



---

**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

**FACULTAD EN CIENCIAS DE LA COMPUTACIÓN**

**SISTEMA DE BASE DE DATOS PARA EL CONTROL  
DE INVENTARIOS DE DULCERIAS UTILIZANDO  
TRIGGERS (SISCOIND)**

**TESIS**

**QUE PARA OBTENER EL GRADO DE**

**LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN**

**PRESENTA:**

**GILBERTO MOLINA MARCELINO**

**ASESOR:**

**M.C. MARCO ANTONIO SORIANO ULLOA**

**PUEBLA, PUEBLA**

# INTRODUCCIÓN

Existen diversas herramientas para el desarrollo de base de datos que pueden ser aplicadas en diferentes sectores como: el comercial, el industrial, el científico, el químico y el automotriz. El control automatizado de información, es la tendencia de las bases de datos actuales ya que aporta grandes beneficios al hombre para realizar el procesamiento de información en forma más eficiente, segura y controlada. Debido a los grandes volúmenes de información que se generan en el sector consumo se requiere de nuevas herramientas que ayuden a facilitar la administración de tal información para lo cual se hace uso de triggers (disparadores) para dicho sector, específicamente en el **control de inventarios**.

Uno de los problemas al que nos enfrentamos actualmente, es cómo podemos aplicar las tendencias actuales de las bases de datos para facilitar la administración de la información específicamente para el control de inventarios, en puntos de ventas, originándose así, el siguiente proyecto: **Sistema de Base de Datos para el Control de Inventarios de Dulcerías utilizando Triggers (SISCOIND)**.

Existen tesis similares a ésta. En la Facultad de Ciencias de la Computación de la BUAP, una de ellas tiene por título: Control de Inventarios para una Farmacia, del autor Javier Conde Iztetzi. La diferencia con este proyecto es que, tal control de inventarios no hace uso de triggers.

Este proyecto se analizará, diseñará e implantará un sistema de Base de Datos que permita realizar el control de inventarios de dulcerías (SISCOIND). A diferencia de otros proyectos ya realizados, en éste se implantarán en la Base de Datos Triggers para monitorear eventos sobre la existencia de productos que se venden en las dulcerías, con la finalidad de verificar condiciones que establezcan acciones que el sistema realizará, para disparar pedidos de productos con un stock mínimo en existencia, de tal manera que facilitará los usuarios la administración de las dulcerías.

El sistema será implantado en Access 2003 y un lenguaje de 4ª. Generación como visual Basic 6 y será diseñado para dulcerías donde existe un gran volumen de información.

Específicamente, el proceso para el control de inventarios de Dulcerías se realiza en base a los siguientes aspectos:

Una dulcería clasifica sus productos en 4 tipos: Galletas, dulces, Chocolates y frituras.

- Cada producto tiene un nombre comercial, código de barras, cantidad, peso y/o piezas, marca, tipo, precio de proveedor, precio de venta y fecha de caducidad.
- La dulcería tiene 2 tipos de usuarios, el administrador y el vendedor. De los que se generan los siguientes datos: nombre completo, dirección, teléfono, e-mail, RFC, CURP.
- El empleado o vendedor, es el encargado de realizar las ventas y consultas de productos. El administrador es el encargado del control total de la dulcería; como la supervisión de mercancía en entradas y salidas, control de personal y el levantamiento de pedidos.
- Los encargados de surtir la mercancía son los proveedores, de quienes se generan los siguientes datos: Nombre, dirección, teléfono, persona de contacto, e-mail, RFC.
- El pedido de la mercancía se obtiene después de hacer un balance general de la Dulcería mediante el conteo, en base a las siguientes características, stock mínimo de cada producto, fecha próxima a caducar.
- La ubicación de productos de la dulcería se hace por orden alfabético, en base al nombre del producto.
- Al comprar el cliente productos, este puede solicitar nota ó factura de tal compra. En caso de solicitar factura deberá proporcionar los siguientes datos: Nombre completo, domicilio, RFC, con los cuales será registrado.
- Al final del día se realiza un inventario con el fin de verificar compras y ventas de mercancía, además de un resumen general de la venta total. Analizando con ello, el Producto con mayor demanda para la realización del pedido correspondiente.

El sistema de este proyecto agilizará enormemente el anterior proceso, debido a que varias tareas realizadas por el administrador serán hechas por el Sistema; por ejemplo, cuando se genere un stock mínimo o un producto éste próximo a caducar, el Sistema detectará tales situaciones y producirá los mensajes correspondientes.

**Objetivo General:** En este proyecto se analiza, diseña e implanta un Sistema de Control de Inventarios de Dulcerías utilizando Triggers, para lo cual se hace uso del Modelo de Cascada de Ingeniería de Software y de herramientas para la administración de Bases de Datos como : Access 2003 y visual Basic 6.

Los **Objetivos particulares** para alcanzar el objetivo general son los siguientes:

- Recopilación de requisitos.
  - Recopilación, análisis y clasificación de la información relacionada con la Ingeniería de Software y con la Teoría de Bases de Datos.
  - Realizar un análisis de requisitos para la especificación del sistema **SISCOIND**.
  - Realizar el diseño del sistema y la base de datos
  - Realizar la implantación del sistema y de la base de datos
  - Pruebas
  - Generar un documento que avale el desarrollo de éste proyecto de tesis.
- 
- **Para la Facultad:** El documento presentado, es material de apoyo para los alumnos de la Facultad de Ciencias de la Computación, con el fin de que puedan desarrollar más proyectos e implantarlos haciendo uso de las nuevas tendencias de Bases de Datos.
  - **Para la sociedad:** Tener sistemas de Bases Datos más funcionales donde la intervención del hombre sea mínima, realizando así tareas de manera eficiente y segura para llevar un mejor Control de Inventarios en los diferentes sectores, aumentando con ello las ganancias económicas.
  - **Para mi como pasante de la FCC-BUAP:** Debido a que los conocimientos adquiridos en la Lic. En Cs. De la Computación y el Diplomado de Bases de Datos de la FCC-BUAP son aplicados hoy en día en muchas diferentes áreas en donde se requiere el procesamiento de información, se justifica la elaboración de este proyecto, ya que en él se hace uso de las tecnologías de Bases de Datos y se aplican en el ámbito comercial, obteniendo así un control preciso, eficiente y seguro, en la elaboración de inventarios, con el fin de facilitar la administración de la información.

# CONTENIDO

**Capítulo 1:** Se encuentra dividido en dos partes; Ingeniería de Software y Teoría de Base de Datos. En la primera parte se muestra la definición de Ingeniería de Software, conceptos y tipos de modelos que existen para el Diseño de Sistemas, mencionando las ventajas y desventajas que tiene cada uno de éstos. En la segunda parte se definen el concepto de Base de Datos y conceptos relacionados, se explica el concepto de sistema manejador de base de datos (DBMS), Bases de Datos Relacionales, Normalización de BD. Definición de triggers y ejemplos.

**Capítulo 2:** Se realiza un Análisis del sistema para el Control de Inventario de Dulcerías (SISCOIND), Se muestra; la especificación de los requisitos del usuario, la descripción del sistema, la especificación del sistema. Se describe el diagrama Entidad- Relación, El diagrama funcional y el diccionario de datos.

**Capítulo 3:** Se muestra el diseño de la Base de datos de “SISCOIND” utilizando el modelo relacional, se realiza el diseño lógico, conversión de esquemas conceptuales a esquemas lógicos, Normalización de la BD, eliminación de asociaciones de muchos a muchos, se analiza el esquema lógico local con el usuario.

**Capítulo 4:** Implantación y Justificación del uso de herramientas para el desarrollo de Bases de Datos Relacionales, pantallas del sistema, la explicación de las funciones más importantes y conclusiones.

**Perspectivas:** Ejemplo de un enfoque orientado a objetos del sistema, se proponen más funciones y un enfoque distribuído.

# CAPÍTULO I

## MARCO TEÓRICO

1.1 Ingeniería de Software.....	1
1.2 Modelar.....	1
1.2.1 Definición de modelo de datos.....	1
1.2.2 Modelo de cascada.....	1
1.3 Críticas al Modelo de Cascada.....	3
1.4 Modelo de prototipos.....	4
1.4.1 Ventajas del paradigma de prototipos.....	6
1.4.2 Desventajas del paradigma de prototipos.....	6
1.5 Teoría de la base de datos.....	7
1.5.1 Definición de base de datos.....	7
1.5.2 Objetivos de los sistemas de base de datos.....	7
1.6 Sistema Manejador de Base de Datos. (DBMS).....	8
1.7 Conceptos fundamentales.....	8
1.8 Bases de Datos Relacionales.....	9
1.8.1 El modelo entidad -relación.....	10
1.9 Formas normales.....	11
1.9.1 Primera forma normal.....	11
1.9.2 Segunda forma normal.....	13
1.9.3 Tercera forma normal.....	14
1.10 Definición de triggers.....	16
1.10.1 Componentes principales.....	16
1.10.2 Tipos de triggers.....	16
1.10.3 Efectos y características.....	16
1.10.4 Ejemplo de triggers.....	17

# CAPÍTULO II

## ANÁLISIS DEL SISTEMA

2.1 Especificación de Requisitos del Usuario.....	18
2.1.1 Encuesta de los Usuarios del Sistema.....	19
2.2 Análisis del Sistema.....	20
2.3 Descripción del Sistema.....	20
2.4 Especificación del Sistema.....	21
2.4.1 Funciones Básicas y Secundarios del SISCOIND.....	22
2.4.2 Diagrama Conceptual del Sistema.....	23
2.4.3 Diagrama Funcional.....	24
2.5 Diseño Conceptual.....	26
2.5.1 Identificación de Entidades.....	26
2.5.2. Identificación de Relaciones.....	26
2.5.3.- Localización de los Identificadores.....	26
2.6 Diccionario de Datos.....	28

# CAPÍTULO III

## DISEÑO DEL SISCOIND

3.1 Revisión de requerimientos.....	30
3.2 Modelo Relacional “Justificación para el SISCOIND”.....	30
3.3 Diseño lógico.....	30
3.3.1 Convertir Esquemas Conceptuales en Esquemas Lógicos.....	30
3.4 Eliminación de Asociaciones Muchos a Muchos.....	34
3.5 Validación de Esquemas Lógicos locales mediante la Normalización de las Entidades.....	37
3.5.1 Primera Forma Normal.....	37
3.5.2 Segunda Forma Normal.....	38
3.5.3 Tercera Forma Normal.....	41
3.6 Restricciones de Integridad de las Entidades.....	41
3.7 Análisis de Esquema Lógico Local con el Usuario.....	41

# CAPÍTULO IV

## IMPLANTACIÓN Y PRUEBAS

4.1 Introducción.....	43
4.2 Implementación del Sistema.....	43
4.2.1 Inicio del Sistema.....	43
4.2.2 Menú Principal del SISCOIND.....	44
4.3 Arquitectura del SISCOIND.....	45
4.3.1 Nivel Interno o Físico.....	46
4.3.2 Nivel Conceptual.....	47
4.3.3 Nivel Externo.....	47
4.4 Herramientas Utilizadas en la Aplicación.....	48
4.4.1 Análisis de la Herramienta Visual Basic 6.0.....	48
4.4.2 Crear un Origen de Datos ODBC.....	50
4.5 Análisis de Microsoft Access 2003.....	54
4.6 Lenguaje de Consulta del SISCOIND.....	55
4.7 Conclusiones.....	56
4.8 Perspectivas.....	57
4.9 Bibliografía.....	58

# ANEXO A

Manual de usuario.....	X
------------------------	---

# CAPÍTULO I

## MARCO TEORICO

### 1.1 Ingeniería de Software.

La Ingeniería de Software es una disciplina de la Informática y de las Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelve problemas de diferentes tipos.

### 1.2 Modelar

Consiste en crear un mundo teórico y abstracto tal, que las conclusiones que se puedan sacar de él coincidan con las manifestaciones aparentes de la realidad.

#### 1.2.1 Definición de modelo de datos.

Conjunto de conceptos, reglas y convenciones que nos permiten describir los datos del universo del discurso, constituyendo una herramienta que facilita la interpretación de nuestro universo del discurso y su representación en forma de datos en nuestro sistema de información.

Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software, así como intenta determinar el orden de las etapas involucradas y los criterios de transición asociados entre estas etapas.

#### 1.2.2 Modelo de cascada

El Modelo Cascada Es el más básico de todos los modelos de ciclo de vida y sirve de base para otros modelos, ve el desarrollo como una secuencia simple de etapas (figura 1.1) en donde cada etapa tiene un conjunto de metas bien definidas y las actividades entre cualquier etapa contribuyen a satisfacer las metas de las etapas subsiguientes. Las flechas muestran el flujo de información entre las etapas. La flecha de avance muestra el flujo normal. Las flechas hacia atrás representan la retroalimentación.

### **Principios básicos del Modelo de Cascada.**

- Planear un proyecto antes de embarcarse en él.

- Definir el comportamiento externo deseado del sistema antes de diseñar su arquitectura.
- Documentar los resultados de cada actividad.
- Diseñar un sistema antes de codificarlo.
- Probar el sistema después de construirlo.

### Fases del Modelo de Cascada

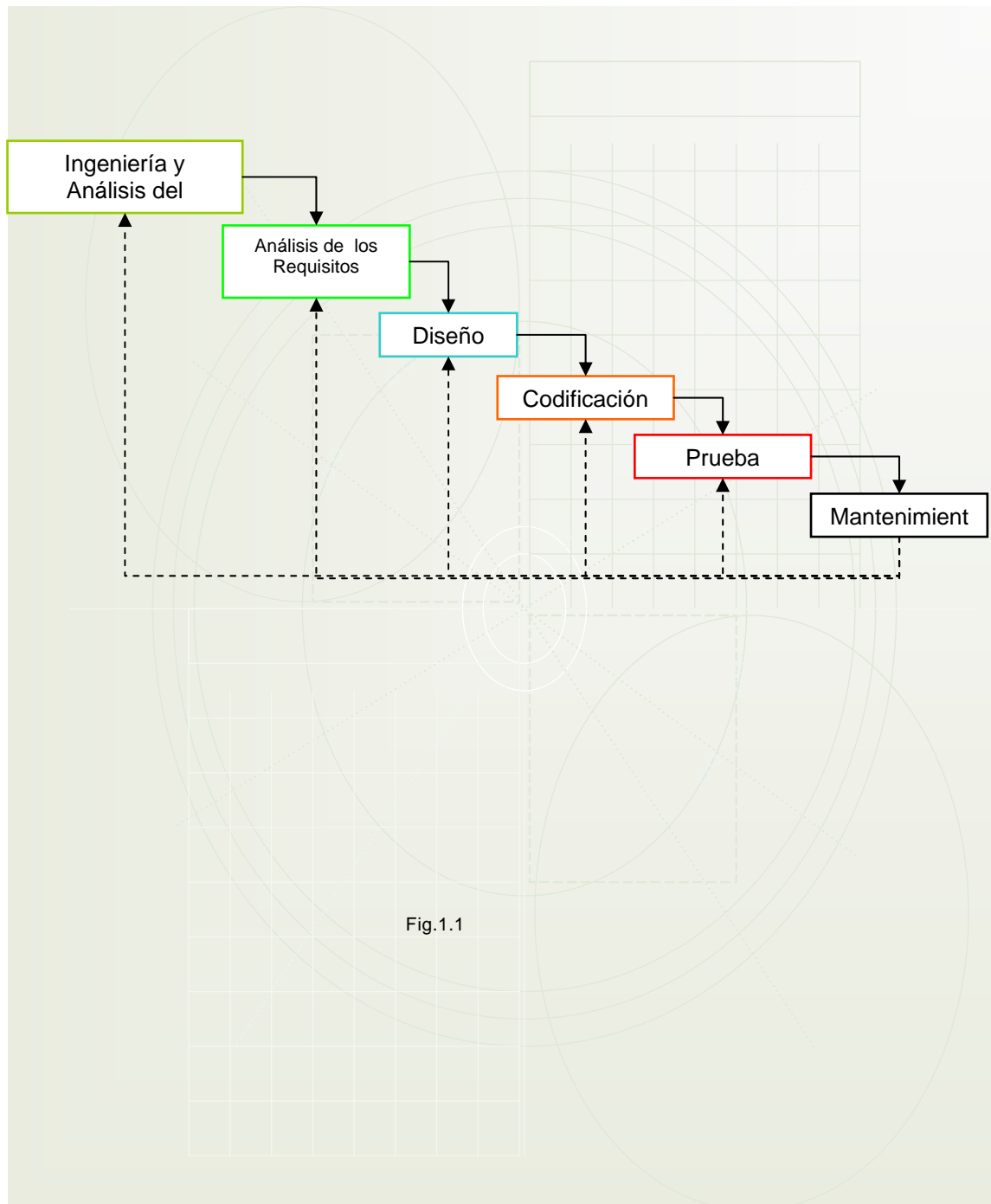


Fig.1.1

## **Planificación del Sistema**

Etapa en la cual se determina la factibilidad del sistema, así como costos aproximados y tiempo en el desarrollo del mismo, estableciendo así la ruta crítica de cada actividad.

