



Agradecimientos

Alejandro Rodríguez Barba

“Para ser grande primero se tiene que aprender a ser pequeño” A.R.B

A Dios:

*Por darme salud, amor, esperanza, fe, sabiduría
Y especialmente porque día con día me enseña que el es, el camino la verdad y la vida.*

A mis abuelos:

*Alejo Rodríguez Vera y Teresa Casares Miranda
Por el gran amor y apoyo que me han dado en toda mi vida.*

A mis padres:

*Rosendo Rodríguez Casares y Silvia Barba Villalpando.
A quienes me han heredado el tesoro mas valioso que pudiera dársele a un hijo, amor, a quienes sin escatimar esfuerzo alguno han sacrificado gran parte de su vida para formarme y educarme, a quienes la ilusión de su vida ha sido convertirme en una persona de provecho, a quienes nunca podre pagar todos sus desvelos ni aun con la riqueza más grande del mundo por eso y más muchas gracias.*

A mi Futura Esposa:

*Monserrat Maldonado Velázquez.
Por ser mi bendición de cada día, mi apoyo incondicional, mi motivación, mi inspiración, mi alegría, mi felicidad muchas gracias, te amo muchísimo más que a mi vida.*

A mis hermanos:

*Fernando Rodríguez Barba y Teresa Rodríguez Barba
Por todos esos pequeños detalles que me han motivado en mis 24 años de vida.*

Al Doctor:

*Adrian Trueba Espinosa
Por todo el apoyo, paciencia, dedicación y conocimiento que me brindo en todo mi desarrollo como profesionista muchas gracias*

Al Licenciado:

Ángel Rafael Quintos Ramírez por brindarme todo el apoyo y conocimiento pero principalmente por ser mi amigo en todo este tiempo de mi formación profesional muchas gracias

A la Familia Maldonado Velázquez.

Por la comprensión, cariño y apoyo que me han brindado en esta importante etapa de mi vida.



Agradecimientos

Montserrat Maldonado Velázquez

“Es fácil, si puedo lo voy a lograr”

A Dios

Por la segunda oportunidad que me ha brindado para seguir adelante, porque día con día me da la vida, las fuerzas y la motivación para seguir sin rendirme, luchar contra todo y contra todos para lograr mis sueños; por que nunca me ha dejado sola y siempre me escucha.

A mis Padres y hermanos

Porque a lo largo de mi vida me han enseñando muchas cosas importantes por todo su apoyo y confianza que me han brindado, porque me dieron la oportunidad de tener una carrera y están conmigo en todos los momentos difíciles que he tenido muchas gracias.

A mi Abuelita Ma. Del Carmen Magos Mejía

Por ser mi modelo a seguir, por ser tan maravillosa, por todo el apoyo que me ha dado en toda mi vida, porque nunca ha dejado de creer en mí y me ha enseñado que cuando las cosas se hacen con amor se hacen bien muchas gracias

A mi Futuro Esposo Alejandro Rodríguez Barba

Por que mas que nunca y que nadie antes, me ha amado por sobre todas las cosas ante toda adversidad, logro, tropiezo, éxito o fracaso, siempre en todo momento él estuvo conmigo, me impulso, ayudó, levanto y en ocasiones me bajaba de mi nube o al contrario, me recompensaba como nunca nadie lo había hecho pero principalmente, confió en mí y logramos el juramento que hace algunos años hicimos, “terminar la carrera juntos”, en verdad y con todo mi corazón y todo mi amor te digo gracias mi cielo.

A la Familia Rodríguez Barba

Porque desde hace algunos años me acogieron en sus corazones, me brindaron amor, esperanza, cariño, comprensión, apoyo y me hacen sentir que soy parte de ellos, gracias por todos esos momentos tan lindos y por todo el apoyo que me han brindado muchas gracias.

Al Dr. Adrian Trueba Espinosa.

Porque en todo momento nos apoyo mucho, siempre ha sido y será un muy excelente profesor al cual admiramos mucho y respetamos.



Contenido

RESUMEN	1
I. INTRODUCCIÓN.....	2
II. PLANTEAMIENTO DEL PROBLEMA.....	3
III. JUSTIFICACIÓN.....	4
IV. OBJETIVOS	5
4.1 Objetivo General.....	5
4.2 Objetivos específicos.	5
V. MARCO TEORICO	6
5.1 UML	6
5.1.1 Definición.....	6
5.1.2 Introducción	6
5.1.3 Necesidad del UML	7
5.1.4 Concepción del UML	8
5.1.5 Diagramas UML.....	9
5.1.5.1 Diagrama de clases.....	9
5.1.5.2 Diagrama de objetos	10
5.1.5.3 Casos de uso	10
5.1.5.3.1 Importancia de los casos de uso	11
5.1.5.4 Diagrama de Casos de Uso	11
5.1.5.4.1 Representación de un modelo de Caso de uso.....	12
5.1.5.4.2 Secuencia en los escenarios.....	13
5.1.5.4.3 Relaciones entre casos de uso	13
5.1.5.4.3.1 Include	13
5.1.5.4.3.2 Extensión.....	14
5.1.5.4.3.3 Generalización.....	14
5.1.5.4.3.4 Agrupamiento	15
5.1.5.4.4 DIAGRAMAS DE CASOS DE USO EN EL PROCESO DE ANALISIS.	15
5.1.5.5 Diagramas de Estados	16
5.1.5.5.1 Sucesos y Acciones	17
5.1.5.5.2 Subestados.....	17
5.1.5.5.3 Importancia del diagrama de Estados.....	18



5.1.5.6	Diagrama de Secuencias.....	18
5.1.5.6.1	Objetos	18
5.1.5.6.2	Mensajes.....	19
5.1.5.6.3	Tiempo.....	19
5.1.5.7	Diagramas de colaboraciones.....	20
5.1.5.7.1	Ejemplo de diagrama de colaboraciones (Interfaz Grafica del Usuario) ...	21
5.1.5.8	Diagrama de Actividades.....	22
5.1.5.8.1	Decisiones	23
5.1.5.8.2	Rutas concurrentes e indicaciones.....	23
5.1.5.9	Diagrama de componentes.....	25
5.1.5.9.1	Tipos de componentes	25
5.2	SISTEMAS DE INFORMACION.....	26
5.2.1	Una tipología de Sistemas	26
5.2.1.1	¿Qué es un sistema?.....	28
5.2.2	Frontera y Entorno.....	29
5.2.3	Entrada, Salida, Interfaz Y Transformación	30
5.2.4	Subsistemas.....	32
5.2.5	Características de un sistema	33
5.2.5.1	Control del sistema	34
5.2.5.2	Retroalimentación.....	34
5.2.5.3	La retroalimentación hacia adelante	34
5.2.5.3.1	Concepto de Retroalimentación hacia adelante.....	34
5.2.5.3.2	Propiedades emergentes.....	35
5.2.6	Información y Sistemas de Información.....	35
5.2.6.1	¿Qué es información?	35
5.2.6.2	¿Qué es un sistema de Información?.....	35
5.2.6.3	Ciclo de vida del desarrollo de sistemas.....	36
5.2.6.4	Investigación Preliminar.	37
5.2.7	Las cuatro etapas del proceso.	38
5.2.7.1	Importancia de las herramientas asistidas por computadora en el desarrollo de sistemas.	38
5.2.7.1.1	Beneficios del empleo de herramientas.....	38
5.2.7.1.2	Beneficios de las herramientas asistidas por computadora	39



5.2.7.1.3	Clasificación de herramientas automatizadas	39
5.2.8	Tipos de sistemas de información.....	40
5.2.9	Modelos de sistemas de información	41
5.2.9.1	Modelo en cascada.	41
5.2.10	Modelo Incremental	41
5.2.11	Modelo de Prototipos	42
5.2.11.1	Efectividad	42
5.2.11.2	Desventajas del Modelo de prototipos	43
5.2.11.3	El prototipo evolutivo.....	43
5.2.12	Modelo en Espiral	44
5.2.12.1	Diferencias entre modelo en espiral y modelos tradicionales.....	44
5.3	BASES DE DATOS.....	45
5.3.1	Antecedentes.....	45
5.3.2	Definiciones	47
5.3.2.1	Dato.....	47
5.3.2.2	Información	47
5.3.2.3	Campo	47
5.3.2.4	Registro.....	48
5.3.2.5	Archivo.....	48
5.3.2.6	Data Warehouse	48
5.3.2.7	Base de datos.....	48
5.3.2.8	Sistema Manejador de Base de Datos. (DBMS).....	48
5.3.2.9	Esquema de base de datos	49
5.3.2.10	Administrador de base de datos (DBA)	49
5.3.3	Objetivos de las Bases de Datos	49
5.3.3.1	Consultas no predefinidas y complejas.....	49
5.3.3.2	Flexibilidad e independencia	49
5.3.3.3	Redundancia e inconsistencia de datos.	50
5.3.3.4	Integridad de los datos.....	50
5.3.3.5	Concurrencia de los usuarios.....	51
5.3.3.6	Seguridad.....	51
5.3.3.7	Otros objetivos.....	51
5.3.4	Sistema Administrador de Bases de Datos.....	52



5.3.4.1	Usos y funciones DBMS.....	52
5.3.5	Modelo de Bases de Datos.....	52
5.3.5.1	Modelo Relacional.....	53
5.3.5.1.1	Estructura de los Datos.....	54
5.3.5.1.1.1	Características y Definición de una Tabla.....	55
5.3.5.1.1.1.1	Tupla.....	55
5.3.5.1.1.1.2	Dominios.....	55
5.3.5.1.1.1.3	Atributo.....	56
5.3.5.1.1.2	Llaves Primarias.....	56
5.3.5.1.1.3	Llaves Secundarias.....	56
5.3.5.2	Modelo Entidad-Relación.....	56
5.3.5.2.1	Entidad.....	57
5.3.5.2.2	Relaciones.....	57
5.3.5.2.3	Atributos.....	57
5.3.5.2.4	Notación del Diagrama Entidad-Relación.....	58
5.3.5.2.5	Cardinalidad.....	58
5.3.5.2.5.1	Uno a uno.....	58
5.3.5.2.5.2	Uno a Muchos.....	59
5.3.5.2.5.3	Muchos a Muchos.....	59
5.3.6	Normalización de Bases de Datos.....	59
5.3.6.1	Concepto de Normalización.....	59
5.3.6.2	Primera Forma Normal (1NF).....	60
5.3.6.3	Segunda Forma Normal (2NF).....	60
5.3.6.4	Tercera Forma Normal (3NF).....	60
5.4	SQL.....	61
5.4.1	Introducción.....	61
5.4.2	Comandos SQL.....	61
5.4.2.1	Comandos DDL.....	61
5.4.2.2	Comandos DML.....	62
5.4.3	Clausulas.....	62
5.4.4	Operadores Lógicos.....	62
5.4.5	Operadores de Comparación.....	63
5.4.6	Funciones de Agregado.....	63



5.4.7	Consultas.....	64
5.4.7.1	Consultas Básicas.....	64
5.4.7.2	Ordenar Registros.....	64
5.4.7.3	Consultas con Predicado.....	65
5.4.7.4	La clausula where.....	65
5.4.8	Consultas de acción.....	65
5.4.8.1	Delete.....	65
5.4.8.2	Insert into.....	66
5.4.8.3	UPDATE.....	66
5.4.9	Instrucción SQL.....	67
VI.	METODOLOGIA.....	67
6.1	Pasos metodológicos.....	67
6.2	Desarrollo de pasos metodológicos.....	68
6.3	Herramientas.....	68
6.4	Casos de Uso en SELLSYS1.0.....	70
6.4.1	Caso de Uso Buscar registro.....	70
6.4.2	Caso de Uso dar de Alta un registro.....	71
6.4.3	Caso de Uso Cambiar un registro.....	71
6.4.4	Caso de uso Eliminar un registro.....	72
6.4.5	Caso de uso Hacer un reporte.....	73
6.5	Diseño de la Base de Datos.....	74
6.5.1	Primera Forma Normal SELLSYS1.0.....	75
6.5.2	Segunda Forma Normal SELLSYS1.0.....	76
6.5.3	Tercera Forma Normal SELLSYS1.0.....	78
6.5.4	Modelo Relacional de SELLSYS1.0.....	80
6.5.5	Conexión con SQL.....	82
6.6	Programación de SELLSYS1.0.....	82
VII.	RESULTADOS.....	82
VIII.	DISCUSION.....	96
IX.	CONCLUSIONES.....	97
X.	BIBLIOGRAFIA.....	98



INDICE DE FIGURAS

Figura 1	Diagrama de una clase-----	10
Figura 2	Diagrama de un objeto-----	10
Figura 3	Modelo de caso de uso.-----	12
Figura 4	Función include-----	13
Figura 5	Relación que tienen los casos de uso con la herramienta extensión -----	14
Figura 6	Generalización de un caso de uso -----	15
Figura 7	Partes del Diagrama de Estados-----	16
Figura 8	Diagrama de estados que captura los estados y las transiciones en la GUI. -----	17
Figura 9	Objeto en Diagrama de secuencias-----	18
Figura 10	Conjunto básico de símbolos del Diagrama de secuencias -----	19
Figura 11	Diagrama de secuencias-----	20
Figura 12	Simbología del diagrama de colaboraciones. -----	21
Figura 13	Forma de representar una secuencia de operaciones. -----	22
Figura 14	Punto inicial y final, así como dos actividades y una transición. -----	22
Figura 15	Formas de tomar una decisión. -----	23
Figura 16	Ejemplo de rutas concurrentes al momento de tomar una decisión. -----	24
Figura 17	Muestra una indicación en un Diagrama de actividades. -----	24
Figura 18	Diagrama de componentes enfocado en un sistema de información. -----	26
Figura 19	Mapa de Sistemas -----	28
Figura 20	Frontera y el entorno de un sistema -----	29
Figura 21	Vista del funcionamiento de la caja negra de un sistema-----	31
Figura 22	Sistemas Cerrados y Abiertos. -----	31
Figura 23	Subsistema -----	32
Figura 24	Acoplamiento de subsistemas -----	33
Figura 25	Ciclo de vida del desarrollo de sistemas -----	36
Figura 26	Clasificación de herramientas automatizadas -----	39
Figura 27	Modelo de la Metodología en Cascada. -----	41
Figura 28	Modelo de la Metodología Incremental-----	42
Figura 29	Modelo de la Metodología de Prototipos -----	44
Figura 30	Modelo de la Metodología en Espiral -----	45
Figura 31	Modelo de Bases de Datos. [28] -----	53
Figura 32	Notación Básica del Modelo Entidad-Relación-----	58
Figura 33	Cardinalidad de Uno a Uno -----	58
Figura 34	Relación de Uno a Muchos -----	59
Figura 35	Relación de Muchos a Muchos-----	59
Figura 36	Caso de uso Buscar-----	70
Figura 37	Caso de uso Alta -----	71
Figura 38	Caso de Uso Cambios. -----	72
Figura 39	Caso de uso Eliminar. -----	73
Figura 40	Caso de uso Reportes. -----	74
Figura 41	Datos requeridos para SELLSYS1.0 -----	75
Figura 42	Esquema de bases de datos de SELLSYS1.0 -----	80
Figura 43	Tablas en MySQL -----	81



Figura 44	Agregar Registros en MySQL.	81
Figura 45	Diseño de las interfaces en Visual Basic.	82
Figura 46	interfaz de contraseña del sistema.....	83
Figura 47	Interfaz de Selección desde el menú principal.	83
Figura 48	Toolbar del Sistema Sellsys1.0.	83
Figura 49	Interfaz para dar de alta un Registro.	84
Figura 50	Interfaz que se utiliza para buscar un registro.	84
Figura 51	Interfaz que muestra como cambiar un registro.	85
Figura 52	Interfaz que muestra cómo eliminar un registro.	85
Figura 53	Interfaz que muestra como limpiar los registros.	86
Figura 54	Interfaz que muestra la automatización del sistema.	86
Figura 55	Interfaz que muestra como se registran en la tabla distribuidores.	87
Figura 56	Interfaz que muestra el estado de cuenta general.	87
Figura 57	Interfaz que muestra el estado de cuenta particular.	88
Figura 58	Interfaz que muestra el resultado de facturas.	88
Figura 59	Interfaz de la tabla Fullerettes..	89
Figura 60	Interfaz de la tabla ofertas.	89
Figura 61	Interfaz de la tabla gerente.	90
Figura 62	Interfaz que muestra cómo funciona la tabla de papelería.	90
Figura 63	Interfaz de la tabla producto.	91
Figura 64	Interfaz de la tabla puntos Fuller.	91
Figura 65	Interfaz de la tabla pedido.	92
Figura 66	Interfaz que hace referencia a los reportes sin fechas definidas.	93
Figura 67	Interfaz que muestra la tabla de reportes con consultas en fechas determinadas.	94
Figura 68	Interfaz que muestra los datos del DATAGRID	94
Figura 69	Interfaz que muestra el reporte de productos.	95
Figura 70	Interfaz que muestra la utilidad de las fechas en los reportes.	95

INDICE DE CUADROS

Cuadro 1	Comparación de Sistemas de Información [11, 12, 14, 16, 17, 19,20].....	40
Cuadro 2	Estructura Básica del Modelo Relacional, tabla estudiante.	55



RESUMEN

Los integrantes de la empresa Fuller Cosmetics hacen sus reportes y asignación de funciones de forma manual, eso tiene como consecuencia una inversión de tiempo que se podría emplear en aumentar el número de clientes y por ende las ventas y las ganancias para vendedoras, gerentes y la propia empresa.

En este trabajo se da una alternativa a esta problemática, el sistema sellsys1.0 es capaz de dar de alta registros de todas las personas que interactúan en la empresa Fuller Cosmetics, así como buscarlos en la base de datos, si es necesario puede cambiar algún dato que ya no se necesite, además es capaz de eliminar registros. El sistema también proporciona reportes con datos específicos que son de utilidad para el personal de la empresa como lo son; las ventas en una campaña, todas las persona que están registradas en la empresa, los de estados de cuenta de los clientes, las ofertas y promociones que hay en determinado tiempo entre otros reportes.



I. INTRODUCCIÓN

El proceso de ventas que se lleva a cabo en la empresa Fuller Cosmetics consiste básicamente en la venta directa de artículos cosméticos, de perfumería y artículos para el hogar, los cuales manufactura con la máxima calidad, innovación y avances tecnológicos; de esta manera brinda a miles de mujeres la oportunidad de tener su propio negocio, formando parte de las Fullernettes de la empresa, es decir de comisionistas mercantiles independientes.

Dentro de este proceso de ventas existen cargos y comisiones a lo largo de la empresa, los cuales son los puntos clave de la existencia misma del flujo de ventas. Dentro de estos cargos esta el principal que es el que lleva los productos hasta el consumidor final que son las Fullernettes, siguiendo hacia adentro con la coordinadora empresarial que tiene la función más importante del proceso de ventas ya que funciona como intermediario entre las Fullernettes y la empresa. La coordinadora se encarga de la entrega y cobro de pedidos a Fullernettes así como de la transmisión a la gerente de las ventas que se realizaron en una campaña que se compone de 14 días; posteriormente tenemos a la gerente de ventas quien tiene a su cargo a todas las coordinadoras empresariales de su distrito la cual recibe toda la información de las ventas realizadas en cada campaña y la transmite hacia la empresa para poder surtir los pedidos realizados.

Fuller Cosmetics maneja su estrategia de ventas de manera seccionada geográficamente, es decir un estado lo divide en distritos dirigidos por un director, estos a su vez están particionados en divisiones organizados por las gerentes que tienen a su cargo varias zonas que controlan las coordinadoras

Empresariales; con esta organización la empresa cubre cada sector de la población para poder obtener mayores ganancias con sus ventas.



Las ventas son uno de los procesos más importantes de las empresas y sus datos deben de estar actualizados diariamente para una buena toma de decisiones pero en la empresa Fuller Cosmetics carece de herramientas para las coordinadoras empresariales en su manejo de datos.

La finalidad de este trabajo es desarrollar un sistema de información que permita lograr una mejor administración, mayor rapidez, eficiencia, calidad, exactitud y eficacia de los datos de las ventas realizadas cada campaña que realicen las coordinadoras empresariales.

II. PLANTEAMIENTO DEL PROBLEMA

En la empresa Fuller Cosmetics hay varios roles laborales, de ellos destaca el de las coordinadoras empresariales, que son parte substancial del flujo de ventas de la empresa; ellas llevan el control de ventas de forma manual y por obligación la vacían en hojas de papel. Este Sistema requiere de mucho tiempo y esfuerzo, corriendo el riesgo de presentar inconsistencia en la información a causa de errores humanos. Esto por la gran cantidad de datos que se manejan diariamente. Se han detectado los siguientes problemas:

- Inconsistencia en los registro de pagos.
- Fugas de capital en la cobranza.
- Falta de organización en el trabajo diario (Cubrir todos los sectores de la zona).
- Falla en el registro de productos.

Considérese que cada coordinadora empresarial tiene a su cargo varios sectores lo cual hace que el trabajo manual se multiplique de acuerdo a cada sector. Cada falla al registro trae como consecuencias pérdidas económicas que



merman las ganancias de la empresa así como de las coordinadoras y por ende tiempo de operación.

III. JUSTIFICACIÓN

Diseñar y programar un sistema de información que ayude a las tareas de la coordinadora empresarial en el tratamiento de los datos de las ventas, permitirá proporcionar datos actualizados diariamente de forma clara y precisa, así como llevar un control de la información para poder ser transmitida a los niveles superiores de la organización, al igual que sirve de apoyo a la coordinadora en las tareas contables que se le solicitan como parte de su reporte.

Con el sistema la coordinadora empresarial va a reducir sus tiempos en papeleo y aumentar su productividad para beneficio de su trabajo, así como, de la empresa reflejándose en las ventas realizadas y las ganancias para ambas partes, ya que el sistema va a ser capaz de proporcionar datos procesados, almacenados y organizados de tal manera que la coordinadora solo se dedique a ingresarlos al sistema, los beneficios se resumen en:

- Proporcionar reportes e informes en tiempo y forma.
- Organizar diariamente el trabajo de la coordinadora por sectores.
- Procesamiento de captura de pedidos más efectiva
- Reducir tiempo de trabajo
- Optimizar tareas contables y administrativas de la coordinadora
- Actualización de los datos diariamente.
- Mayores ganancias para la empresa
- Mayores ganancias para la coordinadora



- Mayores ganancias para las vendedoras

IV. OBJETIVOS

4.1 Objetivo General

- Desarrollar un sistema de información para el control de ventas de cosméticos.

4.2 Objetivos específicos.

- Modelar el sistema de información en UML.
- Analizar y diseñar la base de datos.
- Planteamiento de algoritmos del gestor de Bases de datos.
- Programación de algoritmos.
- Desarrollo de interfaces.



V. MARCO TEORICO

5.1 UML

5.1.1 Definición

UML (Lenguaje Unificado de Modelado) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. En, UML se utiliza para definir un sistema de software. [5]

5.1.2 Introducción

UML (Lenguaje Unificado de Modelado) es una de las herramientas más útiles para el desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que proyecten sus ideas en una forma convencional y fácil de comprender para comunicarse con otras personas. [7]

La comunicación de la idea es de suma importancia, antes del advenimiento del UML, el desarrollo de sistemas era con frecuencia, una propuesta al azar. Los analistas de sistemas evaluaban los requerimientos de los clientes con en algún tipo de notación que ellos mismos comprendieran (Aunque el cliente no lo comprendiera), entregar tal notación a uno o varios programadores y esperar que el producto final cumpliera con lo que el cliente deseaba. [9]

Dado que el desarrollo de sistemas es una actividad humana, hay muchas posibilidades de cometer errores en cualquier etapa del proceso, por ejemplo, el analista pudo haber malentendido al cliente, es decir, probablemente produjo un documento que el cliente no pudo comprender. Tal vez ese documento tampoco fue comprendido por los programadores, quienes por ende, pudieron generar un programa difícil de utilizar y no generar una solución al problema original del cliente; por esta razón surge la necesidad de crear una herramienta de diseño como lo es UML. [1]



5.1.3 Necesidad del UML

En los principios de la computación, los programadores no realizaban análisis muy profundo sobre el problema por resolver. Con frecuencia comenzaban a escribir el programa desde el principio, y el código necesario se escribía conforme se requería. Aunque esta metodología proporcionaba un aura de atrevimiento al desarrollo, en la actualidad es inapropiado en el desarrollo de software.

Para el desarrollo de software es necesario contar con un plan bien analizado. Un cliente tiene que entender que es lo que hará un equipo de desarrolladores; además tiene que ser capaz de señalar cambios, si no se han captado claramente en las necesidades del sistema (o si cambia de opinión durante el proceso). A su vez, el desarrollo es un esfuerzo orientado a equipos de trabajo, por lo que cada uno de sus miembros tiene que saber qué lugar toma en la solución final (así como saber cuál es la solución en general). [3]

Conforme aumenta la globalización la complejidad de los negocios aumenta, los sistemas informáticos también crecen en complejidad. En ellos se encuentran diversas piezas de hardware y software que se comunican a grandes distancias mediante una red, misma que está vinculada a bases de datos, a su vez, contienen grandes volúmenes de información.

La clave para poder manejar la complejidad de los sistemas radica en organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y lo convengan, con el UML se facilita la organización y comprensión de los sistemas de información.

