.- Modificar Clientes

La Figura 2.21 muestra como realizar la modificación de un cliente, el administrador debe seleccionarlo, si el dato es encontrado, entonces se despliega la información del cliente, posteriormente, se realizan los cambios y se guarda la nueva información en el almacén.

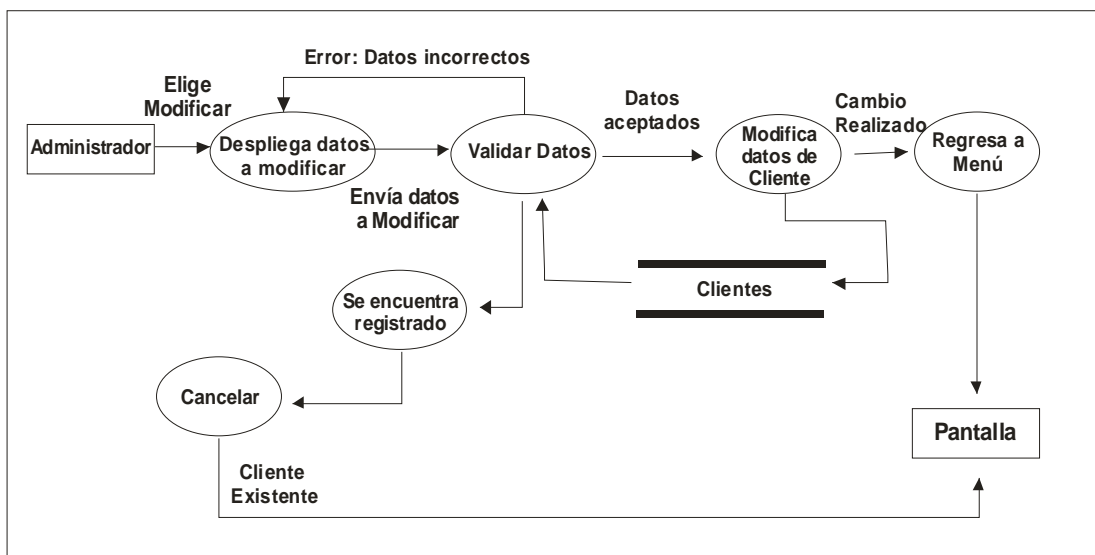


Figura 2.21. Nivel 6.2.- Modificar Clientes

### Nivel 6.3.- Baja de Clientes

En la Figura 2.22 se muestra el proceso para dar de baja un cliente, el administrador selecciona el cliente que desea eliminar, la cual será validado y buscado en el almacén de datos. Cuando el elemento haya sido encontrado, el administrador determinará la eliminación o cancelación de la misma.

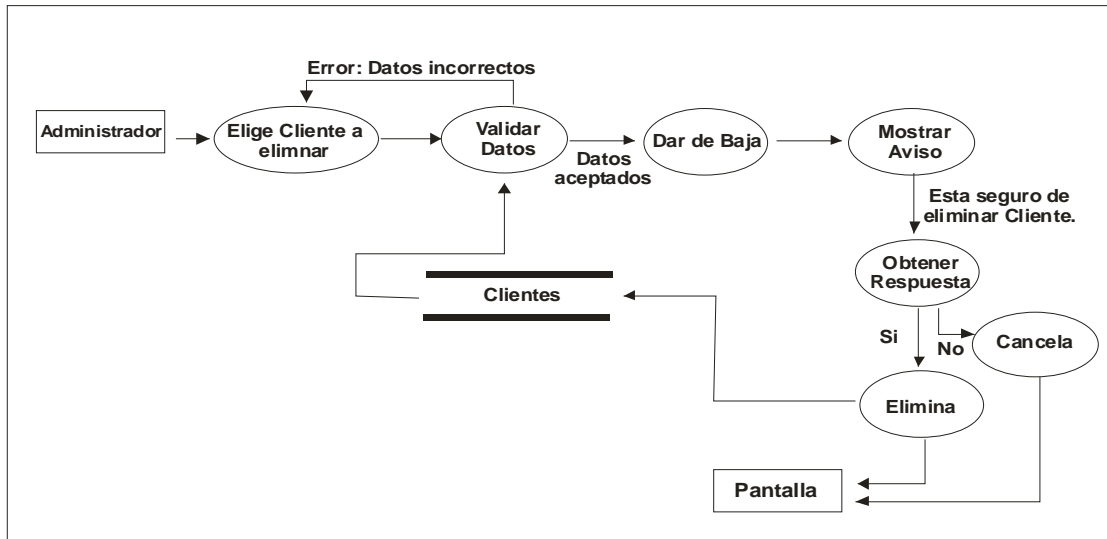


Figura 2.22. Nivel 6.3.- Baja de Clientes

### Nivel 6.4.- Consulta de Clientes

En la Figura 2.23 se muestra la consulta de clientes, el administrador o el empleado pueden realizarla por medio de una selección, se busca en el almacén de datos y se despliega en pantalla la información solicitada en caso de que se encuentre.

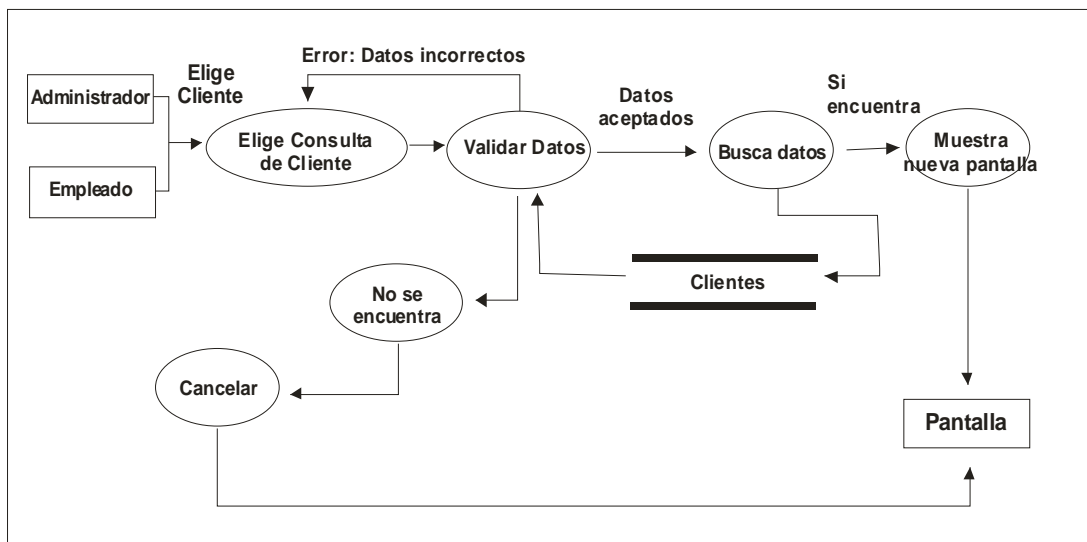


Figura 2.23. Nivel 6.4.- Consulta de Clientes

## Nivel 7.- Inmuebles

En la Figura 2.24 se describe como el administrador ejecuta los submenús de Inmuebles en el Sistema de Base de Datos, como son: Altas, Bajas, Modificaciones y Consultas.

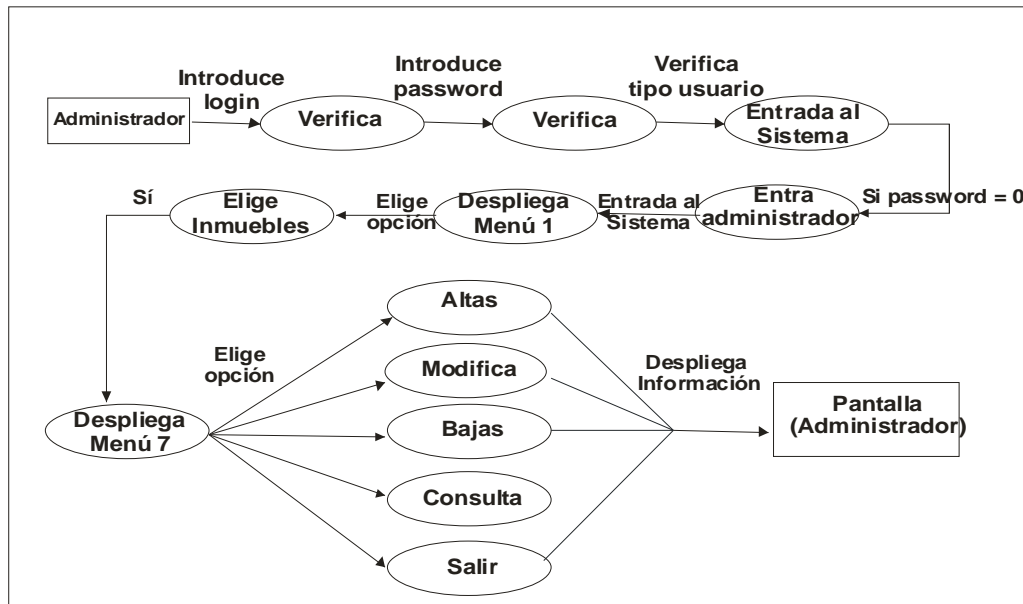


Figura 2.24. Nivel 7.- Inmuebles

### Nivel 7.1.- Alta de Inmuebles

La Figura 2.25 describe como el administrador debe introducir los datos correspondientes al Cliente que se dará de alta en el Sistema, posteriormente, se validan los datos, y en caso de ser correctos se da de alta el cliente en la base de datos.