En esta etapa es muy importante aclarar todos los objetivos, necesidades, problemas que requiere el sistema para evitar un sistema deficiente

## **Análisis**

Etapa en la que se establecen perfectamente los requisitos del software que el cliente requiere, determinando los límites y alcances del proyecto. En esta etapa es muy importante comprender completamente los requisitos del software para desarrollar un sistema funcional.

## **Diseño**

Esta etapa se enfoca a la estructura de los datos, arquitectura del software, la interfaz para poder tener un producto esperado. En este proceso se debe interpretar los requisitos tal como se requieren, para que el sistema obtenga una calidad esperada para el cliente, ya que es una documentación del desarrollo del sistema.

## **Codificación**

Proceso mediante el cual se hace la interpretación en código fuente del diseño del sistema de base de datos.

## **Pruebas**

Se analiza tanto la lógica interna del software así como la verificación de una entrada de datos definida que corresponda a un resultado esperado.

## **Mantenimiento**

Corresponde a la corrección de errores no detectados en la modificación del software.

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

### **1.3. Críticas al Modelo Cascada:**

- Gran énfasis en la producción de documentos completamente elaborados, producto de las fases de análisis y especificación de requerimientos y de diseño.
- No muy aplicable a productos de software altamente interactivos.

- Es difícil tener todos los requerimientos, bien definidos al principio, como lo requiere el modelo y además presenta dificultades para acomodar posibles incertidumbres existentes al comienzo de los proyectos.
- Un error importante no detectado al principio puede ser desastroso.
- Se requiere mucha paciencia por parte del cliente, porque solo hasta las etapas finales del desarrollo podrá tener una versión operativa del producto.
- Se tarda mucho tiempo en pasar por todo el ciclo
- Perpetúa el fracaso de la industria del software en su comunicación con el usuario final.
- El mantenimiento se realiza en el código fuente
- Las revisiones de proyectos de gran complejidad son muy difíciles.
- Impone una estructura de gestión de proyectos

## 1.4 Modelo de prototipos

La construcción de prototipos es un proceso que facilita al programador la creación de un modelo del software.

### **El modelo tomará una de las siguientes formas:**

- 1.-Un prototipo en papel o un modelo basado en computadora que describa la interacción humano-computadora, esto facilitará al usuario la comprensión de cómo se producirá tal interacción.
- 2.-Un prototipo que implementa algunos subconjuntos de la función requerida del programa deseado.
- 3.-Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.
- 4.-Un escenario (simulación del uso del sistema)
- 5.-Una demostración (porciones de código que realizan algunas funciones)
- 6.-Una versión 0 (aplicación liberada que puede usarse bajo condiciones preliminares añadiendo, cambiando o quitando funciones existentes y creándole su documentación)

La siguiente figura 1.2 muestra la secuencia de sucesos del paradigma de construcción de prototipos. La construcción de prototipos inicia con la recolección de los requerimientos. El desarrollador y el cliente se reúnen y definen los objetivos globales para el software, identifican todos los requerimientos conocidos y perfilan las áreas donde será necesario una mayor definición. Luego se produce un “diseño rápido”. El diseño rápido se enfoca sobre la representación de los aspectos del software visibles al usuario. Este diseño conduce a la construcción de un prototipo. Este es evaluado por el cliente/usuario y se utiliza para refinar los requerimientos del software a desarrollar. Se produce un proceso interactivo en el que el prototipo es “afinado” para que satisfaga las necesidades del cliente, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que hay que hacer.

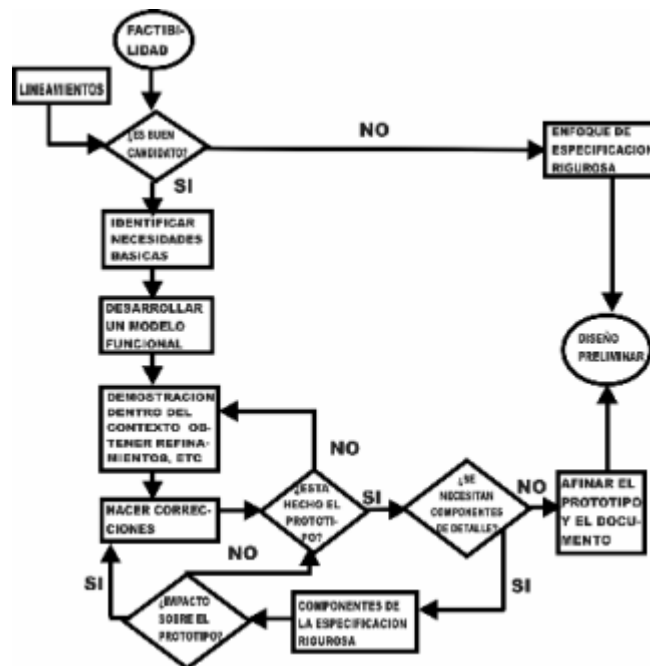


Fig.1.2

El prototipo puede servir como “primer sistema”, al igual que el modelo en cascada, puede ser problemático por las siguientes razones:

1. El cliente ve funcionando lo que parece ser una primera versión del software, no se han considerado los aspectos de calidad o de mantenimiento del software. Se ha construido un sistema con “plastilina y alambres”
2. El desarrollador impone ciertos compromisos de implementación con el fin de obtener un prototipo que funcione rápidamente.

Es un paradigma efectivo para la Ingeniería del Software. La clave está en definir al inicio las reglas del juego; esto es, cliente y desarrollador deben estar de acuerdo en que el prototipo se construya para servir sólo como un mecanismo de definición de los

requerimientos. Posteriormente, se descartará (al menos en parte) y debe construirse el software real, con los ojos bien puestos en la calidad y en el mantenimiento.

Dentro del enfoque de prototipos se pretende que el modelo sea operante, es decir, una colección de programas de computadora que simulan algunas o todas las funciones que el usuario desea. Para lograr lo anterior se utilizan las siguientes herramientas de software:

- Un diccionario de datos integrado
- Un generador de pantallas
- Un generador de reportes no guiado por procedimientos
- Un lenguaje de programación de cuarta generación
- Un lenguaje de consultas no guiado por procedimientos
- Medios poderosos de administración de base de datos

El modelo de prototipos comienza con una actividad de sondeo; de esto sigue una determinación de sí el proyecto es un buen candidato para un enfoque de prototipos. Los buenos candidatos son aquellos que tienen las siguientes características:

- El usuario no puede o no está dispuesto a examinar modelos abstractos en papel, tales como diagramas de flujo de datos.
- El usuario no puede o no está dispuesto a articular sus requerimientos de ninguna forma y sólo se pueden determinar sus requerimientos mediante un proceso de tanteo, o ensayo y error.
- Se tiene la intención de que el sistema sea en línea y con operación total por la pantalla, en contraposición con los sistemas de edición, actualización y reportes operados por lote.
- El sistema no requiere la especificación de grandes cantidades de detalles algorítmicos, ni de muchas especificaciones de procesos para describir los algoritmos con los cuales se obtienen los siguientes resultados.

#### **1.4.1 Ventajas del paradigma de prototipos**

- Son reales y tangibles.
- Permite al cliente aclarar lo que quiere que haga el sistema.
- Siente que es oído y tenido en cuenta para el diseño.  
Asegura que el trabajo se está haciendo bien y cumpliendo los requerimientos del cliente.

#### **1.4.2 Desventajas del paradigma de prototipos**

- El cliente puede creer que el sistema ya está listo y pedir su entrega rápida.
- Crea expectativas más allá de lo que realmente puede hacer.

- Se dificulta la dirección y control del proceso de desarrollo más que en el método clásico.
- La presión por entregar rápido el producto compromete la calidad.
- Se dificulta mantener el entusiasmo del cliente después de aprobado el prototipo porque creará que se desperdicia el tiempo en detalles insignificantes.

## **1.5 Teoría de Bases de Datos**

La evolución de las bases de datos se ha dado desde sistemas de archivos rudimentarios hasta grandes y complejos sistemas estructurados que ofrecen servicios remotos.

### **1.5.1. Definición de Base de Datos.**

Es un conjunto de información relacionada que se encuentra agrupada o estructurada. Un archivo por sí mismo no constituye una base de datos, sino más bien la forma en que está organizada, la información es la que da origen a la base de datos.

Colección de datos organizada para dar servicio a muchas aplicaciones al mismo tiempo al combinar los datos de manera que parezcan estar en una sola ubicación.

### **1.5.2. Objetivos de los Sistemas de Bases de Datos.**

Los objetivos principales de un sistema de base de datos es disminuir los siguientes aspectos:

#### **Redundancia e inconsistencia de datos.**

Esto es debido a que la información almacenada es creada por diferentes programas de aplicación, puede dar motivo a la duplicación de la información, lo cual aumenta los costos de almacenamiento y acceso a los datos, además de que puede originar la inconsistencia de los datos es decir, diversas copias de un mismo dato que no concuerdan entre si.

#### **Independencia lógica.**

Al añadir, modificar o eliminar objetos en la base de datos no debe de repercutir en los demás usuarios que estén accediendo a subconjuntos parciales de estos.

#### **Dificultad para tener acceso a los datos.**

Un sistema de base de datos debe contemplar un entorno de datos que le facilite al usuario el manejo de los mismos, para brindar una fácil consulta de los datos almacenados.

#### **Aislamiento de los datos.**

Puesto que los datos están repartidos en varios archivos, y estos no pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicación para obtener los datos apropiados.

## 1.6 Sistema Manejador de Base de Datos. (DBMS)

Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica.

El objetivo primordial de un sistema manejador de base de datos es proporcionar un contorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfase entre los usuarios y la base de datos. Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar, así como el personal encargado del manejo del sistema.

## 1.7 Conceptos fundamentales

### Entidad.

Una entidad es un objeto concreto o abstracto el cual modela eventos del mundo real que presenta interés para el sistema y sobre el que se recoge información para ser representada en un sistema de base de datos.

### Atributo.

Es una unidad básica e indivisible de información acerca de una entidad o una relación la cual sirve para identificar y describir a las mismas.

### Dominio

Es el conjunto de valores que puede tomar cada uno de los atributos.

### Tabla.

Organización de los datos en forma de filas y columnas. Cada fila se llama tupla, y cada columna dentro de una tupla corresponde al valor de un atributo para esa tupla.

### Relación

Asociación entre entidades. Por ejemplo, un "alumno" "tiene" una "asignatura".

**Tabla relacional:** Debe cumplir las siguientes características:

- Cada fila debe ser única
- Cada columna debe ser única
- Los valores de las columnas deben pertenecer al dominio de cada atributo
- Debe tener un solo tipo de fila, cuyo formato está definido por el esquema de la tabla o relación
- El valor de la columna para cada fila debe ser único

**Clave candidata.**

Atributo o atributos que pueden distinguir de forma unívoca una tupla dentro de una tabla puede haber varias claves candidatas para distinguir una misma entidad. Se elegirá como clave candidata aquel atributo que posea un dominio en el que se tenga valores únicos. Si esto no es posible, entonces usaremos como clave candidata la combinación de varios atributos, de manera que esta combinación sí sea única.

**Clave principal.**

Aquella de las claves candidatas que es designada para distinguir de forma unívoca una tupla dentro de una tabla.

**Clave ajena.**

Se trata de un atributo que es clave principal en otra tabla.

**Vista.**

Una vista es una tabla ficticia cuya definición y tuplas se obtiene a partir de una o más tablas base. Y como características tenemos que:

- Sus columnas se obtienen a partir de varias tablas base.
- Pueden estar definidas a partir de otras vistas.
- Sus datos se obtienen como resultado de una consulta a la base de datos.
- Se puede almacenar su estructura.

Así pues, vemos que se trata de una tabla virtual que no existe como tabla en el disco.

**Inconsistencia.**

Se da cuando se encuentra un valor en una clave ajena no existente en la entidad donde ésta sea clave principal.

## 1.8 Bases de Datos Relacionales

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

- ofrecen sistemas simples y eficaces para representar y manipular los datos

- se basan en un modelo, el relacional, con sólidas bases teóricas

### 1.8.1 El modelo entidad-relación

Es una técnica de diseño de bases de datos gráfica, que incorpora información relativa a los datos y la relación existente entre ellos, para poder así plasmar una visión del mundo real sobre un soporte informático. Sus características fundamentales son:

- Reflejan tan sólo la existencia de los datos sin expresar lo que se hace con ellos.
- Es independiente de las bases de datos y de los sistemas operativos
- Incluye todos los datos que se estudian sin tener en cuenta las aplicaciones que se van a tratar

Las entidades se representan como rectángulos, los atributos como elipses y las relaciones como rombos. El significado de estas palabras así como de otras también importantes las vemos a continuación.

#### Asociaciones entre entidades

Además de los atributos de cada entidad, un modelo de datos debe especificar las asociaciones existentes entre las entidades. Estas asociaciones son las relaciones entre entidades. Por ejemplo, la frase "los clientes compran productos" nos dice que hay dos entidades, "Clientes" y "Productos", que están relacionadas por "comprar".

La gran mayoría de las asociaciones son binarias, como "los clientes compran productos" o "los empleados venden productos". Entre las dos hay una asociación ternaria implícita: "los empleados venden productos a los clientes". Con las dos asociaciones binarias independientemente no podríamos saber a qué clientes se han vendido los productos que ha vendido un cierto empleado: en este caso necesitamos de la asociación ternaria.

Las asociaciones entre dos entidades cualesquiera pueden ser de tres tipos: uno-a-uno, uno-a-muchos y muchos-a-muchos.

#### Asociaciones uno-a-uno.

Si es cierto que cualquier ejemplar de la entidad X se puede asociar con tan sólo un ejemplar de la entidad Y, entonces decimos que la asociación es uno-a-uno. Cuando elegimos una asociación uno-a-uno debemos asegurarnos de que o bien se mantiene la asociación en todo momento, o en caso de que cambie no nos interesan los valores pasados. Por ejemplo: si asumimos que en los despachos de un edificio hay uno por persona, entonces la asociación será uno-a-uno. Pero esta asociación sólo es cierta en un momento dado. A lo largo del tiempo, se irán asignando diferentes empleados en el edificio. Habrá que valorar si el mantenimiento de esta información es útil en nuestro modelo o no.

#### Asociaciones uno-a-muchos.

Es el tipo de asociación más común, donde un solo ejemplar de una entidad se puede asociar con cero, uno o muchos ejemplares de otra entidad. Por ejemplo, una persona puede tener varios números de teléfono.

### Asociaciones muchos-a-muchos.

Los clientes compran en muchas tiendas, una tienda tiene muchos clientes. Como este tipo de relaciones no se puede modelar directamente en una base de datos relacional, se modela usando una tabla intermedia que tenga una asociación uno-a-muchos con cada uno de los participantes originales. Por ejemplo, un pedido puede tener muchos tipos de confección, y un tipo de confección puede aparecer en varios pedidos.

## 1.9 Formas normales

### 1.9.1 Primera forma normal.

Una relación R se encuentra en 1FN si y solo si por cada renglón columna contiene valores atómicos. Abreviada como 1FN, se considera que una relación se encuentra en la primera forma normal cuando cumple lo siguiente:

1. Las celdas de las tablas poseen valores simples y no se permiten grupos ni arreglos repetidos como valores, es decir, contienen un solo valor por cada celda.
2. Todos los ingresos en cualquier columna (atributo) deben ser del mismo tipo.
3. Cada columna debe tener un nombre único, el orden de las columnas en la tabla no es importante.
4. Dos filas o renglones de una misma tabla no deben ser idénticas, aunque el orden de las filas no es importante.

Por lo general la mayoría de las relaciones cumplen con estas características, así que podemos decir que la mayoría de las relaciones se encuentran en la primera forma normal. Para ejemplificar como se representan gráficamente las relaciones en primera forma normal consideremos la relación alumno cursa materia cuyo diagrama E-R es el siguiente:

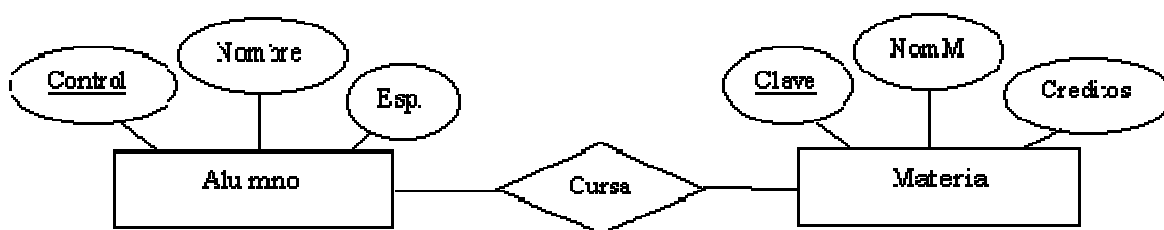


Fig. 1.3

Como esta relación maneja valores atómicos, es decir un solo valor por cada uno de los campos que conforman a los atributos de las entidades, ya se encuentra en primera forma normal, gráficamente así representamos a las relaciones en 1FN.

### **Concepto de dependencia funcional**

Se dice que un atributo B depende funcionalmente de A ( $A \rightarrow B$ ) si cada valor de A se corresponde con un único valor de B o, visto de otra manera, si dado A puedo obtener B. Un caso típico podría ser DNI  $\rightarrow$  Nombre, pues dado un DNI puedo obtener el nombre de la persona con ese DNI.