Un arquitecto no podría crear una compleja estructura de un edificio de oficinas, sin crear primero un anteproyecto detallado; asimismo no se podría generar un complejo sistema en un edificio de oficinas, sin crear un plan de diseño detallado. La idea es que así como un arquitecto hace un anteproyecto para mostrárselo a la persona que lo contrata. Como desarrolladores de sistemas debemos de tener un plan de diseño que contenga un cuidadoso análisis de las necesidades del cliente. [3]

Otra característica del desarrollo de sistemas contemporáneo es reducir el periodo de desarrollo. Cuando los plazos se encuentran muy cerca uno del otro es absolutamente necesario contar con un diseño sólido.



Hay otro aspecto de la vida moderna que demanda un diseño sólido; las adquisiciones corporativas. Cuando una empresa adquiere a otra, la nueva organización debe tener la posibilidad de modificar aspectos importantes de un proyecto de desarrollo que este en progreso (la herramienta de desarrollo, el lenguaje de codificación, y otras cosas). Un anteproyecto bien diseñado facilitara la conversión. Si el diseño es sólido, un cambio en la implementación procederá sin problemas [3]

La necesidad de diseños sólidos ha traído consigo la creación de una notación de diseño estandarizada, que los analistas, desarrolladores y clientes acepten como pauta y guía de desarrollo de sistemas. El UML es esa notación. [3]

5.1.4 Concepción del UML

El UML es la creación de Grady Booch, James Rumbaugh e Ivar Jacobson. Estos tres desarrolladores trabajaban en empresas distintas durante la década de los ochenta y principios de los noventa y cada uno desarrollo su propia metodología para el análisis y diseño orientado a objetos. Sus metodologías predominaron sobre las de sus competidores. A mediados de los años noventa empezaron a intercambiar ideas entre si y decidieron realizar un trabajo en conjunto.

En 1994 Rumbaugh ingreso a Rational software corporation, donde ya trabajaba Booch, Jacobson ingreso a Rational un año después, el resto es historia.

Los anteproyectos del UML empezaron a circular en la industria del software y las reacciones resultantes trajeron consigo considerables modificaciones. Conforme diversos corporativos vieron al UML como una herramienta útil a sus propósitos, se formo un consorcio del UML. Entre los miembros se encuentran DEC, Hewlett Packard, intellicorp, Microsoft, Oracle, Texas Instruments y Rational. En 1997 el consorcio produjo la versión 1.0 del UML y lo puso a la consideración del OMG (Grupo de Administración de Proyectos) como respuesta a su propuesta para un lenguaje de modelado estándar. [5]

El consorcio aumento y genero la versión 1.1, misma que se puso nuevamente a consideración del OMG. El grupo adopto esta versión a finales de 1997. El OMG se encargo de la conservación del UML y produjo otras dos revisiones en 1998. El UML ha llegado a ser el estándar de facto en la industria del software y su evolución continua. [5]



5.1.5 Diagramas UML

UML está compuesto por elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. El modelo UML de un sistema es similar a un modelo a escala de un edificio junto con la interpretación del artista del edificio. Es importante destacar que un modelo UML describe lo que hará un sistema, pero no dice como implementar dicho sistema.

Un método es un proceso disciplinado para generar un conjunto de modelos que describen varios aspectos de un sistema de software en desarrollo, utilizando alguna notación bien definida. [4]

5.1.5.1 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro. [2]

Una clase es una categoría o grupo de cosas que tienen atributos o acciones similares. Cualquier cosa dentro de la clase, por ejemplo en la clase “Usuarios” tienen atributos como lo son id, nombre, fecha de nacimiento, e-mail etc. Entre las acciones de las cosas de esta clase se encuentran “registro”, “password”, “activarse” y “mi cuenta”

La Figura 1.1 muestra un ejemplo de cómo se utiliza el diagrama de clases en UML que captura los atributos y acciones de una clase llamada “usuarios”. Un rectángulo es un símbolo que representa a la clase y se divide en tres áreas. El área superior contiene el nombre, el área central contiene los atributos, y el área inferior las acciones. Un diagrama de clases está formado por varios rectángulos de este tipo conectados por líneas que muestran la manera en que las clases se relacionan entre sí. [2]

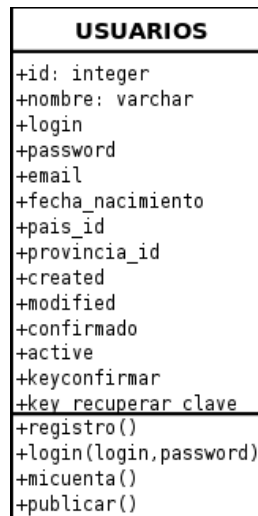


Figura 1 Diagrama de una clase

5.1.5.2 Diagrama de objetos

Un objeto es una instancia de clase (Una entidad que tiene valores específicos de los atributos y de las acciones). Por ejemplo un teléfono celular podría tener la marca Ericsson, el modelo z530, el número de serie g1346326 y una capacidad de 512MB.

La figura 2 muestra la forma en que UML representa un objeto. El símbolo que se utiliza es un rectángulo, como en una clase, pero el nombre esta subrayado. El nombre de la instancia específica que se encuentra ala izquierda de los dos puntos (:), y el nombre de la clase ala derecha. [6]

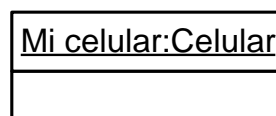


Figura 2 Diagrama de un objeto

5.1.5.3 Casos de uso

Las ideas estáticas ayudan a que un analista se comunique con un cliente. La idea dinámica ayuda al analista a comunicarse con un grupo de desarrolladores, y ayudara a estos a crear programas. El cliente y los desarrolladores forman parte fundamental en un sistema, sin embargo una parte fundamental para el sistema es el usuario. Ni la idea estática ni la idea



dinámica mostraran el comportamiento desde el punto de vista del usuario. Comprender tal punto de vista es clave para generar sistemas que sean tanto útiles como funcionales; esto significa que cumplan con los requerimientos y que sea fácil trabajar con ellos.

Al referirse a un caso de uso se habla de los requerimientos que se necesitan para poder generar un sistema que cumpla las expectativas que se tienen. Este tipo de análisis es particularmente crucial para la fase de análisis del desarrollo de un sistema. La forma en que los usuarios utilicen un sistema da la pauta para quien lo diseñara y creara. [5]

El caso de uso es una estructura que ayuda a los analistas a trabajar con los usuarios para determinar la forma en que se usará un sistema. Con una colección de casos de uso se puede hacer el bosquejo de un sistema en términos de lo que los usuarios intenten hacer con él.

5.1.5.3.1 Importancia de los casos de uso

El caso de uso es una excelente herramienta para estimular a que los usuarios potenciales hablen de un sistema, desde sus propios puntos de vista. La idea es involucrar a los usuarios en las etapas iniciales del análisis y diseño del sistema. Esto aumenta la probabilidad de que el sistema sea de mayor provecho para la gente que supuestamente ayudara, en lugar de ser un recurso computacional incomprensibles e inmanejables por los usuarios finales. [4]

5.1.5.4 Diagrama de Casos de Uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores del sistema, esta es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos de un sistema desde el punto de vista del usuario. Esto es importante si la finalidad es crear un sistema que pueda ser utilizado por la gente en general (no solo por expertos en computación).

El caso de uso es un poderoso concepto que ayuda a un analista a comprender la forma en que un sistema deberá comportarse. Le ayuda a obtener los requerimientos desde el punto de vista del usuario. Esta visualización permite mostrar los casos de uso a los usuarios para que ellos puedan dar mayor información. Una representación visual ayuda a combinar los diagramas de casos de uso con otro tipo de diagramas. [5]

Una de las finalidades del proceso de análisis de un sistema es generar una colección de casos de uso. La idea es tener la posibilidad de catalogar y hacer una referencia a esta colección que sirve como el punto de vista de los usuarios acerca del sistema.

5.1.5.4.1 Representación de un modelo de Caso de uso

Hay un “Actor” que inicia un caso de uso y otro (posiblemente el que inicio, pero no necesariamente) que recibirá algo de valor del. La representación grafica es directa. Una elipse representa a un caso de uso, y el que recibe ala derecha. El nombre del actor aparece justo debajo de el, y el nombre del caso de uso aparece ya sea dentro de la elipse o justo debajo de ella. Una línea asociativa conecta a un actor con el caso de uso, y representa la comunicación entre el actor y el caso de uso.

Uno de los beneficios de los casos de uso es que muestra todo el sistema y el mundo exterior que actúa en el, generalmente, los actores están fuera del sistema mientras que los casos de uso están dentro del sistema. Un rectángulo (con el nombre del sistema en algún lugar dentro de él) para representar los límites del sistema. El rectángulo envuelve a los casos de uso del sistema. Los actores, casos de uso y líneas de interconexión componen un modelo de caso de uso. [5]

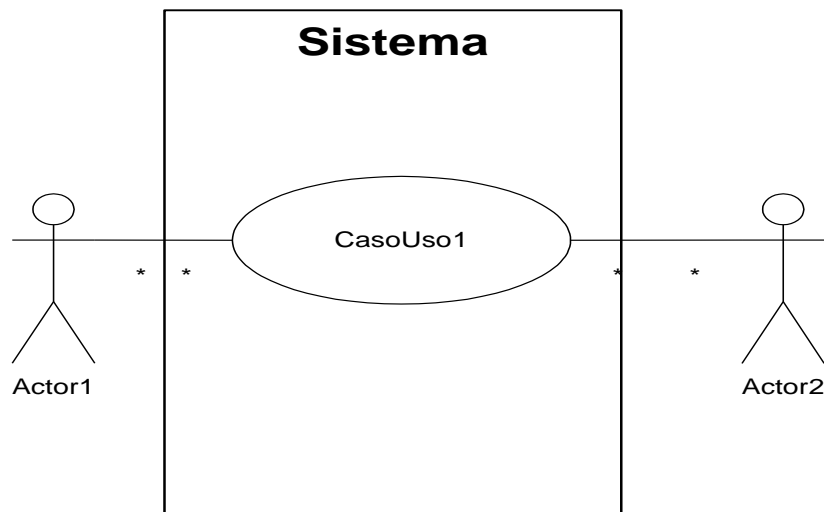


Figura 3 Modelo de caso de uso.

5.1.5.4.2 Secuencia en los escenarios

Cada caso de uso es una colección de escenarios y cada escenario es una secuencia de pasos. El uso de los diagramas de casos de uso será, por lo general, parte de un documento del diseño que el cliente y el equipo de diseño tomarán como referencia. Cada diagrama tendrá su propia página, de igual manera cada escenario de caso de uso tendrá su propia página, se listará en modo de texto a [2]:

- ✓ El actor que inicia el caso de uso
- ✓ Condiciones previas para el caso de uso
- ✓ Pasos en el escenario
- ✓ Condiciones posteriores cuando se finaliza el escenario.
- ✓ El actor se beneficia del caso de uso.

5.1.5.4.3 Relaciones entre casos de uso

5.1.5.4.3.1 *Include*

Se utiliza cuando se tiene una parte del comportamiento común a más de un caso de uso, y no se desea almacenar una copia de cada caso de uso de la descripción de este comportamiento.

Para representar la inclusión se utilizará el símbolo que se usó para la dependencia entre clases: una línea discontinua con una punta de flecha que conecta las clases apuntando hacia la clase independiente. Justo sobre la línea, agregará un estereotipo: la palabra “incluir” bordeada por dos pares de paréntesis angulares. [4]

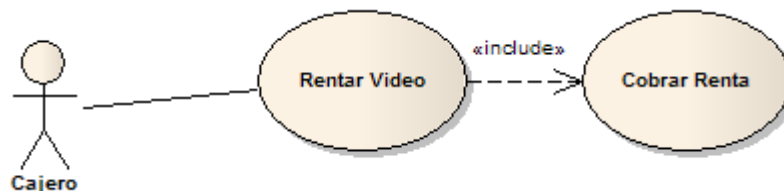


Figura 4 Función include

5.1.5.4.3.2 Extensión

En un caso de uso dado la función “extends” puede extender a otro caso de uso. Esta relación indica que el comportamiento del caso de uso extensión puede ser insertado bajo ciertas condiciones. La extensión solo se puede realizar en puntos indicados de manera específica dentro de la secuencia del caso de uso base. [4]

Como la inclusión, podrá concebir la extensión con una línea de dependencia (Línea discontinua con una punta de flecha), junto con un estereotipo que muestra “Extend” entre paréntesis angulares. Dentro del caso de uso básico, el punto de extensión aparecerá debajo del nombre del caso de uso.



Figura 5 Relación que tienen los casos de uso con la herramienta extensión

5.1.5.4.3.3 Generalización

Las clases pueden heredarse entre sí y esto también aplica a los casos de uso. En la herencia de los casos de uso, el caso de uso secundario hereda las acciones y significado del primario, y además agrega sus propias acciones. Puede aplicar el caso de uso secundario en cualquier lugar donde aplique el primario.

Entonces la Generalización es la actividad de identificar elementos en común entre conceptos y definir las relaciones de una superclase (concepto general) y subclase (concepto especializado). Es una manera de construir clasificaciones taxonómicas entre conceptos que entonces se representan en jerarquías de clases. Las subclases conceptuales conforman las superclases conceptuales en cuanto a la intensión y extensión [2]

En el siguiente ejemplo el caso de uso “Imprimir” que se hereda de “generar reporte”. Se modela la generalización de casos de uso de la misma forma que se hace con la de clases: con líneas continuas y una punta de flecha en forma de triangulo sin rellenar que apunta hacia el caso de uso primario como se muestra en la Figura 6.

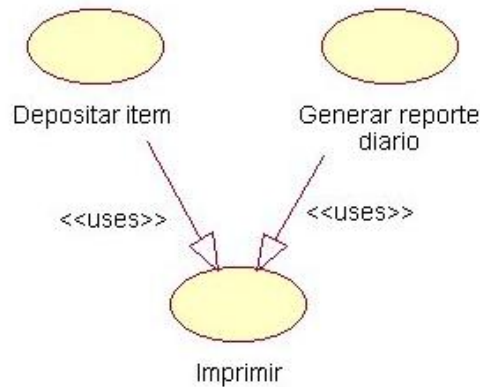


Figura 6 Generalización de un caso de uso

5.1.5.4.3.4 Agrupamiento

En ciertos diagramas de casos de usos, podría tener varios que querrá organizar. Esto puede ocurrir cuando un sistema cuenta con un sistema de subsistemas. Otra posibilidad sería cuando entrevista a los usuarios para obtener los requerimientos de un sistema. Cada requerimiento podría ser representado como un caso de uso por separado. Necesitara alguna forma de ordenar los requerimientos por categorías.

La forma más directa para organizar sería agrupar en un paquete los casos de uso que se relacionen. Recuerde que cada paquete aparece como una carpeta tubular. Los casos de usos agrupados aparecerán dentro de la carpeta. [6]

5.1.5.4.4 DIAGRAMAS DE CASOS DE USO EN EL PROCESO DE ANALISIS.

Las entrevistas con el cliente deberán iniciar el proceso. Estas entrevistas producirán diagramas de clases que fungirán como las bases de su conocimiento para el dominio del sistema (el área en el cual resolverá los problemas). Una vez que conozca la terminología general del área del cliente se está lista para hablar con los usuarios. [2]

Las entrevistas con los usuarios comienzan en la terminología del dominio, aunque deberán alternarse hacia la terminología de los usuarios. Los resultados iniciales de las entrevistas deberán revelar a los actores y a los casos de uso de alto nivel que describirán los requerimientos funcionales en términos generales. Esta información establece los confines y ámbito del sistema. [1]

Las entrevistas posteriores con los usuarios profundizaran en estos requerimientos, lo que dará por resultado modelos de caso de uso que mostrarán los escenarios y las secuencias detalladamente. Esto podría resultar en otros casos de usos que satisfagan las relaciones de inclusión y extensión. En esta fase, es importante confiar en la comprensión del dominio (a partir de los diagramas de clase derivados de las entrevistas con el cliente). Si no se comprende adecuadamente el dominio, podría crear demasiados casos de usos y demasiados detalles (situación que podría obstaculizar el diseño y el desarrollo) [8]

5.1.5.5 Diagramas de Estados

Una manera para caracterizar un cambio en un sistema es decir que los objetos que lo componen modificaron su estado como respuesta a los sucesos y al tiempo.

El diagrama de estados de UML captura los cambios que existan entre los objetos. Presenta los estados en los que puede encontrarse un objeto con las transiciones entre los estados y muestra los puntos inicial y final de una secuencia de cambios de estado. [5]

En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos.

Un estado se representa como una caja redondeada con el nombre del estado en su interior. Una transición se representa como una flecha desde el estado origen al estado destino. La caja de un estado puede tener 1 o 2 compartimentos. En el primer compartimento aparece el nombre del estado. El segundo compartimento es opcional, y en él pueden aparecer acciones de entrada, de salida y acciones internas. [6]

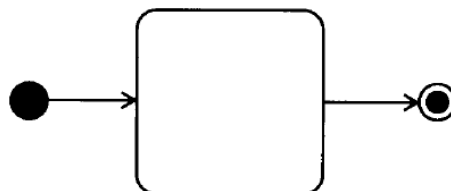


Figura 7 Partes del Diagrama de Estados

5.1.5.5.1 Sucesos y Acciones

También se pueden agregar ciertos detalles a las líneas de transición. Puede indicar un suceso que provoque una transición (desencadenar un suceso), y la actividad de computo (la acción) que se ejecute y haga que suceda la modificación del estado. Los sucesos y acciones los escribirá cerca de la línea de transición mediante una diagonal para separar un suceso desencadenado de la acción. En ocasiones un evento causara una transición sin una acción asociada y algunas veces alguna transición sucederá dado que un estado finalizara una actividad (en lugar de hacerlo por suceso). A este tipo de transición se le conoce como transición no desencadenada. La GUI (interfaz grafica del usuario) con que interactué le dará ejemplos de detalle de transición por el momento se asume que la GUI se puede establecer en uno de tres estados:

- Inicialización
- Operación
- Apagar

Por ejemplo cuando encienda su PC, se ejecutara un proceso de arranque; al encenderlo se desencadena un suceso que provoca que la GUI aparezca luego de una transición desde el estado de inicialización, y el arranque, es una acción que se realiza durante la transición. Como resultado de las actividades en el estado de inicialización, la GUI entra en modo de operación. Cuando desea apagar su PC, desencadena un suceso que provoca la transición hacia el estado de apagado, y con ello la PC se apaga. [6]

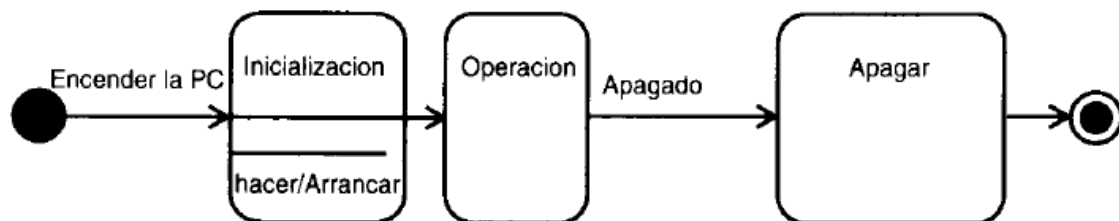


Figura 8 Diagrama de estados que captura los estados y las transiciones en la GUI.

5.1.5.5.2 Subestados

Los subestados son estados que se encuentran dentro de otros en el UML existen dos tipos de subestados para poder definir el sistema que son:

- Secuenciales: Como su nombre lo indican los subestados secuenciales suceden uno detrás del otro.
- Concurrente: Son los subestados que están operando sincronizadamente uno con otro. [6]

5.1.5.5.3 Importancia del diagrama de Estados

El diagrama de estados del UML proporciona una gran variedad de símbolos y abarca varias ideas (todas para modelar los cambios por los que pasa un objeto). Este tipo de diagrama tiene el potencial de convertirse en algo complejo con rapidez.

Es necesario contar con diagramas de estados dado que permiten a los analistas, diseñadores y desarrolladores comprender el comportamiento de los objetos de un sistema. [4]

5.1.5.6 Diagrama de Secuencias

El diagrama de secuencias consta de objetos que muestran de forma usual; rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

5.1.5.6.1 Objetos

Los objetos se colocan cerca del área superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La extensión que está debajo (y en forma descendente) de cada objeto será una línea discontinua conocida como línea de la vida de un objeto. Junto con la línea de vida de un objeto se encuentra un pequeño rectángulo conocido como activación, el cual representa la ejecución de una operación que realiza el objeto. La longitud del rectángulo se interpreta como la duración de la activación. [5]



Figura 9 Objeto en Diagrama de secuencias

5.1.5.6.2 Mensajes

Un mensaje que va de un objeto a otro pasa de la línea de vida de un objeto a la de otro. Un objeto puede enviarse un mensaje así mismo (es decir, desde su línea de vida hacia su propia línea de vida).

Un mensaje puede ser simple, sincrónico y asíncrono. Un mensaje simple es la transferencia de control de un objeto a otro. Si un objeto envía un mensaje sincrónico, esperara la respuesta a tal mensaje antes de continuar con su trabajo. Si un objeto envía un mensaje asíncrono, no esperara una respuesta antes de continuar. En el diagrama de secuencias, los símbolos de mensaje varían, por ejemplo: la punta de la flecha de un mensaje simple está formada por dos líneas, la punta de la flecha de un mensaje sincrónico esta rellena y la de un asíncrono tiene una sola línea. [10]

5.1.5.6.3 Tiempo

El diagrama representa al tiempo en dirección vertical. El tiempo se inicia en la parte superior y avanza hacia la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno esté cerca de la parte inferior. Con ello, el diagrama de secuencias tiene dos dimensiones. La dimensión horizontal es la disposición de los objetos y la dimensión vertical muestra el paso del tiempo. La Figura 10 muestra un conjunto básico de de símbolos de diagrama de secuencias, con los símbolos en funcionamiento conjunto. La Figura 10 muestra a un actor que inicia la secuencia, aunque, en sentido estricto, muestra a un actor que a un actor que inicia la secuencia, aunque, es un conjunto de símbolos del diagrama de secuencias. [5]

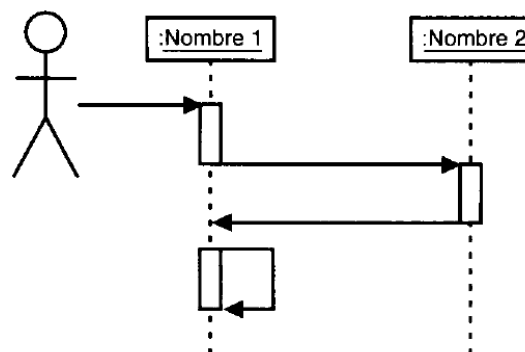


Figura 10 Conjunto básico de símbolos del Diagrama de secuencias

Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria. Durante

el análisis inicial, el modelador típicamente coloca el nombre 'business' de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre 'business' es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado, o invocado, pertenece a la definición de la clase instanciada por el objeto en la recepción final del mensaje. La Figura 11 muestra un ejemplo de un diagrama de secuencias. [4]

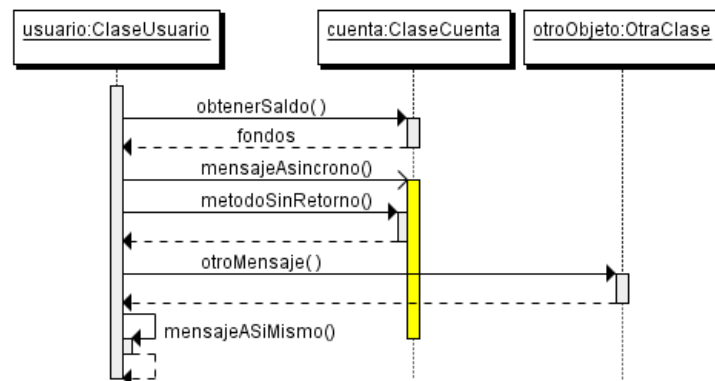


Figura 11 Diagrama de secuencias

5.1.5.7 Diagramas de colaboraciones

Los diagramas de colaboraciones muestran la forma en que los objetos colaboran entre si, tal como sucede con un diagrama de secuencias. Muestran los objetos junto con los mensajes que se envían entre ellos. Si el diagrama de secuencias hace eso ¿Por qué UML requiere otro diagrama? Si el de secuencias es muy similar ambos diagramas, de hecho son semánticamente equivalentes. Esto significa que representan la misma información, y se puede convertir un diagrama de secuencias en un diagrama de colaboraciones y viceversa. [4]

Los diagramas de secuencias destacan la sucesión de las interacciones. Los diagramas de colaboraciones destacan el contexto y organización general de los objetos que lo interactúan. Otra diferencia es que el diagrama de secuencias se organiza de acuerdo al tiempo, y el de colaboraciones de acuerdo al espacio. [2]

Un diagrama de colaboraciones es una extensión de unos objetos. Además de la relación de entre objetos, el diagrama de colaboraciones muestran los mensajes que se envían los objetos entre sí. Por lo general evita la multiplicidad, dado que podría ser fuente de confusión.