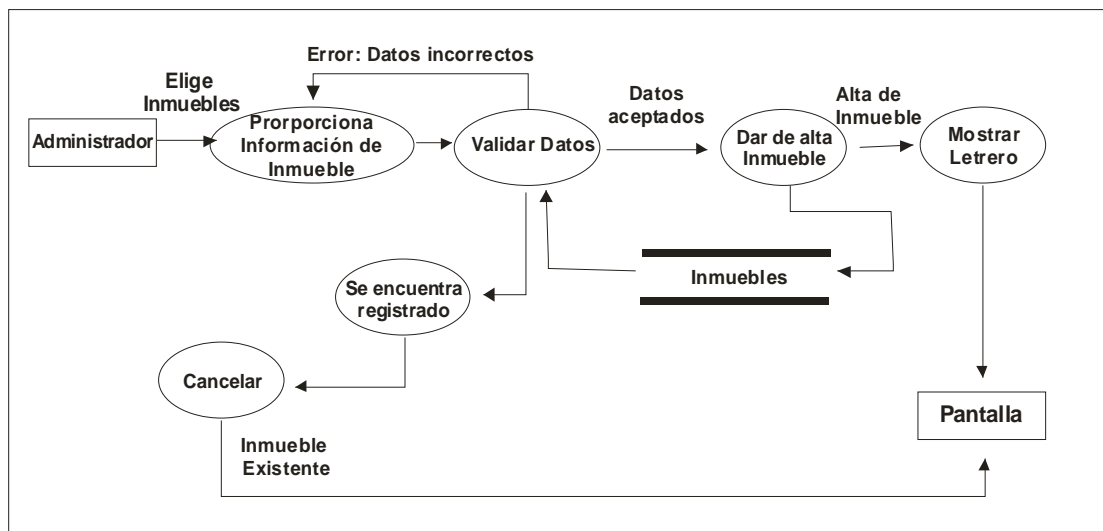


Figura 2.25. Nivel 7.1.- Alta de Inmuebles

## Nivel 7.2.- Modificar Inmuebles

Para realizar la modificación de un inmueble, el administrador selecciona el inmueble a modificar, si el dato es encontrado, entonces se despliega la información del inmueble, se realizan los cambios y se guarda la nueva información en el almacén como lo muestra la Figura 2.26.

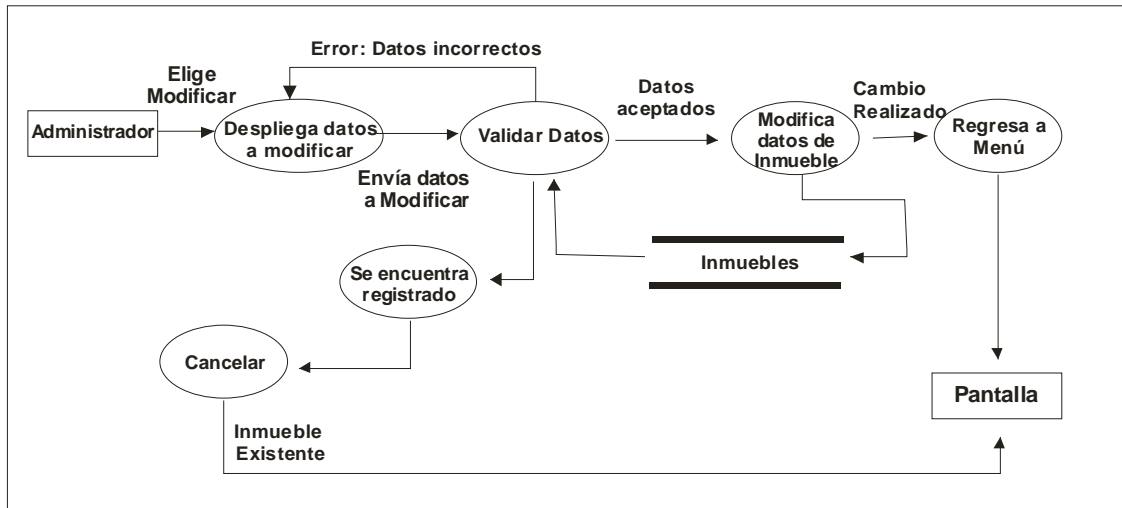


Figura 2.26. Nivel 7.2.- Modificar Inmuebles

## Nivel 7.3.- Baja de Inmuebles

La Figura 2.27 muestra el proceso para eliminar cualquier inmueble, para ello, el administrador realiza una selección de inmueble, el cuál será validado, una vez aceptado, el sistema buscará en el almacén para eliminarla.

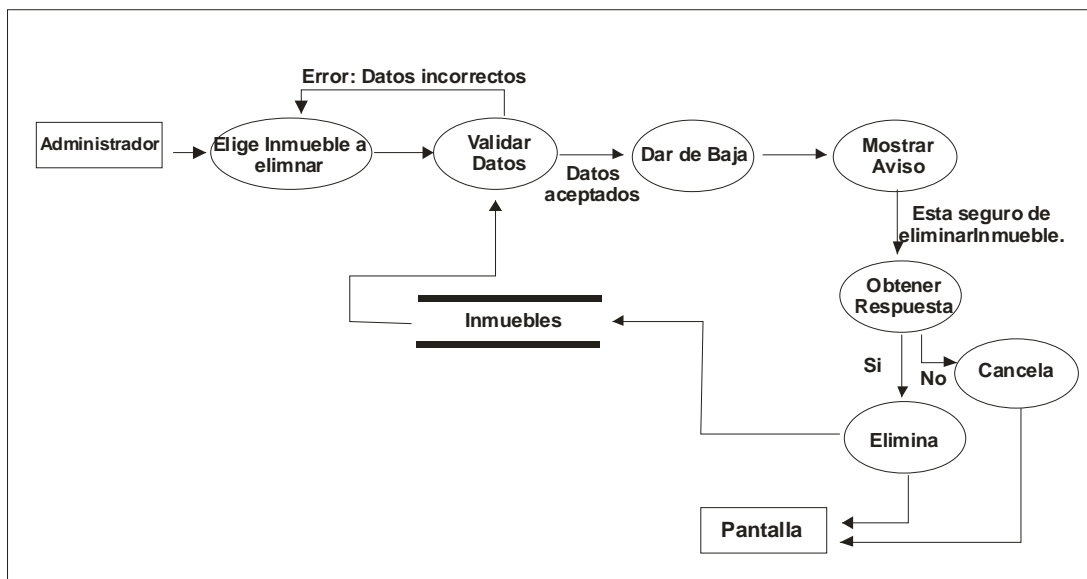


Figura 2.27. Nivel 7.3.- Baja de Inmuebles

## Nivel 7.4.- Consulta de Inmuebles

Las consultas de los inmuebles se realizan a través de la selección que realice el administrador dentro de la pantalla, una vez encontrado, se despliega la información en la pantalla tal como lo muestra la Figura 2.28.

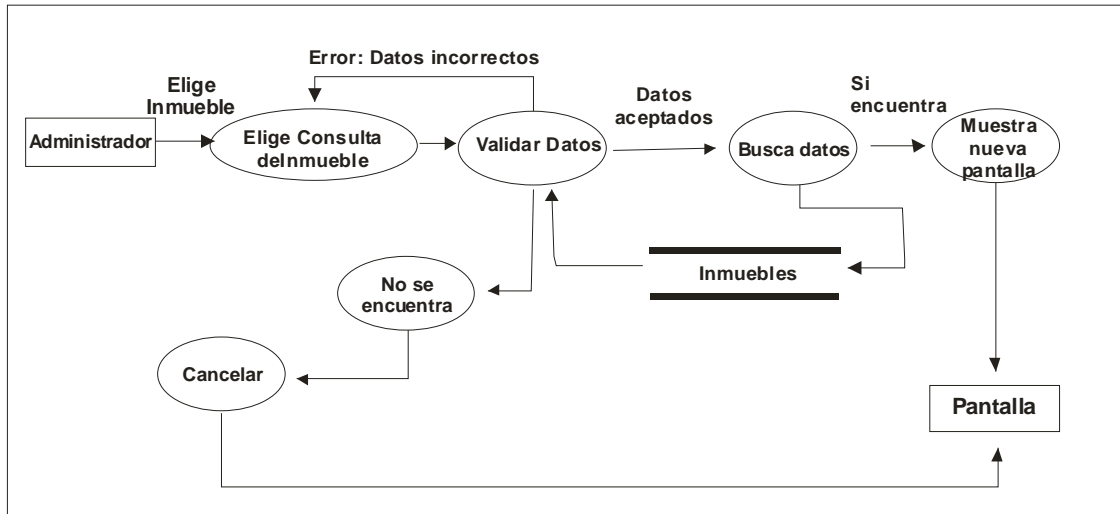


Figura 2.28. Nivel 7.4.- Consulta de Inmuebles

## Nivel 8.- Usuarios

En la Figura 2.29 se muestran las funciones que tiene el Sistema de Base de Datos para los Usuarios, que son los que manejarán el Sistema.