Vamos a ver unas reglas que se pueden realizar entre atributos para poder obtener dependencias funcionales adicionalmente. Vamos a suponer que T es una tabla relacional y X, Y, Z son subconjuntos de atributos de la tabla T.

#### **Reflexibilidad.**

Si los valores de un subconjunto de atributos Y están incluidos dentro de un subconjunto de atributos X, se dice que X depende funcionalmente de Y ( $Y \rightarrow X$ )

#### **Aumentación.**

Si un subconjunto X depende funcionalmente de otro Y, dicha dependencia se mantendrá aunque se añada otro atributo a los dos subconjuntos ( $Y \rightarrow X$  entonces  $Y.a \rightarrow X.a$ )

#### **Transitividad**

Si Y depende funcionalmente de X y Z depende funcionalmente de Y, entonces Z depende funcionalmente de X ( $X \rightarrow Y$  e  $Y \rightarrow Z$  entonces  $X \rightarrow Z$ ). Por ejemplo, DNI  $\rightarrow$  NOMBRE y NOMBRE  $\rightarrow$  DIRECCIÓN, luego DNI  $\rightarrow$  DIRECCIÓN

#### **Dependencia funcional total.**

Un atributo Y tiene una dependencia funcional total con otro atributo X si tiene una dependencia funcional con X y no depende funcionalmente de ningún subconjunto de X. Por ejemplo, supongamos que una empresa tiene empleados y que una persona puede ser empleado de varias empresas. Según esto, podríamos decir que DNI.EMPRESA  $\rightarrow$  NOMBRE, pero esta dependencia no es total porque también es cierto que DNI  $\rightarrow$

NOMBRE. Sin embargo, no se puede identificar el sueldo de un empleado sin saber a qué empresa pertenece, por tanto, DNI.EMPRESA  $\rightarrow$  SUELDO sí es una dependencia funcional total.

### 1.9.2. Segunda forma normal

Para definir formalmente la segunda forma normal requerimos saber que es una **dependencia funcional**: Consiste en edificar que atributos dependen de otro(s) atributo(s).

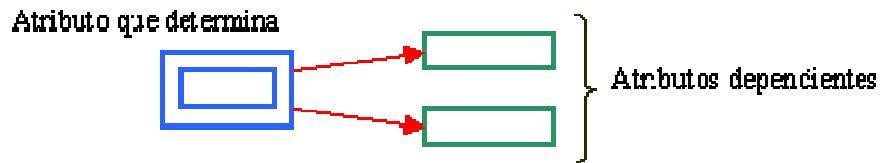


Fig. 1.4

#### Definición formal.

Una relación R está en 2FN si y solo si está en 1FN y los atributos no primos dependen funcionalmente de la llave primaria.

Una relación se encuentra en segunda forma normal, cuando cumple con las reglas de la primera forma normal y todos sus atributos que no son claves (llaves) dependen por completo de la clave. De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, está en segunda forma normal.

La segunda forma normal se representa por dependencias funcionales como:

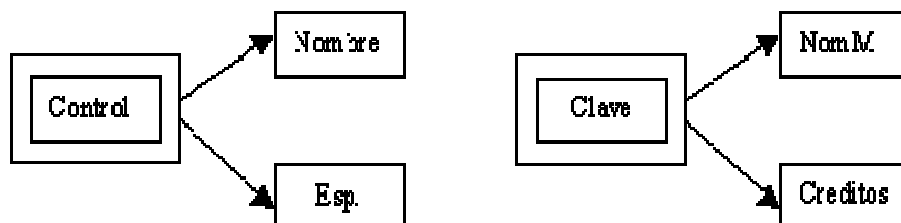


Fig.1.5

Nótese que las llaves primarias están representadas con doble cuadro, las flechas nos indican que de estos atributos se puede referenciar a los otros atributos que dependen funcionalmente de la llave primaria.

### 1.9.3 Tercera forma normal

Una relación R está en 3FN si y solo si está en 2FN y todos sus atributos no primos dependen no transitivamente de la llave primaria.

Consiste en eliminar la dependencia transitiva que queda en una segunda forma normal, en pocas palabras una relación está en tercera forma normal si está en segunda forma normal y no existen dependencias transitivas entre los atributos, nos referimos a dependencias transitivas cuando existe más de una forma de llegar a referencias a un atributo de una relación. Por ejemplo, consideremos el siguiente caso:

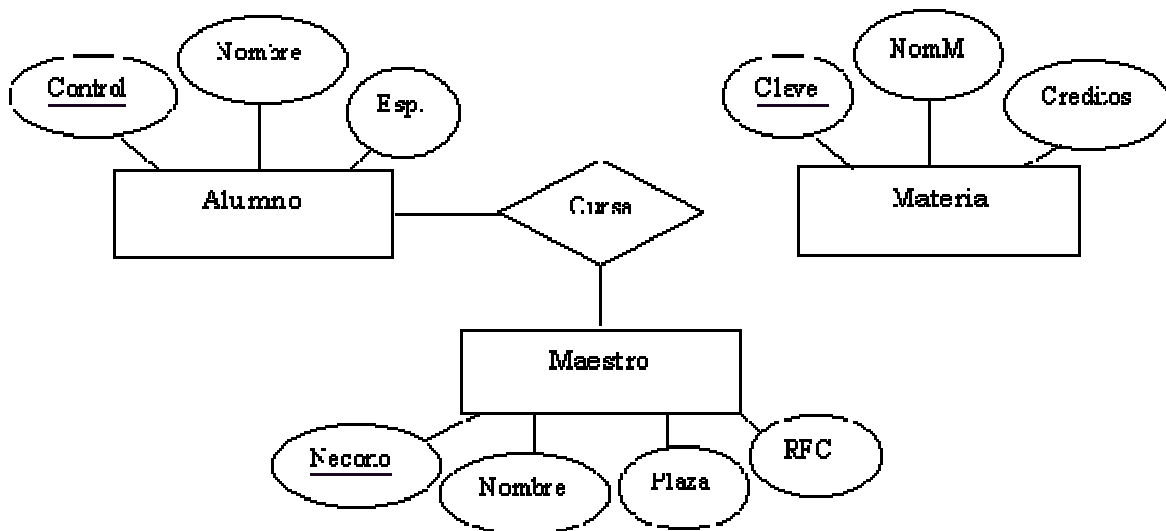


Fig.1.6

Tenemos la relación alumno-curso-materia manejada anteriormente, pero ahora consideramos al elemento maestro, gráficamente lo podemos representar de la siguiente manera:

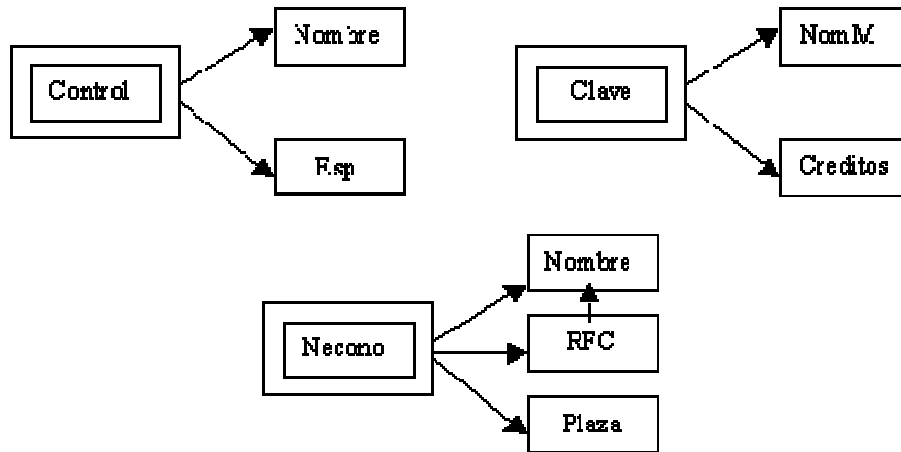


Fig.1.7

Podemos darnos cuenta que se encuentra graficado en segunda forma normal, es decir que todos los atributos llave están indicados en doble cuadro indicando los atributos que dependen de dichas llaves, sin embargo en la llave Necono tiene como dependientes a 3 atributos en el cual el nombre puede ser referenciado por dos atributos: Necono y RFC (Existe dependencia transitiva), podemos solucionar esto aplicando la tercera forma normal que consiste en eliminar estas dependencias separando los atributos, entonces tenemos:

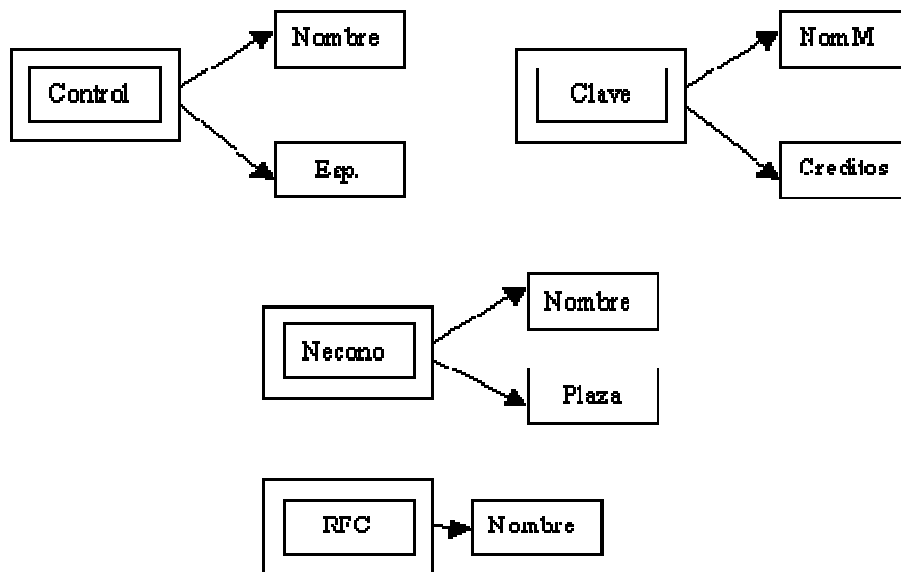


Fig.1.8

## 1.10 Definición de Triggers

Un trigger (o disparador) en una Base de datos , es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE). Es una clase especial de procedimiento almacenado que se ejecuta automáticamente cuando se produce un evento en el servidor de bases de datos.

Son usados para mejorar la administración de la Base de datos, sin necesidad de contar con que el usuario ejecute la sentencia de SQL.

Además, pueden generar valores de columnas, previene errores de datos, sincroniza tablas, modifica valores de una vista, etc.

Permite implementar programas basados en paradigma lógico (sistemas expertos, deducción).

### 1.10.1 Componentes principales

La estructura básica de un *trigger* es:

- **Llamada de activación:** es la sentencia que permite "disparar" el código a ejecutar.
- **Restricción:** es la condición necesaria para realizar el código. Esta restricción puede ser de tipo condicional o de tipo nulidad.
- **Acción a ejecutar:** es la secuencia de instrucciones a ejecutar una vez que se han cumplido las condiciones iniciales.

### 1.10.2 Tipos de triggers

Existen dos tipos de disparadores que se clasifican según la cantidad de ejecuciones a realizar:

- **Row Triggers** (o Disparadores de fila): son aquellas que se ejecutarán n-veces si se llama n-veces desde la tabla asociada al trigger
- **Statement Triggers** (o Disparadores de secuencia): son aquellos que sin importar la cantidad de veces que se cumpla con la condición, su ejecución es única.

### 1.10.3 Efectos y características

- No aceptan parámetros o argumentos (pero podrían almacenar los datos afectados en tablas temporales)
- No pueden ejecutar las operaciones *COMMIT* o *ROLLBACK* por que estas son parte de la sentencia SQL del disparador (únicamente a través de transacciones autónomas)

- Pueden causar errores de mutaciones en las tablas, si se han escrito de manera deficiente.

#### 1.10.4 Ejemplo de triggers

Un sencillo ejemplo sería crear un trigger para insertar un pedido de algún producto cuando la cantidad de éste en nuestro almacén sea inferior a un valor dado.

```
BEFORE UPDATE ON tabla_almacen
FOR ALL records
  IF :NEW.producto < 100 THEN
    INSERT INTO tabla_pedidos(producto) VALUES ('1000');
  END IF;
SELECT DBO.POLVE.TEST
END
```

## CAPÍTULO II

### ANÁLISIS DEL SISTEMA

#### 2.1 Especificación de requisitos del usuario.

El sistema estará enfocado a la administración de la información generada en los inventarios de dulcerías, en las que se lleva a cabo la compra y venta de dulces, frituras, galletas y chocolates, la venta puede ser por mayoreo y/o menudeo. Según lo que requiera el cliente.

Cuando la Dulcería adquiere un producto, se generan los siguientes datos: nombre producto, marca, tipo, peso y/o cantidad, precio de proveedor, precio de venta, fecha de compra, fecha de caducidad, nombre del proveedor y observaciones. Basándose en la fecha de compra y de caducidad, se generará un reporte del producto próximo a caducar.

Se requieren 4 tipos de consultas: Producto, Usuario, Cliente y Proveedor. En caso de consulta por Producto, ésta se puede realizar por, el nombre o por la marca. Para el Usuario se puede consultar por el Nombre o el Identificador de usuario. Para el Cliente se puede consultar por el Nombre o Identificador. Para el Proveedor se puede consultar por el Nombre o el Identificador de proveedor.

La adquisición de productos se realizará diariamente, por lo cual es necesario que el sistema al final del día genere un reporte del inventario y se envíen los pedidos a los proveedores vía e-mail, debido a la variedad de mercancía que existe y la gran cantidad de operaciones que se llevan a cabo durante el día, se requiere que el sistema realice reportes de estas operaciones de una manera más rápida, segura y confiable. Todas estas funciones son controladas por un administrador.

Existen dos tipos de usuarios administrador y empleado (vendedor). El primero es el que deberá tener acceso a toda la información sobre el inventario así como la revisión del reporte de compras y ventas durante el día, podrá modificar, dar de alta productos, precios e información del personal. El segundo sólo podrá realizar consultas y salida de mercancía.

El vendedor debe poder realizar la consulta por nombre del producto, marca o identificador, si el producto no se encuentra, se lleva un control de los productos que son más pedidos para anexarlos en el reporte que se realiza todos los días.

### 2.1.1 Encuesta de los usuarios del sistema

**P. ¿Qué preparación necesita una persona para atender una Dulcería?**

R. Estudios de educación media superior (bachillerato).

**P. ¿Cuál es el mínimo de personas que pueden atender la Dulcería?**

R Por lo menos un administrador y 3 vendedores.

**P. ¿Existe un identificador clave para cada producto?**

R. Es a través del nombre del producto con el que se maneja la gestión.

**P. Cuando un cliente pide un producto, ¿cuál es el proceso?**

R. Los productos están ordenados por tipo de producto y después por orden alfabético por lo cual se busca de esta manera hasta encontrarlo.

**P. Cuando un producto no se encuentra en existencia, ¿que es lo que se hace?**

R. Se lleva un registro de los productos que no se tiene, para así ponerlos en el pedido.

**P. ¿En base a qué, se levantan los pedidos?**

R. Al final del día se levanta un inventario manual de todo el almacén, los productos que tengan un stock mínimo se ponen en el pedido.

**P. Cuando existe un producto próximo a caducar ¿qué es lo que hace?**

R. Generalmente se tira el producto debido a que es imposible estar verificando la fecha de caducidad, solo es visible al momento de venderlo.

**P. ¿Cada qué tiempo se realiza el inventario?**

R. El inventario se realiza al final de la venta, para que al final del día se genere y se manden los pedidos a los proveedores vía e-mail.

**P. ¿Cuál es el proceso de venta?**

R. El cliente pide el producto, el vendedor busca en el almacén si existe el producto, una vez encontrado se verifica la fecha de caducidad después se manda a caja para realizar las operaciones correspondientes.

**P. ¿Cómo se dan cuenta, que producto debe ser pedido con un mayor volumen?**

R. Es difícil darse cuenta ya que se manejan demasiados productos y no se lleva un control.

**P. ¿Qué es lo que esperas del sistema?**

R. Disponer de una herramienta que haga más fácil el proceso antes mencionado, además de manera rápida y segura donde la intervención del hombre sea mínima, reduciendo así la mano de obra.

**P. ¿Qué tiene que hacer el sistema?**

R. El sistema deberá almacenar toda la información que se genera en la Dulcería, realizar operaciones de los productos, usuarios, proveedores, tiene que facilitar las tareas que diariamente se realizan y que son demasiado laboriosas, de una manera confiable para así generar el pedido y tener una mejor administración de toda la información.

**2.2 Análisis del sistema**

El sistema SISCOIND es una herramienta que facilita la administración y manejo de operaciones en controles de inventarios. Uno de los grandes problemas que enfrentan los diferentes sectores, en particular el sector consumo es la gran cantidad de información y control de ésta, por tal motivo se pensó en un sistema que facilite el manejo de la información realizando algunas operaciones de forma “automática”. En específico la generación de reportes de stocks y de pedidos del inventario de dulcerías mediante triggers(disparadores), lo cual es el punto clave en el desarrollo del sistema SISCOIND, ya que se hará uso de nuevas herramientas para realizar el proceso antes mencionado en forma más eficiente y segura.