Para representar un mensaje, dibujara una flecha cerca de la línea de asociación entre dos objetos, esta flecha apunta al objeto receptor. El tipo de mensaje se mostrara en una etiqueta cerca de la flecha; por lo general, el mensaje indicara al objeto receptor que ejecute una de sus operaciones. El mensaje finalizara con un par de paréntesis; dentro de los cuales colocara los parámetros (si los hay) con los que funcionara la operación. [10]

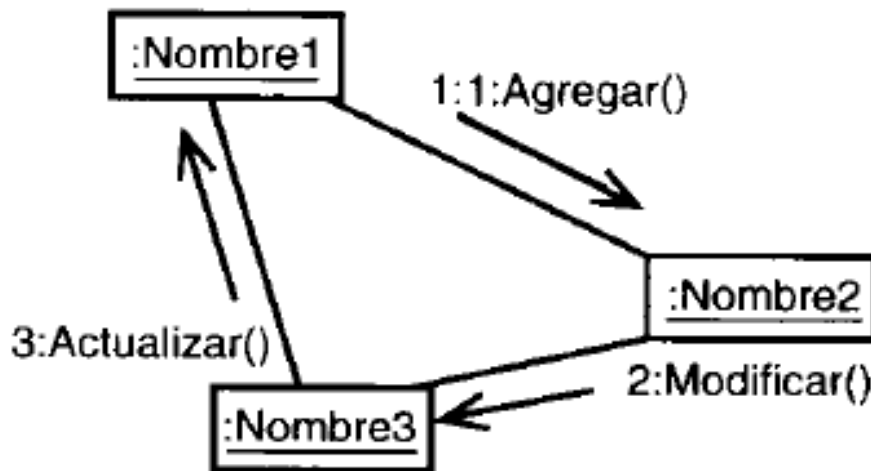


Figura 12 Simbología del diagrama de colaboraciones.

5.1.5.7.1 Ejemplo de diagrama de colaboraciones (Interfaz Grafica del Usuario)

Un actor inicia la secuencia de interacción al oprimir una tecla, con lo que los mensajes ocurrirán. Con lo que los mensajes ocurrirán de manera secuencial.

- 1.- La GUI notifica al sistema operativo que se oprimió una tecla
- 2.- El sistema operativo le notifica ala CPU
- 3.- El sistema operativo actualiza la GUI
- 4.- L a CPU le notifica a la Tarjeta de Video
- 5.- La tarjeta de video envía un mensaje al monitor
- 6.- El monitor presenta el carácter alfanumérico en la pantalla con lo que se hará evidente al usuario [1]

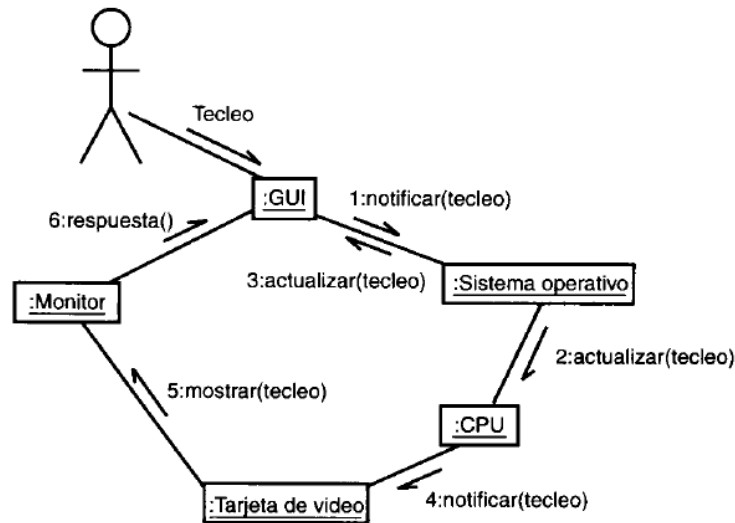


Figura 13 Forma de representar una secuencia de operaciones.

5.1.5.8 Diagrama de Actividades

Un diagrama de actividades está diseñado para mostrar una vista simplificada de lo que ocurre durante una operación o proceso. Es una extensión de un diagrama de estados. El diagrama de estados muestra los estados de un objeto y representa las actividades como flechas que conectan a los estados. El diagrama de actividades resalta precisamente las actividades. [5]

A cada actividad se le representa por un rectángulo con las esquinas redondeadas (más angosto y ovalado que la representación del estado). El procesamiento dentro de una actividad se lleva a cabo y al realizarse, se continúa con la siguiente actividad. Una flecha representa la transición de una a otra actividad. Al igual que el diagrama de estados el de actividad cuenta con un punto inicial (representado por un círculo relleno y uno final representado por una diana). [9]



Figura 14 Punto inicial y final, así como dos actividades y una transición.

5.1.5.8.1 Decisiones

Casi siempre una secuencia de actividades llegara a un punto donde se realizara alguna decisión. Ciertas condiciones llevaran por un camino y otras por otro (pero ambas son mutuamente exclusivas). [1]Podrá representar un punto de decisión de una de dos formas: la primera es mostrar las rutas posibles que parten directamente de una actividad y la segunda es llevar la transición hacia un rombo (reminiscencias del símbolo de decisión en un diagrama de flujo) y que de allí salgan las rutas de decisión. Se indicara la condición con una instrucción entre corchetes junto a la ruta correspondiente [1]

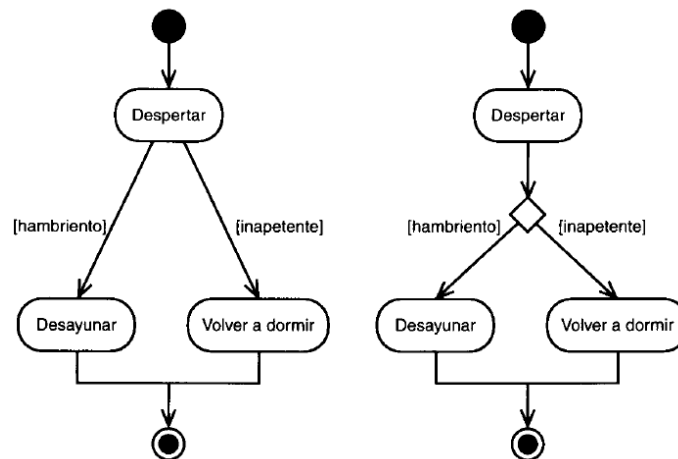


Figura 15 Formas de tomar una decisión.

5.1.5.8.2 Rutas concurrentes e indicaciones

Conforme modele actividades tendrá la oportunidad de separar una transición en dos rutas que se ejecuten al mismo tiempo (es decir, en forma concurrente) y luego se reúnan. Para representar esta división, se utilizara una línea gruesa perpendicular a la transición y las rutas partirán de ella. Para representar la reincorporación, ambas rutas apuntaran a otra línea gruesa. [1]

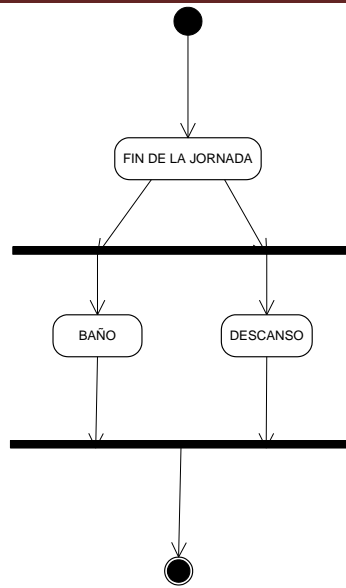


Figura 16 Ejemplo de rutas concurrentes al momento de tomar una decisión.

Durante una secuencia de actividades, es posible enviar una indicación. Cuando se recibe, la indicación provocara que se ejecute una actividad. El símbolo para enviar una indicación es un pentágono convexo, y el que la recibe es un pentágono cóncavo. En UML el pentágono convexo simbolizo al envío de un evento; el cóncavo simboliza la recepción del evento. [7]

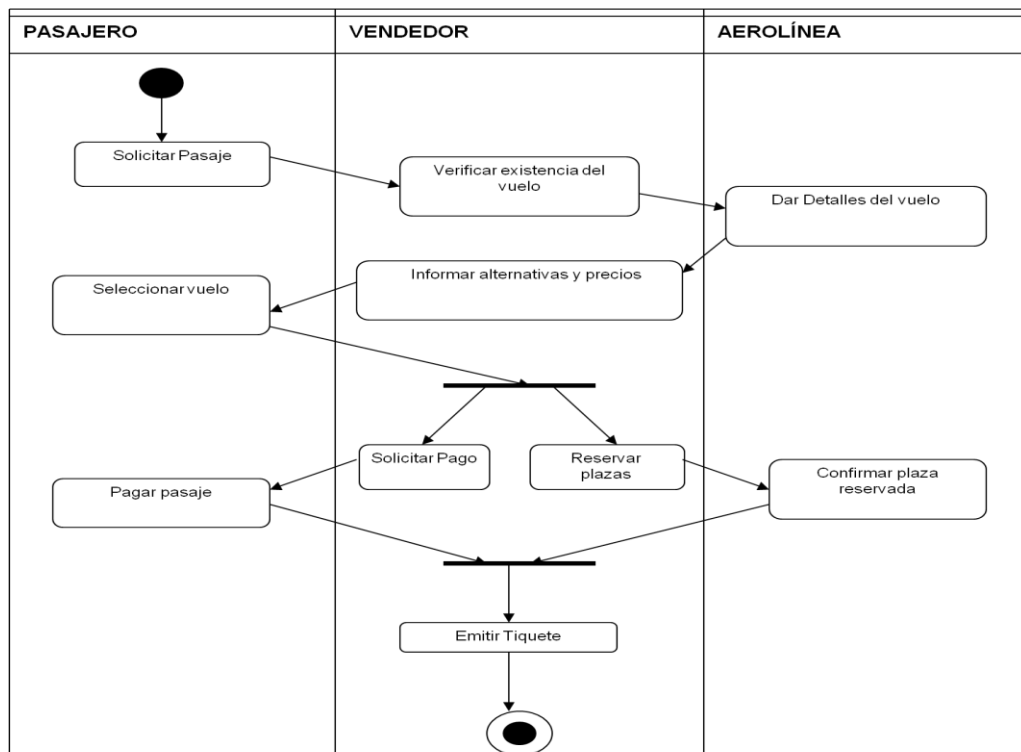


Figura 17 Muestra una indicación en un Diagrama de actividades.



En un diagrama de actividades se puede representar las actividades de acuerdo con la responsabilidad asignada. Esto se realiza con marcos de responsabilidad, mismos que son segmentos paralelos que corresponden a los responsables de realizar cada tarea. Es posible combinar el diagrama de actividades con símbolos de otros diagramas con lo que producirán diagramas híbridos. [9]

5.1.5.9 Diagrama de componentes

Un componente de software es una parte física de un sistema y se encuentra en una computadora, no en la mente del analista. Un componente puede ser: una tabla, archivo de datos, ejecutable, biblioteca de vínculos dinámicos, documentos etc. [4]

Uno de los aspectos más importantes de los componentes es el potencial que tienen de volver a ser utilizados. Con las necesidades actuales de los negocios de soluciones rápidas, entre más rápido presente un sistema para producción mayor será su competitividad. Si se puede crear un componente para un sistema y puede volver a reutilizarlo en otro, se habrá contribuido a esa competitividad. [10]

5.1.5.9.1 Tipos de componentes

Existen tres tipos de componentes:

1. Componentes de Distribución: Conforman el fundamento de los sistemas ejecutables (por ejemplo, DLL, ejecutables, Controles Active X y Java Beans).
2. Componentes para trabajar en el producto: a partir de los cuales se han creado los componentes de distribución(como archivos de bases de datos y de código)
3. Componentes de ejecución: creados como resultado de un sistema en ejecución. [8]

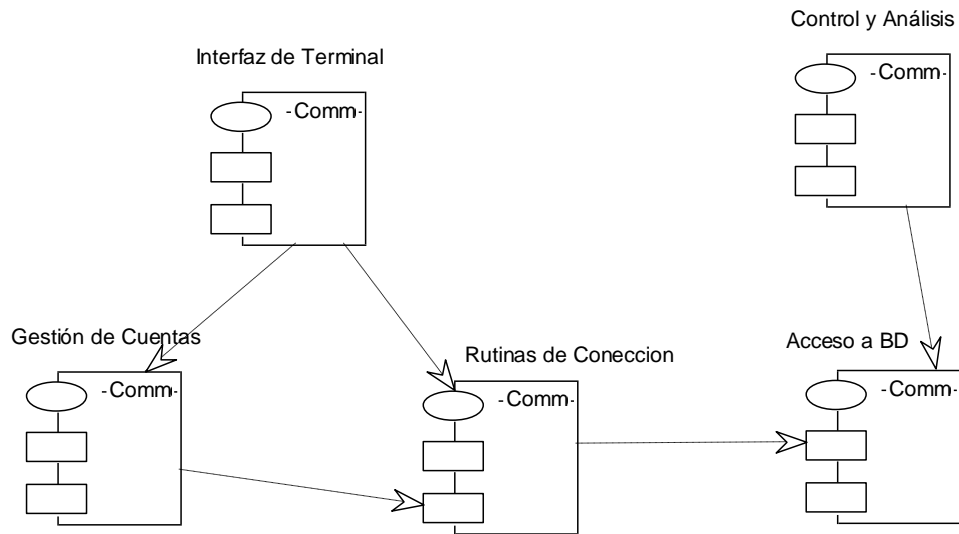


Figura 18 Diagrama de componentes enfocados en un sistema de información.

5.2 SISTEMAS DE INFORMACION

5.2.1 Una tipología de Sistemas

Los sistemas físicos que aparentemente maquillan al universo, desde los sistemas subatómicos de núcleos atómicos (como los que describe la física), pasando por el marco físico de este y otros planetas y los sistemas vivientes observados en la tierra, hasta sistemas galácticos en el otro extremo; todos estos son *sistemas naturales*, sistemas cuyos orígenes están en el origen del universo y que son resultado de las fuerzas y procesos que caracterizan a este universo; son sistemas que no pueden ser algo distinto de lo que son, dado un universo cuyos patrones y leyes no son caprichosos. [11,14]

Los sistemas naturales son los todos irreductibles, hechos por la evolución, que un observador puede observar y describir como tales, resultado de la agrupación de otras entidades que tiene relaciones mutuas. Son “irreductibles” en el sentido de que sigue siendo verdad incluso si podemos describir sus componentes y las relaciones entre los componentes con una cierta precisión. El bióxido de carbono no es reducible, en este sentido, a carbono y oxígeno, ya que no obstante cuanto sepamos acerca de las distancias interatómicas y ángulos de enlace, el bióxido de carbono sigue siendo un todo de nivel superior, que tiene propiedades exclusivas de sí. [11,12]

Existen también muchas otras entidades observadas que son similares a los sistemas naturales en aspectos diferentes a este último. Estos son los sistemas



que son el resultado de un diseño consciente. Ellos son los *sistemas físico diseñados* que el hombre ha hecho, su clase va desde los martillos, vía de tranvías, hasta cohetes espaciales. Ellos son diseñados como resultado de algún propósito humano, que es su origen y existen para servir a un propósito, incluso aunque, como en el caso de pintura de un artista, por ejemplo, sea difícil de definir explícitamente cual es el propósito. [18,11]

La capacidad de diseño del hombre no está restringida a la construcción de artefactos físicos. También vemos en el mundo un gran número de lo que se podría describir, como *sistemas abstractos diseñados* como la matemática o los poemas, o las filosofías, ellos representan el producto consciente ordenado de la mente humana. Son en si mismos sistemas abstractos, aunque gracias a una actividad de diseño exitosa previa, ahora se pueden capturar en sistemas físicos diseñados como libros, filmes, grabaciones, heliografías. [11, 16,14]

El acto humano de diseño en si un ejemplo de una cuarta clase de sistema posible: el *sistema de actividad humana*. Estos son sistemas menos tangibles que los sistemas naturales y diseñados, sin embargo, se pueden observar claramente en el mundo innumerables grupos de actividades humanas ordenadas más o menos conscientemente en todos, como resultado de algún propósito o misión fundamental. Por ejemplo, un “sistema de placer” consistirá en actividades humanas que involucren distintos sistemas naturales y de diseño físico y (o) sistemas abstractos como los campos de juego, los bastones de cricket, las reglas de juego, etc., sin embargo es mejor el restringir la definición del sistema de actividad humana a las actividades mismas, nombrando y describiendo otros sistemas asociados si es adecuado en ese momento. [11, 15,18]

Más allá de los sistemas de actividad humana de diseño abstracto, diseño físico o natural, debe haber una categoría que incluya a los sistemas más allá del conocimiento; así que podríamos denominar a estos como sistemas trascendentales. [17, 11,20]

El mapa de sistemas (como se muestra en la Figura 19) sugiere que el número mínimo absoluto de clases de sistemas necesarias para describir la realidad es de cuatro: sistemas de actividad humana, de diseño abstracto, de diseño físico y natural. [16]

Es importante observar que el mapa teológico es en si mismos un sistema abstracto diseñado. Proporciona no tanto como un reporte de la realidad, sino más bien un grupo de tipos conceptuales a emplearse en las descripciones de la realidad con base en sistemas. [11]

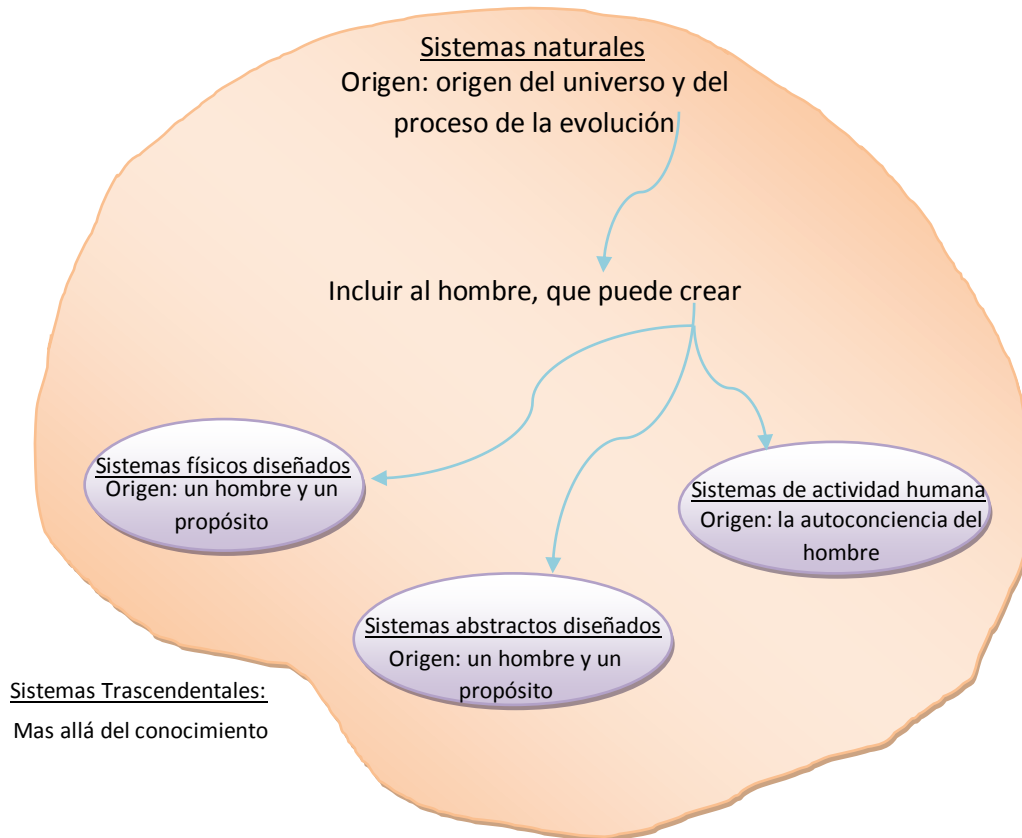


Figura 19 Mapa de Sistemas

5.2.1.1 ¿Qué es un sistema?

Sistema es una de las palabras más populares en los círculos informáticos, y por este motivo, la aproximación a los sistemas, tal como se les llama, es un término muy difundido en la totalidad de los trabajos de diseño e implementación de los sistemas informáticos. [13]

Un sistema es un conjunto de componentes que interaccionan entre sí para lograr un objetivo común. La finalidad de un sistema es la razón de su existencia. Todo sistema organizacional depende, en mayor o menor medida, de una entidad abstracta denominada sistema de información. Este es el medio por el cual los datos fluyen de una persona o departamento hacia otros y puede ser cualquier cosa, desde la comunicación interna entre los diferentes componentes

de la organización y líneas telefónicas hasta sistemas de cómputo que generan reportes periódicos para varios usuarios. Los sistemas de información proporcionan servicio a todos los demás sistemas de una organización y enlazan todos sus componentes en forma tal que estos trabajen con eficiencia para alcanzar el mismo objetivo. [16]

Muy pocos sistemas son totalmente auto contenidos, muchos existen en un entorno que los afecta de una manera u otra, sin embargo la interacción con el entorno no es tan fuerte como la interacción entra los compontes del sistema, y usualmente es posible identificar un límite entre el sistema y su entorno. [12]

5.2.2 Frontera y Entorno

Frontera y Entorno son conceptos estrechamente relacionados y entenderlos resulta esencial para poder modernizar cualquier sistema; de hecho, el primer paso para comprender un sistema es elegirlo, y esa tarea está fuertemente relacionada con la elección de su frontera como se muestra en la figura 20. [18]



Figura 20 Frontera y el entorno de un sistema

Por ejemplo, un biólogo celular puede pensar en una celda humana como en un sistema; su interés está limitado por la membrana celular. Un bioquímico puede estar interesado en un sistema ligeramente mayor, tal vez en una reacción química que tenga lugar en un determinado grupo de células. De esta forma podríamos seguir ampliando el concepto de sistema, quizás hasta alcanzar el punto de vista de un cosmólogo (cuyo interés puede estar limitado



por las fronteras del universo físico) o de un teólogo (cuyo interés puede ser incluso más amplio). Pero la selección de un sistema que se corresponda con un asunto de interés no es solo cuestión de escala. Los sistemas pueden solapar entre sí y este es también un problema a la hora de definir los límites. Por ejemplo un sociólogo puede pensar que una familia es un conjunto de sistemas que se solapan: un sistema del cuidado de los hijos, un sistema económico, un sistema de mantenimiento doméstico, etc. Estas visiones de la familia como sistema cuentan con límites puramente conceptuales, ya que los miembros de la familia formarían parte de los distintos sistemas con independencia de la distancia que existe entre ellos. [16]

Las fronteras de los distintos sistemas pueden solaparse o coincidir; de hecho, dos sistemas pueden estar estrechamente relacionados, pueden tener fronteras idénticas y aun así, ser distintos. [11]

5.2.3 Entrada, Salida, Interfaz Y Transformación

Los sistemas tienen interacciones con su entorno, por ejemplo, las células humanas toman alimento y oxígeno y lo transforman en proteínas, energía y otros productos; también emiten bióxido de carbono y otros residuos. Algunas células producen sustancias que resultan útiles para el cuerpo. Por ejemplo, el sistema nervioso recibe información sensitiva en forma de luz, sonido, tacto, etc., y transforma esta información en señales eléctricas, algunas de estas señales actúan como entradas a los músculos del cuerpo, generando movimiento o el habla. Cada interacción está basada en un conjunto de entradas y produce algunas salidas. Las entradas se originan fuera del sistema y se introducen en él de alguna forma. Las salidas son creadas por el sistema y se envían al entorno con el fin de tener un efecto en algún lugar. Las salidas actúan para alcanzar algún propósito en el sistema. [13]

La transformación de entradas en salidas es una característica importante de los sistemas que tienen algún propósito, tales como las empresas; es la forma en que cumplen sus objetivos. Por ejemplo el cliente de un sistema bancario no estará, posiblemente, interesado en lo que ocurra detrás del telón. Su objetivo principal será su interacción con el sistema, en otras palabras las entradas que deberá proporcionar por ejemplo los datos de una cuenta bancaria y las salidas que reciba como la información acerca de a quién pertenece, tipo de acciones que puede realizar, etc., esta acción se denomina técnica de la caja negra, ya que

oculta por completo las tareas internas del sistema, todo lo que se puede ver es lo que entra o sale del sistema como lo muestra la Figura 21.

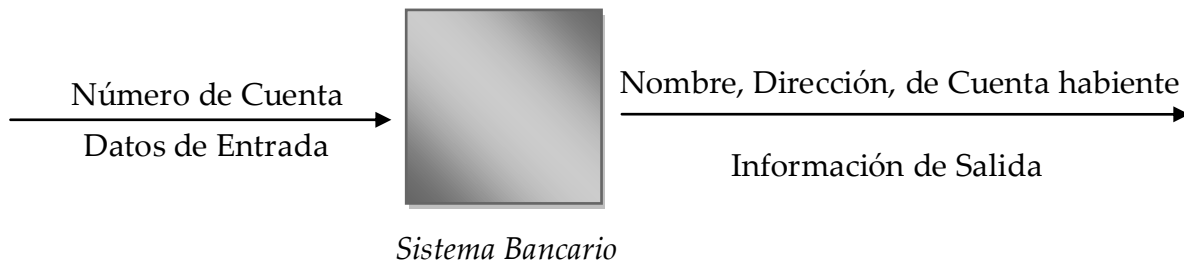


Figura 21 Vista del funcionamiento de la caja negra de un sistema

Una salida de un sistema puede ser simultáneamente la entrada otro. Los dos sistemas comparten parte de su frontera, a lo largo de la cual las entradas y salidas pasan de uno a otro. La frontera compartida es una interfaz. [13]

Una subclase de sistemas son los sistemas significativos, es decir, los sistemas que tienen un objetivo bien definido. Un sistema significativo puede ser considerado como un proceso que transforma las entradas en salidas. Un sistema abierto es uno con entradas y salidas. Un sistema cerrado es uno que no tenga ni entradas ni salidas, es decir, que este completamente aislado de su entorno como se muestra en la Figura 22.