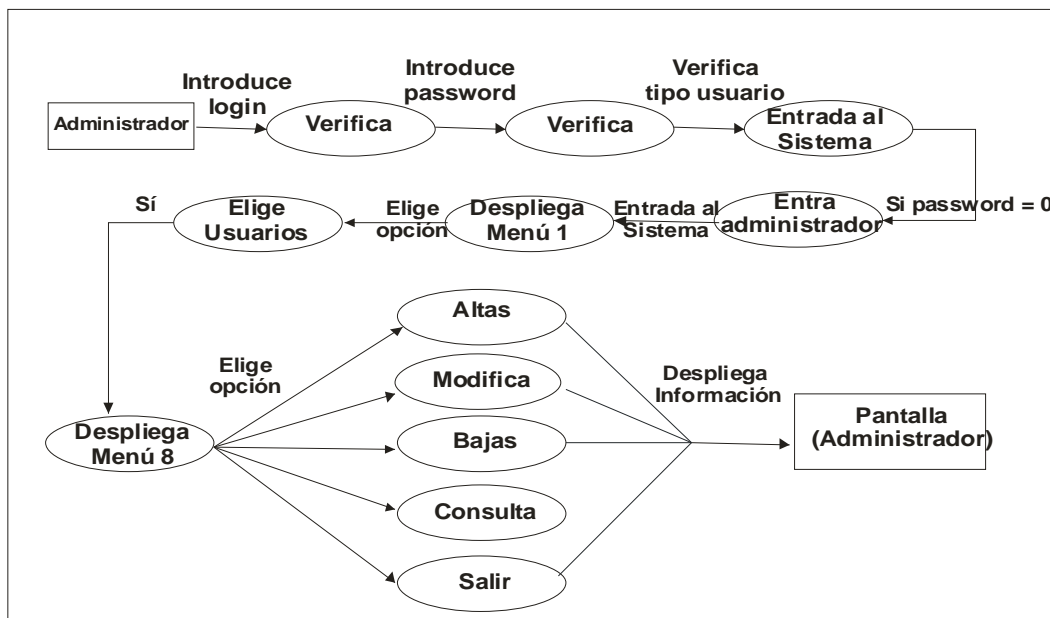


Figura 2.29. Nivel 8.- Usuarios

## Nivel 8.1.- Alta de Usuarios

La Figura 2.30 muestra como dar de alta a un usuario, para esto, el administrador introduce los datos correspondientes al usuario que se dará de alta, en este menú se podrá decir si es administrador o empleado, después se validan los datos, y si están correctos se da de alta en la base de datos.

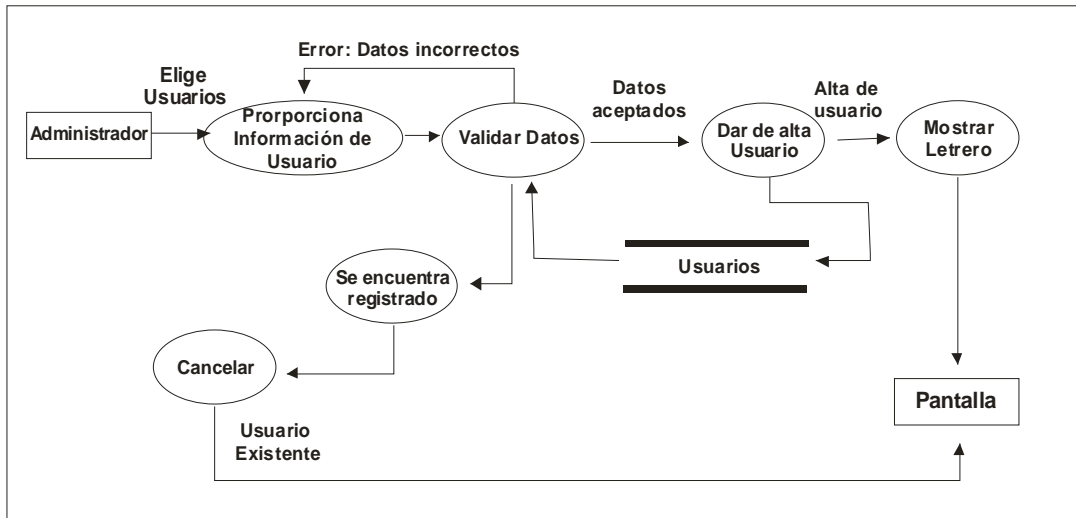


Figura 2.30. Nivel 8.1.- Alta de Usuarios

## Nivel 8.2.- Modificar Usuarios

Para modificar un usuario, el administrador debe seleccionarlo, una vez hecho esto, se desplegarán los datos del usuario para realizar la modificación, posteriormente, se valida y si es encontrado el dato, se realizan los cambios y se guarda la nueva información en el almacén, como lo muestra la Figura 2.31.

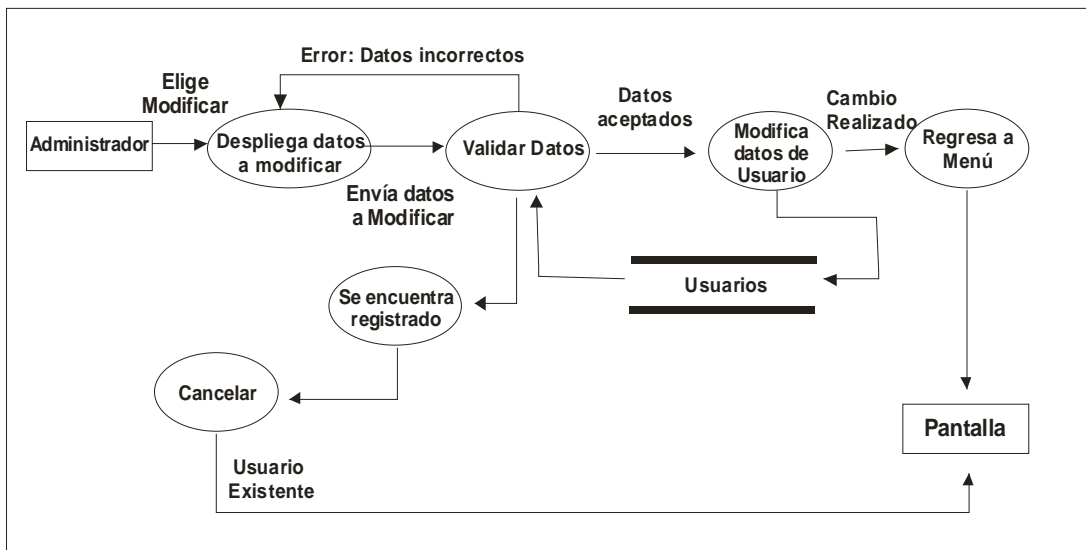


Figura 2.31. Nivel 8.2.- Modificar Usuarios

### Nivel 8.3.- Baja de Usuarios

La Figura 2.32 describe el proceso para dar de baja un usuario, en el cual, el administrador selecciona al usuario que desea eliminar, el cual será validado y buscado en el almacén de datos. Cuando el elemento haya sido encontrado, el administrador determinará la eliminación o la cancelación de la misma.

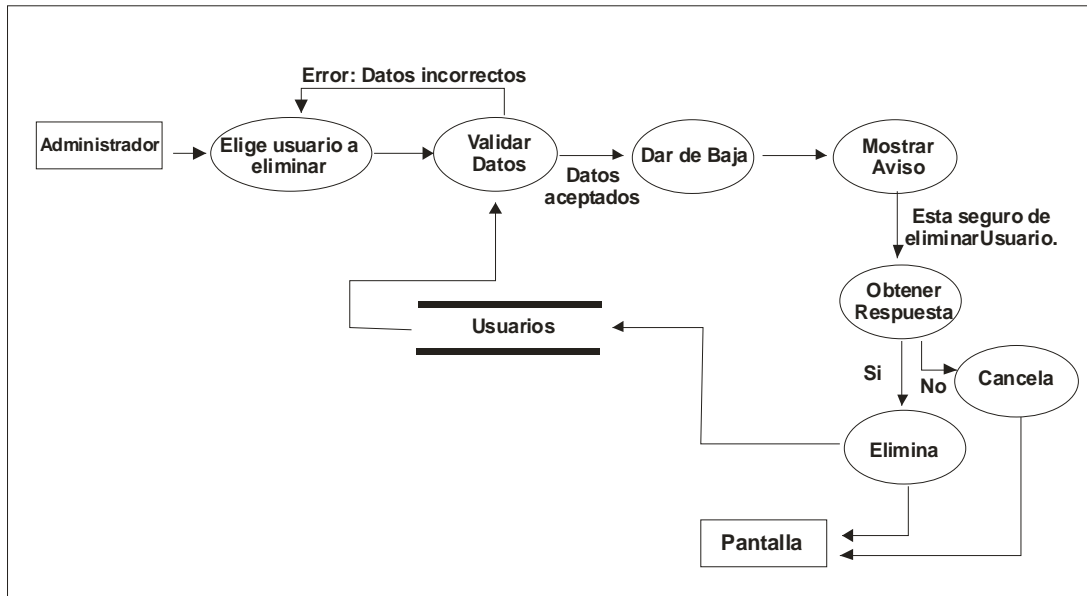


Figura 2.32. Nivel 8.3.- Baja de Usuarios

### Nivel 8.4.- Consulta de Usuarios

La Figura 2.33 muestra como el administrador puede realizar la consulta de un usuario, para ello, se selecciona el usuario a consultar y finalmente, se desplegarán los datos correspondientes.