En una dulcería se manejan 4 tipos de productos: dulces, galletas, frituras y chocolates. Las operaciones que se realizan son las siguientes: catálogos, compras, ventas, , consultas y reportes, siendo el vendedor y el administrador quienes se encargan de estas funciones. La información se ira almacenando en la base de datos, el sistema registrara estos valores, para que al final del día se genere un reporte de stocks de productos existentes en el almacén y así determinar los productos que serán adquiridos. Una vez realizado esto, el sistema generará el pedido correspondiente de manera “automática”, tomando en cuenta los productos que tienen un stock mínimo.

El control de personal, se hace con el fin de tener información de; clientes, proveedores así como del personal que labora en la dulcería. Los clientes o usuarios que hayan realizado más de una vez, alguna operación serán almacenados en la base de datos para evitar volver a pedirles sus datos. Como se dijo anteriormente el sistema generará información de manera controlada y segura.

**2.3. Descripción del sistema**

En lo que sigue se da una explicación de las principales funciones del sistema SISCOIND con el fin de que cada usuario pueda entenderlas y ejecutarlas de forma clara y sencilla

**Producto:** Es la entidad principal del sistema, la cual estará dividida en 4 tipos. Sus atributos son: identificador del producto, nombre, marca, tipo, peso y/o cantidad, precio de compra, precio de venta, fecha de caducidad.

En esta entidad se almacenarán los datos antes mencionados. El tipo de producto puede ser clasificado en las siguientes opciones: galletas, dulces, chocolates o frituras, según corresponda.

**Usuario:** En esta entidad, es donde se almacenarán los dos tipos de usuarios: administrador y vendedor. Sus atributos son: identificador único, nombre completo, dirección, teléfono, RFC, tipo de usuario, e-mail, fecha de nacimiento.

Esta entidad se divide en dos tipos de usuarios: el administrador (que deberá tener todos los privilegios para realizar cualquier operación) y el vendedor (que sólo podrá realizar consultas y ventas de productos).

**Cliente:** Entidad que se refiere a los tipos de clientes de la Dulcería, sus atributos son: identificador único, nombre completo, dirección, teléfono, e-mail, RFC (en caso de facturación).

**Proveedor:** Entidad que se refiere a la persona física o moral que suministra la mercancía, de la cual se obtienen los siguientes atributos: identificador único (RFC), Nombre completo, dirección, teléfono, contacto, e-mail, ciudad.

**Pedido:** Entidad que se refiere al reporte que se genera después de hacer una verificación de stock de mercancía que se encuentra en el almacén. Sus atributos son: identificador de pedido, fecha, nombre, número de piezas, fecha de entrega, hora, cantidad total, nombre ó identificador del proveedor.

## 2.4. Especificación del sistema

El sistema SISCOIND realizará operaciones controladas para facilitar la administración de la información de los inventarios de Dulcerías, esto con la finalidad de que, tanto el usuario como el cliente realicen sus procesos de una forma más rápida y segura.

### El Vendedor:

Es el que realiza las consultas de: Productos, Clientes, Proveedor o Reporte de existencias. En caso de consulta por producto será por nombre o identificador del producto.

Para realizar ventas de productos, el vendedor entrega al cliente factura o ticket según lo requiera, en caso de ser un nuevo usuario se le pedirán sus datos personales, o en su defecto sólo su identificador de usuario si ya está registrado, en caso de que éste requiera factura.

**Administrador:**

Generará al final del día un reporte del inventario tomando en cuenta los productos de mayor demanda, los productos más pedidos así como el total en productos vendidos:

- Puede realizar modificaciones de Productos, Ventas o Usuarios.
- Verifica el total de productos vendidos por cada vendedor así como el total de ventas por usuario.
- Realiza reportes generales del inventario de la Dulcería.

**2.4.1. Funciones Básicas y Secundarias de SISCOIND.**

Las funciones principales que se llevan a cabo por SISCOIND son las siguientes:

- **Control de Productos**
- **Control de Personal**
- **Control de Clientes**
- **Control de Proveedores**
- **Control de Pedidos**
- **Reportes Generales**
- **Búsqueda General**

Funciones secundarias son las siguientes:

- **Consulta de productos**
  - Por nombre
  - Por id de producto
- **Consulta de proveedores**
  - Por nombre
  - Por id del proveedor
- **Consulta de de inventario**
  - Por id pedido
  - Fecha de pedido
- **Alta de información**
  - De producto
  - De cliente
  - De proveedor
  - De usuario

### 2.4.2 Diagrama Conceptual del sistema

El siguiente diagrama muestra las principales entidades de nuestro sistema de forma general.

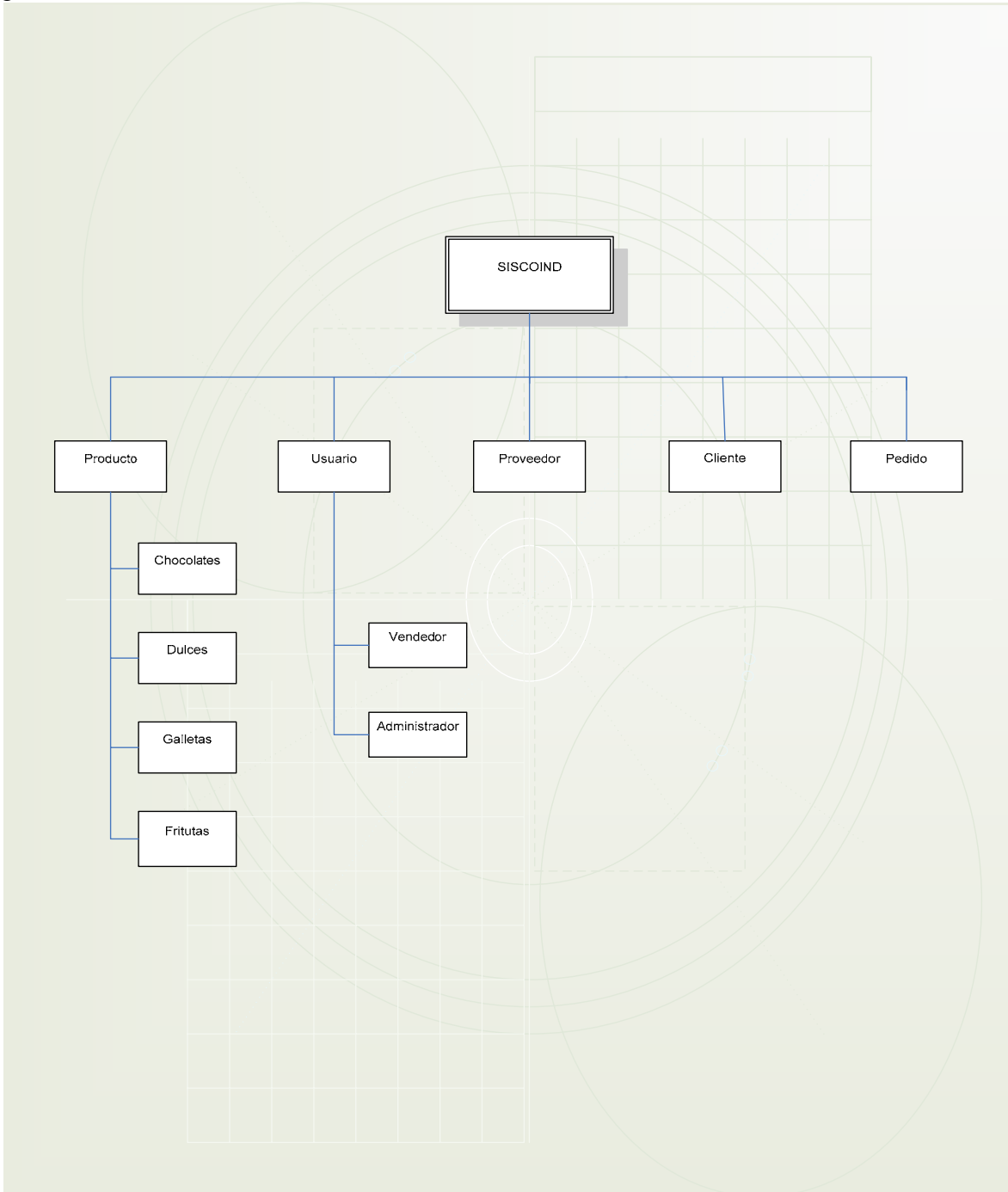
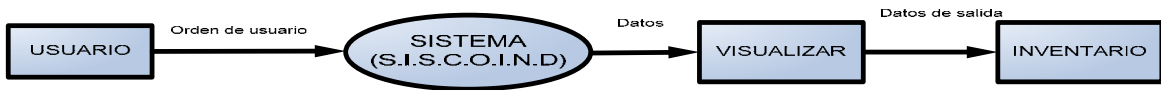


Fig.2.2

2.4.3 Diagrama funcional

La elaboración del diagrama funcional es importante para ver como fluye la información de entrada y salida del sistema, pasando por los 3 niveles de normalización que a continuación se muestran:

Diagrama de flujo de datos nivel 0



DFD Nivel 0

Fig. 2.3

Diagrama de flujo de datos nivel 1

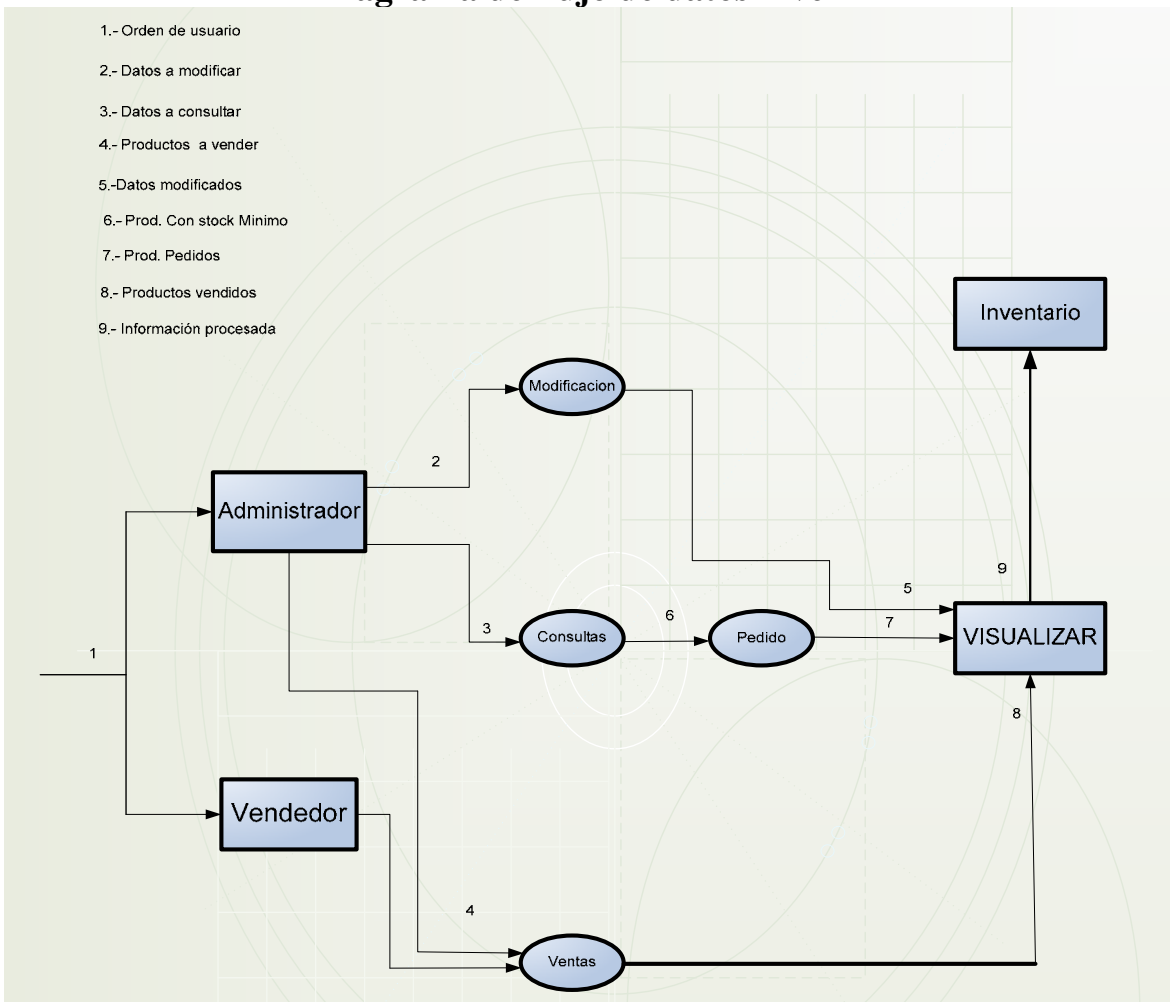


Fig. 2.4

Diagrama de flujo de datos nivel 2

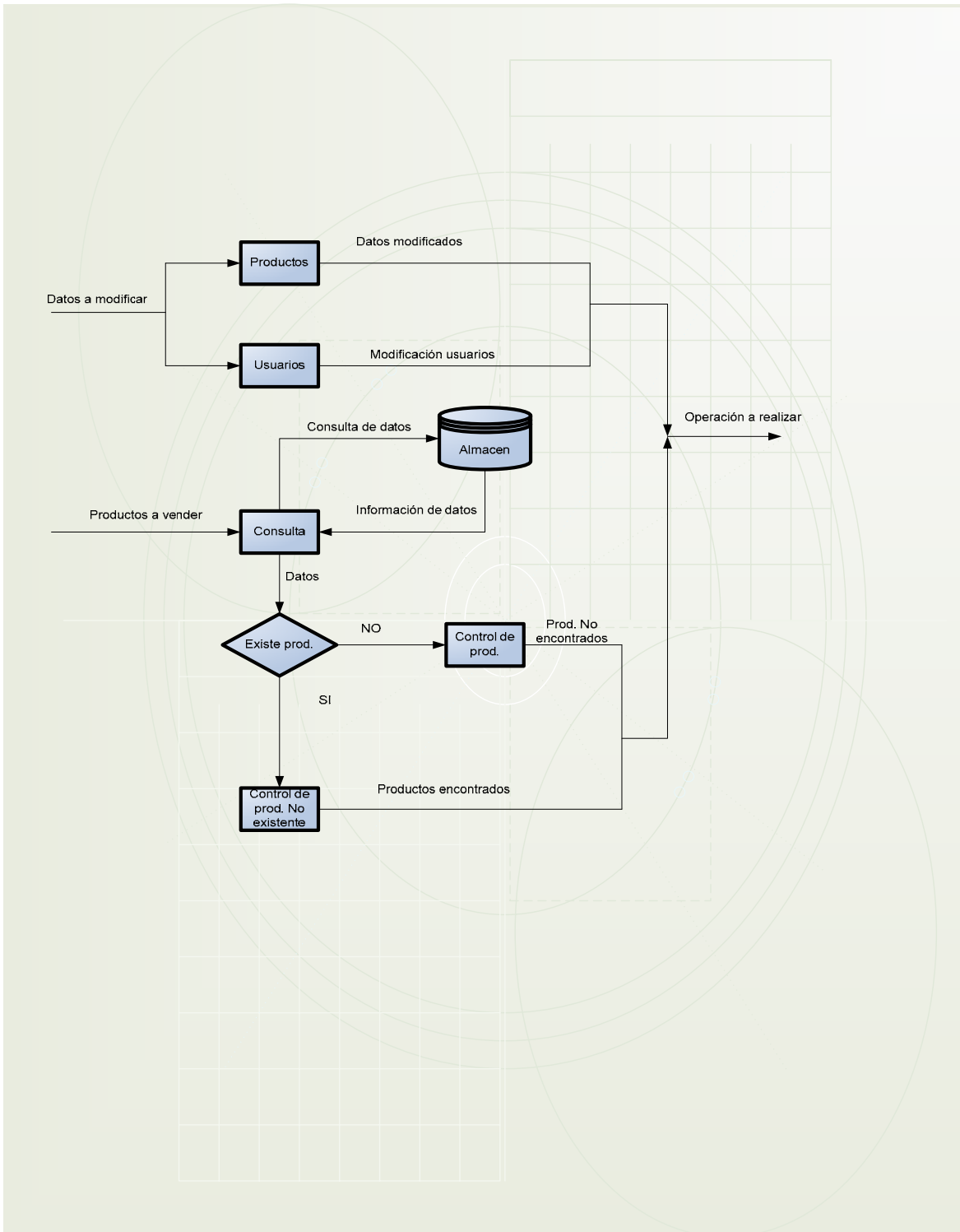


Fig.2.5

## 2.5 Diseño conceptual

Ya obtenidos todos los requisitos del usuario es posible plantear el diseño conceptual de la Base de Datos, que es la información que se utiliza en la empresa, a través del modelo Entidad-Relación, para ello definamos primero los elementos de lo que se denomina el “Patrón Elemental” y posteriormente se presenta el Diagrama del Diseño Conceptual en la Figura 2.6

### 2.5.1 Identificación de entidades

A continuación se muestran las entidades obtenidas del análisis del SISCOIND.