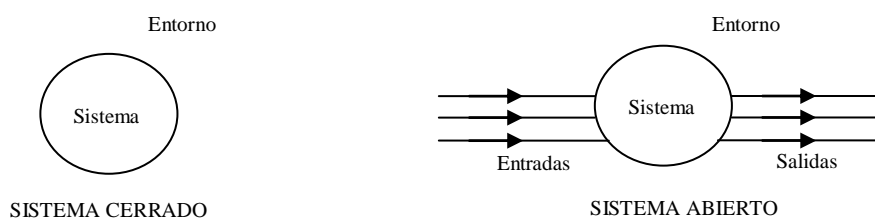


Figura 22 Sistemas Cerrados y Abiertos.

La distinción entre sistemas abiertos y cerrados es poco útil, sin embargo es bueno recordar que algunos sistemas son más abiertos que otros; también es bueno contrastar las formas muy restrictivas en que los ordenadores pueden aceptar la información (por ejemplo, un grupo de mandatos entrados por medio del teclado) con los medios más variados de la percepción humana. Los sistemas tales como los sistemas informáticos se definen a veces como sistemas relativamente cerrados. [14]

5.2.4 Subsistemas.

Muchos sistemas reales son extremadamente complejos, y es muy útil dividirlos en sistemas más pequeños interconectados llamados *subsistemas*. Estos también son sistemas en el sentido de la definición anterior, y ellos mismos también pueden subdividirse en subsistemas menores, hasta que eventualmente se alcanza un nivel conceptual conveniente de subdivisión o se llega a los componentes y se para, como lo muestra la Figura 23.

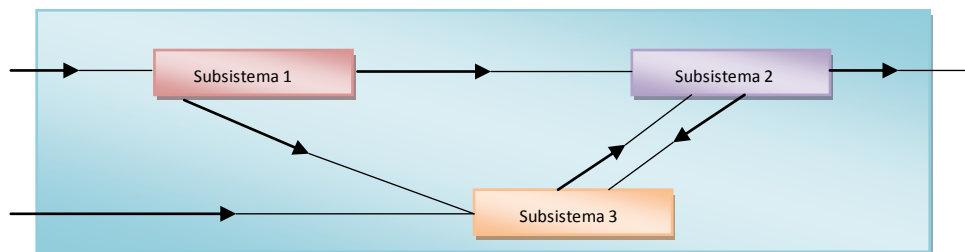


Figura 23 Subsistema

La elección de los subsistemas se determina en el análisis final por la visión global que queremos adoptar sobre dicho sistema, la que a su vez, depende, en general, de lo que se está contemplando, pero existen además otras consideraciones. Igual que en el propio sistema, queremos que las interacciones dentro de cada subsistema sean más fuertes que las existentes con el entorno (es decir, en los otros subsistemas). Al nivel de interpretación entre subsistemas se le conoce como *acoplamiento* (Figura 24).

Al desacoplar subsistemas, intentamos:

- ✓ Minimizar el número de entradas y salidas a partir de cada subsistema, y también minimizar el número de distintos subsistemas que son fuentes y destinos de estas entradas y salidas;
- ✓ Asegurar que el tiempo en que se realizan estos procesos en un subsistema, no venga impuesto por otro subsistema.[11]

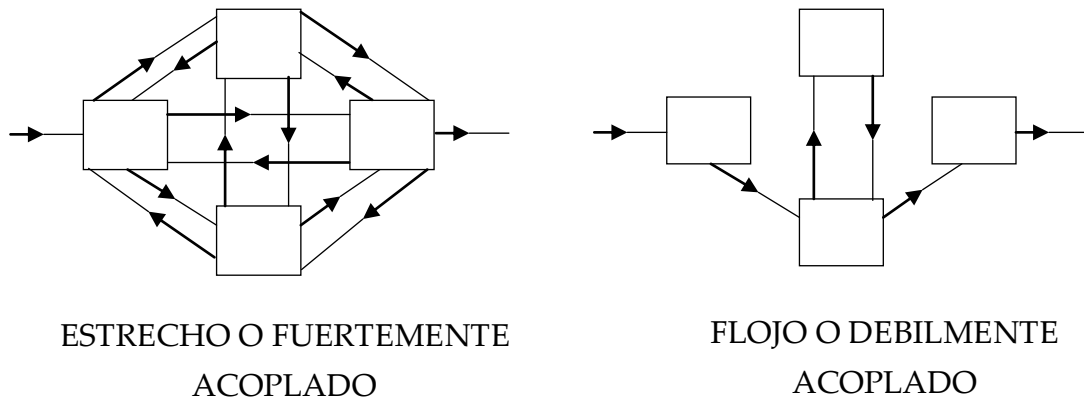


Figura 24 Acoplamiento de subsistemas

5.2.5 Características de un sistema

La finalidad de un sistema es la razón de su existencia. Para alcanzar sus objetivos los sistemas interactúan con su medio ambiente, el cual está formado por todos los objetos que se encuentran fuera de las fronteras de los sistemas.

Todo sistema tiene un conjunto de características que son las siguientes:

- ✓ Existen dentro de un entorno
- ✓ Se encuentran separados de su entorno por algún tipo de frontera
- ✓ Tienen entradas y salidas (reciben entradas desde su entorno y envían salidas a su entorno)
- ✓ Transforman sus entradas de alguna forma para producir sus salidas
- ✓ Disponen de interfaces que permiten la comunicación entre los sistemas
- ✓ Un sistema puede estar formado por subsistemas y a la vez el sistema pertenece a otros sistemas llamado súper sistema.
- ✓ Todo sistema que realiza funciones, dispone de un mecanismo de control, el que a su vez está formado por la retroalimentación. La retroalimentación maneja información sobre las operaciones del sistema o su entorno, que luego se pasará al mecanismo de control.
- ✓ Disponen de ciertas propiedades que no dependen directamente de las propiedades de sus partes. Estas se denominan propiedades emergentes ya que solo aparecen a nivel del sistema como un todo. [13]



5.2.5.1 Control del sistema

El control del sistema suele estar basado en una comparación entre dos o más valores de entrada, cuya similitud o diferencia obliga al sistema a tomar una decisión sobre si hace falta realizar alguna acción de control.

Los sistemas emplean un modelo de control básico que consiste en:

- ♣ Un estándar para lograr un desempeño aceptable
- ♣ Un método para medir el desempeño aceptable
- ♣ Un medio para comparar el desempeño actual contra el estándar
- ♣ Un método de retroalimentación [14]

5.2.5.2 Retroalimentación.

El término de retroalimentación indica que uno o más salidas del sistema van a ser analizadas y literalmente, volverán a alimentar a la unidad de control.

La denominada retroalimentación negativa su objetivo es mantener el equilibrio del sistema oponiéndose a las desviaciones con respecto a alguna norma, se utiliza ampliamente en los sistemas físico, tales como los equipos electrónicos, sistemas de fabricación, etc. Como contraste la retroalimentación positiva funciona reforzando las desviaciones en lugar de oponerse a ellas y, por lo tanto, tiende a aumentar los desplazamientos alejándolos del punto de equilibrio. Los sistemas gobernados por retroalimentación positiva son inherentemente inestable, lo cual resulta indeseable cuando la desviación que se está reforzando resulta en si misma indeseable.

La retroalimentación positiva y negativa se produce también en los sistemas sociales y empresariales pero, en estos casos, su funcionamiento no suele ser tan claro o sencillo. La retroalimentación positiva es beneficiosa cuando un estado estacionario resulta indeseable; no implica, necesariamente, que cada desviación de la norma va a ser reforzada, se puede permitir que algunas de ellas mueran ante la ausencia de un estímulo. [12]

5.2.5.3 La retroalimentación hacia adelante

5.2.5.3.1 Concepto de Retroalimentación hacia adelante

La retroalimentación hacia adelante de la información se basa en analizar las entradas del sistema, el lugar de sus salidas. Por ejemplo la venta de juguetes en navidad, se puede llegar a la conclusión de que el uso de la retroalimentación en las ventas (una medida de la salida) no les permite reaccionar con la



suficiente rapidez como para adaptarse a las condiciones del mercado. Este hecho puede provocar que haya existencias que no se vendan, que se realicen compras a fabricantes antes de que puedan constatar que la demanda se ha desplomado. Idealmente, la empresa debería ajustar su fabricación para adecuarse al nivel de la demanda y deberían poder usar las conclusiones de investigaciones de mercado para prever que juguetes serán populares entre los niños ese año.

Aunque la información de control obtenida de la retroalimentación hacia adelante puede ayudar a que un sistema responda mejor a las fluctuaciones del entorno, no resulta siempre fácil de implementar o gestionar en una organización empresarial.

5.2.5.3.2 Propiedades emergentes

Una de las características auténticamente distintivas de un sistema es que es algo más que la suma de sus partes. El sistema posee alguna característica o capacidad en sí mismo que no se encuentra presente en ninguno de sus componentes. A eso se denomina *propiedades emergentes*. Este reconocimiento de que los sistemas cuentan con propiedades emergentes es el motivo principal de por qué los sistemas se describen como *holísticos* (integral). Esto implica pensar en cada sistema como un todo y donde se pasaran por alto importantes aspectos si pensamos solo en sus partes de forma aislada unas de otras.

La vista integral de los sistemas se manifiesta en técnicas tales como la *imagen rica*. Este elemento intenta capturar en una sola imagen o diagrama todo lo que resulta esencial para comprender, inicialmente, un sistema. [12]

5.2.6 Información y Sistemas de Información.

5.2.6.1 ¿Qué es información?

La información se describe como el proceso por el cual los hechos en bruto alcanzan utilidad (es decir, se convierten en información) a través de una secuencia de etapas. El significado de un hecho siempre depende de la importancia relativa que tenga para el observador. La información son datos a los que se les ha dado una forma que tiene sentido y es útil para los humanos. [18]

5.2.6.2 ¿Qué es un sistema de Información?

Un sistema de información puede definirse como un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la

información para apoyar la toma de decisiones, el control, análisis y visión en una institución. [19]

Tres actividades de un sistema de información producen la información que la institución requiere para la toma de decisiones, el control de las operaciones, el análisis de los problemas y la creación de nuevo productos y servicios. Estas actividades son las de insumo, procesamiento y producto. [19]

5.2.6.3 *Ciclo de vida del desarrollo de sistemas.*

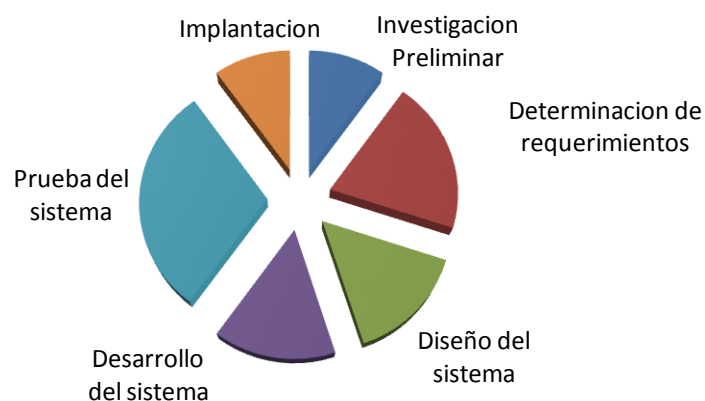


Figura 25 Ciclo de vida del desarrollo de sistemas

El método del ciclo de vida para desarrollo de sistemas (SDLC) (que se muestra en la Figura 25) es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. En la mayor parte de las situaciones dentro de una empresa todas las actividades están muy relacionadas en general son inseparables, y quizá sea difícil determinar el orden de los pasos que se siguen para efectuarlas. Las diversas partes del proyecto pueden encontrarse al mismo tiempo en distintas fases de desarrollo; algunos componentes en la fase de análisis mientras que otros en etapas avanzadas de diseño. [19]

- ♣ El método del ciclo de vida para desarrollo de sistemas consta de las siguientes actividades:
- ♣ Investigación preliminar
- ♣ Determinación de los requerimientos del sistemas
- ♣ Diseño del sistema
- ♣ Desarrollo de software
- ♣ Prueba de los sistemas implantación y evaluación[19]



5.2.6.4 Investigación Preliminar.

Cuando se formula la solicitud comienza la primera actividad de sistemas, la *investigación preliminar*. Esta actividad tiene tres partes:

- ♥ **Aclaración de la solicitud :** muchas solicitudes que provienen de empleados y usuarios no están formuladas de manera clara, por consiguiente, antes de considerar cualquier investigación de sistema, la solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante desea; antes de seguir adelante, la solicitud de proyecto debe estar claramente planeada.
- ♥ **Estudio de factibilidad:** un resultado importante de la investigación preliminar es la determinación de que el sistema solicitado sea factible. En la investigación preliminar existen tres aspectos relacionados con el estudio de factibilidad:
 - f* **Factibilidad Técnica:** el trabajo para el proyecto, ¿puede realizarse con el equipo actual, la tecnología existente de software y el personal disponible? Si se necesita nueva tecnología, ¿cuál es la posibilidad de desarrollarla?
 - f* **Factibilidad Económica:** al crear el sistema, ¿los beneficios que se obtienen serán suficientes para los costos?, ¿los costos asociados con la decisión de *no* crear el sistema son tan grandes que se debe aceptar el proyecto?
 - f* **Factibilidad Operacional:** si se desarrolla e implanta, ¿será utilizado el sistema?, ¿existirá cierta resistencia al cambio por parte de los usuarios, que dé como resultado una disminución de los posibles beneficios de la aplicación?

El resultado de factibilidad lo lleva a cabo un pequeño grupo de personas que son responsables de evaluar la factibilidad y son analistas capacitados o directivos.

- ♥ **Aprobación de la solicitud:** no todos los proyectos solicitados son deseables o factibles, en ocasiones algunos proyectos pueden comenzar inmediatamente, aunque lo común es que los miembros del equipo de sistemas se encuentren ocupados con otros proyectos y den prioridad a los sistemas conforme le asignen el orden. Después de aprobar la solicitud de un proyecto se estima su costo, el tiempo necesario para



terminarlo y las necesidades de personal; con esta información se determina donde ubicarlo dentro de la lista existente de proyectos. [19]

5.2.7 Las cuatro etapas del proceso.

Todos los sistemas de información operan básicamente de la misma forma, incluyan una computadora o no. Sin embargo, la computadora proporciona un medio conveniente para ejecutar las cuatro operaciones principales de un sistema de información:

- ❖ Introducción de datos en el SI (**entradas**)
- ❖ Cambio y manipulación de los datos en el SI (**procesamiento de datos**)
- ❖ Obtención de información fuera del SI (**salidas**)
- ❖ Almacenamiento de datos e información (**almacenamiento**)

Una transacción es una actividad de un negocio: una venta, una compra, la contratación de un nuevo empleado, etc. Las transacciones pueden registrarse en papel y más adelante capturarse en un sistema de cómputo o registrarse directamente en terminales de sistemas de procesamiento de transacciones (STP), como cajas registradoras y terminales de entrada de pedidos. A menudo el mismo sistema también procesa las transacciones, al resumir y enviar la información a otros sistemas, por lo tanto, estos son sistemas de *procesamiento* de transacciones, no solo de registro de transacciones. [13]

5.2.7.1 *Importancia de las herramientas asistidas por computadora en el desarrollo de sistemas.*

Las herramientas son esenciales para el análisis de sistemas. Ellas mejoran la forma en que ocurre el desarrollo y tiene influencia sobre la calidad del resultado final.

5.2.7.1.1 Beneficios del empleo de herramientas.

Tanto el proceso de desarrollo de sistemas como el producto que se obtiene con él, pueden mejorarse con el uso de herramientas apropiadas. Con el uso de herramientas podemos lograr excelentes mejoras en nuestro sistema por ejemplo:

- Mejora en la productividad
- Mejora en la eficiencia
- Mejora en la calidad del sistema de información

5.2.7.1.2 Beneficios de las herramientas asistidas por computadora

La automatización mejora los beneficios que se pueden obtener con el empleo de herramientas. Con ella disminuye el tiempo necesario para llevar a cabo las tareas, se reduce la intensidad del trabajo, y el seguimiento de todos los procedimientos se lleva a cabo de manera consistente; también se capturan los datos que describen el sistema para tenerlos almacenados en un formato que pueda leer una computadora. [14]

5.2.7.1.3 Clasificación de herramientas automatizadas

Las *herramientas de tipo front-end*, automatizan las primeras actividades del proceso de desarrollo de sistemas. (Como se muestra en la Figura 26)

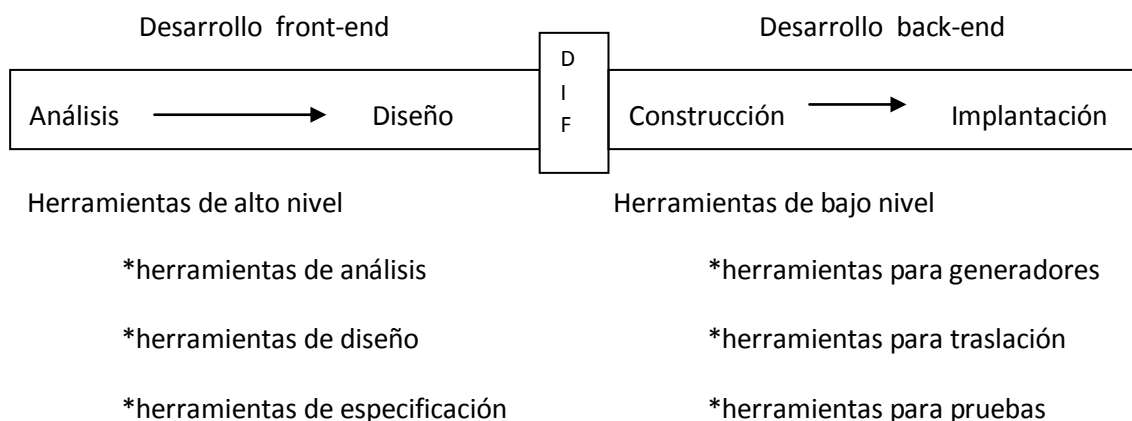


Figura 26 Clasificación de herramientas automatizadas

Las herramientas de tipo front-end proporcionan a menudo, soporte para el desarrollo de modelos gráficos de sistemas y procesos. Los diagramas de flujo de datos son representativos de este tipo de herramienta; los diagramas de flujo de datos representan en forma gráfica (más que por escrito) los procesos y flujos de datos del sistema.

Las *herramientas de tipo back-en* tiene como finalidad ayudar al analista a formular la lógica del programa, los algoritmos de procesamiento y la descripción física de datos, también ayudan a la interacción con los dispositivos (para entrada y salida), etc. [15]



5.2.8 Tipos de sistemas de información.

Cuadro 1 Comparación de Sistemas de Información [11, 12, 14, 16, 17, 19,20]

NOMBRE DEL SISTEMAS	CARACTERISTICAS DEL SISTEMA
SISTEMA DE PROCESAMIENTO DE TRANSACCIONES (TPS)	Gestiona información referente a las transacciones producidas de una empresa.
SISTEMA DE INFORMACION GERENCIAL (MIS)	Dan servicio a funciones a nivel administrativo Apoyan a nivel administrativo a la organización proveyendo de información a los gerentes y en algunos casos de acceso en línea al desempeño real y registros históricos de la organización.
SISTEMAS DE APOYO A DECISIONES (DSS)	Brinda herramientas para realizar el análisis de las diferentes variables de negocio con la finalidad de apoyar el proceso de toma de decisiones.
SISTEMAS DE APOYO EDUCATIVO (ESS)	Auxilian en las decisiones no rutinarias que requieren juicio, evaluación y comprensión porque no hay un procedimiento convenio para llegar a una solución.
SISTEMA DE AUTOMATIZACION DE OFICINAS (OAS)	Sus aplicaciones son destinadas a ayudar al trabajo diario del administrativo de una empresa u organización.
SISTEMAS DE VENTAS Y MARKETING	Ayudan a la empresa a identificar a los clientes para los productos o servicios de la empresa, desarrollar productos y servicios para satisfacer las necesidades de los clientes.
SISTEMAS EXPERTOS (SE)	Emula o copia el comportamiento de un experto en un dominio concreto.
SISTEMA DE PLANIFICACION DE RECURSOS (ERP)	Integra la información y los proceso de una organización en un solo sistema.
SISTEMA DE FINANZAS Y CONTABILIDAD	Dan seguimiento a los activos financieros dela empresa y flujos de fondo.
SISTEMAS DE MANUFACTURA Y PRODUCCION	Tratan de la planeación, desarrollo y elaboración de productos y servicios, del control y flujo de la producción.
SISTEMAS DE RECURSOS HUMANOS	Sistemas que llevan registros de los empleados, dan seguimiento a las habilidades, desempeño del trabajo y capacitación del empleado y apoyan la planeación de la remuneración y desarrollo profesional de los empleados.
SISTEMAS A NIVEL OPERATIVO	Supervisan las actividades elementales y las transacciones de la organización.
SISTEMAS A NIVEL CONOCIMIENTO	Apoyan a los trabajadores del conocimiento y de datos de una organización.
SISTEMAS A NIVEL ADMINISTRATIVO	Apoyan a las actividades de supervisión, control, de toma de decisiones y administrativas de los gerentes de nivel medio.

5.2.9 Modelos de sistemas de información

5.2.9.1 Modelo en cascada.

- No refleja realmente el proceso de desarrollo del software
- Se tarda mucho tiempo en pasar por todo el ciclo
- Perpetua 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 administración de proyectos [20]

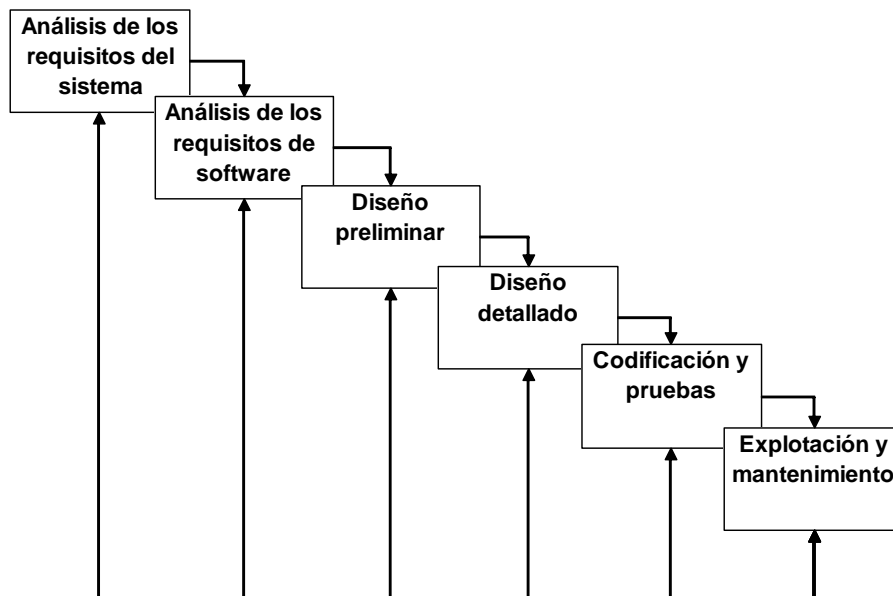


Figura 27 Modelo de la Metodología en Cascada.

5.2.10 Modelo Incremental

- Se evitan proyectos largos y se entrega “Algo de valor” a los usuarios con cierta frecuencia
- El usuario se involucra más
- Difícil de evaluar el costo total
- Difícil de aplicar a sistemas transaccionales que tienden a ser integrados y a operar como un todo

- Requiere desarrolladores experimentados
- Los errores en los requisitos se detectan tarde
- El resultado puede ser muy positivo [18]

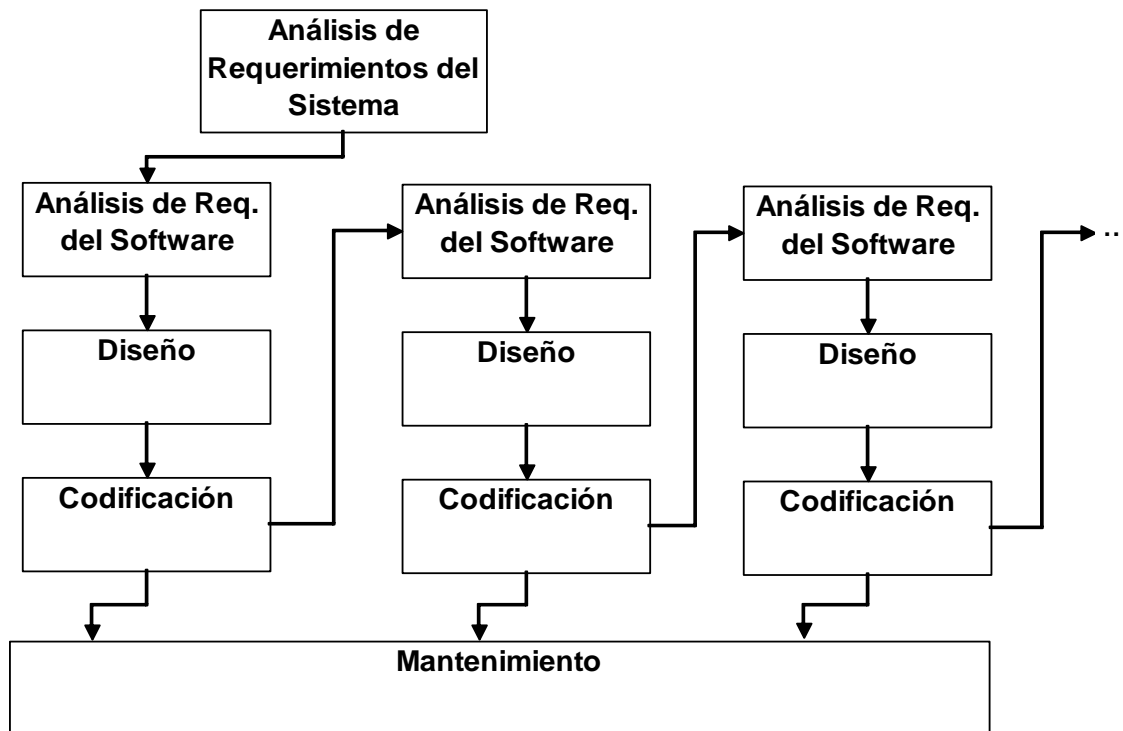


Figura 28 Modelo de la Metodología Incremental

5.2.11 Modelo de Prototipos

- No modifica el flujo del ciclo de vida
- Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios
- Reduce costos y aumenta la probabilidad de éxito
- Exige disponer de las herramientas adecuadas
- No presenta calidad ni robustez
- Una vez identificados todos los requisitos mediante prototipo, se construye el producto de ingeniería.