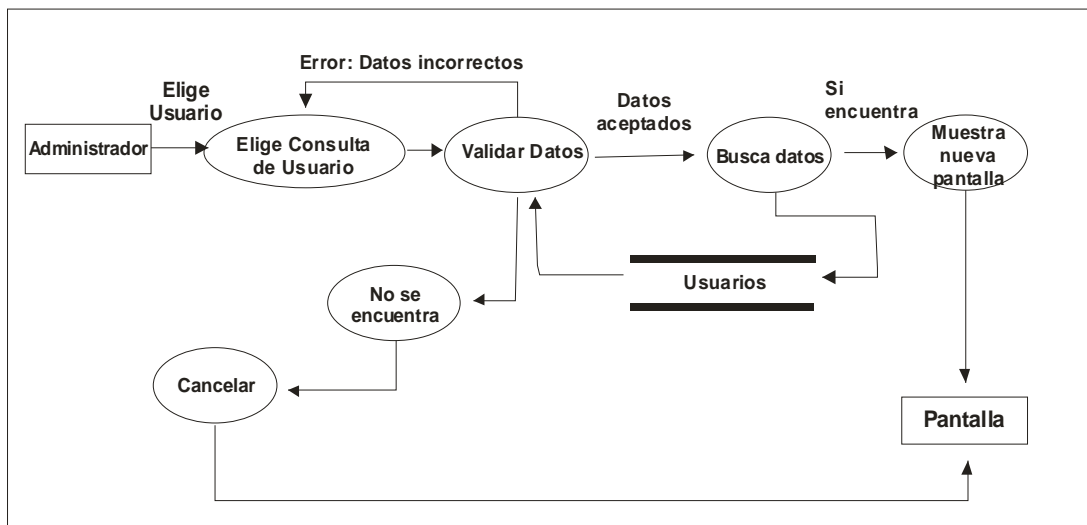


Figura 2.33. Nivel 8.4.- Consulta de Usuarios

## DISEÑO DEL SISTEMA

En éste capítulo, se realizó el diseño de la base de datos, a través de los diagramas de flujo, el modelo Entidad-Relación, el modelo relacional y la normalización. También se describen cada una de las entidades, atributos y relaciones en el diccionario de datos.

### 3.1 Modelo Conceptual

El primer paso en el diseño de una base de datos es la producción del esquema conceptual, normalmente, se construyen varios esquemas conceptuales, cada uno para representar las distintas visiones que los usuarios tienen de la información. Cada una de estas visiones suelen corresponder a las diferentes áreas funcionales de la empresa.

A los esquemas conceptuales correspondientes a cada vista de usuario se les denomina esquemas conceptuales locales. Cada uno de estos esquemas se compone de entidades, relaciones, atributos, dominios de atributos e identificadores. El esquema conceptual también tendrá una documentación, que se irá produciendo durante su desarrollo.

### 3.2 Modelo Entidad-Relación

Dentro del modelo Entidad-Relación se realizó la identificación de entidades, atributos y relaciones correspondientes.

#### Identificación de Entidades

Las entidades que se identificaron se muestran en la Tabla 3.1, así como sus atributos.

Entidad	Atributos
Zona	Id_zona, precio, ubicación
Avalúo	Id_avaluo, alumbrado,drenaje, pavimentacion, mts_totales, mts_construídos, num_plantas, avaluo
Inmueble	Id_inmueble, tipo_inmueble, direccion, monto_inicial,

	tipo_negocio
Cliente	Id_cliente, nombre, direccion, telefono, email, rfc
Usuario	Id_usuario, nombre, direccion, telefono, tipo_usuario, email, login, password
venta	Id_venta, monto_mensual, monto_total, num_pago, enganche, num_plazos, fecha_venta
Renta	Id_renta, monto_renta, fecha_ini, fecha_fin, deposito
compra	Id_compra, monto_total, fecha

Tabla 3.1 Identificación de Entidades

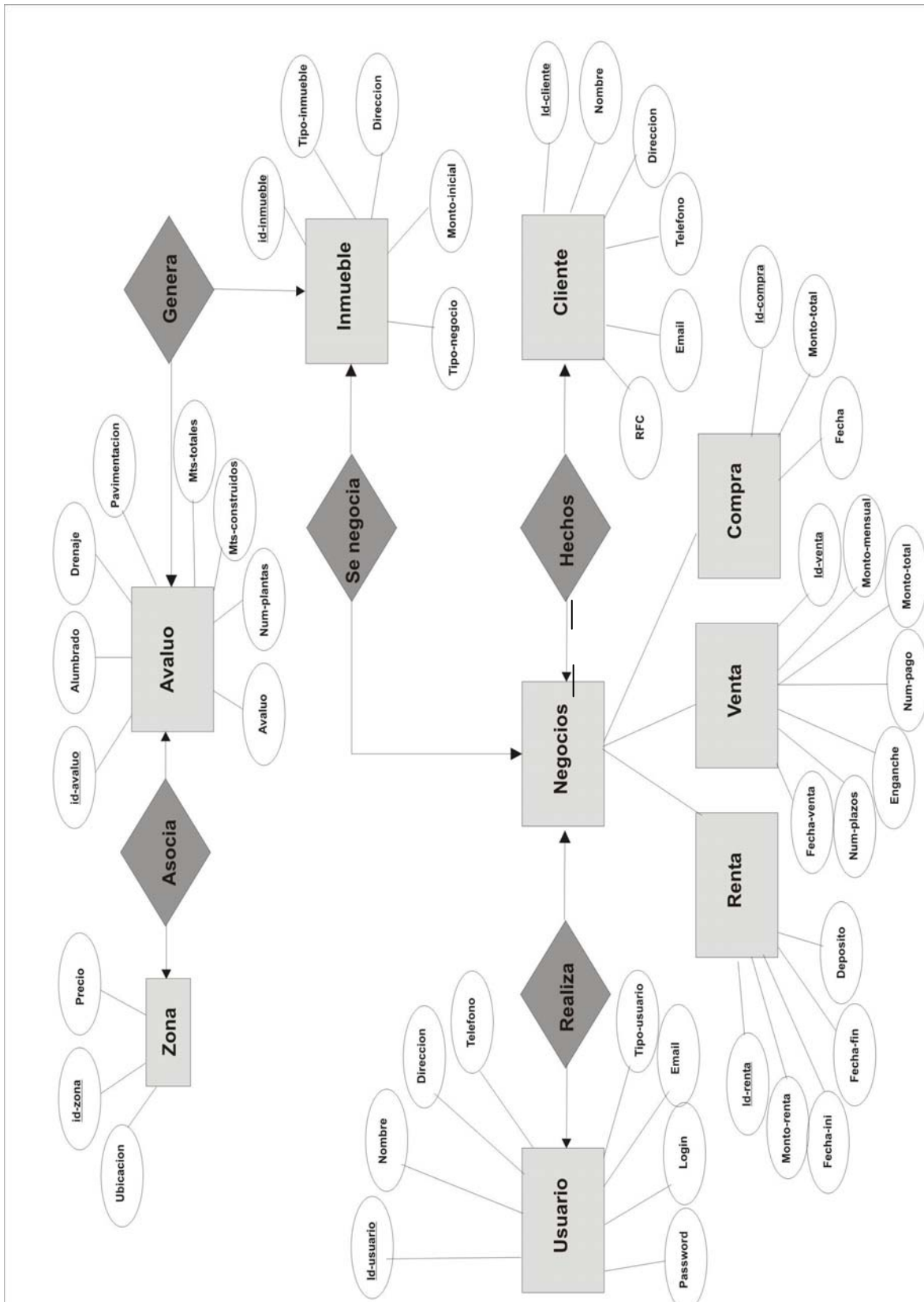
### Identificación de Relaciones

Las relaciones que se muestran en la Tabla 3.2 son las que permitieron diseñar el diagrama Entidad-Relación.

ENTIDAD	RELACION	CONECTIVIDAD	ENTIDAD
usuario	realiza_c	1 a M	Compra
usuario	realiza_v	1 a M	venta
usuario	realiza_r	1 a M	renta
cliente	Hace_c	1 a M	Compra
cliente	Hace_v	1 a M	venta
cliente	Hace_r	1 a M	Renta
compra	se_negocia_c	1 a M	Inmueble
venta	se_negocia_v	1 a M	inmueble
renta	se_negocia_r	1 a M	inmueble
inmueble	Genera	1 a 1	avaluo
avaluo	Asocia	1 a 1	zona

Tabla 3.2 Tabla de Relaciones

# Diagrama Entidad-Relación



### 3.3. Diccionario de datos

Los principales datos se explican en un documento llamado Diccionario de Datos, dentro del cuál se definen los elementos que conforman la base de datos de nuestro Sistema.

- **USUARIOS**

La Tabla 3.3 muestra la entidad usuarios que son aquellas personas que laboran en la inmobiliaria y en su momento pueden hacer modificaciones, consultas, o generar un reporte dependiendo del tipo de usuario que sea.