- **Producto**
- **Usuario**
- **Cliente**
- **Proveedor**
- **Pedido**

Una vez identificadas las Entidades, procedemos a encontrar las relaciones que existen entre estas, para así realizar un análisis de los tipos de relaciones que existen para evita los problemas que puedan surgir.

### 2.5.2. Identificación de relaciones

Una vez analizadas e identificadas las entidades de nuestro sistema, proseguimos a identificar las asociaciones que existen, y verificar que tipo de asociación es:

- Un usuario vende uno o más productos
- Un cliente compra uno o más productos
- Un usuario consulta un producto o más productos
- Un proveedor surte uno o más productos
- Un proveedor realiza uno o más pedidos
- Administrador genera uno o más pedidos

### 2.5.3 Localización de los identificadores

En base al análisis realizado, se determinan los “identificadores únicos” para cada entidad.

Nombre del identificador	Entidad	Tipo
<b>Id_pro</b>	<b>Producto</b>	<b>Numérico</b>
<b>Id_user</b>	<b>Usuario</b>	<b>Numérico</b>
<b>Id_cliente</b>	<b>Cliente</b>	<b>Numérico</b>
<b>Id_proveedor</b>	<b>Proveedor</b>	<b>Numérico</b>
<b>Id_pedido</b>	<b>Pedido</b>	<b>Numérico</b>

Una vez realizado el análisis completo del SISCOIND y establecido lo anterior, se muestra el diagrama “Entidad – Relación” de la Base de Datos involucrada en tal sistema.

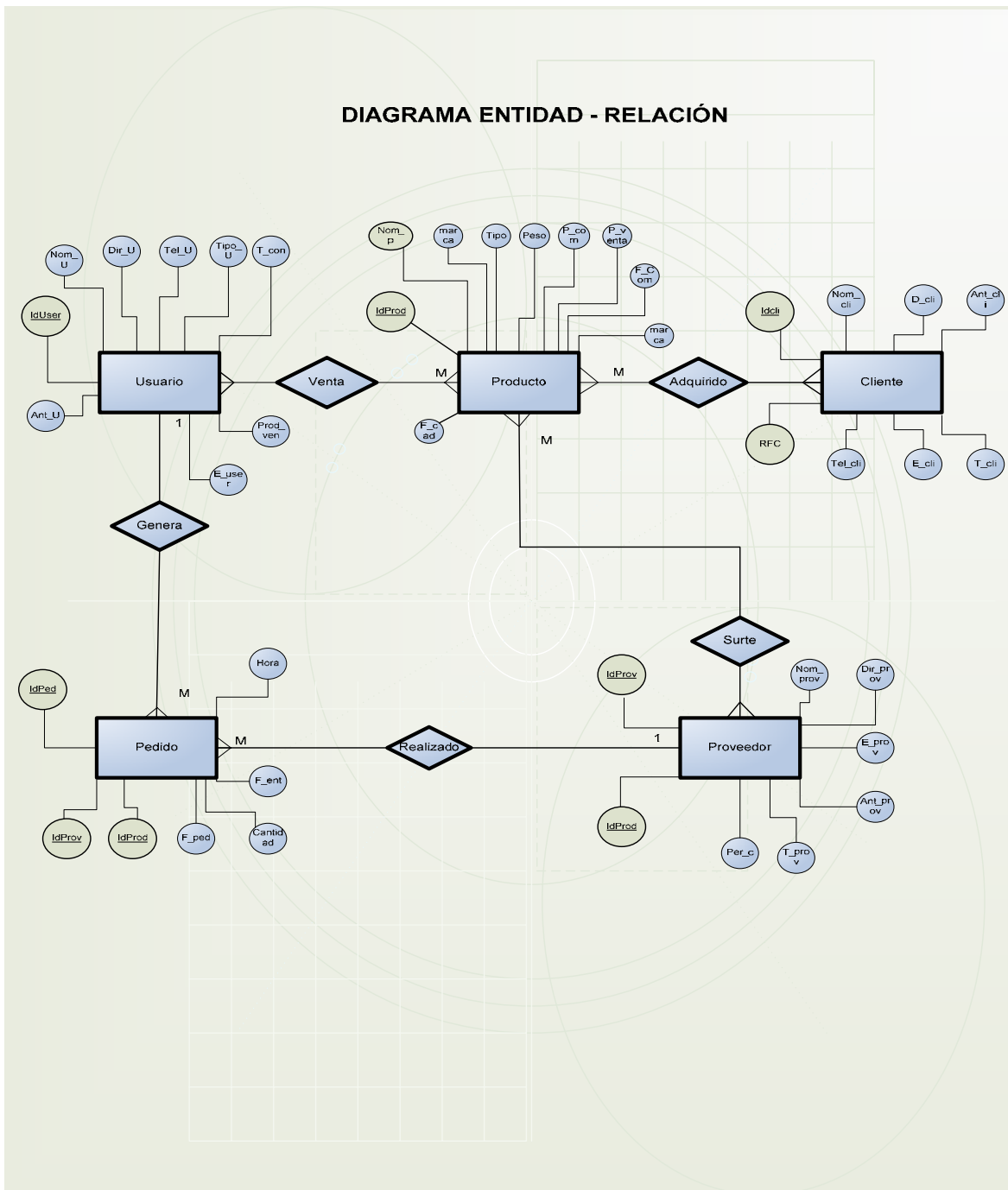


Fig.2.6

**2.6 DICCIONARIO DE DATOS.**

Una vez analizado el sistema y definido claramente los atributos es necesario que se dé la descripción de cada entidad junto con sus atributos, así como el tipo de cada uno de éstos, como se muestra en las tablas siguientes:

**Entidad PRODUCTO**

Nombre	Tipo	Descripción
Identificador del producto (idProd)	Entero numérico	Numérico
Nombre del producto (Nom_p)	Cadena 60	Caracteres
Marca del producto (marca)	Cadena 30	Caracteres
Tipo del producto(Tipo)	Cadena 50	Caracteres
Peso del producto(Peso)	Entero numérico	Numérico
Cantidad (cantidad)	Entero Numérico	Numérico
Precio de compra ( P_com)	Entero numérico	Entero/ Moneda
Precio de Venta (P_venta)	Entero Numérico	Entero/ Moneda
Fecha de compra ( F_Com)	Fecha/hora	Fecha y hora completa
Fecha de caducidad (f_cad)	Fecha/hora	Fecha y hora completa
Descripción del producto (Desc_prod)	Cadena 100	Caracteres

**Entidad USUARIO**

Nombre	Tipo	Descripción
Identificador de usuario (Iduser)	Entero numérico	Numérico
Nombre del usuario ( Nom_U)	Cadena 60	Caracteres
Dirección (Dir_U)	Cadena 60	Caracteres
Teléfono del usuario (tel_U)	Numérico 15	Tira de 15 dígitos
Tipo de usuario (Tipo_U)	Cadena 30	Caracteres
e-mail de usuario (E-user)	Cadena 50	Caracteres
Fecha de nacimiento (F_nac)	Fecha	Fecha

**Entidad CLIENTE**

Nombre	Tipo	Descripción
Identificador del cliente (IdCli)	Entero numérico	Numérico
Nombre cliente ( Nom_cli)	Cadena 60	Caracteres
Dirección cliente (D_cli)	Cadena 60	Caracteres
Teléfono del cliente (Tel_cli)	Numérico 15	Tira de 15 dígitos
e-mail del cliente (e_cliente)	Cadena 30	Caracteres
Tipo de cliente ( T_cli)	Cadena 30	Caracteres
RFC	Cadena 30	Caracteres

**Entidad PROVEEDOR**

Nombre	Tipo	Descripción
Identificador del proveedor (Idpro)	Cadena 30	RFC
Nombre del proveedor (Nom_pro)	Cadena 60	Caracteres
Dirección del proveedor (dir_pro)	Cadena 60	Caracteres
Telefono del proveedor (T_pro)	Numérico 15	Tira de 15 dígitos
Persona de contacto ( Per_c)	Cadena 60	Caracteres
e-mail del proveedor(e_prov)	Cadena 30	Caracteres
Ciudad del proveedor (ciudad)	Cadena 40	Caracteres

**Entidad PEDIDO**

Nombre	Tipo	Descripción
Identificador de pedido (IdPed)	Entero numérico	Numérico
Identificador de proveedor (IdProv)	Entero numérico	Numérico
Identificador de producto (IdPro)	Entero numérico	Numérico
Identificador usuario(IdUser)	Entero numérico	Numérico
Fecha de pedido (F_ped)	Fecha/hora	Fecha y hora completa
Fecha de entrega (F_ent)	Fecha/hora	Fecha y hora completa
Descripción del pedido(Desc_ped)	Cadena 150	Caracteres

## CAPITULO III

### DISEÑO DEL SISCOIND.

#### 3.1 Revisión de requerimientos.

Una vez realizado el análisis del sistema SISCOIND, encontramos detalles que deben mejorarse a través del Diseño Conceptual y durante el proceso de mapeo al diseño lógico, con el fin de eliminar la redundancia y garantizar la integridad de los datos.

#### 3.2 Modelo Relacional “Justificación para el SISCOIND”

Uno de los principales problemas en las bases de datos es la administración de grandes volúmenes de información, lo cual nos conduce a utilizar un modelo de consultas orientadas a conjuntos de datos (tablas relacionales), como el modelo relacional con el que será implantado el SISCOIND. Con el uso de tablas relacionales se puede distinguir y manipular de una manera rápida y eficiente la información en la Base de Datos. El modelo relacional se basa en la teoría de conjuntos y el álgebra relacional, por lo que se pueden aplicar varias operaciones en las tablas.

#### 3.3 Diseño lógico

Se realiza el diseño lógico tomando como referencia el modelo relacional esto con el fin de que cada entidad base se convierta en relación base, así como sus vínculos. Se construye un esquema de la información que utiliza la empresa, basándose en el modelo entidad relación

##### 3.3.1 Convertir esquemas conceptuales en esquemas lógicos

Partiendo de las vistas conceptuales locales que se toman del Diseño Conceptual de nuestro sistema SISCOIND, sus correspondientes tablas se reducen a las siguientes formas:

#### Vista Usuario - Producto



Fig. 3.1

Tabla **USUARIO**

IdUser	Nom_U	Dir_u	Mail_u	Cargo	Username	password	Estado	Tel_U
<b>PK</b>								

Tabla **PRODUCTO**

IdProd	Nom_p	Marca	tipo	Peso	Cantidad	P_Com	P_ven	F_com	F_cad
<b>PK</b>									

Tabla **Venta**

IdUserIdProd	T_venta	Fecha_v	Total

**Vista producto - Cliente**



Fig. 3.2

Tabla **PRODUCTO**

IdProd	Nom_p	Marca	tipo	Peso	Cantidad	P_Com	P_ven	F_com	F_cad
<b>PK</b>									

Tabla **CLIENTE**

IdCli	Nom_cli	D_cli	Tel_cli	e_cli	RFC
<b>PK</b>					

Tabla **ADQUIRIDO**

IdProdIdCli	Cantidad_p	Fecha_A	Total_A

Vista Usuario – Pedido

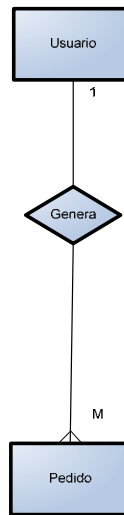


Fig. 3.3

Tabla **USUARIO**

IdUser	Nom_U	Dir_u	Mail_u	Cargo	Username	password	Estado	Tel_U
<b>PK</b>								

Tabla **PEDIDO**

Idped	IdProv	Idprod	F_ped	Cantidad	Hora	F_Ent
<b>PK</b>						

Tabla **GENERA**

idUserIdped	Total_G

**Vista Producto - Proveedor**

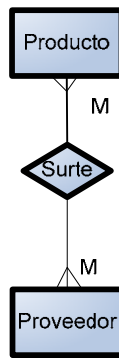


Fig. 3.4

**Tabla Producto**

IdProd	Nom_p	Marca	tipo	Peso	Cantidad	P_Com	P_ven	F_com	F_cad
<b>PK</b>									

**Tabla Proveedor**

IdProv	idProd	Nom_prov	Dir_prov	T_prov	Per_c	e_prov	Ant_prov
<b>PK</b>							

**Tabla SURTE**

IdProd	IdProv	Cantidad_s	Total_s

**Vista Pedido - Proveedor**



Fig. 3.5

Tabla **PEDIDO**

Idped	IdProv	Idprod	F_ped	Cantidad	Hora	F_Ent
<b>PK</b>						

Tabla **Proveedor**

IdProv	idProd	Nom_prov	Dir_prov	T_prov	Per_c	e_prov	Ant_prov
<b>PK</b>							

Tabla **Realizado**

IdUser	IdProv	Total_R

### 3.4. Eliminación de asociaciones de muchos a muchos

Uno de los problemas que se deben resolver en el Diseño Conceptual de las bases de datos relacionales, son las asociaciones de muchos a muchos, por lo que vamos a identificar las entidades donde se presentan tales casos para eliminarlas y sustituirlas por una nueva entidad débil, misma que depende de la ocurrencia de las entidades originales.

Como se puede observar en las figuras anteriores 3.1, 3.2 y 3.4 ahí existen asociaciones de muchos a muchos las cuales deben ser eliminadas, el objetivo es producir una relación de 1 a muchos en cada caso para poder obtener los esquemas lógicos locales correspondientes.

De la fig. 3.1 tenemos que al eliminar la asociación muchos a muchos, obtenemos:



Fig.3.1.1

**Venta**

IdUserIdProd	T_venta	Fecha_v	Total

De la fig. 3.2 tenemos que al eliminar la asociación muchos a muchos, obtenemos:

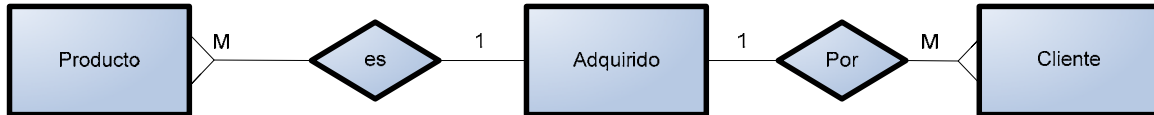


Fig.3.2.1

**ADQUIRIDO**

IdProdIdCli	Cantidad_p	Fecha_A	Total_A

De la **fig.3.4** tenemos que al eliminar la asociación muchos a muchos, obtenemos:

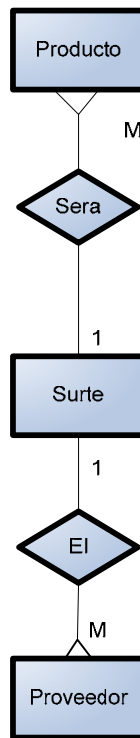


Fig.3.4.1

**SURTE**

IdProd IdProv	Cantidad_s	Total_s

Las tablas que se obtuvieron de la eliminación de asociaciones de muchos a muchos son:

**Venta**

IdUserIdProd	T_venta	Fecha_v	Total

**ADQUIRIDO**

IdProdIdCli	Cantidad_p	Fecha_A	Total_A

Tabla **PRODUCTO**

Id_Prod	Nom_p	Marca	tipo	Peso	Cantidad	P_Com	P_ven	F_com	F_cad	Desc_prod
<b>PK</b>										

- Id\_Prod= Identificador del producto
- Nom\_p= nombre del producto
- Marca= marca del producto
- P\_com= precio de compra
- P\_ven= precio de venta
- F\_com= fecha de compra
- F\_cad=fecha de caducidad
- Des\_prod = Descripción del producto

Tabla **USUARIO**

Id_User	Nom_U	Dir_u	Lugar_N	Tel_U	Tipo_u	email_user	Ant_user	F_nac	RFC	Salario
<b>PK</b>										

- IdUser= Identificador de usuario
- Nom\_U= Nombre de usuario
- Dir\_u= Dirección del usuario
- Lugar\_N=Lugar de nacimiento
- Tel\_u= telefono del usuario
- Tipo\_u= tipo de usuario
- E\_user= email de usuario
- Ant\_user=antigüedad de usuario
- F\_nac= Fecha de nacimiento

Tabla **CLIENTE**

idCli	Nom_cli	D_cli	Tel_cli	e_cli	T_cli	Ant_cli	RFC
<b>PK</b>							

IdCli= Identificador del cliente  
 Nom\_cli= Nombre del cliente  
 D\_cli= Direccion del cliente  
 Tel\_cli=Telefono del cliente  
 e\_cliente= e\_mail cliente  
 T\_cli= tipo de cliente  
 Ant\_cli= antigüedad del cliente  
 RFC

Tabla **PROVEEDOR**

Id_Prov	Nom_prov	Dir_prov	T_prov	Per_c	e_prov	Ant_prov	Ciudad	Edo/provincia
<b>PK</b>								

Idpro= identificador del proveedor que es el RFC  
 nom\_pro= nombre del proveedor  
 dir\_pro= direccion del proveedor  
 T\_prov= telefono del proveedor  
 Per\_c= Persona de contacto  
 E\_prov= e-mail del proveedor  
 Ant\_prov= antigüedad del proveedor

Tabla **PEDIDO**

Id_ped	F_ped	F_entrega	Nom_producto	Cantidad	Total_pedido	Total_productos
<b>PK</b>						

IdPed= identificador de pedido  
 F\_ped= Fecha de pedido  
 F\_ent= Fecha de entrega  
 Nom\_producto= Nombre de los productos que se compran y se encuentran dentro del pedido  
 Cantidad= Cantidad del producto  
 Total\_pedido=costo total del pedido  
 Total\_productos= total de productos que se compraron en dicho pedido.