5.2.11.1 Efectividad

- Debe ser un sistema con el que se pueda experimentar



-
- Debe ser comparativamente barato (< 10%)
 - Debe desarrollarse rápidamente
 - Énfasis en la interfaz de usuario
 - Equipo de desarrollo reducido
 - Herramientas y lenguajes adecuados
 - “El prototipito es un medio excelente para recoger el ‘feedback’ (retroalimentación) del usuario final”

5.2.11.2 Desventajas del Modelo de prototipos

- El cliente ve funcionando lo que para él, es la primera versión del prototipo que ha sido construido con “plastilina y alambres”, y puede desilusionarse al decirle que el sistema aun no ha sido construido.
- El desarrollador puede caer en la tentación de ampliar el prototipo para construir el sistema final sin tener en cuenta los compromisos de calidad y de mantenimiento que tiene con el cliente

5.2.11.3 El prototipo evolutivo

- Construcción de una implementación parcial que cubre los requisitos conocidos, para ir aprendiendo el resto y, paulatinamente, incorporarlos al sistema
- Reduce el riesgo y aumenta la probabilidad de éxito
- No se conocen niveles apropiados de calidad y documentación
- Problemas de gestión de configuración
- Construir software para que pueda ser modificado fácilmente es un “arte desconocido” [19]

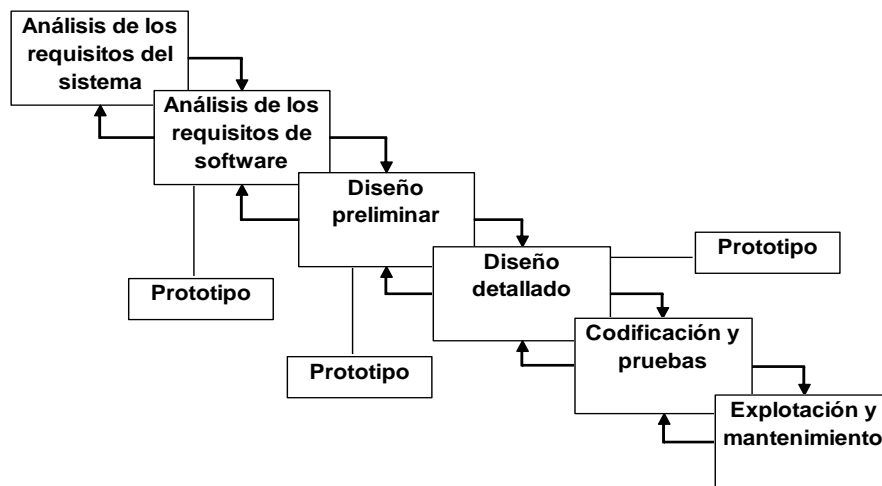


Figura 29 Modelo de la Metodología de Prototipos

5.2.12 Modelo en Espiral

- Trata de mejorar los ciclos de vida clásicos y prototipos.
- Permite acomodar otros modelos
- Incorpora objetivos de calidad y gestión de riesgos
- Elimina errores y alternativas no atractivas al comienzo
- Permite iteraciones, vuelta atrás y finalizaciones rápidas
- Cada ciclo empieza identificando:
 - Los objetivos de la porción correspondiente
 - Las alternativas
 - Restricciones
- Cada ciclo se completa con una revisión que incluye todo el ciclo anterior y el plan para el siguiente

5.2.12.1 Diferencias entre modelo en espiral y modelos tradicionales

- Reconocimiento explícito de las diferentes alternativas.
- Identificación de riesgos para cada alternativa desde el comienzo.
- Al dividir el proyecto en ciclos, al final de cada uno existe un acuerdo para los cambios que hay que realizar en el sistema.

- El modelo se adapta a cualquier tipo de actividad adicional [15]

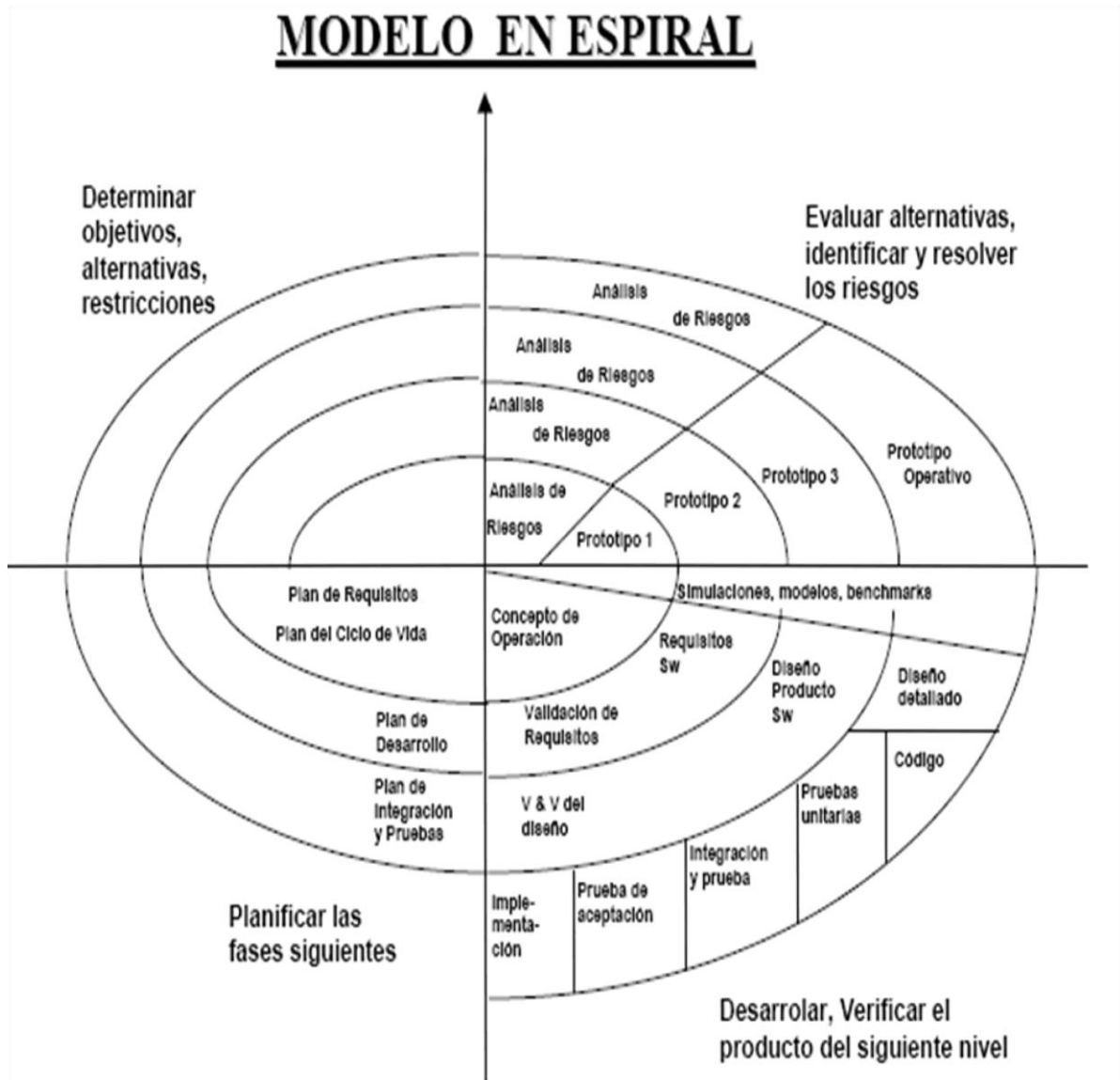


Figura 30 Modelo de la Metodología en Espiral

5.3 BASES DE DATOS

5.3.1 Antecedentes.

Las aplicaciones informáticas de los años sesenta acostumbraban a darse totalmente por lotes (batch) y estaban pensadas para una tarea muy específica relacionada con muy pocas entidades. Cada aplicación utilizaba archivos de movimientos para actualizar (creando una copia nueva) o para consultar uno o



dos archivos maestros o, excepcionalmente, más de dos. Cada programa trataba como máximo un fichero maestro, que solía estar sobre cinta magnética y, en consecuencia, se trabajaba con acceso secuencial. Cada vez que se le quería añadir una aplicación que requería el uso de algunos de los datos que ya existían y de otros nuevos, se diseñaba un archivo nuevo con todos los datos necesarios (algo que provocaba redundancia) para evitar que los programas tuviesen que leer muchos archivo. [21]

El software de gestión de archivo era demasiado elemental para dar satisfacción a las necesidades de las empresas en esos años. Por ejemplo, el tratamiento de las interrelaciones no estaba previsto, no era posible que varios usuarios actualizaran datos simultáneamente, etc. La utilización de estos conjuntos de ficheros por parte de los programas de aplicación era excesivamente compleja, de modo que, especialmente durante la segunda mitad de los años setenta, fueron saliendo al mercado Aplicaciones informáticas tales como: La emisión de facturas, control de pedidos pendientes de servir, el mantenimiento del fichero de productos o la nómina del personal eran algunas de las aplicaciones habituales. [23]

Los primeros SGBD (Sistemas Gestores de Bases de Datos) en los años sesenta todavía no se les denominaba así estaban orientados a facilitar la utilización de grandes conjuntos de datos en los que las interrelaciones eran complejas. [23, 24,25]

Los ordenadores minis, en primer lugar, y después los ordenadores micros, extendieron la informática a prácticamente todas las empresas e instituciones. Los SGBD de los años sesenta y setenta (IMS de IBM, IDS de Bull, DMSde Univac, etc.) eran sistemas totalmente centralizados, como corresponde a los sistemas operativos de aquellos años, y al hardware para el que estaban hechos: un gran ordenador para toda la empresa y una red de terminales sin inteligencia ni memoria. Las aplicaciones típicas eran la reserva/compra de boletos a las compañías aéreas y de ferrocarriles y, un poco más tarde, las cuentas de clientes en el mundo bancario. Esto exigía que el desarrollo de aplicaciones fuese más sencillo. Los SGBD de los años ochenta eran demasiado complejos e inflexibles, y sólo los podía utilizar un personal muy calificado. [21]

Los tipos de datos que se pueden definir en los SGBD relacionales de los años ochenta y noventa son muy limitados. La incorporación de tecnologías



multimedia imagen y sonido en los SI (Sistemas de Información) hace necesario que los SGBD relacionales acepten atributos de estos tipos. [25]

La rápida adopción de la web a los SI hace que los SGBD incorporen recursos para ser servidores de páginas web, como por ejemplo la inclusión de SQL en guiones HTML, SQL incorporado en Java, etc. Esto hace que en el mundo de la web sean habituales los datos multimedia y la OO.

Hoy día, los SGBD relacionales están en plena transformación para adaptarse a tres tecnologías de éxito reciente, fuertemente relacionadas: la multimedia, la de orientación a objetos (OO) e Internet y la web. [27]

Durante estos últimos años se ha empezado a extender un tipo de aplicación de las BD denominado Data Warehouse, o almacén de datos, que también produce algunos cambios en los SGBD relacionales del mercado. A lo largo de los años que han trabajado con BD de distintas aplicaciones, las empresas han ido acumulando gran cantidad de datos de todo tipo. Si estos datos se analizan convenientemente pueden dar información valiosa. [24]

5.3.2 Definiciones

5.3.2.1 Dato

Es un conjunto de caracteres con algún significado, pueden ser numéricos, alfabéticos, o alfanuméricos. [21]

5.3.2.2 Información

Es un conjunto ordenado de datos los cuales son manejados según la necesidad del usuario, para que un conjunto de datos pueda ser procesado eficientemente y pueda dar lugar a información, primero se debe guardar lógicamente en archivos. [21, 22,23]

5.3.2.3 Campo

Es la unidad más pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del programador representa una característica de un individuo u objeto. [21]



5.3.2.4 Registro

Colección de campos de iguales o de diferentes tipos. [26,22]

5.3.2.5 Archivo

Colección de registros almacenados siguiendo una estructura homogénea. [29]

5.3.2.6 Data Warehouse

Un Datawarehouse es una base de datos corporativa que se caracteriza por integrar y depurar información de una o más fuentes distintas, para luego procesarla permitiendo su análisis desde infinidad de perspectivas y con grandes velocidades de respuesta. La creación de un datawarehouse representa en la mayoría de las ocasiones el primer paso, desde el punto de vista técnico, para implantar una solución completa y fiable de Business Intelligence. [28,30]

La ventaja principal de este tipo de bases de datos radica en las estructuras en las que se almacena la información (modelos de tablas en estrella, en copo de nieve, cubos relacionales etc.). Este tipo de persistencia de la información es homogénea y fiable, y permite la consulta y el tratamiento jerarquizado de la misma (siempre en un entorno diferente a los sistemas operacionales). [28]

5.3.2.7 Base de datos

Es una colección de archivos interrelacionados, son creados con un DBMS. El contenido de una base de datos engloba a la información concerniente (almacenadas en archivos) de una organización, de tal manera que los datos estén disponibles para los usuarios, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla. 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. [29]

5.3.2.8 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. [30, 24,25]

El objetivo primordial de un sistema manejador 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 interface entre los usuarios y la base de datos. [22]

5.3.2.9 Esquema de base de datos

Es la estructura por la que está formada la base de datos, se especifica por medio de un conjunto de definiciones que se expresa mediante un lenguaje especial llamado lenguaje de definición de datos. (DDL) [24]

5.3.2.10 Administrador de base de datos (DBA)

Es la persona o equipo de personas profesionales responsables del control y manejo del sistema de base de datos, generalmente tiene(n) experiencia en DBMS, diseño de bases de datos, Sistemas operativos, comunicación de datos, hardware y programación. [25]

5.3.3 Objetivos de las Bases de Datos

Los objetivos principales de un sistema gestor de base de datos son los siguientes: [27,28]

5.3.3.1 Consultas no predefinidas y complejas

El objetivo fundamental de las SGBD es permitir que se hagan consultas no predefinidas (ad hoc) y complejas. [23]

El usuario debe formular la consulta con un lenguaje sencillo (que se quede, obviamente, en el nivel lógico), que el sistema debe interpretar directamente. Sin embargo, esto no significa que no se puedan escribir programas con consultas incorporadas (por ejemplo, para procesos repetitivos). [23]

Los usuarios podrán hacer consultas de cualquier tipo y complejidad directamente al SGBD. El SGBD tendrá que responder inmediatamente sin que estas consultas estén preestablecidas; es decir, sin que se tenga que escribir, compilar y ejecutar un programa específico para cada consulta. [21]

La solución estándar para alcanzar este doble objetivo (consultas no predefinidas y complejas) es el lenguaje SQL. [26]

5.3.3.2 Flexibilidad e independencia

La complejidad de las BD y la necesidad de ir las adaptando a la evolución del SI hacen que un objetivo básico de los SGBD sea dar flexibilidad a los cambios. Interesa obtener la máxima independencia posible entre los datos y los procesos



usuarios para que se pueda llevar a cabo todo tipo de cambios tecnológicos. Los usuarios podrán hacer consultas de cualquier tipo y complejidad directamente al SGBD. El SGBD tendrá que responder inmediatamente sin que estas consultas estén preestablecidas; es decir, sin que se tenga que escribir, compilar y ejecutar un programa específico para cada consulta. [21]

En los Archivos tradicionales, cada vez que se quería hacer una consulta se tenía que escribir un programa a medida, y variaciones en la descripción de la BD, sin que se deban modificar los programas de aplicación ya escritos ni cambiar la forma de escribir las consultas (o actualizaciones) directas. [21,22]

Para conseguir esta independencia, tanto los usuarios que hacen consultas(o actualizaciones) directas como los profesionales informáticos que escriben programas que las llevan incorporadas, deben poder desconocerlas características físicas de la BD con que trabajan. No necesitan saber nada sobre el soporte físico, ni estar al corriente de qué SO se utiliza, qué índices hay, la compresión o no compresión de datos, etc. De este modo, se pueden hacer cambios de tecnología y cambios físicos para mejorar el rendimiento sin afectar a nadie. Este tipo de independencia recibe el nombre de independencia física de los datos. [21, 24,26]

5.3.3.3 Redundancia e inconsistencia de datos.

Puesto que los archivos que mantienen almacenada la información son creados por diferentes tipos de programas de aplicación existe la posibilidad de que si no se controla detalladamente el almacenamiento, se pueda originar un duplicado de información, es decir que la misma información sea más de una vez en un dispositivo de almacenamiento. Esto 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 no concuerdan entre sí, por ejemplo: que se actualiza la dirección de un cliente en un archivo y que en otros archivos permanezca la anterior. [28]

5.3.3.4 Integridad de los datos

Nos interesará que los SGBD aseguren el mantenimiento de la calidad de los datos en cualquier circunstancia. Acabamos de ver que la redundancia puede provocar pérdida de integridad de los datos, pero no es la única causa posible. Se podría perder la corrección o la consistencia de los datos por muchas otras



razones: errores de programas, errores de operación humana, avería de disco, transacciones incompletas por corte de alimentación eléctrica, etc. [28,30]

5.3.3.5 Concurrencia de los usuarios

Un objetivo fundamental de los SGBD es permitir que varios usuarios puedan acceder concurrentemente a la misma BD. Cuando los accesos concurrentes son todos de lectura (es decir, cuando la BD sólo se consulta), el problema que se produce es simplemente de rendimiento, causado por las limitaciones de los soportes de que se dispone: pocos mecanismos de acceso independientes, movimiento del brazo y del giro del disco que son demasiado lentos, buffers locales demasiado pequeños, etc. Cuando un usuario o más de uno están actualizando los datos, se pueden producir problemas de interferencia que tengan como consecuencia la obtención de datos erróneos y la pérdida de integridad de la BD. [25]

Para tratar los accesos concurrentes, los SGBD utilizan el concepto de transacción de BD, concepto de especial utilidad para todo aquello que hace referencia a la integridad de los datos. [25,26]

5.3.3.6 Seguridad

La información de toda empresa es importante, aunque unos datos lo son más que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un sistema de base de datos sea confiable debe mantener un grado de seguridad que garantice la autenticación y protección de los datos. En un banco por ejemplo, el personal de nóminas sólo necesita ver la parte de la base de datos que tiene información acerca de los distintos empleados del banco y no a otro tipo de información. [27]

5.3.3.7 Otros objetivos

Acabamos de ver los objetivos fundamentales de los SGBD actuales. Sin embargo, a medida que los SGBD evolucionan, se imponen nuevos objetivos adaptados a las nuevas necesidades y las nuevas tecnologías. Como ya hemos visto, en estos momentos podríamos citar como objetivos nuevos o recientes los siguientes:

- 1) Servir eficientemente los Data Warehouse.
- 2) Adaptarse al desarrollo orientado a objetos.



3) Incorporar el tiempo como un elemento de caracterización de la información.

4) Adaptarse al mundo de Internet. [28]

5.3.4 Sistema Administrador de Bases de Datos

Un sistema de administración de bases de datos DBMS (Data base Management System, por sus siglas en Inglés) es un sistema basado en computador (software) que maneja una base de datos, o una colección de bases de datos o archivos. La persona que administra un DBMS es conocida como el DBA (Data base Administrator, por sus siglas en ingles). [21]

5.3.4.1 Usos y funciones DBMS

Los sistemas de administración de bases de datos son usados para:

- ✓ Permitir a los usuarios acceder y manipular la base de datos proveyendo métodos para construir sistemas de procesamiento de datos para aplicaciones que requieran acceso a los datos.
- ✓ Proveer a los administradores las herramientas que les permitan ejecutar tareas de mantenimiento y administración de los datos.

Algunas de las funciones de un DBMS son:

- ✓ Definición de la base de datos - como la información va a ser almacenada y organizada.
- ✓ Creación de la base de datos - almacenamiento de datos en una base de datos definida.
- ✓ Recuperación de los datos - consultas y reportes.
- ✓ Actualización de los datos - cambiar los contenidos de la base de datos.
- ✓ Programación de aplicaciones de para el desarrollo de software.
- ✓ Control de la integridad de la base de datos.
- ✓ Monitoreo del comportamiento de la base de datos. [23]

5.3.5 Modelo de Bases de Datos

Una BD es una representación de la realidad (de la parte de la realidad que nos interesa en nuestro SI). Dicho de otro modo, una BD se puede considerar un modelo de la realidad. El componente fundamental utilizado para modelar en un SGBD relacional son las tablas (denominadas relaciones en el mundo teórico). [28]

El conjunto de componentes o herramientas conceptuales que un SGBD proporciona para modelar recibe el nombre de modelo de BD. Los cuatro modelos de BD más utilizados en los SI son el modelo relacional, el modelo jerárquico, el modelo en red y el modelo relacional con objetos.

Todo modelo de BD nos proporciona tres tipos de herramientas:

- a) Estructuras de datos con las que se puede construir la BD: tablas, árboles, etc.
- b) Diferentes tipos de restricciones (o reglas) de integridad que el SGBD tendrá que hacer cumplir a los datos: dominios, claves, etc.
- c) Una serie de operaciones para trabajar con los datos. Un ejemplo de ello, en el modelo relacional, es la operación SELECT, que sirve para seleccionar (o leer) las filas que cumplen alguna condición. Un ejemplo de operación típica del modelo jerárquico y del modelo en red podría ser la que dice si un determinado registro tiene “hijos” o no. [28, 24,21]

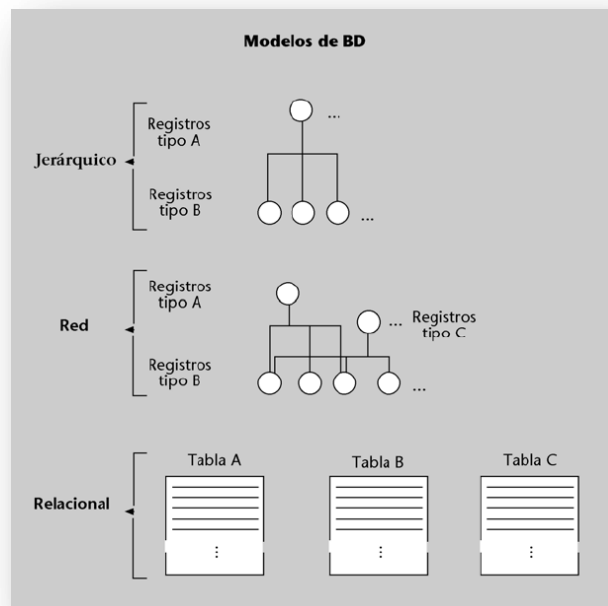


Figura 31 Modelo de Bases de Datos. [28]

5.3.5.1 Modelo Relacional

El modelo relacional es un modelo de datos y, como tal, tiene en cuenta los tres aspectos siguientes de los datos:

- 1) La estructura, que debe permitir representar la información que nos interesa del mundo real.



2) La manipulación, a la que da apoyo mediante las operaciones de actualización y consulta de los datos.

3) La integridad, que es facilitada mediante el establecimiento de reglas de integridad; es decir, condiciones que los datos deben cumplir.

El principal objetivo del modelo de datos relacional es facilitar que la base de datos sea percibida o vista por el usuario como una estructura lógica que consiste en un conjunto de relaciones y no como una estructura física de implementación. Esto ayuda a conseguir un alto grado de independencia de los datos. [29, 21, 22]

Un objetivo adicional del modelo es conseguir que esta estructura lógica con la que se percibe la base de datos sea simple y uniforme. Con el fin de proporcionar simplicidad y uniformidad, toda la información se representa de una única manera: mediante valores explícitos que contienen las relaciones (no se utilizan conceptos, como por ejemplo, apuntadores entre las relaciones). Con el mismo propósito, todos los valores de datos se consideran atómicos; es decir, no es posible descomponerlos. [25, 28]

5.3.5.1.1 Estructura de los Datos

El modelo relacional proporciona una estructura de los datos que consiste en un conjunto de relaciones con objeto de representar la información que nos interesa del mundo real. [30]

Una base de datos relacional consta de un conjunto de relaciones, cada una de las cuales se puede visualizar de este modo tan sencillo. []

La estructura de los datos del modelo relacional resulta fácil de entender para el usuario. []

La relación es el elemento básico en el modelo relacional. Las relaciones se definen de la siguiente manera:

ESTUDIANTE (Clave, Nombre, Nacionalidad, Institución)

y se puede representar como una tabla, como vemos en el siguiente ejemplo



Cuadro 2 Estructura Básica del Modelo Relacional, tabla estudiante.