ATRIBUTOS	TIPO_ATRIBUTO	DESCRIPCIÓN	DOMINIO	TAMAÑO	RESTRICCIÓN
id_usuario	Simple, monovalorado	Clave de identificación del usuario	Varchar	4	No nulo
nombre	Simple, monovalorado	Es el nombre del usuario	Varchar	30	No nulo
dirección	Simple, monovalorado	Es la dirección del usuario	Varchar	30	No nulo
teléfono	Simple, multivalorado	Teléfono del usuario	Varchar	13	No nulo
tipo_usuario	Simple, monovalorado	Es la identificación del tipo de usuario	Int	4	No nulo
email	Simple, monovalorado	Es el e-mail del usuario	Varchar	30	Ninguna
login	Simple, monovalorado	Nombre con el que se reconoce el usuario	Varchar	30	No nulo
password	Simple, monovalorado	Clave para acceder al sistema	Varchar	30	No nulo

Tabla 3.3 Diccionario de datos *Usuarios*

- **CLIENTES**

La Tabla 3.4 muestra la entidad de los clientes, son las personas con las cuales se realiza el contrato de compra-venta-renta de un inmueble.

ATRIBUTOS	TIPO_ATRIBUTO	DESCRIPCIÓN	DOMINIO	TAMAÑO	RESTRICCIÓN
Id_cliente	Simple, monovalorado	Es la clave de identificación del cliente	Int	4	No nulo
nombre	Simple, monovalorado	Es el nombre del cliente	Varchar	30	No nulo
dirección	Simple, monovalorado	Es la dirección del cliente	Varchar	40	No nulo
teléfono	Simple, multivalorado	Es el teléfono que proporciona el cliente	Varchar	13	No nulo
email	Simple, multivalorado	Es el e-mail del cliente	Varchar	30	Ninguna
Rfc	Simple, monovalorado	Es el RFC del cliente	Varchar	14	No nulo

Tabla 3.4 Diccionario de datos *Clientes*

- **VENTA**

La Tabla 3.5 describe la entidad Venta, esta entidad me dice que inmueble esta en Venta.

ATRIBUTOS	TIPO_ATRIBUTO	DESCRIPCIÓN	DOMINIO	TAMAÑO	RESTRICCIÓN
id_venta	Simple, monovalorado	Es la clave de identificación de la venta	int	4	No nulo
monto_mensual	Simple, monovalorado	Monto que se pagará mensual	int	9	No nulo
monto_total	Simple, monovalorado	Es el monto en el que se vende el inmueble	int	9	No nulo
Num_pago	Simple, monovalorado	Es el número de pago al corriente	int	9	No nulo
enganche	Simple, monovalorado	Es la cantidad que se da al principio	Int	9	No nulo
num_plazos	Simple, monovalorado	Es el número de meses que se da	int	3	No nulo
fecha_venta	Simple, monovalorado	Es la fecha en que se hace la venta	date	10	No nulo

Tabla 3.5 Diccionario de datos *Venta*

- **RENTA**

La Tabla 3.6 muestra la entidad Renta, esta entidad me dice que inmueble esta en Renta.

ATRIBUTOS	TIPO_ATRIBUTO	DESCRIPCIÓN	DOMINIO	TAMAÑO	RESTRICCIÓN
id_renta	Simple, monovalorado	Es la clave de identificación de la renta	int	4	No nulo
monto_renta	Simple, monovalorado	Monto de renta del inmueble	int	9	No nulo
fecha_ini	Simple, monovalorado	Fecha en que inicia la renta	date	10	No nulo
fecha_fin	Simple, monovalorado	Fecha en que termina la renta	date	10	No nulo
deposito	Simple, monovalorado	Cantidad que se da de inicio	int	9	No nulo

Tabla 3.6 Diccionario de datos *Renta*

- **COMPRA**

La entidad Compra es mostrada por la Tabla 3.7, esta entidad me dice que inmueble esta en Compra.

ATRIBUTOS	TIPO_ATRIBUTO	DESCRIPCIÓN	DOMINIO	TAMAÑO	RESTRICCIÓN
id_compra	Simple, monovalorado	Es la clave de identificación de la compra	int	4	No nulo
monto_total	Simple, monovalorado	Monto en que se compra el inmueble	int	9	No nulo
Fecha	Simple, monovalorado	Es la fecha en que se realiza la compra	Date	10	No nulo

Tabla 3.7 Diccionario de datos *Compra*

- **AVALÚO**

La Tabla 3.8 describe la entidad Avalúo, en la que me dice el avalúo de un inmueble.

<b>ATRIBUTOS</b>	<b>TIPO_ATRIBUTO</b>	<b>DESCRIPCIÓN</b>	<b>DOMINIO</b>	<b>TAMAÑO</b>	<b>RESTRICCIÓN</b>
id_avaluo	Simple, monovalorado	Clave de identificación del avalúo	int	4	No nulo
alumbrado	Simple, monovalorado	Si el inmueble cuenta con alumbrado	int	4	No nulo
drenaje	Simple, monovalorado	Si el inmueble cuenta con drenaje	int	4	No nulo
pavimentación	Simple, monovalorado	Si el inmueble cuenta con pavimentación	int	4	No nulo
mts_totales	Simple, monovalorado	Total de metros cuadrados	int	9	No nulo
mts_construidos	Simple, monovalorado	Total de metros construídos	int	9	No nulo
num_plantas	Simple, monovalorado	Total de número de plantas extras	int	3	No nulo
avaluo	Simple, monovalorado	Total de avalúo del inmueble	int	9	No nulo

Tabla 3.8 Diccionario de datos *Avalúo*

- **INMUEBLE**

La Tabla 3.9 muestra la entidad inmueble, este puede ser terreno, casa o local que va a estar en Compra-Venta-Renta.

<b>ATRIBUTOS</b>	<b>TIPO_ATRIBUTO</b>	<b>DESCRIPCIÓN</b>	<b>DOMINIO</b>	<b>TAMAÑO</b>	<b>RESTRICCIÓN</b>
id_inmueble	Simple, monovalorado	Es la clave de identificación del inmueble	int	4	No nulo
tipo_inmueble	Simple, monovalorado	Es el tipo de inmueble que se está tratando	Varchar	30	No nulo
direccion	Simple, monovalorado	Es la dirección del inmueble	Varchar	30	No nulo
monto_inicial	Simple, monovalorado	Monto con el que se inicia el avalúo del inmueble	int	9	No nulo
tipo_negocio	Simple, monovalorado		Varchar	10	No nulo

Tabla 3.9 Diccionario de datos *Inmueble*

- **ZONA**

La Tabla 3.10 describe la entidad Zona, la cual nos ayuda a realizar el avalúo del inmueble.

ATRIBUTOS	TIPO_ATRIBUTO	DESCRIPCIÓN	DOMINIO	TAMAÑO	RESTRICCIÓN
id_zona	Simple, monovalorado	Clave de identificación de la zona	int	4	No nulo
precio	Simple, monovalorado	Es lo que vale el inmueble en esa zona	int	9	No nulo
ubicación	Simple, monovalorado	Donde se encuentra la zona	Varchar	30	No nulo

Tabla 3.10 Diccionario de datos *Zona*

### 3.4. Modelo Relacional

El modelo relacional representa la segunda generación de los SGBD. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica. Pero detrás de esa simple estructura hay un fundamento teórico importante del que carecen los SGBD de la primera generación, lo que constituye otro punto a su favor.

Dada la popularidad del modelo relacional, muchos sistemas de la primera generación se han modificado para proporcionar una interfaz de usuario relacional, con independencia del modelo lógico que soportan (de red o jerárquico).

En los últimos años, se han propuesto algunas extensiones al modelo relacional para capturar mejor el significado de los datos, para disponer de los conceptos de la orientación a objetos y para disponer de capacidad deductiva.

El modelo relacional se diseñó a partir del modelo Entidad-Relación. Se identificaron entidades, atributos y relaciones, que posteriormente se convirtieron en tablas en el modelo relacional las cuáles son relacionadas por medio de llaves foráneas.

#### **Esquema relacional:**

La Tabla 3.11 muestra el esquema de la base de datos para el Sistema de la inmobiliaria:

<b>Usuario:</b>							
<u>id_usuario</u>	nombre	dirección	teléfono	tipo_usuario	email	login	password
<b>Cliente:</b>							
<u>id_cliente</u>	nombre	dirección	teléfono	email	rfc		
<b>Venta:</b>							
<u>id_venta</u>	monto_mensual	monto_total	num_pago	enganche	num_plazos		
fecha_venta	<u>id_usuario</u>	<u>id_cliente</u>	<u>id_inmueble</u>				
<b>Renta:</b>							
<u>id_renta</u>	monto_renta	fecha_ini	fecha_fin	deposito	<u>id_usuario</u>	<u>id_cliente</u>	
<u>id_inmueble</u>							
<b>Compra:</b>							
<u>id_compra</u>	monto_total	fecha	<u>id_usuario</u>	<u>id_cliente</u>	<u>id_inmueble</u>		
<b>Avalúo:</b>							
<u>id_avaluo</u>	alumbrado	drenaje	pavimentación	mts_totales	mts_construidos		
Num_plantas	avaluo	<u>id_inmueble</u>	<u>id_zona</u>				
<b>Inmueble:</b>							
<u>id_inmueble</u>	tipo_inmueble	direccion	monto_inicial	tipo_negocio			
<b>Zona:</b>							
<u>id_zona</u>	precio	ubicación					

Tabla 3.11 Esquema relacional

### 3.5. Normalización

El proceso de normalización consiste en verificar el cumplimiento de ciertas reglas que aseguran la eliminación de redundancias e inconsistencias. Esto se hace mediante la aplicación de ciertos procedimientos y en ocasiones se traduce en la separación de los datos en diferentes relaciones.