### 3.5 Validación de esquemas lógicos locales mediante la normalización de las entidades.

Una vez determinados los identificadores para cada tabla, aplicamos las tres primeras Formas Normales a cada una de las relaciones antes mencionada.

#### 3.5.1 Primera Forma Normal (1FN).

Una relación R se encuentra en 1FN si y sólo si cada renglón columna contiene valores atómicos. Se revisa que cada tabla no contenga atributos repetidos.

Tabla **USUARIO**

IdUser	Nom_U	Dir_u	Lugar_N	Tel_U
<b>PK</b>				

Tabla **PRODUCTO**

IdProd	Id_tipo_p	Nom_p	Marca	Peso	Cantidad	P_Com	P_ven	F_com	F_cad
<b>PK</b>									

**Tipo**

Id_tipo_p	Tipo

Tabla **CLIENTE**

IdCli	Nom_cli	D_cli	e_cli	Ant_cli	RFC	IDTel_cli	ID_T_cli
<b>PK</b>							

**Tels\_cli**

IDTel_cli	Tel_cli

relT\_cliente

Id_T_cli	T_cli

Tabla **PROVEEDOR**

IdProv	Nom_prov	Dir_prov	Per_c	e_prov	Ant_prov	Ciudad	Edo/provincia	IdT_prov
<b>PK</b>								

**Ts\_prov**

IdT_prov	T_prov

Observamos que cada una de las tablas cumple con la primera 1FN, ya que eliminamos los grupos repetitivos de las tablas individuales. Se crearon tablas separadas por cada grupo de datos relacionados. Se encuentra plenamente identificado cada grupo de datos está relacionado con una llave primaria.

### 3.5.2 Segunda Forma Normal (2FN).

Una relación se encuentra en 2FN, cuando cumple con las reglas de la 1FN y todos sus atributos que no son llaves dependen funcionalmente de la llave primaria. En caso de que no cumplir con lo antes mencionado, se tiene que crear una nueva subtabla.

Para la tabla:

Id_Prod	Nom_p	Marca	Tipo	Peso	Cantidad	P_Com	P_ven	F_com	F_cad	Desc_prod
<b>PK</b>										

En la tabla **PRODUCTO**, Observamos que esta cumple con la 1FN y no tiene información repetida, así como todos los atributos dependen funcionalmente de la llave primaria (Id\_prod), pero observamos que esta tabla tiene una relación con la tabla proveedor, por lo que tenemos que relacionarlo mediante una llave externa.

#### Tabla **PRODUCTO**

IdProd	Nom_p	Marca	Tipo	Peso	Cantidad	P_Com	P_ven	F_com	F_cad	Desc_prod	IdProv
<b>PK</b>											

#### Tabla **USUARIO**

Id_User	Nom_U	Dir_u	Lugar_N	Tel_U	Tipo_u	email_user	Ant_user	F_nac	RFC	Salario
<b>PK</b>										

Para la tabla usuario tenemos que en el atributo “ tipo\_u” tipo de usuario existe información repetida, para que cumpla con la 2FN tenemos que separar este campo en una nueva tabla llamada Tipo\_U

#### **Tipo\_u**

Id_tipo_u	Rel_id_user	Tipo_usuario

Así las nuevas tablas correctas serian las siguientes:

Id_User	Nom_U	Dir_u	Lugar_N	Tel_U	email_user	Ant_user	F_nac	RFC	Salario
<b>PK</b>									

Id_tipo_u	Rel_id_user	Tipo_usuario

Para la siguiente tabla tenemos que:

Entidad **CLIENTE**

IdCli	Nom_cli	D_cli	Tel_cli	e_cli	T_cli	Ant_cli	RFC
<b>PK</b>							

Observamos que en esta tabla todos los atributos dependen de la llave primaria, pero tenemos un problema, cuando se requiera más de un teléfono del cliente no podremos anexar más campos para éste y en el atributo en tipo de cliente habrá duplicidad de información, por lo que se tendrá que agregar una subtabla:

Tabla **CLIENTE**

IdCli	Nom_cli	D_cli	e_cli	Ant_cli	RFC	IDTel_cli
<b>PK</b>						

**Tels\_cli**

IDTel_cli	Rel_idcli	Tel_cli

Así corregimos el problema, cuando se requiera de más números telefónicos para el cliente, podremos anexarlos sin ningún problema, una vez realizado esto, podemos decir que esta entidad se encuentra en 2FN.

Lo mismo pasa con la siguiente tabla, por lo que se aplica el mismo método:

Tabla **PROVEEDOR**

IdProv	Nom_prov	Dir_prov	Per_c	e_prov	Ant_prov	Ciudad	Edo/provincia
--------	----------	----------	-------	--------	----------	--------	---------------

PK							
----	--	--	--	--	--	--	--

**Id\_prov**

IdT_prov	Rel_idprov	T_prov

Es decir se eliminan aquellos campos que no dependan de la llave primaria.

Observamos que cada una de las tablas antes vistas cumplen con la 1FN, y no todos los campos dependen funcionalmente de la llave primaria, por lo que podemos decir que todas las tablas antes mencionadas se encuentran en **2FN**.

**3.5.3 Tercera Forma Normal (3FN).**

La 3FN consiste en eliminar la dependencia transitiva que queda en una segunda forma normal, en pocas palabras una relación está en 3FN si está en 2FN y no existen dependencias transitivas entre los atributos, nos referimos a dependencias transitivas cuando existe más de una forma de llegar a las referencias de un atributo de una relación. Es decir se eliminan aquellos campos que no dependan funcionalmente y totalmente de la llave primaria.

Cada una de las tablas se encuentra en segunda forma normal, y cada atributo depende de la clave principal por lo que podemos decir que se encuentran todas las tablas en **tercera Forma Normal**.

En cada una de las tablas anteriores observamos que no existen dependencias transitivas entre los atributos, por lo que concluimos que se encuentra en 3FN.

**3.6. Restricciones de integridad de las entidades.**

**Usuario.**- Cada usuario tiene un identificador único, no es permitido valores nulos, no permite la duplicidad de usuarios con el mismo nombre.

**Producto.** Para cada producto se tiene una clave única, no se permite valores nulos.

**Ciente.** Cada cliente tiene un identificador único y no acepta valores nulos.

**Pedido.**-Cada pedido tiene identificador único, no acepta valores nulos.

**Proveedor.** Identificador único que es el RFC, no permite valores nulos.

**3.7 Análisis de esquema lógico local con el usuario**

El movimiento de datos dentro de nuestro sistema (SISCOIND) es de extrema importancia para un buen funcionamiento del mismo, el diagrama de flujo de datos que se presentó en el capítulo II, debe corresponder con los sistemas locales para que sea un sistema eficiente.

Para el control de productos de la dulcería que se muestra en nuestro diagrama de flujo de datos, el principal contacto es el control de ventas, el empleado es el que las realiza, esto es lo que corresponde a una de las vistas de los esquemas locales.

El control de pedidos es el que se genera después de realizar el inventario total de mercancía en la dulcería que se realiza al final del día, lo cual interactúa con control de ventas.

El control de clientes interactúa con el control de ventas, el cliente es el que recibe la mercancía que adquiere en la dulcería, el control de proveedores es el que surte los pedidos que se generan de la dulcería, este pedido es proporcionado por el administrador.

El control de usuarios comprende empleados y administradores. Los empleados son quienes realizan la transacción de operaciones y el administrador es quien realiza la transacción de pedidos e inventarios, es aquí donde se pone mayor énfasis, ya que el control de estos es el objetivo del sistema SISCOIND.

## CAPITULO IV

### IMPLANTACION Y PRUEBAS

#### 4.1 Introducción.

A partir del análisis y diseño, tanto de la base de datos como del sistema **SISCOIND**, en este capítulo se desarrolla la implantación y las pruebas del mismo.

En la actualidad existen varias herramientas para el desarrollo de sistemas de base de datos relacionales. El sistema está diseñado para negocios pequeños, la base de datos de este sistema será local, tales características son de suma importancia para su diseño ya que de aquí se partió para su realización. Tomando en cuenta lo anterior, se determina que el sistema será implantado en Visual Basic 6.0 y el gestor de la base de datos será Microsoft Access 2003, descritos en las secciones siguientes.

#### 4.2 Implementación del sistema

En esta sección se muestra una descripción de los elementos más relevantes que conforman el sistema **SISCOIND**.

##### 4.2.1 Inicio del sistema

El sistema inicia pidiendo el nombre del usuario y su Password. Existen dos tipos de usuarios: el vendedor o el administrador, dependiendo del usuario tendrá los permisos correspondientes para usar el sistema, como lo muestra la pantalla siguiente:



#### 4.2.2 Menú principal del SISCOIND.

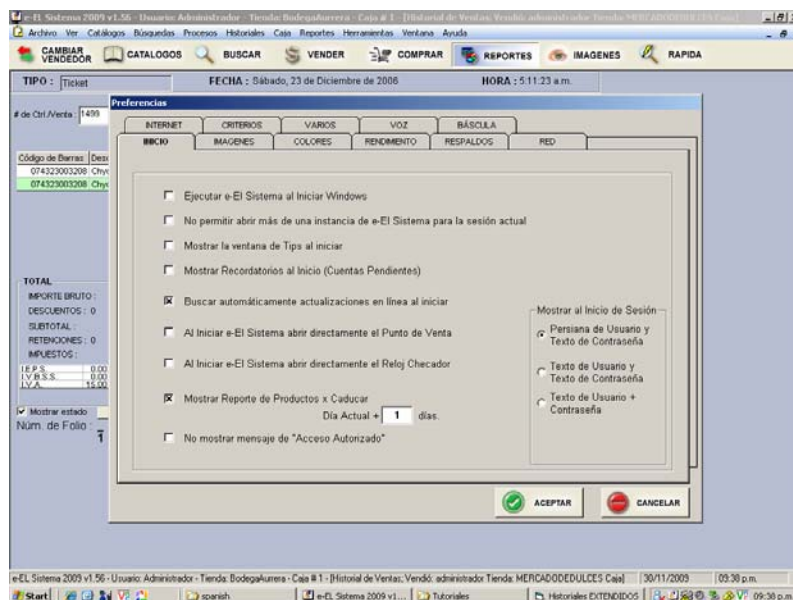
Una herramienta para el control de ventas esta enfocado para empresas que requieren un sólido punto de venta con un amplio respaldo de operaciones, dentro del almacén, la administración de proveedores y clientes, La seguridad por medio de privilegios.

La pantalla principal del usuario administrador es la que tiene permiso a todas las opciones para la administración del sistema.



ID	Código de Barras	S.K.U.	Descripción	Existencia Actual
453	MINI BUBULUBU	MINI BUBULUBU	MINI BUBULUBU	5
454	7501030423119	WINK	WINK	46
455	7503001379067	VELA LECHE 100	VELA LECHE 100	14
456	7503001379012	VELA AGUA 100	VELA AGUA 100	15
457	719886020025	CHIZPIKAS	CHIZPIKAS	24
458	7501000310630	Q-BO	Q-BO	12
459	75030005897307	MIGUELITO SABORES	MIGUELITO SABORES	25
460	7501061900214	CANICA PERICO	CANICA PERICO	5
461	7501061900115	CANICA TEBOL	CANICA TEBOL	5
462	7501061900023	CANICA AGUITA	CANICA AGUITA	5
463	7501058913489	TRIDENT FRESA	TRIDENT FRESA	10
464	7506105610784	BUBBALO MISTERIO	BUBBALO MISTERIO	19
465	038629001160	SAURIS COM	SAURIS COM	94
466	038629001009	SAURIS CHOCO	SAURIS CHOCO	99
467	7501011170872	GUDUPOP MULTI	GUDUPOP MULTI	15
468	7501011170865	GUDUPOP CHILE	GUDUPOP CHILE	15
469	7501011192379	ROCKALETA	ROCKALETA	10
470	7702025100644	WAFERS CHOCO	WAFERS CHOCO	22
471	7702025107544	WAFERS VAINILLA	WAFERS VAINILLA	22
472	725226000992	MONSTRO	MONSTRO	23
473	7501008766806	CHAVALIN	CHAVALIN	19
474	CHICOTE	CHICOTE	CHICOTE	50
475	GARAPIÑADO	GARAPIÑADO MEDIANO	GARAPIÑADO MEDIANO	22

Cont = 474



### 4.3 Arquitectura del SISCOIND.

La arquitectura de nuestro sistema gestor de base de datos está estructurada de acuerdo a la arquitectura llamada ANSI/SPARC, la cual se divide en 3 niveles muy importantes que se describen brevemente a continuación:

- **Nivel interno:** Es aquel que está más cerca del almacenamiento físico, es decir el que tiene que ver con la forma en que los datos están almacenados físicamente.
- **Nivel Externo:** Es la forma en que el usuario ve los datos.
- **Nivel Conceptual:** Es el nivel de intermediación entre los dos anteriores o en el cual se concibe el diseño de la aplicación.

#### 4.3.1 Nivel Interno o Físico

Los datos del SISCOIND se almacenan en tablas relacionales para un acceso rápido a la información, existen registros de datos de; personas, productos e información en general, por lo que no se requiere de una gran complejidad de almacenamiento.

La mejor forma de almacenamiento de las tablas de usuarios y de productos se hace mediante el uso de identificadores únicos o llaves primarias.

Debido a que la seguridad del sistema es una parte importante del mismo, al inicio de éste se solicita la forma de acceder a él, por lo que tendrán diferentes privilegios, tanto el administrador como el vendedor.

Cada uno de los triggers incluidos en el sistema SISCOIND involucra procesos tales como:

- La validación de información de usuarios y clientes.
- La ejecución de procesos que realizan, a una determinada hora, una consulta de productos para actualizar la base de datos o para realizar un pedido.

Un ejemplo sería crear un trigger para insertar un pedido de algún producto cuando el stock de éste en nuestro almacén sea inferior a un valor dado.

```
BEFORE UPDATE ON tabla_almacen
FOR ALL records
  IF :NEW.producto < 100 THEN
    INSERT INTO tabla_pedidos(producto) VALUES ('1000');
  END IF;
SELECT DBO.POLVE.TEST
END
```

Ejemplo de un triggers que realiza una consulta de los productos que están próximos a caducar, una vez encontrado le asigna una etiqueta de próximo a caducar:

```

CREATE TRIGGER producto
BEFORE INSERT OR UPDATE OF Fecha_cadu
ON Producto
FOR EACH ROW
WHEN (:new.Fecha_cadu > sysdate-3);
BEGIN
  UPDATE producto
  SET fecha_cadu = caducado
  WHERE id_producto = :new.id_producto;
END;

```

### 4.3.2 Nivel conceptual

Las tablas de la base de datos del SISCOIND, implantada en Access 2003, son identificadas por una llave primaria y están agrupadas por columnas como a continuación se muestra:

La siguiente tabla muestra la estructura de Productos en la base de datos:

	Id. de producto	Nombre de producto	Proveedor	Categoría	Cantidad	Precio por unidad
+	1	Ferrero	La gran bodega	Chocolate	7	\$ 58.00
+	2	tupsi	Ribera	Dulce	50	\$ 60.00
+	3	chipileta	Ribera	Dulce	20	\$ 54.00
+	4	pelon	La gran bodega	Dulce	10	\$ 48.00

Cada uno de estos campos está previamente definido de acuerdo al atributo que tenga la entidad, ya que Access no da esa opción de definir los atributos por el usuario.

La siguiente tabla muestra la estructura de Proveedor en la base de datos:

	IdProveedor	NombreCompañía	NombreContacto	CargoContacto	Dirección	Ciudad	Región	CódPos
+	1	Ribera	Martin Glz.	Gerente de compras	ksdpofsad	Puebla	Puebla	45235
+	2	La gran bodega	Jorge Emilio	Administrador de compras	okjdorfooi	Puebla	Puebla	70117
+	3	DCM	Regina Murillo	Representante de ventas	uidhfil qwo9	Puebla	Puebla	48104
+	4	Magadan Suc 1	Blanca Luna	Gerente de marketing	mdsajfwsdj	Puebla	Puebla	58524
+	5	Magadan Suc 2	Antonio del Val	Administrador de compras	pokwfwer	Puebla	Puebla	33007
+	6	Magadan Suc 3	Karla Torres	Representante de ventas	hsas,ewfk	Puebla	Puebla	545

### 4.3.3 Nivel externo.

Debido a que las Dulcerías para las que se creó este sistema ya contaban con Visual Basic, se decidió implantar el SISCOIND con tal herramienta. Algunas consideraciones adicionales de tal decisión fueron las siguientes:

- Visual Basic contiene herramientas fáciles de manejar y de gran utilidad (por mencionar alguna de ellas, citemos la herramienta gráfica que puede ser usada para realizar interfaces con gestores de bases de datos Access 2003).
- Visual Basic proporciona el motor de base de datos JET que es el mismo que utiliza Access.
- La fácil manipulación de datos en Access y Visual Basic.
- Acceso a las bases de datos a través del control de datos ADO, utilizando el motor de Access y controladores de ODBC.
- Biblioteca para SQL, lo cual permite la manipulación de datos en Access.
- Un administrador visual de datos que permitirá crear y manipular bases de datos en Access y ficheros de texto.