Clave	Nombre	Nacionalidad	Institución
01	Pepe	México	U.A.E.M
02	John	EE.UU.	U.N.A.M
03	Pierre	Francia	U.A.C.L.A

Se puede obtener una buena idea intuitiva de lo que es una relación si la visualizamos como una tabla. En la tabla 2 se muestra la visualización tabular de una relación que contiene datos de estudiantes. Cada fila de la tabla contiene una colección de valores de datos relacionados entre sí; en nuestro ejemplo, son los datos correspondientes a un mismo estudiante. La tabla tiene un nombre (ESTUDIANTES) y también tiene un nombre cada una de sus columnas (Clave, Nombre, Nacionalidad, Institución). El nombre de la tabla y los de las columnas ayudan a entender el significado de los valores que contiene la tabla. Cada columna contiene valores de un cierto dominio; por ejemplo, la columna Clave contiene valores del dominio Clave.

5.3.5.1.1.1 Características y Definición de una Tabla

Las tablas se denominan entidades y están formadas por un conjunto de *tuplas* o instancias de cada relación de datos atómicos, llamados *dominios*.

5.3.5.1.1.1.1 Tupla

A menudo se le llama también *registro* o *fila*, físicamente es cada una de las líneas de la relación. Equivale al concepto de entidad del modelo E-R, y define un objeto real, ya sea abstracto, concreto o imaginario

5.3.5.1.1.1.2 Dominios

Un dominio D es un conjunto de valores atómicos. Por lo que respecta al modelo relacional, *atómico* significa indivisible; es decir, que por muy complejo o largo que sea un valor atómico, no tiene una estructuración interna para un SGBD relacional. Por ejemplo el dominio "nacionalidades" tiene valores: Española, francés, Italiana,... etc.



5.3.5.1.1.3 Atributo

También denominado *campo* o *columna*, corresponde con las divisiones verticales de la relación. Corresponde al concepto de atributo del modelo E-R y contiene cada una de las características que definen una entidad u objeto.

Cada una de las características que posee una entidad, y que agrupadas permiten distinguirla de otras entidades del mismo conjunto.

Al igual que en el modelo E-R, cada atributo tiene asignado un nombre y un dominio. El conjunto de todos los atributos es lo que define a una entidad completa, y es lo que compone una Tupla.

5.3.5.1.1.2 Llaves Primarias.

Una llave primaria es aquel atributo el cual consideramos clave para la identificación de los demás atributos que describen a la entidad. Por ejemplo, si consideramos la entidad ALUMNO del Instituto Tecnológico de La Paz, podríamos tener los siguientes atributos: Nombre, Semestre, Especialidad, Dirección, Teléfono, Número de control, de todos estos atributos el que podremos designar como llave primaria es el número de control, ya que es diferente para cada alumno y este nos identifica en la institución.

5.3.5.1.1.3 Llaves Secundarias

Una llave es llamada secundaria si no puede identificar en forma única a un registro. Las llaves secundarias pueden usarse para seleccionar un grupo de registros que pertenecen a un conjunto.

Cuando no es posible identificar unos registros en forma única mediante el uso de uno de los conceptos de datos que se encuentran en un registro, se puede construir una clave seleccionando dos o más atributos y combinándolos. A esto se le llama una llave concatenada.

5.3.5.2 Modelo Entidad-Relación.

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relaciones entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema que representa la estructura lógica completa de una base de datos. El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo yace en la representación del significado de los datos. El modelo E-R es extremadamente



útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R.

5.3.5.2.1 Entidad

Una entidad es una cosa u objeto en el mundo real que es distinguible de todos los demás objetos. Por ejemplo, cada persona en un desarrollo es una entidad. Una entidad tiene un conjunto de propiedades, y los valores para algún conjunto de propiedades pueden identificar una entidad de forma unívoca. Por ejemplo, la clave 03259 identifica unívocamente una persona particular en la empresa. Análogamente, se puede pensar en los préstamos bancarios como entidades, y un número de préstamo P-15 en la sucursal de Banamex identifica unívocamente una entidad de préstamo. Una entidad puede ser concreta, como una persona o un libro, o puede ser abstracta, como un préstamo, unas vacaciones o un concepto.

5.3.5.2.2 Relaciones

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan entidades participantes. El número de participantes en una relación es lo que se denomina grado de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación binaria; si son tres las entidades participantes, la relación es ternaria; etc. Una relación recursiva es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

5.3.5.2.3 Atributos

Los atributos describen propiedades que posee cada miembro de un conjunto de entidades. La designación de un atributo para un conjunto de entidades expresa que la base de datos almacena información similar concerniente a cada entidad del conjunto de entidades; sin embargo, cada entidad puede tener su propio valor para cada atributo. Posibles atributos del conjunto de entidades cliente son id-cliente, nombre-cliente, calle-cliente y ciudad-cliente. En la vida real, habría más atributos, tales como el número de la calle, el número del portal, la provincia, el código postal, y la comunidad

autónoma, pero no se incluyen en el ejemplo simple. Posibles atributos del conjunto de entidades préstamo son número-préstamo e importe.

5.3.5.2.4 Notación del Diagrama Entidad-Relación.

<u>Símbolo</u>	<u>Significado</u>	<u>Símbolo</u>	<u>Significado</u>
	Tipo de Entidades		Atributo Compuesto
	Tipo de Entidades Débil		Atributo Derivado
	Tipo de vinculos (relaciones)		Participación total de E2 en R
	Tipo de vínculo identificador		Razón de cardinalidad 1:N para E1:E2 en R
	Atributo		
	Atributo Clave		
	Atributo multievaluado		

Figura 32 Notación Básica del Modelo Entidad-Relación

5.3.5.2.5 Cardinalidad.

Expresa el número máximo de entidades que están relacionadas con una única entidad del otro conjunto de entidades que interviene en la relación. Aunque normalmente nos interesa sólo la Cardinalidad máxima, a veces es útil especificar la Cardinalidad mínima.

5.3.5.2.5.1 Uno a uno

Cuando un registro de una tabla sólo puede estar relacionado con un único registro de la otra tabla y viceversa.

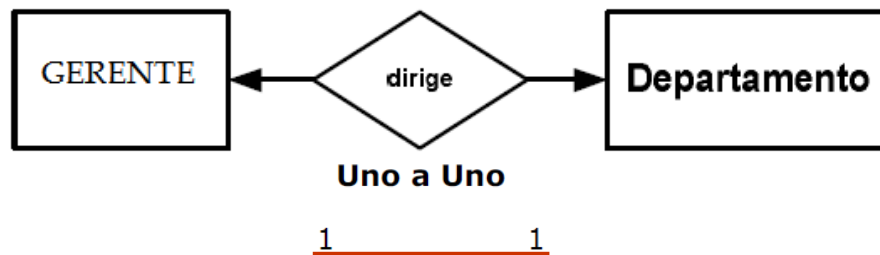


Figura 33 Cardinalidad de Uno a Uno

5.3.5.2.5.2 Uno a Muchos

Cuando un registro de una tabla (tabla secundaria) sólo puede estar relacionado con un único registro de la otra tabla (tabla principal) y un registro de la tabla principal puede tener más de un registro relacionado en la tabla secundaria.



Figura 34 Relación de Uno a Muchos

5.3.5.2.5.3 Muchos a Muchos

Cuando un registro de una tabla puede estar relacionado con más de un registro de la otra tabla y viceversa. En este caso las dos tablas no pueden estar relacionadas directamente, se tiene que añadir una tabla entre las dos (Tabla débil o de vinculación) que incluya los pares de valores relacionados entre sí.

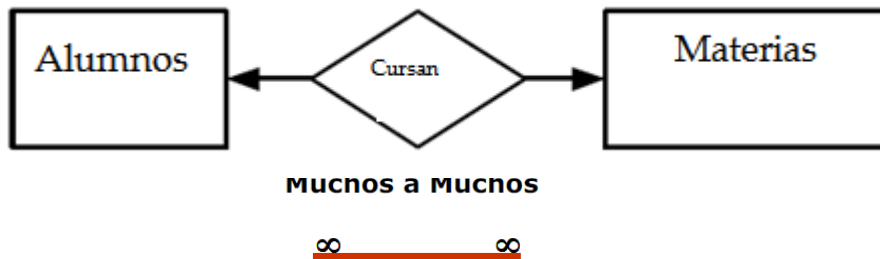


Figura 35 Relación de Muchos a Muchos

5.3.6 Normalización de Bases de Datos.

5.3.6.1 Concepto de Normalización.

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataban de manipular los datos.



La normalización también hace las cosas fáciles de entender. Los seres humanos tenemos la tendencia de simplificar las cosas al máximo. Lo hacemos con casi todo, desde los animales hasta con los automóviles. Vemos una imagen de gran tamaño y la hacemos más simple agrupando cosas similares juntas. Las guías que la normalización provee crean el marco de referencia para simplificar una estructura de datos compleja.

Otra ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco.

5.3.6.2 Primera Forma Normal (1NF)

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

Poner la base de datos en la Primera Forma Normal resuelve el problema de los encabezados de columna múltiples. Muy a menudo, los diseñadores de bases de datos inexpertos harán algo similar a la tabla no normalizada. Una y otra vez, crearán columnas que representen los mismos datos. La normalización ayuda a clarificar la base de datos y a organizarla en partes más pequeñas y más fáciles de entender. En lugar de tener que entender una tabla gigantesca y monolítica que tiene muchos diferentes aspectos, sólo tenemos que entender los objetos pequeños y más tangibles, así como las relaciones que guardan con otros objetos también pequeños.

5.3.6.3 Segunda Forma Normal (2NF)

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos.

Una vez alcanzado el nivel de la Segunda Forma Normal, se controlan la mayoría de los problemas de lógica. Podemos insertar un registro sin un exceso de datos en la mayoría de las tablas.

5.3.6.4 Tercera Forma Normal (3NF)

Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Comentamos anteriormente que una dependencia transitiva es aquella en la cual existen columnas que no son llave que dependen



de otras columnas que tampoco son llave. Cuando las tablas están en la Tercera Forma Normal se previenen errores de lógica cuando se insertan o borran registros. Cada columna en una tabla está identificada de manera única por la llave primaria, y no debe haber datos repetidos. Esto provee un esquema limpio y elegante, que es fácil de trabajar y expandir.

5.4 SQL

5.4.1 Introducción.

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft Jet. SQL se utiliza para crear objetos QueryDef, como el argumento de origen del método OpenRecordSet y como la propiedad RecordSource del control de datos. También se puede utilizar con el método Execute para crear y manipular directamente las bases de datos Jet y crear consultas SQL de paso a través para manipular bases de datos remotas cliente - servidor.

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

5.4.2 Comandos SQL.

Existen dos tipos de comandos SQL:

- los DDL que permiten crear y definir nuevas bases de datos, campos e índices.
- los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

5.4.2.1 Comandos DDL.

CREATE:

Utilizado para crear nuevas tablas, campos e índices

DROP:

Empleado para eliminar tablas e índices

ALTER:

Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.



5.4.2.2 Comandos DML

SELECT:

Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.

INSERT:

Utilizado para cargar lotes de datos en la base de datos en una única operación.

UPDATE:

Utilizado para modificar los valores de los campos y registros especificados.

DELETE:

Utilizado para eliminar registros de una tabla de una base de datos.

5.4.3 Clausulas.

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

FROM:

Utilizada para especificar la tabla de la cual se van a seleccionar los registros.

WHERE:

Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar

GROUP BY:

Utilizada para separar los registros seleccionados en grupos específicos.

HAVING:

Utilizada para expresar la condición que debe satisfacer cada grupo.

ORDER BY:

Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico.

5.4.4 Operadores Lógicos.

AND:

Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.



OR:

Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.

NOT:

Negación lógica. Devuelve el valor contrario de la expresión.

5.4.5 Operadores de Comparación.

< Su uso es: Menor que

> Su uso es: Mayor que

<> Su uso es: Distinto de

<= Su uso es: Menor ó Igual que

>= Su uso es: Mayor ó Igual que

= Su uso es: Igual que

BETWEEN: Utilizado para especificar un intervalo de valores.

LIKE: Utilizado en la comparación de un modelo.

In: Utilizado para especificar registros de una base de datos.

5.4.6 Funciones de Agregado.

Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

AVG:

Utilizada para calcular el promedio de los valores de un campo determinado.

COUNT:

Utilizada para devolver el número de registros de la selección.

SUM:

Utilizada para devolver la suma de todos los valores de un campo determinado.

MAX:

Utilizada para devolver el valor más alto de un campo especificado

MIN:



Utilizada para devolver el valor más bajo de un campo especificado.

5.4.7 Consultas.

Las consultas de selección se utilizan para indicar al motor de datos que devuelva información de las bases de datos, esta información es devuelta en forma de conjunto de registros que se pueden almacenar en un objeto recordset. Este conjunto de registros es modificable.

5.4.7.1 Consultas Básicas.

La sintaxis básica de una consulta de selección es la siguiente:

```
SELECT Campos FROM Tabla;
```

En donde campos es la lista de campos que se deseen recuperar y tabla es el origen de los mismos, por ejemplo:

```
SELECT Nombre, Teléfono FROM Clientes;
```

Esta consulta devuelve un recordset con el campo nombre y teléfono de la tabla clientes.

5.4.7.2 Ordenar Registros.

Adicionalmente se puede especificar el orden en que se desean recuperar los registros de las tablas mediante la cláusula *ORDER BY* Lista de Campos. En donde Lista de campos representa los campos a ordenar. Ejemplo:

```
SELECT CodigoPostal, Nombre, Teléfono FROM Clientes ORDER BY  
Nombre;
```

Esta consulta devuelve los campos CodigoPostal, Nombre, Teléfono de la tabla Clientes ordenados por el campo Nombre. Se pueden ordenar los registros por mas de un campo, como por ejemplo:

```
SELECT CodigoPostal, Nombre, Teléfono FROM Clientes ORDER BY  
CodigoPostal, Nombre;
```

Incluso se puede especificar el orden de los registros: ascendente mediante la cláusula (*ASC* -se toma este valor por defecto) ó descendente (*DESC*)

```
SELECT CodigoPostal, Nombre, Teléfono FROM Clientes ORDER BY  
CodigoPostal DESC, Nombre ASC;
```



5.4.7.3 Consultas con Predicado.

El predicado se incluye entre la cláusula y el primer nombre del campo a recuperar, los posibles predicados son:

ALL:

Devuelve todos los campos de la tabla.

TOP:

Devuelve un determinado número de registros de la tabla.

DISTINCT:

Omite los registros cuyos campos seleccionados coincidan totalmente.

DISTINCTROW:

Omite los registros duplicados basándose en la totalidad del registro y no sólo en los campos seleccionados.

5.4.7.4 La cláusula where.

La cláusula WHERE puede usarse para determinar qué registros de las tablas enumeradas en la cláusula FROM aparecerán en los resultados de la instrucción SELECT. Después de escribir esta cláusula se deben especificar las condiciones expuestas en los partados 3.1 y 3.2. Si no se emplea esta cláusula, la consulta devolverá todas las filas de la tabla. WHERE es opcional, pero cuando aparece debe ir a continuación de FROM.

SELECT Apellidos, Salario **FROM** Empleados **WHERE** Salario > 21000;

SELECT Id Producto, Existencias **FROM** Productos **WHERE** Existencias <= Nuevo_Pedido;

SELECT * **FROM** Pedidos **WHERE** Fecha_Envio = #5/10/94#;

5.4.8 Consultas de acción.

Las consultas de acción son aquellas que no devuelven ningún registro, son las encargadas de acciones como añadir y borrar y modificar registros.

5.4.8.1 Delete

Crea una consulta de eliminación que elimina los registros de una o más de las tablas listadas en la cláusula FROM que satisfagan la cláusula WHERE. Esta



consulta elimina los registros completos, no es posible eliminar el contenido de algún campo en concreto. Su sintaxis es:

```
DELETE Tabla.* FROM Tabla WHERE criterio
```

DELETE es especialmente útil cuando se desea eliminar varios registros. En una instrucción DELETE con múltiples tablas, debe incluir el nombre de tabla (Tabla.*). Si especifica más de una tabla desde la que eliminar registros, todas deben ser tablas de muchos a uno. Si desea eliminar todos los registros de una tabla, eliminar la propia tabla es más eficiente que ejecutar una consulta de borrado.

Se puede utilizar DELETE para eliminar registros de una única tabla o desde varios lados de una relación uno a muchos. Las operaciones de eliminación en cascada en una consulta únicamente eliminan desde varios lados de una relación. Por ejemplo, en la relación entre las tablas Clientes y Pedidos, la tabla Pedidos es la parte de muchos por lo que las operaciones en cascada solo afectarán a la tabla Pedidos. Una consulta de borrado elimina los registros completos, no únicamente los datos en campos específicos. Si desea eliminar valores en un campo especificado, crear una consulta de actualización que cambie los valores a Null.

5.4.8.2 Insert into

Agrega un registro en una tabla. Se la conoce como una consulta de datos añadidos. Esta consulta puede ser de dos tipos:

Insertar un único registro ó Insertar en una tabla los registros contenidos en otra tabla.

5.4.8.3 UPDATE

Crea una consulta de actualización que cambia los valores de los campos de una tabla especificada basándose en un criterio específico. Su sintaxis es:

```
UPDATE Tabla SET Campo1=Valor1, Campo2=Valor2, ...  
CampoN=ValorN  
WHERE Criterio;
```

UPDATE es especialmente útil cuando se desea cambiar un gran número de registros o cuando éstos se encuentran en múltiples tablas. Puede cambiar varios campos a la vez. El ejemplo siguiente incrementa los valores Cantidad pedidos en un 10 por ciento y los valores Transporte en un 3 por ciento para aquellos que se hayan enviado al Reino Unido.:



UPDATE Pedidos *SET* Pedido = Pedidos * 1.1, Transporte =
Transporte * 1.03
WHERE PaisEnvío = 'ES';

5.4.9 Instrucción SQL

Es una instrucción *SELECT*, que sigue el mismo formato y reglas que cualquier otra instrucción *SELECT*. Debe ir entre paréntesis. Se puede utilizar una subconsulta en lugar de una expresión en la lista de campos de una instrucción *SELECT* o en una cláusula *WHERE* o *HAVING*. En una subconsulta, se utiliza una instrucción *SELECT* para proporcionar un conjunto de uno o más valores especificados para evaluar en la expresión de la cláusula *WHERE* o *HAVING*. Se puede utilizar el predicado *ANY* o *SOME*, los cuales son sinónimos, para recuperar registros de la consulta principal, que satisfagan la comparación con cualquier otro registro recuperado en la subconsulta.

El ejemplo siguiente devuelve todos los productos cuyo precio unitario es mayor que el de cualquier producto vendido con un descuento igual o mayor al 25 por ciento.:

SELECT * *FROM* Productos *WHERE* PrecioUnidad > ANY

(*SELECT* PrecioUnidad *FROM* DetallePedido *WHERE* Descuento >= 0 .25);

VI. METODOLOGIA

6.1 Pasos metodológicos

Paso 1: Encuesta y entrevista a la gerente de la empresa.

Paso 2: Análisis de requerimientos de la empresa para el sistema.

Paso 3: Identificación de requerimientos de reportes y datos.

Paso 4: Modelaje de los casos de uso del sistema.

Paso 5: Normalización de la base de datos.

Paso 6: Esquema de la base de datos relacional.



Paso 7: Implementación física de la base de datos.

Paso 8: Desarrollo de las vistas.

Paso 9: Planteamiento de los algoritmos.

Paso 10: Programación de los algoritmos.

Paso 11: Estructuración del sistema de información

Paso 12: Validación del sistema de información.

Paso 13: Liberación del sistema de información.

6.2 Desarrollo de pasos metodológicos.

6.3 Herramientas.

Para el desarrollo del sistema Sellsys1.0 se utilizó el Lenguaje de programación Visual Basic 6.0 Edición Profesional que permite la programación que tiene las siguientes características:

- ✓ Visual Basic 6.0 es un lenguaje de programación visual, también llamado lenguaje de 4ª generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.
- ✓ Visual Basic 6.0 utiliza objetos con propiedades y métodos, pero carece de los mecanismos de herencia y polimorfismo propios de los verdaderos lenguajes orientados a objetos como Java y C++.
- ✓ Permite usar con facilidad la plataforma de los sistemas Windows dado que tiene acceso prácticamente total a la API de Windows incluidas en librerías actuales.
- ✓ Sirve para generar de manera automática formularios que administran registros de tablas o consultas pertenecientes a una base de datos, hoja de cálculo u objeto (ADO-ACTIVE DATA OBJECT).
- ✓ Las interfaces múltiples son una característica del modelo de objetos componente (COM) y permiten que los programas evolucionen con el tiempo, agregando nueva funcionalidad sin afectar al código existente.
- ✓ Los objetos de Visual Basic están encapsulados; es decir, contienen su propio código y sus propios datos.



Visual Basic 6.0 Edición Profesional es una versión comercial que solo se puede encontrar con licencia y se puede instalar en cualquier sistema operativo.

El sistema Manejador de Bases de Datos que se utilizó en la creación de Sellsys1.0 fue MySQL 5.1 que permite la creación de tablas diseñadas con el modelo Relacional.

Para el enlace con Visual Basic se utilizó el MySQL ODBC connector 5.1.4 para permitir la conexión entre el sistema “sellsys1.0” y el sistema gestor de bases de datos “MySQL”.

Las principales características de estos componentes son:

- ✓ Portabilidad entre sistemas.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- ✓ Gestión de usuarios y password, manteniendo un muy buen nivel de seguridad en los datos.
- ✓ Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
- ✓ Fácil instalación y configuración.
- ✓ MySQL, es el manejador de base de datos considerado como el más rápido de Internet.
- ✓ Facilidad de uso.
- ✓ Condición de open source de MySQL hace que la utilización sea gratuita y se puede modificar con total libertad

Se utilizó DBDesigner4 en la creación del modelo relacional y los Scripts SQL.

DBDesigner 4 es un sistema visual de diseño de bases de datos que integra el diseño de bases de datos, modelado, creación y mantenimiento en un ambiente sencillo y sin complicaciones. Se distribuye bajo la Licencia Pública General.

DBDesigner 4 está desarrollado y optimizado para la Base de datos de código abierto MySQL para dar soporte a usuarios de My SQL con una herramienta de diseño poderosa y disponible en forma gratuita.

6.4 Casos de Uso en SELLSYS1.0

Para modelar el sistema se SELLSYS 1.0 se empleo el UML. En la siguiente figura 36 se muestra el caso de uso buscar que funciona con el botón BUSCAR que se encuentra en las interfaces del sistema, donde se hace la interacción con el usuario y el sistema, por ejemplo en la acción borrar un registro, primero se debe especificar que registro es el que se busca, para que el sistema lo encuentre y muestre al usuario.

6.4.1 Caso de Uso Buscar registro.

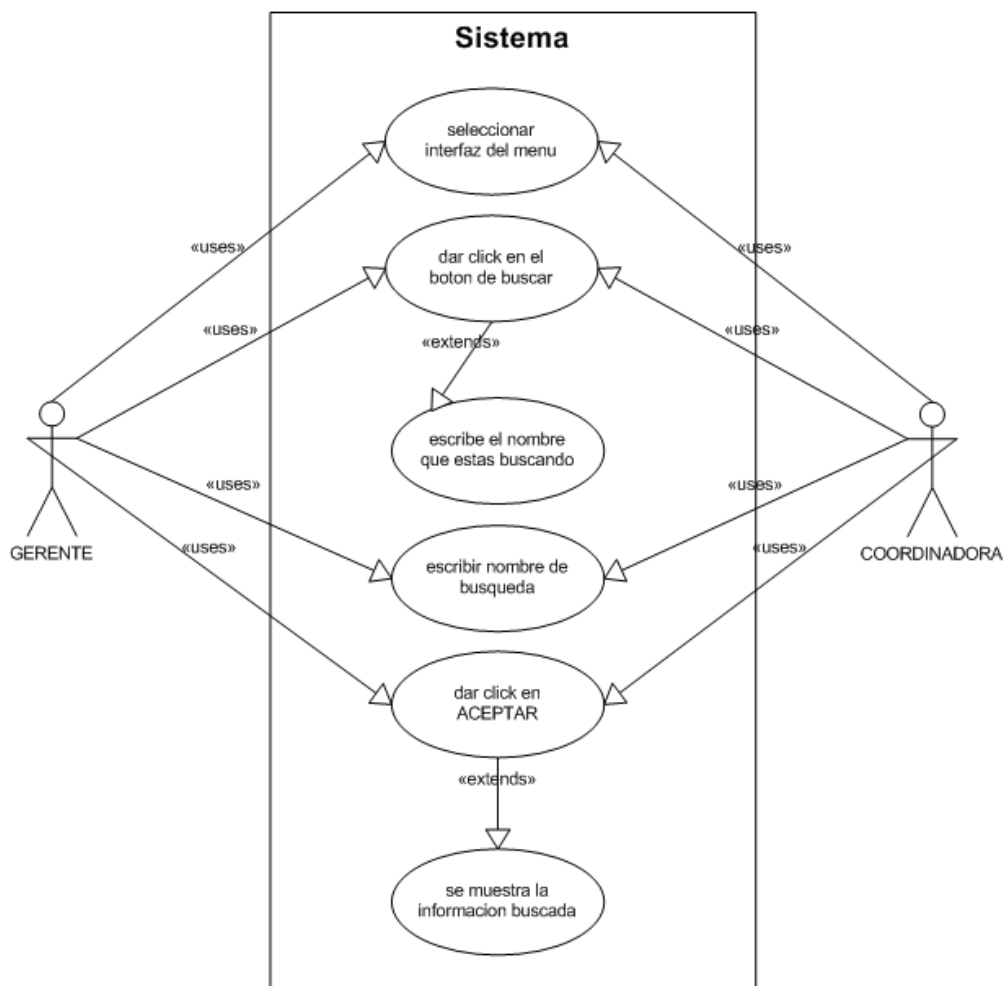


Figura 36 Caso de uso Buscar

6.4.2 Caso de Uso dar de Alta un registro.

En la figura 37 se muestra el funcionamiento del botón de ALTA en las interfaces, es decir que cuando se desee dar de ALTA un registro primero se deben de llenar los espacios de manera que no queden espacios en blanco, después se da click en el botón de alta y el sistema mostrará un recuadro con la leyenda, el registro se dio de alta; donde la base de datos se actualiza, después presione el icono superior de la interfaz que dice GUARDAR.