Las tablas presentadas en el modelo relacional se encuentran en primera forma normal, ya que sus atributos son atómicos que significa "indivisible", es decir, cada atributo debe contener un único valor del dominio, además, cada atributo tiene un nombre único, no tienen valores del mismo dominio o de dominios diferentes y tampoco existen tuplas idénticas.

También las tablas del modelo relacional se encuentran en segunda forma normal, debido a que cada tabla tiene un atributo único como clave primaria.

No existe una dependencia transitiva, por lo que se puede verificar que se cumplen primera, segunda y tercera forma normal.

## IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

Una vez terminada la etapa de análisis y diseño del Sistema, se realizó la implementación de éste, a través de un sistema administrador de bases de datos, lenguajes de programación y herramientas del diseño para su funcionamiento.

En la etapa de la implementación se pusieron en funcionamiento los formularios diseñados anteriormente del menú principal (Ventas, Rentas, Compras, Clientes, Inmuebles y Usuarios) así como de las diferentes funciones de cada módulo (altas, bajas, consultas, modificaciones).

Se creó la interfaz principal, que sirve de medio de comunicación entre el usuario y la base de datos para tener acceso al Sistema. El Sistema se implementó sobre MySQL, PHP y HTML sobre Windows.

### 4.1 Herramientas para el desarrollo del Sistema

Para realizar la implementación del Sistema, las herramientas que se utilizaron fueron las siguientes:

#### ❖ **Manejador de Bases de Datos**

El administrador de bases de datos que se utilizó es **MySQL**, ya que es sencillo de usar y es muy rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Algunas de las características principales de MySQL son:

- 1) Es un gestor de base de datos
- 2) El código fuente se puede descargar y está accesible a cualquiera
- 3) Es un programa interactivo que permite conectarnos a un servidor, ejecutar consultas y ver los resultados.

## ❖ Lenguaje de Programación

El lenguaje de programación que se utilizó para desarrollar el Sistema es **PHP** ya que es un lenguaje de programación usado frecuentemente para la creación de sitios web, se pueden programar las páginas html y los códigos fuente, es usado para la creación de aplicaciones para servidores.

Debido al diseño de PHP, también es posible crear aplicaciones con una interfaz gráfica para el usuario, su interpretación y ejecución se da en el servidor web, en el cual se encuentra almacenado el script y el cliente sólo recibe el resultado de la ejecución.

Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas.

## ❖ Herramientas de Diseño

Para realizar el diseño la herramienta utilizada fue **Dreamweaver** ya que:

- Es un editor de HTML visual
- Hace muy fácil el crear páginas
- Genera HTML dinámico y el código resultante es compatible con las últimas versiones de los navegadores actuales.

Con la unión de todas estas herramientas, se pudo desarrollar la interfaz principal y los formularios del Sistema de Bases de Datos

## 4.2 Conexión de MySQL con PHP

Para realizar esta conexión se realizó la clase **conexión**, Clase que sirve para conectarse a una base de datos de mysql tiene un constructor con valores por defecto los cuales pueden sustituirse por otros valores para acceder a un host diferente, otra base de datos, otro usuario y otra contraseña.

- Código de la clase Conexión:

```
<?PHP
class conexion //Clase conexión
{
var $db_host; // Host al que conectar, habitualmente es el 'localhost'
var $db_nombre;// Nombre de la Base de Datos que se desea utilizar
var $db_user; // Nombre del usuario con permisos para acceder
var $db_pass; // Contraseña de dicho usuario
```

```

var $lnk; //Identificador de conexión a la base de datos

//Constructor de la clase conexión
function __construct($host="localhost", $db="inmobiliaria", $user="root", $pass="root")
{
    $this->db_host = $host;
    $this->db_nombre = $db;
    $this->db_user = $user;
    $this->db_pass = $pass;
}

//Función para conectarse a la base de datos
function conectar()
{
    // Ahora estamos realizando una conexión y la llamamos '$link'
    $this->lnk = mysql_connect($this->db_host, $this->db_user, $this->db_pass) or die
    ("Error conectando al servidor.");
    // Seleccionamos la base de datos que nos interesa
    mysql_select_db($this->db_nombre, $this->lnk) or die("Error seleccionando la base de
    datos.");
}

//Función para desconectarse de la base de datos
function desconectar()
{
    //Cerramos la conexión
    mysql_close($this->lnk) or die("Error al intentar desconectarse del servidor...");
}
?>

```

- Código para realizar Altas:

```

function nuevo($nombre,$direccion,$telefono,$email,$rfc)
{ //Consulta sql de inserción
    $sql = "INSERT INTO cliente (nombre,direccion,telefono,email,rfc) VALUES
    ('$nombre','$direccion','$telefono','$email','$rfc)";
    mysql_query($sql) or die("error al insertar...$sql");
}

```

- Código para realizar Bajas:

```
function elimina($id)
{
    $sql = "DELETE FROM cliente WHERE id_cliente=$id";
    mysql_query($sql);
}
```

- Código para realizar Consultas:

```
function consulta_id($ident)
{
    $sql = "SELECT * FROM cliente WHERE id_cliente=$ident";
    $res = mysql_query($sql);
    $v = mysql_fetch_array($res);
        //Almaceno en las variables del objeto los resultados de la consulta
    $this->id_cliente = $v['id_cliente'];
    $this->nombre = $v['nombre'];
    $this->direccion = $v['direccion'];
    $this->telefono = $v['telefono'];
    $this->email = $v['email'];
    $this->rfc = $v['rfc'];
}
```

- Código para realizar Modificaciones:

```
function modifica($nombre,$direccion,$telefono,$email,$rfc,$id)
{
    //Consulta sql de modificación
    // $sql = "UPDATE venta SET monto_mensual='$monto_mensual',monto_total=
    '$monto_total',num_pago='$num_pago',enganche='$enganche',num_plazos=
    '$num_plazos',fecha_venta='$fecha_venta',id_usuario='$id_usuario',id_cliente=
    '$id_cliente' WHERE id_venta=$id";
    $sql = "UPDATE cliente SET nombre='$nombre',direccion='$direccion',telefono=
    '$telefono',email='$email',rfc='$rfc' WHERE id_cliente=$id";
    mysql_query($sql) or die("Error al intentar modificar los datos...");
}
```

### 4.3 Interfaces del Sistema De Bases de Datos

El acceso al Sistema de Bases de Datos, comienza por la pantalla de autenticación, mostrada en la Figura 4.1, donde el usuario debe escribir su nombre y contraseña ya que se tienen dos tipos de usuario: el administrador y los empleados.

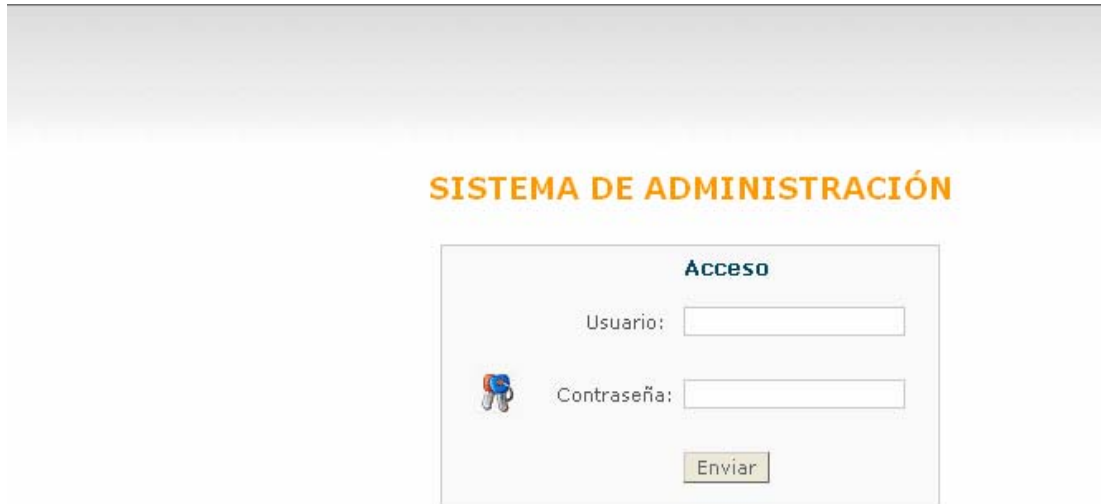


Figura 4.1 Pantalla de acceso al Sistema

Si el login y el password son incorrectos, aparecerá una pantalla que dirá usuario o contraseña incorrectos, como lo muestra la Figura 4.2