## 4.4 Herramientas utilizadas en la aplicación

Para la implantación del sistema **SISCOIND**, es importante describir la arquitectura del gestor de base de datos (Access 2003), y el lenguaje que se utilizó para la interfaz del mismo (visual Basic 6.0), debido a que estas dos herramientas tienen características en común se concluyó que son las más apropiadas para el desarrollo del **SISCOIND**.

### 4.4.1 Análisis de la herramienta visual Basic 6.0

Es un lenguaje de programación que se ha diseñado para facilitar el desarrollo de aplicaciones en un entorno gráfico (GUI-GRAPHICAL USER INTERFACE), como Windows 98 o superior y en donde se pueden manipular bases de datos Microsoft Access.

Las bases de datos relacionales que se utilizan constan de un conjunto de tablas divididas en filas y columnas. El acceso a estas se hace mediante el lenguaje SQL (structured Query lenguaje- lenguaje de consultas estructurado). Por otra parte para que los programadores puedan acceder de forma estándar a una base de datos, los fabricantes suelen desarrollar junto con la base de datos el controlador ODBC (Open Database Connectivity- Conectividad abierta de la base de datos) de acceso correspondiente, que proporciona al programador un conjunto de funciones estándar (API application Programming Interface - interfaz de programación de aplicaciones) para acceder al motor de la base.

Visual Basic proporciona varias formas de acceso a las bases de datos remotas, las cuales se describen a continuación:

- Objetos ActiveX de acceso a datos (ADO- ActiveX Data Objects). Microsoft, a partir de la versión 6 de visual Basic, introduce el modelo de objetos ADO para acceso a la base de datos, es una interfaz común para acceso a datos locales y remotos, conjuntos de registros( record set) remotos y locales, una interfaz de

enlace con los datos accesible para el usuario y un conjunto de registros jerárquicos.

- Control de datos ADO. Permite crear una conexión con la base de datos de una forma fácil y rápida mediante objetos ActiveX de acceso de datos.
- Objetos de acceso a datos (ADO-Data access Objects). El modelo de objetos ADO admite dos entornos diferentes de base de datos o espacios de trabajo. Microsoft Jet permite acceder a base de datos Microsoft Access.
- Control de datos (DC-Data control). Implementa el acceso a datos mediante el motor de base de datos de Microsoft Access.
- RDO. Es una interfaz de acceso de datos mediante ODBC orientada a objetos, que incorpora un estilo sencillo de ADO y cuya interfaz expone prácticamente toda la flexibilidad y eficacia de bajo nivel de ODBC.

Conectividad Abierta de base de datos (ODBC-Open database connectivity). Se trata de una interfaz de programación para acceso a servidores de base de datos que proporciona un lenguaje común para las aplicaciones de Windows que necesitan acceder a una base de datos en una red.

VBSQL. Es una biblioteca de visual Basic para SQL Server. Concretamente se trata de una interfaz de programación (API) para DB-Library.

¿Cuales son las ventajas de uno u otro? Básicamente las ventajas dependen del tipo de usuario para el que se realice la aplicación. El motor Jet funciona bien en monousuario y en redes pequeñas, tiene sus problemas de bloqueo cuando varios usuarios acceden a una misma información o incluso a informaciones colindantes. Con bases de datos grandes es conveniente trabajar con ODBC. La ventaja de trabajar con el motor Jet es que no necesitamos depender de ningún driver ODBC externo, ya que las aplicaciones son mas sencillas y dan menos problemas de configuración. Utilizar RDO es similar en muchos aspectos a utilizar ADO; no obstante hay algunas diferencias, RDO está implementado y diseñado, para utilizarse estrictamente con bases de datos relacionales. RDO no tiene ningún procesador de consultas propio; depende del origen de datos para procesar todas las consultas y crear los conjuntos de resultados. Los objetos de datos propiamente dichos se generan a partir de los conjuntos de resultados y los cursores devueltos por el controlador ODBC. Finalmente, el acceso a datos basado en OLEDB y ADO es adecuado para una gama amplia de aplicaciones cliente-servidor. Las principales ventajas son de fácil utilización, gran velocidad, uso de poca memoria y poca utilización de disco.

De lo anterior concluimos que el modelo de objetos a utilizar es el ADO debido a su manejo sencillo.

### Origen de conexión

Para que un control ADO de una aplicación pueda establecer una conexión con un origen de datos es necesario crear el origen de conexión. Para realizar esta operación hay que asignar a la propiedad ConnectionString del control de uno de los valores siguientes:

- 1.- Un fichero de vínculo a datos.
- 2.- Un nombre de origen de datos ODBC( Es la que utilizaremos).
- 3.- Una cadena de conexión.

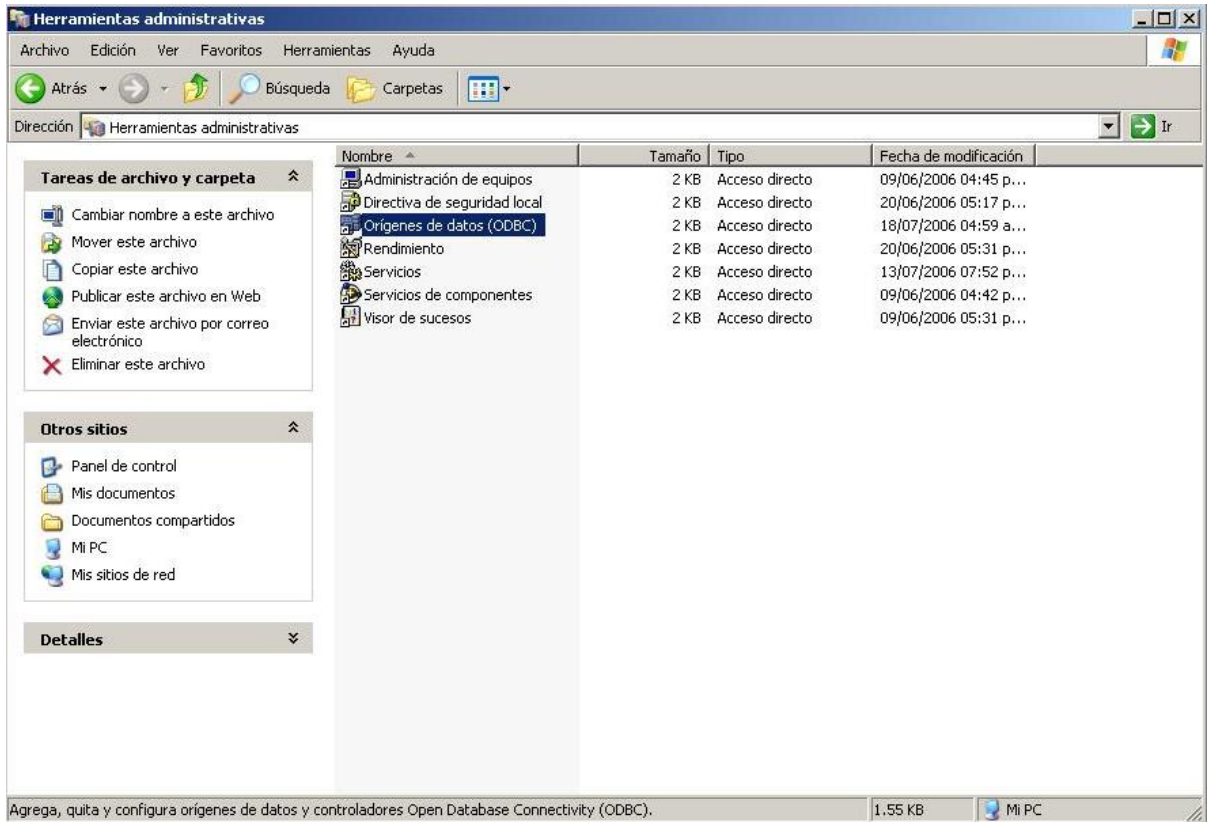
#### Crear un vínculo de datos OLE DB

Para crear un origen de datos OLE DB para cada base de datos a la que se desea tener acceso a través de ADO, se tienen que seguir los siguientes pasos:

- A)** Abrir el explorador de Windows, sitúese en la carpeta donde desea crear el origen de datos OLE DB. Donde se tiene ubicada la base de datos.
- B)** Con el botón derecho del mouse hacer clip en el panel derecho del explorador y seleccionar la instrucción, nuevo del menú contextual. De la lista de tipos de archivo, haga clip en Microsoft data link. Después cambie el nombre del nuevo archivo por el nombre dulcería. udl.
  - 1) Haga clip con el botón derecho del mouse sobre el nuevo archivo dulcería y seleccionar propiedades en el menú contextual.
  - 2) Haga clip en la pestaña Proveedor y seleccione proveedor OLE DB Microsoft jet 3.5.1. haga clip en el botón siguiente para ir a la página de conexión.
  - 3) Haga clip en el botón explorar y seleccione la base de datos dulceria.mdb. Haga clip en probar conexión para comprobar la conexión Si la conexión es correcta, haga clip en aplicar. Haga clip en la pestaña General. Establezca el atributo de sólo lectura y aplicar.

#### 4.4.2 Crear un origen de datos ODBC

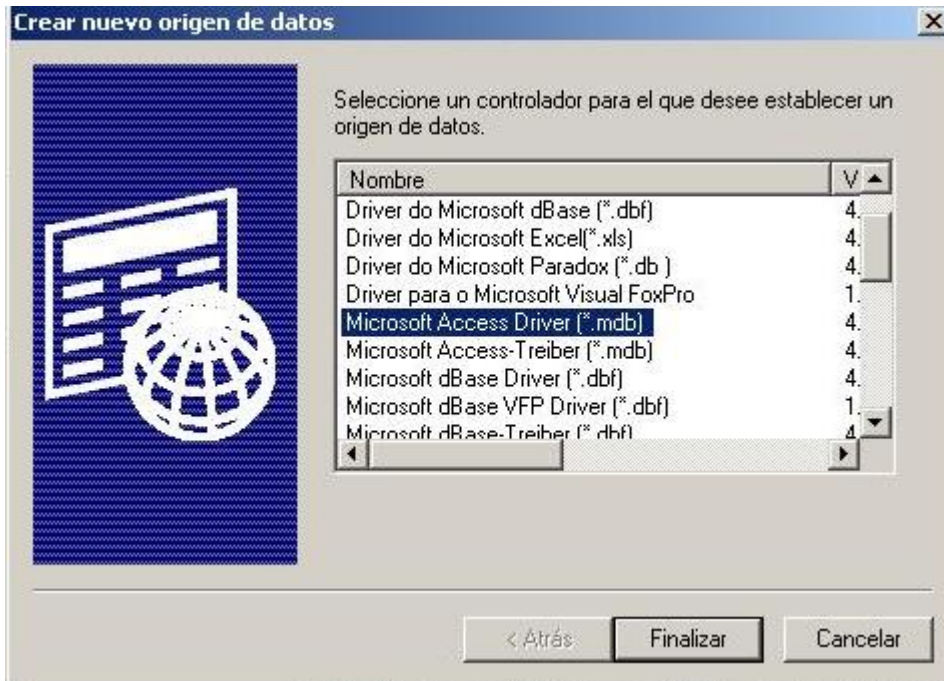
Para crear-registrar la base de datos como origen de datos ODBC, copiar primero la base de datos en el directorio de trabajo. A continuación diríjase al panel de control de Windows y ejecute el administrador ODBC de 32 bits, como se muestra a continuación:



- a) Hacer clip en el botón **Agregar** de la ventana **DNS de usuario** como se muestra a continuación:



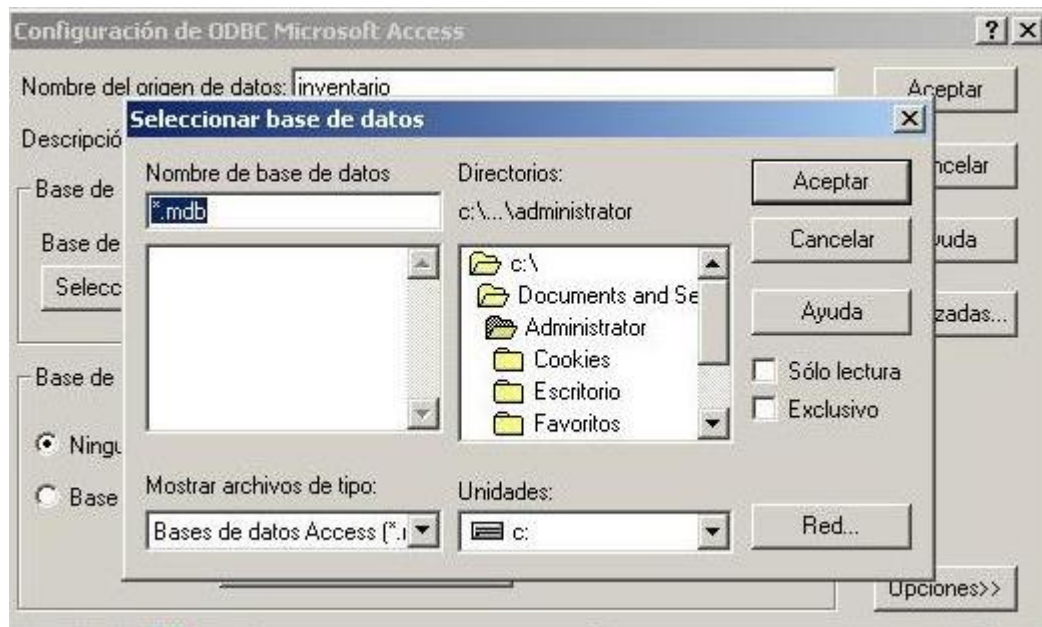
- b) En la ventana crear un nuevo origen de datos que se presenta a continuación, elija controlador ODBC a utilizar y en nuestro caso Access.



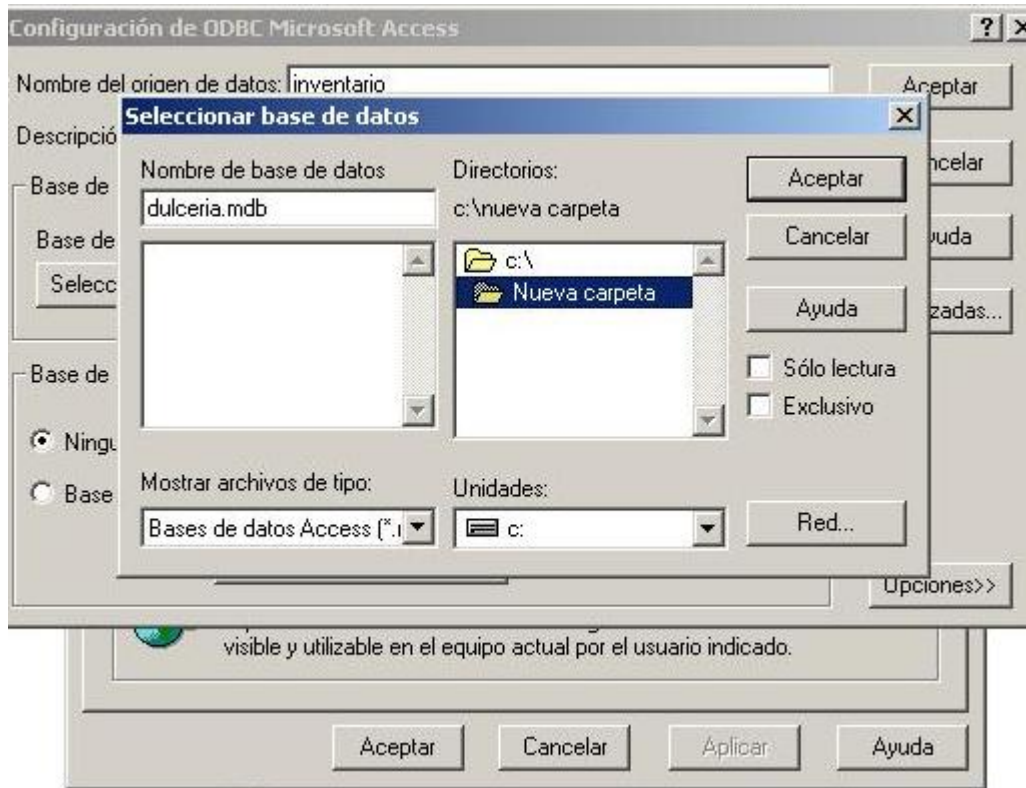
- c) En la ventana Configuración de ODBC, se escribe un nombre para la base de datos (inventario), este nombre será utilizado posteriormente por nuestra aplicación.