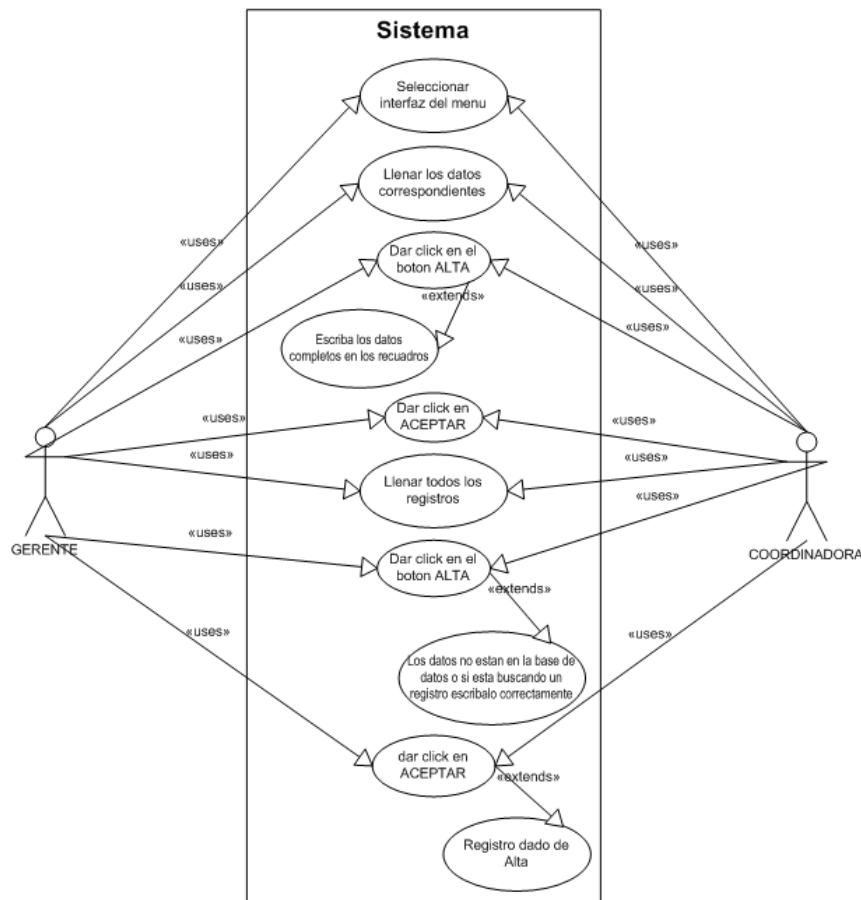


Figura 37 Caso de uso Alta

6.4.3 Caso de Uso Cambiar un registro.

La figura 38 modela el funcionamiento del botón CAMBIOS, antes es necesario realizar la búsqueda del registro al que se desea hacer cambios, cuando ya está editado en pantalla el registro deseado, se realizan los cambios necesarios, enseguida se presiona el botón de cambios, el sistema enviara un recuadro que diga que los cambios se han efectuado correctamente.

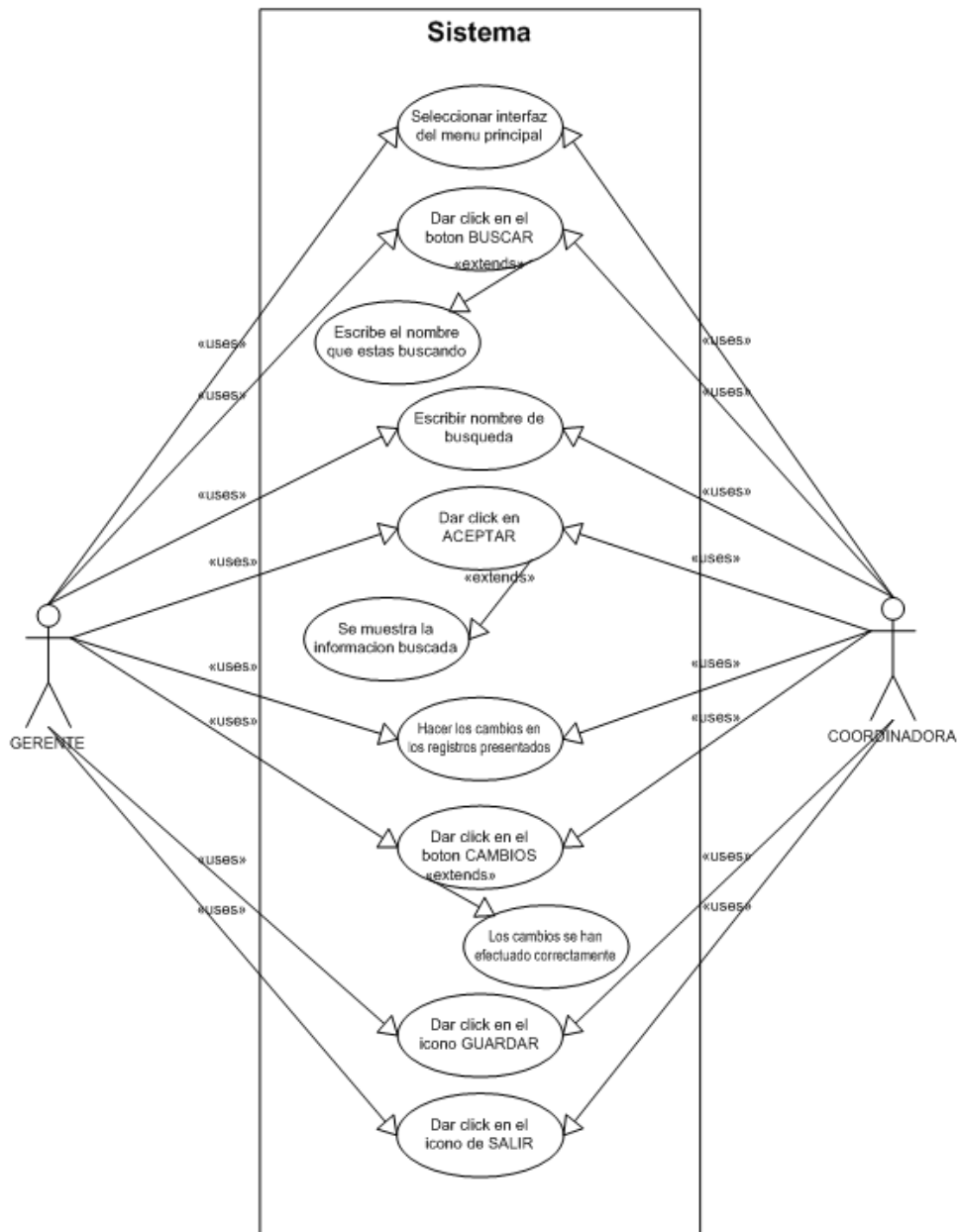


Figura 38 Caso de Uso Cambios.

6.4.4 Caso de uso Eliminar un registro.

En la figura 39 muestra el caso de uso del botón ELININAR, donde antes hay que realizar la búsqueda del registro que se desee eliminar, una vez editado el registro se podrá eliminar el registro, antes se mostrara un recuadro en donde se cuestiona al usuario si en realidad está seguro de que se elimine el registro para posteriormente visualizar en pantalla un cuadro que diga que se ha eliminado dicho registro.

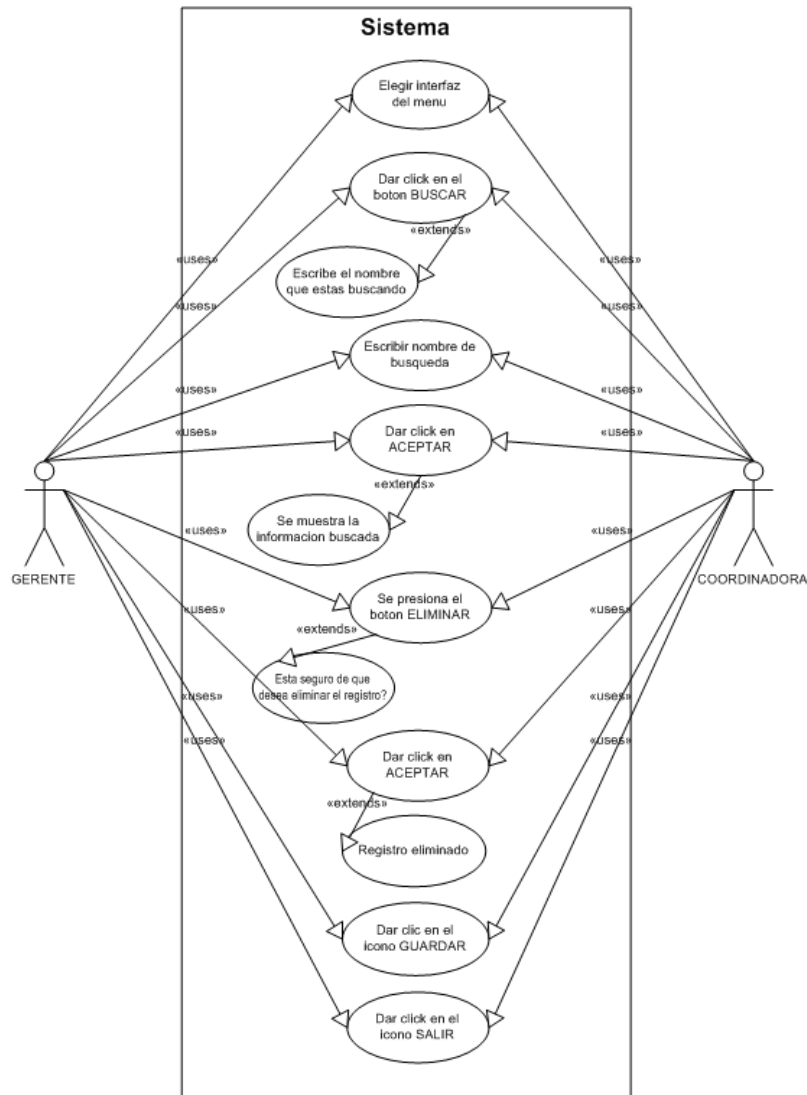


Figura 39 Caso de uso Eliminar.

6.4.5 Caso de uso Hacer un reporte.

Para entrar a la interfaz de los reportes es necesario hacerlo desde el menú principal, una vez estando en dicha interfaz, se seleccionara el que este más acorde a las necesidades del usuario; una vez seleccionada la consulta, se da click en el botón de consulta que se ha elegido, se tiene que especificar una fecha, de otra manera se hará una consulta generalizada; estos reportes se pueden imprimir al dar click en el icono imprimir (figura 40)

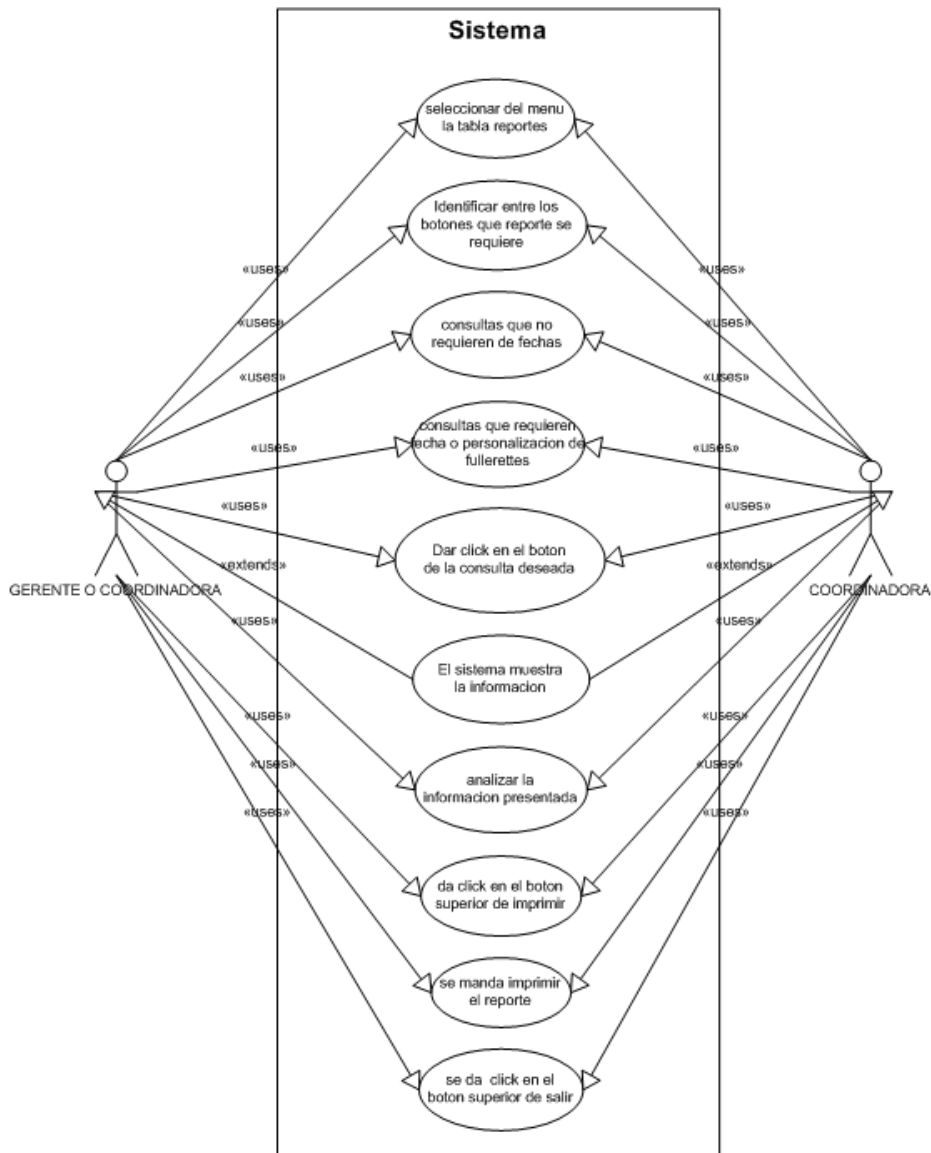


Figura 40 Caso de uso Reportes.

6.5 Diseño de la Base de Datos.

Para el diseño de la base de datos, primero se determinaron los requerimientos del cliente y se elaboro una lista con todos los datos que contiene son necesarios para cumplir con las necesidades del sistema que requiere la empresa. Estos datos se reflejan en la figura 41.



"DESARROLLO DEL SISTEMA DE INFORMACIÓN (SELLSYS1.0) PARA EL CONTROL DE VENTAS DE LA EMPRESA FULLER COSMETICS"

ventas_neta	nom_fuller	id_full	Desc_Ver	saldo_full	premio_ganado_full	dir_full	tel_full	alta_full	baja_full	campana_de_venta
5,642.00	Rosio Anda	178546-6	1692.6	12681.3	viaje bahamas	Abasolo 121	9564137	14/05/1997	16/08/2008	C-7
23654	Petra Solis	162543-9	7096.2	15612.2	computadora	Silverio Perez 53	9546572	02/01/2000	-	C-9
8,972.00	Carmen Diaz	165223-9	2691.6	5971.25	batidora/licuadora	Colon 431	9587614	15/09/2003	-	C-9

dir_coord	tel_coord	ID_coord	id_gerente	desc_cambios	no_antico	cant_antico	cant_cobranz	desc_cobranza	id_cobranza
chapulin 65	5564897216	32658974-94	45879-6	32.56	56	5699	126546.98	56.88	1354
Canto 165	5598764170	26547891-12	26548-9	21.78	23	7498	49876.32	97.4	5698
MA. Sorte 10	5532156479	65877916-03	23647-8	56.89	98	3498.2	65879.1	55.77	7854

desc_cambios	cant_cambid	id_mapas_tr	premio_gana	ventas_parce	prom_ventas	convencion_coor	fecha_ventas	prom_ventas	id_ventas	id_distribuid	nom_distribuidor	zona_distribuid
123.88	89.99	2569	colcha	5987	67%	Los Angeles	12/05/2008	30%	25478961	37894	Rogelio	9
65.71	56.99	4587	plancha	5698	56%	Chicago	30/07/2007	45%	32654780	36587	Camilo	8
59.66	79.99	6981	colcha	1456	82%	Brasil	04/06/2005	66%	13085970	32159	Efrain	3

nom_gerente	dir_gerente	tel_gerente	desc_promo_full	desc_promo_coor	No_factura	Can_factura	concepto_fa	desc_memo	nom_coord	dir_coord
Susana Navarro	Patricia Ocampo	045-5536412879	Jelly Roll	crema facial	125487-96	11,854.55	venta C-3	32.56	Carmen Soto	chapulin 65
Susana Navarro	Patricia Ocampo	045-5536412879	Nekara	crema facial	365489-27	16267.21	venta C-8	45.87	Andrea Torre	Canto 165
Susana Navarro	Patricia Ocampo	045-5536412879	Beautyvert	colonia dama	215473-98	9854.61	venta C-8	29.88	Juana Perez	MA. Sorte 10

ID_premios	id_promo	distribuidor_coo	id_informes	id_estado	saldo_coord	id_factura
125478963	502	Juana Perez	6985134	1564	56897.5	12548796
365478952	608	Silvia Barba	2654781	5987	109568.87	25648791
264590873	739	Petra Ramirez	9582246	3651	42761.3	86435179

Figura 41 Datos requeridos para SELLSYS1.0

Con los datos se procedió a realizar la Normalización (1FN, 2FN Y 3FN).

6.5.1 Primera Forma Normal SELLSYS1.0

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas.

Tabla de Ventas. (1FN)

Ventas									
ventas_neta	desc_ventas	campana_d	id_venta	prom_ven	prom_v	ventas_parcia	fecha_venta	estimado_de_ventas	
5,642.00	1692.6	C-7	564	30%	67%	5987	12/05/2008	25000	
23654	7096.2	C-9	321	45%	56%	5698	30/07/2007	25000	
8,972.00	2691.6	C-9	982	66%	82%	1456	04/06/2005	25000	

Tabla de Fullerette. (1FN)

nom_fuller	id_full	saldo_full	premio_ganado_fu	desc_promo_full	dir_full	tel_full	alta_full	baja_full	distribuidor_fu	zona_full
Rosio Anda	178546-6	12681.3	Jelly Roll	Jelly Roll	Abasolo 121	9564137	14/05/1997	16/08/2008	coordinadora	9
Petra Solis	162543-9	15612.15	Nekara	Nekara	Silverio Perez 53	9546572	02/01/2000	-	camioneta	7
Carmen Diaz	165223-9	5971.25	Beautyvert	Beautyvert	Colon 431	9587614	15/09/2003	-	camioneta	16



Tabla de Distribuidor. (1FN)

distribuidor_f	Estimado_venta	desc_info_coor	no_pedido	desc_cancelacio	no_cancelacio	id_cancelac	division_coor	distrito_gere	nom_gerente
coordinadora	25000	Silvia zona9	12364987	1236.24	si	125478	13-Sep	Hortensias	Susana Navarro
camioneta	25000	Silvia zona9	12364001	5697.13	no	-	13-Sep	Hortensias	Susana Navarro
camioneta	25000	Silvia zona9	12364652	2345.44	si	654789	13-Sep	Hortensias	Susana Navarro

Tabla de Gerente. (1FN)

Distrito_gerent	nom_gerente	dir_gerente	tel_gerente	id_gerente
Hortensias	Susana Navarro	Patricia Ocampo	045-5536412879	45879-6
Hortensias	Susana Navarro	Patricia Ocampo	045-5536412879	26548-9
Hortensias	Susana Navarro	Patricia Ocampo	045-5536412879	23647-8

Tabla de Coordinadora. (1FN)

desc_cambios	cant_cambios	id_cambios	id_distribuido	nom_distrib	zona_distrib	no_anticipo	cant_anticip
123.88	89.99	569	37894	Rogelio	9	56	5699
65.71	56.99	55	36587	Camilo	8	23	7498
59.66	79.99	789	32159	Efrain	3	98	3498.2

desc_cambios	cant_cambios	id_cambios	id_distribuido	nom_distrib	zona_distrib	no_anticipo	cant_anticip
123.88	89.99	569	37894	Rogelio	9	56	5699
65.71	56.99	55	36587	Camilo	8	23	7498
59.66	79.99	789	32159	Efrain	3	98	3498.2

nom_coord	dir_coord	desc_promo_coo	desc_memo	tel_coord	id_coord	premio_ganado	convencion_co	desc_info_coord	id_mapas_de traba
Carmen Soto	chapulin 65	crema facial	32.56	5564897216	32658974-94	viaje bahamas	Los Angeles	56987	2569
Andrea Torre	Canto 165	crema facial	45.87	5598764170	26547891-12	computadora	Chicago	31564	4587
Juana Perez	MA. Sorte 10	colonia dama	29.88	5532156479	55877916-03	batidora/licuad	Brasil	69787	6981

6.5.2 Segunda Forma Normal SELLSYS1.0

cant_cobranza	desc_cobranza	id_cobranza	division_cod	desc_divisio	desc_distrito	distribuidor_coor	No_factura	Can_factura	concepto_fac	id_factura
126546.98	56.88	1354	hortensias	2658	789456	Juana Perez	23658-987	4587961	ventas c-6	12548796
49876.32	97.4	5698	Lirios	56987	236548	Silvia Barba	48791-678	5698743	ventas c-8	25648791
65879.1	55.77	7854	Violetas	23885	698761	Petra Ramirez	46872-165	8761254	ventas c-7	86435179

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la clave de la tabla para identificarlos.



Tabla de Ventas (2FN)

Ventas								
Id_ventas	Ventas_netas	desc_venta	campana	promedio	promed	ventas_parcia	fecha_venta	estimado_de_ventas
564	5,642.00	1692.6	C-7	30%	67%	5987	12/05/2008	25000
321	23654	7096.2	C-9	45%	56%	5698	30/07/2007	25000
982	8,972.00	2691.6	C-9	66%	82%	1456	04/06/2005	25000

Tabla Fullerette (2FN).

FULLERETTE							
ID_full	nom_full	premio_ganado_full	desc_promo_full	dir_full	tel_full	alta_full	baja_full
178546-6	Rosio Anda	Jelly Roll	Jelly Roll	Abasolo	9564137	14/05/1997	16/08/2008
162543-9	Petra Solis	Nekara	Nekara	Silverio Per	9546572	02/01/2000	-
165223-9	Carmen Diaz	Beautyvert	Beautyvert	Colon 43	9587614	15/09/2003	-

Tabla Distribuidor(2FN).

DISTRIBUIDOR			
distribuidor	ID_distribuido	nom_distrin	zona_distribuid
coordinadora	37894	Rogelio	9
camioneta	36587	Camilo	8
camioneta	32159	Efrain	3

Tabla Gerente (2FN)

GERENTES				
Distrito_gerente	nom_gerente	dir_gerente	tel_gerente	id_gerente
Hortensias	Susana Navarro	Patricia Ocampo	45-553641287	45879-6
Hortensias	Susana Navarro	Patricia Ocampo	45-553641287	26548-9
Hortensias	Susana Navarro	Patricia Ocampo	45-553641287	23647-8

Tabla Mapa trabajo (2FN)

MAPA DE TRABAJO		
ID_mapade	desc_divisiones	desc_distrit
2569	2658	789456
4587	56987	236548
6981	23885	698761

Tabla Coordinadora (2FN).

COORDINADORAS									
nom_coord	dir_coord	desc_promo_co	desc_memo	tel_coord	id_coord	premio_gana	convencion_coord	desc_info_coord	no_antipico
Carmen Soto	chapulin 65	crema facial	32.56	5564897216	32658974-94		viaje bahamas	56987	56
Andrea Torre	Canto 165	crema facial	45.87	5598764170	26547891-12		computadora	31564	23
Juana Perez	MA. Sorte 10	colonia dama	29.88	5532156479	65877916-03		batidora/licuadora	69787	98

cant_antipico	cant_cobran	desc_cobran	id_cobranza	division_coord
5699	126546.98	56.88	1354	hortensias
7498	49876.32	97.4	5698	Lirios
3498.2	65879.1	55.77	7854	Violetas



Tabla Cambios (2FN).