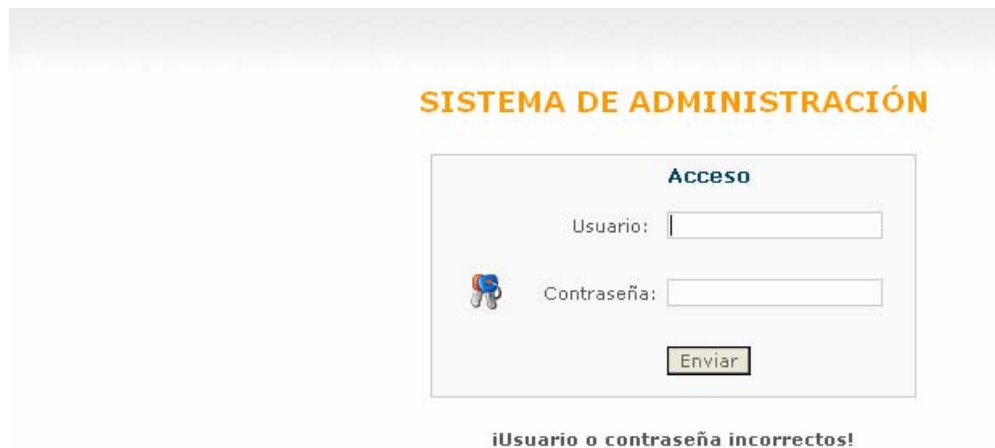


Figura 4.2 Pantalla de datos Incorrectos

En el caso de que los datos sean correctos y pertenezca al administrador, aparecerá una ventana como la que se muestra en la Figura 4.3, éste va a poder tener control sobre todo el Sistema, de lo contrario, el empleado sólo realiza algunas funciones como se muestra en la Figura 4.4.



Figura 4.3 Pantalla de Administrador



Figura 4.4 Pantalla de Empleado

La Figura 4.5 muestra la interfaz de Ventas en la que el administrador puede realizar Altas, Bajas, Consultas y modificaciones, en caso de ser empleado, solo puede realizar altas y consultar.

Fecha	Enganche	Monto Total	
2007-03-01	222	111	[Hand] [Trash] [Add]
2007-03-08	3323	333	[Hand] [Trash] [Add]
2007-03-01	111	111	[Hand] [Trash] [Add]
2007-04-05	99995	99995	[Hand] [Trash] [Add]

Figura 4.5 Interfaz de Ventas

La interfaz de Rentas es mostrada por la Figura 4.6 en la que el administrador puede utilizar todos los elementos y si es empleado solo puede consultar y dar de alta.

Monto de Renta	Fecha Inicial	Fecha Final	
23000	2007-03-01	2007-03-31	[Hand] [Trash] [Add]
99996	0000-00-00	0000-00-00	[Hand] [Trash] [Add]
3333	2007-04-03	2007-04-03	[Hand] [Trash] [Add]

Figura 4.6 Interfaz de Rentas

La Figura 4.7 muestra la interfaz de Compras, esta opción sólo aparece en la consola del administrador.



Figura 4.7 Interfaz de Compras

La Figura 4.8 muestra la interfaz de Clientes, esta opción aparece tanto en la consola del administrador como en la del empleado, pero con privilegios diferentes.



Figura 4.8 Interfaz de Clientes

La Figura 4.9 muestra la interfaz de Inmuebles, opción en la que sólo tiene acceso el administrador.

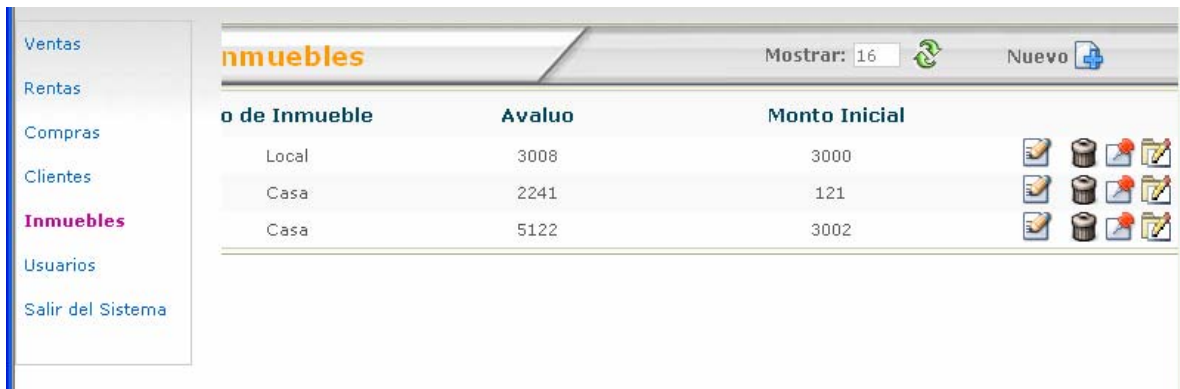


Figura 4.9 Interfaz de Inmuebles

La Figura 4.10 muestra la interfaz de Usuarios en la que sólo el administrador tiene uso de las funciones.



Figura 4.10 Interfaz de Usuarios

Cada uno de los módulos que aparecen en la interfaz principal, realiza las funciones de: altas, bajas, consultar y modificar. La Figura 4.11 muestra la interfaz para dar de alta cualquier elemento, en esta interfaz, se introducen los datos necesarios y posteriormente se le da clic en el botón de “Inserta venta”, de lo contrario, se da clic en el botón “Cancelar”.

**Ventas**

- Rentas
- Compras
- Clientes
- Inmuebles
- Usuarios
- Salir del Sistema

**Ventas**

**Datos:**

Monto Mensual:

Monto Total:

Numero de pago:

Enganche:

Número de plazos:

Fecha de Venta:

Usuario:

Cliente:

Tipo de Inmueble:

Figura 4.11 Pantalla de Alta

Para la opción Modificar aparece la interfaz como lo muestra la Figura 4.12, en esta pantalla aparecen los datos originales, entonces se debe modificar el dato necesario y pulsar el botón “Modificar venta”, de lo contrario, se pulsará el botón “Cancelar”.

**Ventas**

- Rentas
- Compras
- Clientes
- Inmuebles
- Usuarios
- Salir del Sistema

**Ventas**

**Datos:**

Monto Mensual:

Monto Total:

Numero de pago:

Enganche:

Número de plazos:

Fecha de Venta:

Usuario:

Cliente:

Tipo de Inmueble:

Figura 4.12 Pantalla de “Modificar”

Al seleccionar Bajas aparece la interfaz que muestra la Figura 4.13 en la que muestra el letrero para poder eliminar, en caso de que sí se quiera eliminar se da clic en “Aceptar” de lo contrario, se da clic en “Cancelar”

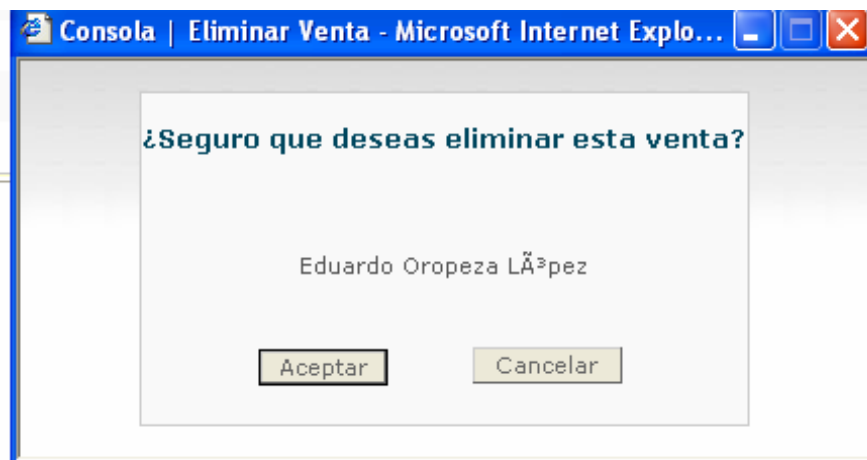


Figura 4.13 Pantalla de “Baja”

Cuando seleccionamos Consulta se despliega una ventana con todos los datos de la consulta requerida, así como lo muestra la Figura 4.14



Figura 4.14 Pantalla de “Consulta”

Para el caso del empleado, se obtienen las mismas pantallas, con la diferencia de que sólo trabajará con el menú: Ventas, Rentas y Clientes.

#### 4.4 Pruebas

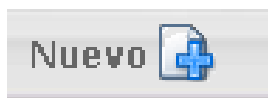
Dentro de la pantalla del administrador, se tiene el menú, que consta de: Ventas, Rentas, Compras, Clientes, Inmuebles, Usuarios, y Salir del Sistema, como lo muestra la Figura 4.15



Figura 4.15 Menú Principal

En el acceso a cualquiera de los submenús las funciones serán las mismas y se presentan a continuación con el icono que las define dentro del Sistema:

- Altas:



- Modificar:



- Bajas:



- Consultas



En el submenú de Ventas, accesamos al módulo de Altas, la Figura 4.16 muestra los campos que deben ser llenados posteriormente se le da clic en el botón de “Inserta venta”, de lo contrario, se da clic en el botón “Cancelar”.