- d) Haga clic en el botón seleccionar base de datos de la ventana mostrada anteriormente.



- e) En la ventana, de la pantalla anterior seleccionar base de datos elija el fichero .mdb de base de datos ( dulceria.mdb) y aceptar.



#### 4.5 Análisis de Microsoft Access 2003

**Microsoft Access 2003.** Es un sistema gestor de bases de datos relacionales (SGBD). Una base de datos suele definirse como un conjunto de información organizada sistemáticamente. En la terminología propia de las bases de datos hay tres conceptos claves dentro de las tablas: campo, registro y dato. Con **Microsoft Access 2003** es posible:

- Vincular tablas para tener acceso a datos de varias bases de datos a la vez en los formularios e informes.
- Vincular tablas de otras bases de datos de Access, hojas de cálculo de Microsoft Excel, orígenes de datos ODBC, bases de datos de Microsoft SQL Server y otros orígenes de datos.
- Buscar rápidamente tablas, consultas, formularios o informes que dependan de un objeto de base de datos determinado.
- Cambiar la propiedad de un campo de una tabla y actualizar automáticamente todos los formularios o informes que tengan controles enlazados a él.
- Guardar en otra ubicación una copia de la base de datos en la que se está trabajando.
- Realizar aplicaciones personales de escritorio usando el lenguaje de programación VisualBasic6.
- Compartir datos a través de cualquier plataforma tanto en una intranet como a través de Internet.

## 4.6 Lenguaje de consulta del SISCOIND.

En el sistema SISCOIND es posible usar SQL para realizar las consultas, ya que SQL proporciona todas las herramientas que el sistema utilizará en la ejecución de sus procesos.

Las funciones para la manipulación de datos que se utilizan son las siguientes:

Insert  
Delete  
Update  
Select  
New

En lo que se refiere en la definición de datos tenemos las siguientes sentencias:

Create\_table  
Create\_index  
Truncate: Borra todas las filas de una tabla

## CONCLUSIONES

En la actualidad la tecnología de las Bases de Datos Relacionales con triggers es de vital importancia en diversos sectores tales como: el Automotriz, el químico, el farmacéutico y el de consumo, entre otros, debido a que tales herramientas ayudan a realizar sistemas más funcionales, eficientes y seguros.

Las bases de datos con triggers son capaces de monitorear y reaccionar ante ciertos eventos de manera oportuna y eficiente, se caracterizan por su reacción ante ciertas condiciones que se ejecutan de forma “independiente”.

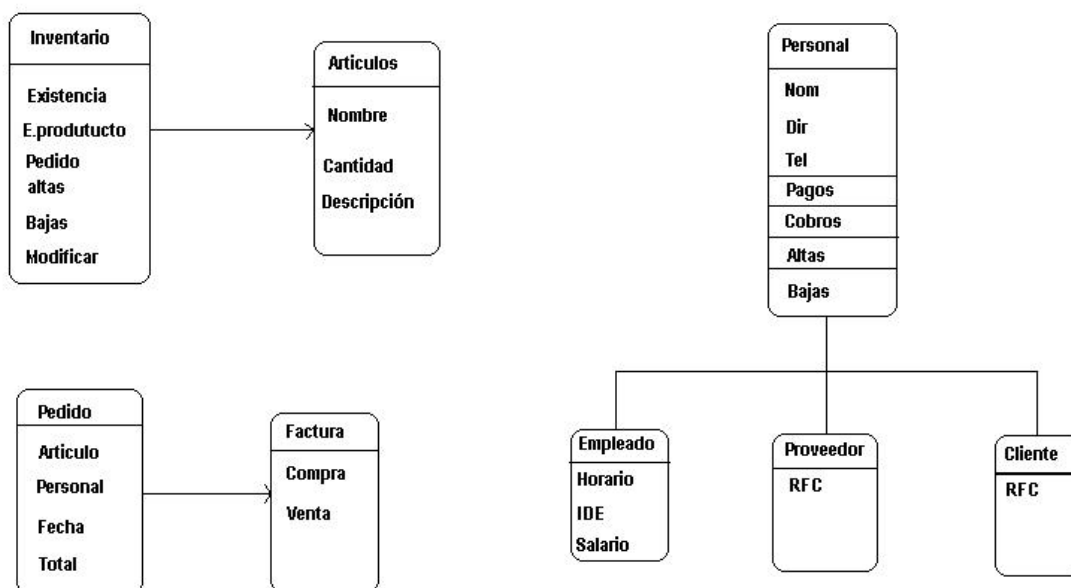
A continuación se presentan las conclusiones obtenidas en el análisis y desarrollo del presente trabajo:

- Se desarrolló un sistema, llamado **SISCOIND**, haciendo uso de los triggers (disparadores), con lo cual se logró que dicho sistema proporcione beneficios en la ejecución de procesos de una manera más rápida, segura, y eficiente. Este sistema está diseñado para dos tipos de usuario, a saber; el usuario administrador y el usuario vendedor, permite al administrador llevar un mejor control de los procesos involucrados con el inventario de dulcerías, lo que redundó en mayores ganancias para la empresa y menores pérdidas económicas.
  
- A través de las pruebas se logró comprobar que el sistema es eficiente, confiable y robusto. Cumpliendo así, los objetivos planteados al inicio de este documento de tesis y el 100% de los requerimientos solicitados por los usuarios del mismo.

## PERSPECTIVAS

El sistema **SISCOIND** puede tener un mayor control en las funciones involucradas con el proceso de inventarios, haciendo uso de reglas activas y no sólo triggers, por ejemplo:

- En el control de productos es necesario determinar quienes son los más vendidos y los menos vendidos por lo que se requiere realizar un análisis al final del día, dicho análisis puede evitarse si el sistema por sí sólo genere reportes y produzca graficas para el administrador.
- Para el control de usuarios: sacar un análisis de la venta de empleados para proporcionar un premio al mejor empleado.
- Se propone que el SISCOIND sea implantado en un modelo cliente – servidor, para distribuir los grandes volúmenes de información que se generen en las varias sucursales locales y nacionales, de una misma empresa, donde las tablas principales puedan estar en el servidor y el resto de forma local y además se produzca el envío automático de reportes de pedidos a los proveedores vía e-mail.
- El paradigma orientado a objetos podría ser otra de las opciones para migrar la base de datos de este sistema a un diseño orientado a objetos, a continuación se da un posible esquema de lo que se refiere a tal propuesta:



# Bibliografía y referencia

[www.todoaccess.com/](http://www.todoaccess.com/)

[www.mundotutoriales.com/tutorial\\_visual\\_basic+\\_access+\\_sql-mdtutorial755687.htm](http://www.mundotutoriales.com/tutorial_visual_basic+_access+_sql-mdtutorial755687.htm).

Tesis, CONTROL DE INVENTARIOS PARA UNA FARMACIA, Javier Conde Iztetzi

# Manual de usuario de SISCOIND

## Iniciar sesión en el sistema

**1 Iniciar el sistema:** Al iniciar SISCOIND le pedirá que se autentifique con un usuario y password, dependiendo de el tipo de usuario tendrá los permisos asignados.



**2 Introducir "Nombre de Usuario":** solo permite introducir nombres de usuario que ya estén registrados por lo tanto basta con elegir el nombre de la lista que aparece en el combo de usuario, contiene el usuario de administrador y el password es "sistemas" posteriormente podrá crear los usuarios que quiera con el perfil correspondiente.

**3 Introducir "Contraseña":** Deberá introducir la contraseña correspondiente a el nombre de usuario.

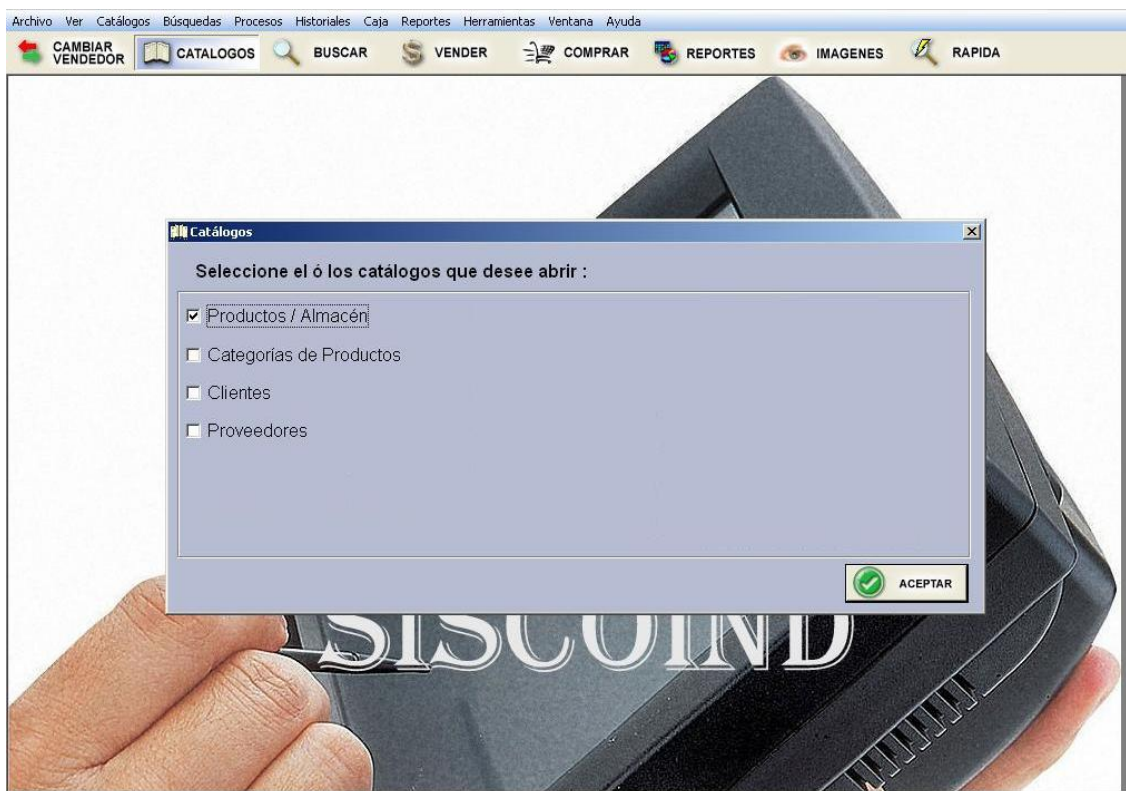
### 4 "Aceptar"

**5 Acceso Autorizado o Acceso Denegado:** Según como el SISCOIND verifique su identidad, emitirá dos mensajes diferentes según sea el caso: "Acceso Autorizado" ó "Acceso Denegado"

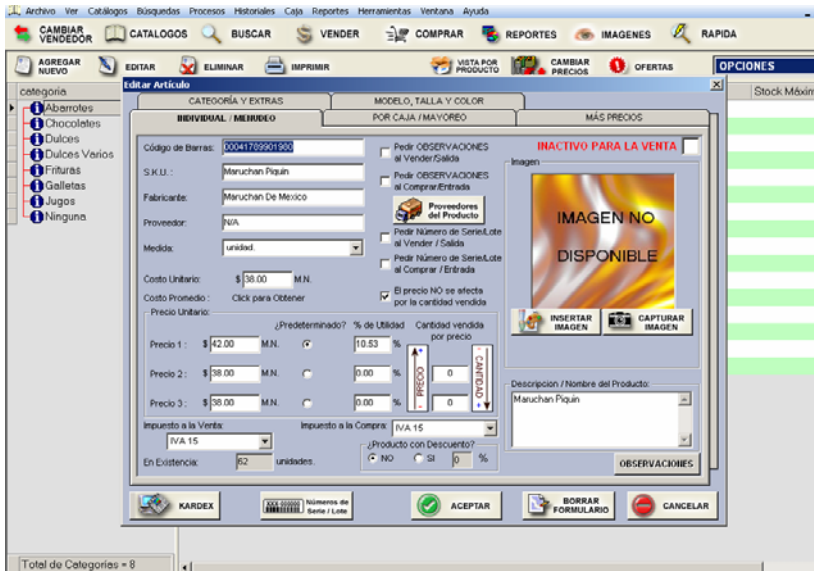
Una vez aceptado el usuario y password muestra la siguiente pantalla, la cual contiene los procesos más comunes que se pueden realizar en el sistema.



Para dar de alta un producto, tenemos que ir a la ventana de Catálogos donde encontraremos a productos, clientes y proveedores, desde esta ventana se pueden dar de alta, como lo muestra la siguiente imagen:

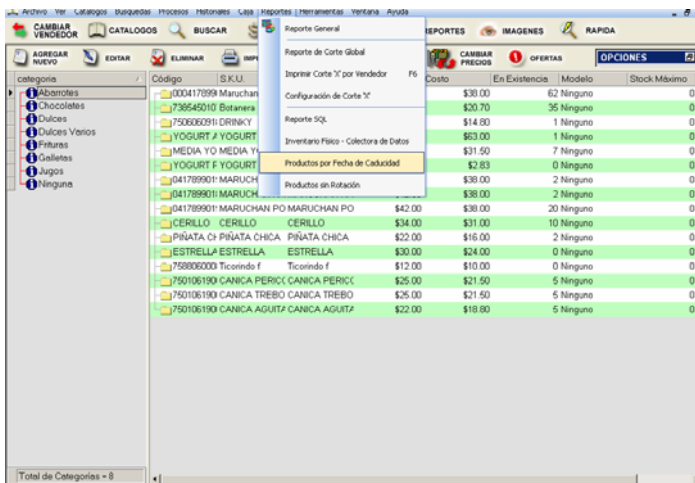


Ir al menú, agregar nuevo producto, poner la descripción de éste con los siguientes campos, donde es de suma importancia el campo de Fecha de caducidad ya que el sistema la toma en cuenta para generar los reportes de productos próximos a caducar.



## Generación de reportes

Existen dos formas de generar reportes de productos: los más vendidos o los próximos a caducar. El Usuario- Administrador puede generarlos de manera manual en base a la fecha de caducidad de cada producto, que se obtuvo en el pedido. Los pasos a seguir son: Ingresar al menú, Reportes, donde despliega la opción de generar reportes de productos próximos a caducar.



Este reporte puede ser obtenido dentro de un rango de fechas que el usuario asigne, una vez obtenido puede ser exportado a dos tipos de formatos: texto o archivos de Microsoft Excel.

Archivo Ver Catálogos Búsquedas Procesos Historiales Caja Reportes Herramientas Ventana Ayuda

CAMBIAR VENDEDOR CATALOGOS BUSCAR VENDER COMPRAR REPORTES IMAGENES RAPIDA

IMPRIMIR EXPORTAR OPCIONES

Productos por Fecha de Caducidad

ARRASTRE AQUÍ UNA COLUMNA PARA AGRUPAR

Fecha de C...	Código	Descripción	Número de ...	Cantidad	Status	No. del pro...	Proveedor
25/11/2009	00041789901...	Maruchan Piquin	00041789901...	\$1.00	COMPRADO	0	N/A
26/11/2009	00041789901...	Maruchan Piquin	no parte	\$20.00	COMPRADO	0	N/A
30/11/2009	7506060918086	DRINKY	2	\$1.00	COMPRADO	0	N/A

Cont=3 Cont=3 Total= \$22

El otro reporte que se puede realizar es, cuando el sistema lo genere en un día y hora determinados por el administrador de forma automática, por lo que este proceso se estará ejecutando de manera periódica en base a la fecha de caducidad que se le dio en la entrada de productos.

## Módulo de ventas.

El módulo de Ventas del SISCOIND, donde se pueden realizar diferentes operaciones de punto de venta para productos que se tienen en la dulcería, donde se muestra el identificador de cada producto, la descripción, precio y total de cada uno, como se muestra en la siguiente figura:

# de Ctrl./Venta: 1500    Tipo: Ticket    Fecha: Martes, 01 de Diciembre de 2009    Hora: 04:16:36 p.m.

**VENTA - DATOS DEL CLIENTE : - CLIENTE DE MOSTRADOR**

Código	Descripción	Medida	Precio	Cant.	Desc.	P.U.	Importe	Cac
AGRANEL	AGRANEL	unidad.	12.00	1	0	12.00	12.00	1980
AGRANEL CHIL	AGRANEL CHILE	unidad.	12.00	1	0	12.00	12.00	1980
AGRANEL CHIL	AGRANEL CHILE	unidad.	12.00	1	0	12.00	12.00	1980

**TOTAL**

IMPORTE BRUTO: \$ 31.29

DESCUENTOS: 0 % \$ 0.00

SUBTOTAL: \$ 31.29

RETENCIONES: 0 % \$ 0.00

IMPUESTOS: \$ 4.71

I.E.P.S. 0.00 % Precio Base IVA 15

I.V.B.S.S. 0.00 % Precio Base

I.V.A. 15.00 % Precio Base

IMPRIMIR

Incluye el impuesto

Serie: 2    Número de Folio: 1    Búsqueda Fonética:

**TOTAL : \$ 36.00**

**CANT. CON LETRA:**

(TREINTA Y SEIS PESOS 00/100 M.N.)

EXPORTAR

PROCESAR VENTA

CERRAR