**Ventas**

**Datos:**

Monto Mensual:

Monto Total:

Numero de pago:

Enganche:

Número de plazos:

Fecha de Venta:  

Usuario:  

Cliente:  


Tipo de Inmueble:  

Figura 4.16 Pantalla de Altas de Ventas

En la interfaz principal de ventas se pueden observar sólo algunos de los datos que fueron introducidos como lo muestra la Figura 4.17

**Ventas** Mostrar:  

Fecha	Enganche	Monto Total
2007-03-01	222	111
2007-03-08	3323	333
<b>2007-03-01</b>	<b>111</b>	<b>111</b>
2007-04-05	99995	99995

Figura 4.17 Pantalla principal de Ventas

Pero al seleccionar Consulta se despliega una ventana con todos los datos de la Venta, así como lo muestra la Figura 4.18, en la que se confirma que la venta fue dada de alta, ya que todos los campos contienen la información de los registros que se llenaron para darse de alta.



The screenshot shows a window titled 'Consulta' with a list of sale details. The details are as follows:

Monto Total:	111
Monto mensual:	111
Numero de Pago:	222
Enganche:	222
Número de plazos:	222
Fecha de Venta:	2007-03-01
Usuario:	
Cliente:	Eduardo Oropeza LÃ³pez
Tipo de Inmueble:	Local

Figura 4.18 Interfaz de Consulta de Ventas

La Figura 4.19 muestra la interfaz principal de Clientes, en la cual se observan los datos de los clientes, dentro de esta pantalla se marcó el teléfono en un recuadro para verificar que en realidad se modifique.



The screenshot shows the 'Clientes' main interface. At the top, there is a header with the title 'Clientes', a 'Mostrar: 16' dropdown, a refresh icon, and a 'Nuevo +' button. Below the header is a table with columns for 'Nombre', 'Dirección', and 'Teléfono'. The first row shows 'Eduardo Oropeza LÃ³pez' with address '47 Sur 2010' and phone number '2010203'. The second row shows 'ursula gomez flores' with address 'cu' and phone number '3256487'. To the right of each row are three icons: a pencil (edit), a trash can (delete), and a document (print).

Nombre	Dirección	Teléfono
Eduardo Oropeza LÃ³pez	47 Sur 2010	2010203
ursula gomez flores	cu	3256487

Figura 4.19 Interfaz Principal de Clientes

La Figura 4.20 muestra los campos a modificar, en esta pantalla aparecen los datos originales del cliente, entonces se debe modificar el dato necesario y pulsar el botón "Modificar venta", de lo contrario, se pulsará el botón "Cancelar".

The screenshot shows a web application interface with a header bar labeled 'Clientes'. Below the header, there is a section titled 'Datos:' containing several input fields for client information. The fields are: 'Nombre:' with the value 'Eduardo Oropeza LÃ³pez', 'Dirección:' with '47 Sur 2010', 'Teléfono:' with '2989796', 'e-mail:' with 'lalo@hotmail.com', and 'RFC:' with 'ORLE561203BPL'. At the bottom of the form, there are two buttons: 'Modificar Cliente' and 'Cancelar'.

Figura 4.20 Pantalla para Modificar datos de Cliente

La Figura 4.21 muestra la pantalla principal con el dato modificado

The screenshot shows the main 'Clientes' list. The header bar includes the title 'Clientes', a 'Mostrar: 16' dropdown, a refresh icon, and a 'Nuevo' button with a plus icon. The table below has three columns: 'Nombre', 'Dirección', and 'Teléfono'. The first row shows 'Eduardo Oropeza LÃ³pez', '47 Sur 2010', and '2989796'. The second row shows 'ursula gomez flores', 'cu', and '3256487'. Each row has a set of icons on the right for editing, deleting, and viewing details.

Nombre	Dirección	Teléfono
Eduardo Oropeza LÃ³pez	47 Sur 2010	2989796
ursula gomez flores	cu	3256487

Figura 4.21 Pantalla que muestra la Modificación de datos de Cliente

La Figura 4.22 muestra dos clientes de los cuales uno se eliminará.

Nombre	Dirección	Teléfono			
Eduardo Oropeza LÃ³pez	47 Sur 2010	2989796			
ursula gomez flores	cu	3256487			

Figura 4.22 Pantalla principal de Clientes

Al dar un clic en Bajas, aparece la pantalla que muestra la Figura 4.23 en la que muestra el letrero para poder eliminar al Cliente cuyo nombre aparecerá en la ventana, en caso de que sí se quiera eliminar se da clic en “Aceptar” de lo contrario, se da clic en “Cancelar”

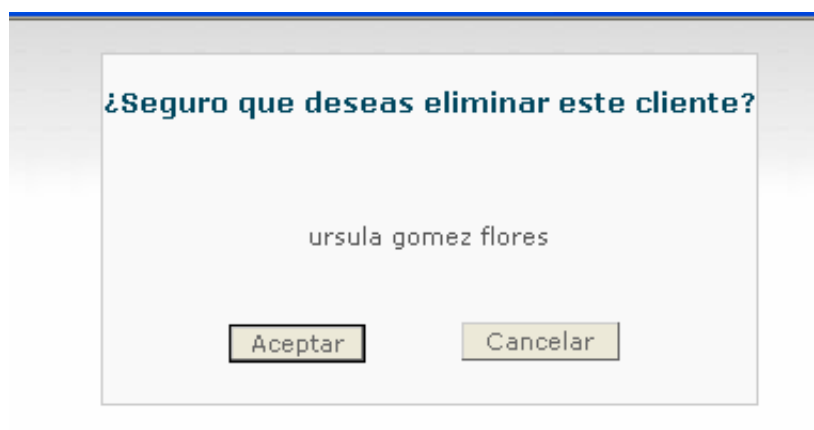


Figura 4.23 Pantalla de Baja de cliente

La Figura 4.24 muestra que efectivamente, el cliente fue eliminado del Sistema.

Nombre	Dirección	Teléfono			
Eduardo Oropeza LÃ³pez	47 Sur 2010	2989796			

Figura 4.24 Pantalla principal de Clientes

# **CONCLUSIONES**

---

Después de realizar pruebas a cerca del funcionamiento, el Sistema está listo para ser utilizado, ya que se apegó a los requerimientos planteados.

El Sistema cuenta con seguridad de acceso, ya que existen 2 tipos de usuario y de este modo se evita un mal uso del Sistema.

Se presenta como un Sistema abierto en código e Interfaces para su mejoramiento y ampliación.

El Sistema de Bases de Datos presenta una interfaz amigable, ya que cuenta con pantallas de fácil uso, además presenta un menú principal que contiene los módulos principales: Ventas, Rentas, Compras, Clientes, Inmuebles, Usuarios y Salir del Sistema, de los que se despliegan los submenús que se plantearon: altas, bajas, consultas y modificaciones. Con estos módulos el usuario podrá manipular la información de manera más fácil y sencilla.

El Sistema aporta la reducción de papel y de tiempo, ya que con el Sistema se pueden localizar fácilmente los datos que se requieran.

Durante el desarrollo del Sistema, reafirmé conocimientos a cerca de Bases de Datos y de Ingeniería de Software, así también adquirí conocimientos de MySQL, Apache y PHP.

El Sistema contribuirá al mejor manejo de la Inmobiliaria ya que las funciones cubren las necesidades de ésta.

# **PERSPECTIVAS**

---

- Sería conveniente realizar un catálogo de los inmuebles que se manejan.
- Sería adecuado implementar un módulo que realice reportes de las diferentes consultas.
- Implementación de la impresión de reportes.
- Sería conveniente considerar intereses por pagos atrasados o vencidos en el módulo de Ventas.
- Implementación del respaldo de la información que se maneja.

Estas son algunas perspectivas para la Inmobiliaria ya que a través del tiempo las necesidades de esta podrían ser otras.

# **BIBLIOGRAFÍA**

---

- [1] Somerville Ian (2002).: Ingeniería de Software, Editorial Addison Wesley, 6ª Edición, México.
- [2] Miguel, A. de y Piattini, M. (1997).: Fundamentos y Modelos de Bases de Datos. Ed. RAMA, Madrid.
- [3] Korth, H.F.; Siberschatz, A.; Sudarshan S. (1998).: Fundamentos de bases de datos. Madrid, McGraw-Hill.
- [4] Jesús Bobadilla S. Alonso (2000).: HTML Dinámico a través de ejemplos, Ed. Alfaomega-Rama.
- [5] Roger S. Pressman. (1998).: Ingeniería de Software:Un enfoque Práctico, Ed. Mc Graw-Hill, 4ª Edición.
- [6] Sitio oficial de MySQL, <http://www.mysql.com/>
- [7] Sitio oficial de PHP, <http://www.php.net/>
- [8] Sitio oficial de Apache, <http://httpd.apache.org/>
- [9] Bases de Datos, <http://www.monografias.com/trabajos12/basdat/>
- [10] Ingeniería de Software, [www.lania.mx/~jalba/mcc/Software.doc](http://www.lania.mx/~jalba/mcc/Software.doc)
- [11] Paul DuBois (1999).: MySQL, Ed. New Riders, 1ª edición.
- [12] <http://html.rincondelvago.com/el-ciclo-de-vida-del-software.html>
- [13] <http://delta.cs.cinvestav.mx/~pmejia/softeng/nuevo1.ppt>

[14] <http://www.infor.uva.es/~chernan/Ingenieria/Teoria/Tema2A.pdf>