CAMBIOS		
desc_cambios	cant_cambid	id_cambios
123.88	89.99	569
65.71	56.99	55
59.66	79.99	789

6.5.3 Tercera Forma Normal SELLSYS1.0

La regla de la Tercera Forma Normal señala que hay que eliminar y separar cualquier dato que no sea clave. El valor de esta columna debe depender de la clave. Todos los valores deben identificarse únicamente por la clave.

Tabla Ventas (3FN).

Tabla Estado de cuenta part. (3FN)

Ventas						ESTADO DE CUENTA PARTICULAR		
Id_ventas	Ventas_netas	desc_venta	campana	promedio	promed	ventas_parcia	fecha_venta	estimado_de_ventas
564	5,642.00	1692.6	C-7	30%	67%	5987	12/05/2008	25000
321	23654	7096.2	C-9	45%	56%	5698	30/07/2007	25000
982	8,972.00	2691.6	C-9	66%	82%	1456	04/06/2005	25000

Tabla Distribuidor (3FN)

Tabla Gerente (3FN)

DISTRIBUIDOR					GERENTES				
ID_distri	nom_distribui	zona_distrib	distribuidor	distribuidor_coord	id_gerente	nom_gerente	dir_gerente	tel_gerente	distrito_geren
37894	Rogelio	9	coordinadora	Juana Perez	45879-6	Susana Navarro	Patricia Ocampo	045-5536412879	Hortensias
36587	Camilo	8	camioneta	Silvia Barba	26548-9	Susana Navarro	Patricia Ocampo	045-5536412879	Hortensias
32159	Efrain	3	camioneta	Petra Ramirez	23647-8	Susana Navarro	Patricia Ocampo	045-5536412879	Hortensias

Tabla Mapa de Trabajo(3FN)

Tabla Coordinadoras(3FN)

MAPA DE TRABAJO			COORDINADORAS				
ID_mapadet	desc_divisio	desc_distri	nom_coord	dir_coord	tel_coord	id_coord	division_coord
2569	2658	789456	Carmen Soto	chapulin 65	5564897216	32658974-94	hortensias
4587	56987	236548	Andrea Torre	Canto 165	5598764170	26547891-12	Lirios
6981	23885	698761	Juana Perez	MA. Sorte 10	5532156479	65877916-03	Violetas

Tabla Cambios (3FN)

Tabla Cambios (3FN)

CAMBIOS			PREMIOS			
desc_cambios	cant_cambios	id_cambios	ID_Premios	premio_ganado_fu	premio_ganado_coo	convencion_coord
123.88	89.99	569	1254897	Jelly Roll		Los Angeles
65.71	56.99	55	1256697	Nekara		Chicago
59.66	79.99	789	125483	Beautyvert		Brasil



Tabla Promociones (3FN)

Tabla Cancelaciones (3FN)

PROMOCIONES			CANCELACIONES			
ID_promo	desc_promo_full	desc_promo_coord	desc_cancel	no_cancelac	id_cancelacion	
23654	Jelly Roll	crema facial	1236.24	si	125478	
98756	Nekara	crema facial	5697.13	no	-	
15789	Beautyvert	olonia dama	2345.44	si	654789	

Tabla Facturas (3FN)

Tabla Informes (3FN)

FACTURACION			INFORMES				
No_factura	Can_factura	concepto_fac	id	ID_informes	desc_me	desc_info_coord	
23658-987	4587961	ventas c-6	#	569874	32.56	56987	
48791-678	5698743	ventas c-8	#	236548	45.87	31564	
46872-165	8761254	ventas c-7	#	965884	29.88	69787	

Tabla Cobranza (3FN)

Tabla Estado de Cuenta (3FN)

COBRANZA				ESTADO DE CUENTA		
cant_cobranz	desc_cobranz	id_cobranza		ID_estado	saldo_full	saldo_coord
126546.98	56.88	1354		12546	12681.3	56897.5
49876.32	97.4	5698		36548	15612.15	109568.87
65879.1	55.77	7854		41656	5971.25	42761.3

Tabla Papeleria (3FN)

Tabla Puntos Fuller (3FN)

PAPELERIA			PUNTOS FULLER			
ID_papeleria	desc_papeleria	iCantidad	ID_Puntos	premio_ganado_fu	premio_ganado_coo	convencion_coord
123.88	89.99	569	1254897	Auto		1
65.71	56.99	55	1256697	Moto		5
59.66	79.99	789	125483	Vajilla		345

Al término de Normalizar los datos, se hace la implementación física con DBDesigner4 para ser cargara en MySQL.



6.5.4 Modelo Relacional de SELLSYS1.0.

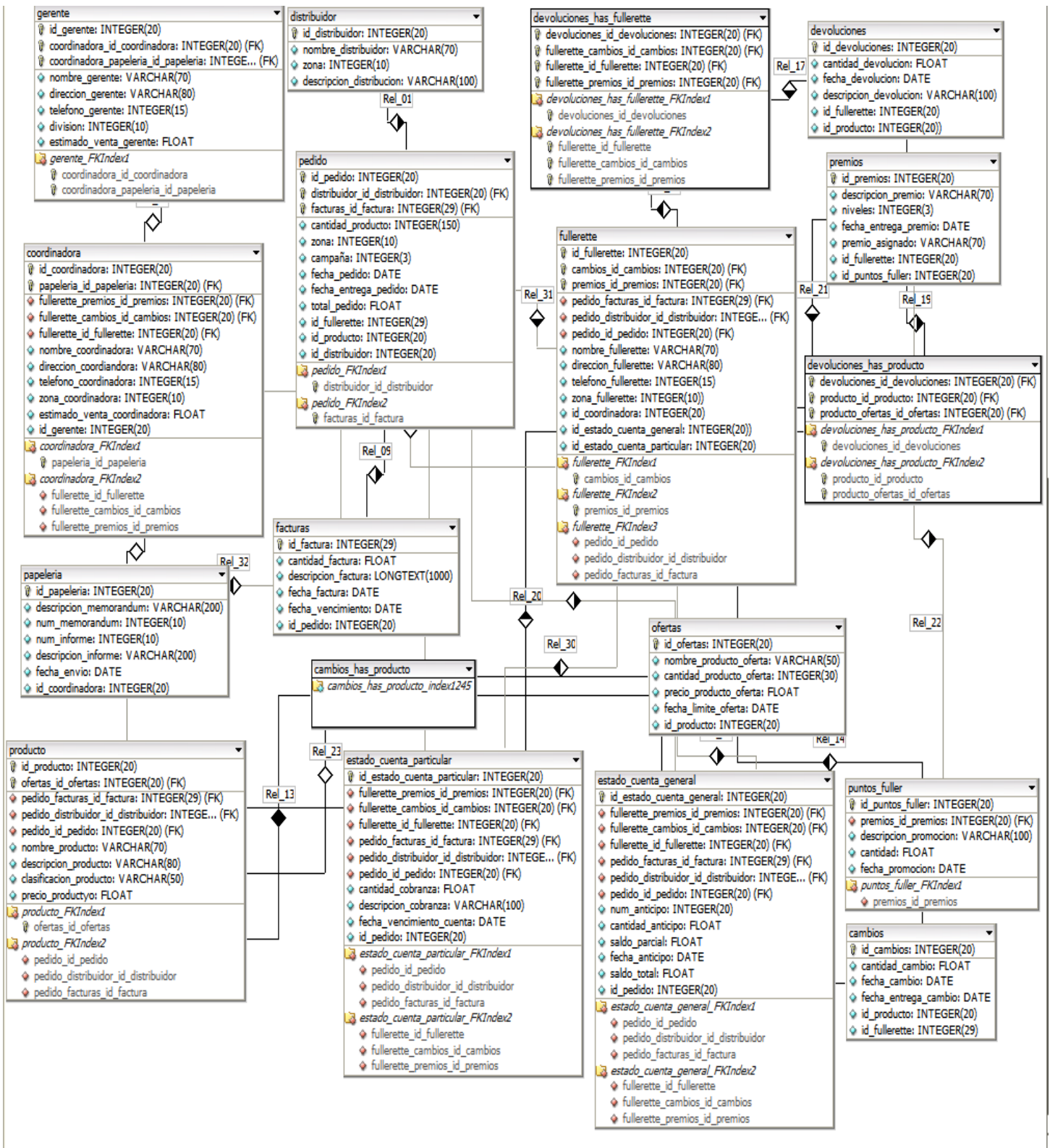


Figura 42 Esquema de bases de datos de SELLSYS1.0



Después de agregar la base de datos en MySQL se puede comenzar a trabajar.

```
mysql> describe gerente
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id_gerente     | int(11)       | NO   | PRI | NULL    | auto_increment |
| nombre_gerente | varchar(70)   | YES  |     | NULL    |                |
| direccion_gerente | varchar(80)  | YES  |     | NULL    |                |
| telefono_gerente | int(11)       | YES  |     | NULL    |                |
| division       | varchar(25)   | YES  |     | NULL    |                |
| estimado_venta_gerente | float        | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.05 sec)

mysql>
```

Figura 43 Tablas en MySQL

En la Figura 42 y 43 se muestran las tablas en MySQL, estas tablas ya contienen los datos normalizados y los tipos de datos que se van a manejar en los campos, así como las llaves primarias y secundarias.

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.38-community MySQL Community Server (GPL)

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

mysql> use ventas;
Database changed
mysql> describe pedido;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id_pedido     | int(11)       | NO   | PRI | NULL    | auto_increment |
| distribuidor_id_distribuidor | int(11)       | NO   | MUL | NULL    |                |
| fullerette_id_fullerette     | int(11)       | NO   | MUL | NULL    |                |
| producto_id_producto         | int(11)       | NO   | MUL | NULL    |                |
| cantidad_producto            | int(11)       | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figura 44 Agregar Registros en MySQL.

En MySQL se agregan registros en las tablas y se hacen consultas para hacer pruebas de buen funcionamiento.



6.5.5 Conexión con SQL.

La conexión de VISUAL BASIC con MySQL se realizó con MySQL Connector 5.1. y desarrollo del algoritmo de conexión.

6.6 Programación de SELLSYS1.0

La interfaz se creó con VISUAL BASIC a partir del lenguaje de programación y del ambiente gráfico, teniendo como resultados vistas amigables e interactivas. Como se muestra en la figura 45. En cada vista de resultados se reflejan los algoritmos de transacciones.

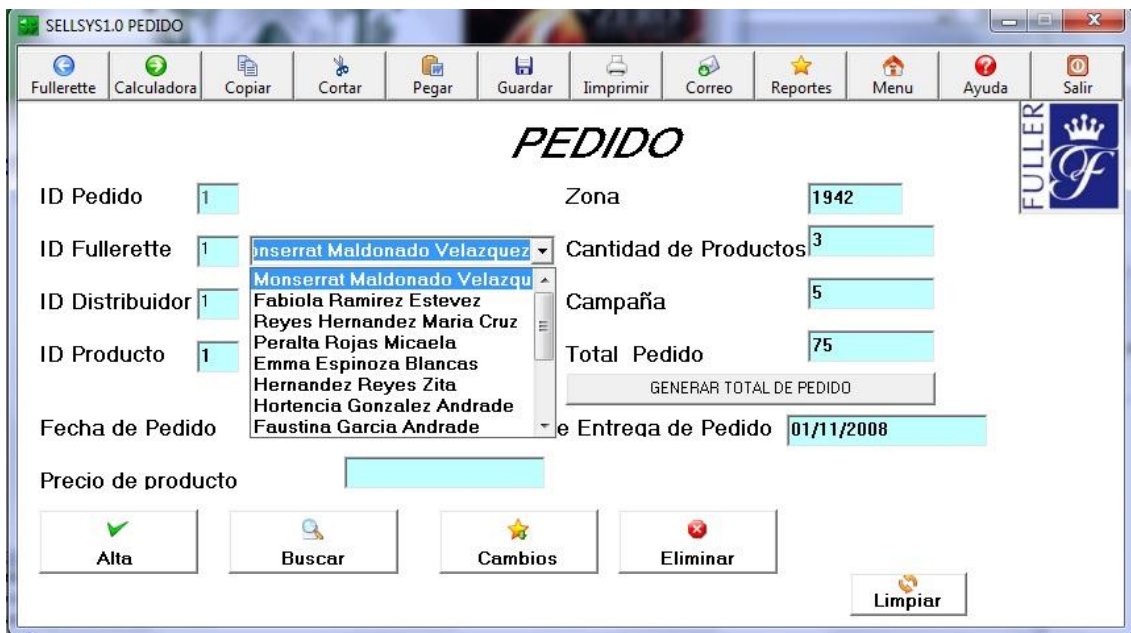


Figura 45 Diseño de las interfaces en Visual Basic.

VII. RESULTADOS

INGRESAR AL SISTEMA

Para comenzar a trabajar con SELLSYS1.0 el usuario deberá registrarse en la pantalla inicial colocando su nombre de usuario al igual su contraseña y darle click en el botón con el símbolo de SELLSYS1.0 (trébol). Figura 46. Si no es un usuario autorizado, no podrá seguir.



Figura 46 interfaz de contraseña del sistema

BIENVENIDA Y SELECCIÓN DESDE EL MENU PRINCIPAL

Al dar click en ingresar al sistema se dará la bienvenida y dando click en la pestaña MENU, se desplegarán todas las opciones que podemos escoger para comenzar a trabajar.



Figura 47 Interfaz de Selección desde el menú principal.

En la parte superior de todas las vistas del sistema se encuentra un conjunto de botones como el que se muestra en la figura 48, con ellos se pueden realizar las acciones que los mismos botones indican, por ejemplo: copiar, cortar, pegar, guardar, menú, ayuda, salir, etc.



Figura 48 Toolbar del Sistema Sellsys1.0.

Estos botones ayudan a hacer que las tareas sean más accesibles, dinámicas y divertidas.



DAR DE ALTA UN REGISTRO

Para dar de alta un registro se debe llenar completamente los datos que se requieren y posteriormente dar click en el botón alta y el sistema avisara que el registro esta dado de alta (ver figura 49).

The screenshot shows a software window titled 'SELLSYS1.0 FULLERETTES'. It features a menu bar with options like 'Pedido', 'Calculadora', 'Copiar', 'Cortar', 'Pegar', 'Guardar', 'Imprimir', 'Correo', 'Reportes', 'Menu', 'Ayuda', and 'Salir'. The main area is titled 'FULLERETTES' and contains a form with the following fields: 'ID Fullerette' (value: 1), 'ID Estado de Cuenta Particular' (value: 1), 'ID Estado de Cuenta General' (value: 1), 'ID Coordinadora' (value: 1), 'Nombre' (value: 'Montserrat Maldonado Velazquez'), 'Direccion' (value: 'abasolo #113'), 'Telefono' (value: '59557482'), and 'Zona' (value: '1942'). Below the form are five buttons: 'Alta' (highlighted with a red dashed circle and an arrow), 'Buscar', 'Cambios', 'Eliminar', and 'Limpiar'. A 'FULLER' logo is visible in the top right corner.

Botón Alta

Figura 49 Interfaz para dar de alta un Registro.

BUSCAR UN REGISTRO

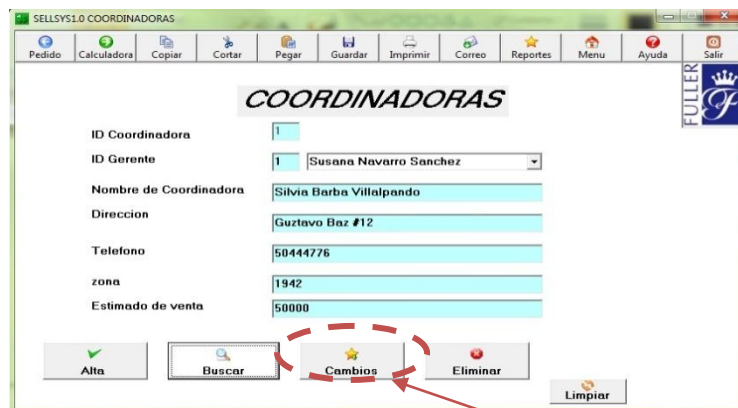
Para buscar un registro en la base de datos es necesario primero darle click en el botón buscar y automáticamente aparecerá un recuadro donde se requiere ingresar el nombre del trabajador que estamos buscando en la base de datos de SELLSYS1.0 vea figura 50.

The screenshot shows a software window titled 'SELLSYS1.0 COORDINADORAS'. It features a menu bar with options like 'Pedido', 'Calculadora', 'Copiar', 'Cortar', 'Pegar', 'Guardar', 'Imprimir', 'Correo', 'Reportes', 'Menu', 'Ayuda', and 'Salir'. The main area is titled 'COORDINADORAS' and contains a form with the following fields: 'ID Coordinadora', 'ID Gerente', 'Nombre de Coordinadora', 'Direccion', 'Telefono', 'zona', and 'Estimado de venta'. Below the form are five buttons: 'Alta', 'Buscar', 'Cambios', 'Eliminar', and 'Limpiar'. A search dialog box is open over the 'Buscar' button, titled 'SELLSYS1.0' and containing the text 'Escribe el nombre de la COORDINADORA' and a text input field with the value 'silvia barba'. The dialog box has 'Aceptar' and 'Cancelar' buttons. A 'FULLER' logo is visible in the top right corner.

Figura 50 Interfaz que se utiliza para buscar un registro.

CAMBIAR UN REGISTRO

Para realizar cambios en un registro en la base de datos es necesario primero tener los datos, hacer los cambios que se deseen y después se le da click en el botón cambiar y automáticamente aparecerá un recuadro donde cuestiona al usuario si desea hacer los cambios, de ser así, se da click en el botón ACEPTAR y muestra otro recuadro que avisa que los cambios se han realizado correctamente figura 51.

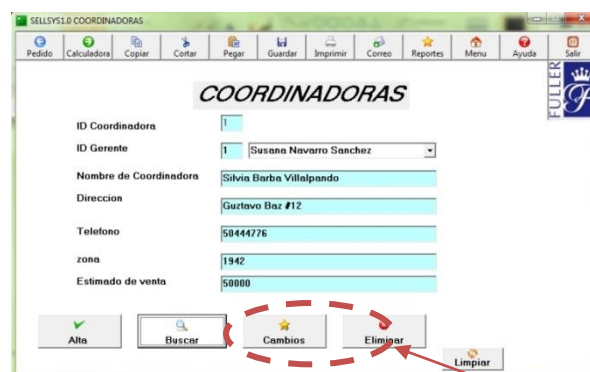


Botón Cambios

Figura 51 Interfaz que muestra como cambiar un registro.

ELIMINAR UN REGISTRO

La figura 52 muestra la interface para eliminar un registro en la base de datos, para ello es necesario primero realizar la búsqueda del registro que deseamos eliminar; una vez sabiendo lo que se eliminará procedemos a darle click en el botón de eliminar y automáticamente aparecerá un recuadro donde se requiere ingresar el nombre del registro que pensamos eliminar de la base de datos de *SELLSYS1.0*, después se mostrará en un pequeño recuadro si le eliminación se ha realizado correctamente.



Botón Eliminar

Figura 52 Interfaz que muestra cómo eliminar



LIMPIAR UN REGISTRO

El botón limpiar sirve para dejar por completo limpias las casillas en tono azul, lo cual ayuda para que ingresemos nuevos datos al sistema, figura 53.

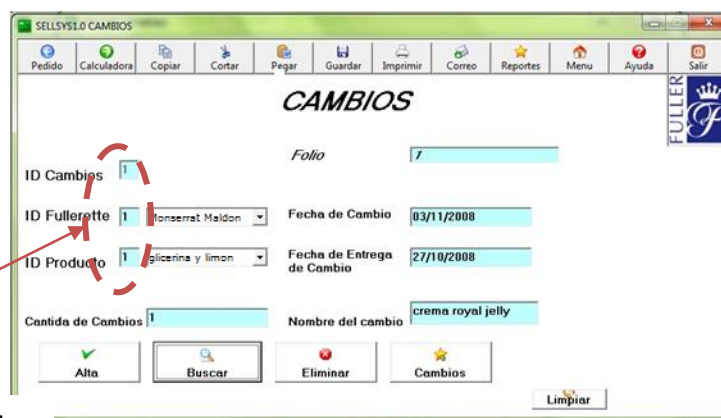


Botón Limpiar

Figura 53 Interfaz que muestra como limpiar los registros.

INTERFAZ QUE MUESTRA LA AUTOMATIZACION DEL SISTEMA.

En la interfaz de figura 54 se podrán capturar los cambios de productos que se vayan a realizar, se seleccionan las opciones en donde están las fechas, la fecha en que se hace el cambio y fecha de cuando se entrega el cambio, la cantidad de cambios que se van a hacer, el nombre del producto de cambio y también se escribe el numero de folio del cambio que se va a realizar, una vez llenados los datos, se da click en el botón de alta y automáticamente se exponen los datos en una pantalla de confirmación de alta en la base de datos de *SELLSYS1.0*.



Los ID aparecen automáticamente

Figura 54 Interfaz que muestra la automatización del sistema.

INTERFAZ QUE MUESTRA COMO SE REGISTRAN LOS DISTRIBUIDORES.

En la tabla distribuidores se captura a los diferentes distribuidores que la empresa tiene, así como la descripción de lo que distribuyen, la zona que le toca reparto y de igual manera el id de distribuidor se coloca automáticamente, posteriormente pasamos a dar de alta el nuevo registro en la base de datos de *SELLSYS1.0* figura 55

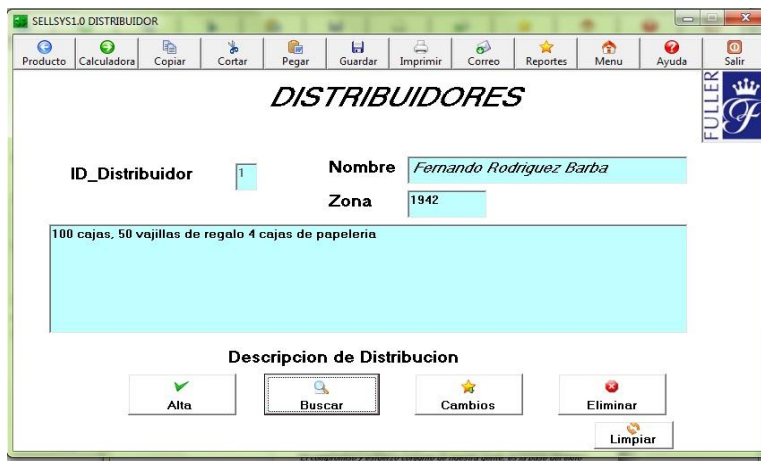
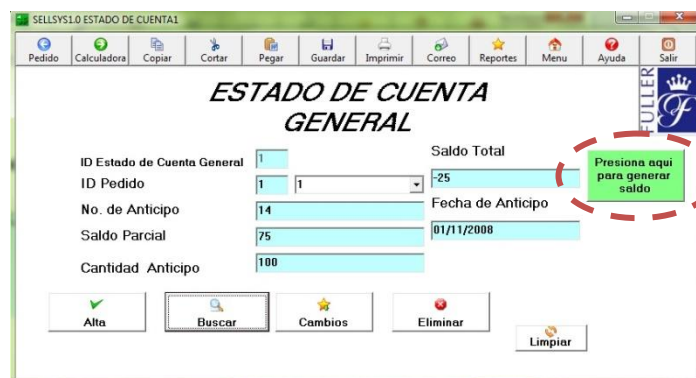


Figura 55 Interfaz que muestra como se registran en la tabla distribuidores.

INTERFAZ QUE MUESTRA EL ESTADO DE CUENTA

En la figura 56 se puede observar la situación de los pedidos, es decir, si el saldo está cubierto o no, si se han realizado algunos abonos y en qué fecha, para generar el saldo se presiona el botón verde, es el que realiza una operación matemática ayudando así a las coordinadoras para evitar realizarlas a mano, con lo que automáticamente se da de alta el nuevo registro en la base de datos de *SELLSYS1.0*.



Este botón genera el saldo total automáticamente

Figura 56 Interfaz que muestra el estado de cuenta general.



INTERFAZ QUE MUESTRA EL ESTADO DE CUENTA PARTICULAR.

En la figura 57 se muestra el pedido con la descripción de cobranzas, es decir con la descripción del contenido del pedido, la fecha de vencimiento de cobro y el monto que falta por cobrar.

Figura 57 Interfaz que muestra el estado de cuenta particular.

INTERFAZ QUE MUESTRA EL RESULTADO DE FACTURAS.

Aquí se anotan los datos de facturación de los pedidos, como lo son la fecha de facturación, cantidad en la factura, fecha de vencimiento del pedido facturado y los productos que contiene el pedido, el id de factura. (Figura 58).

Figura 58 Interfaz que muestra el resultado de facturas.



INTERFAZ PARA IDENTIFICACIÓN DE FULLERETTES.

Con la interfaz de la figura 59 se capturan de todas las fullerettes, sus datos personales y automáticamente se anota un id de fullerette que servirá en posteriores vistas para la identificación de la fullerette.

Figura 59 Interfaz de la tabla Fullerettes..

INTERFAZ DE OFERTAS

La interfaz de ofertas muestra la descripción del producto que esta de oferta en cada campaña. En esta tabla aparecen los precios de manera automática así como también los números de id de los productos, según la figura 60.

Figura 60 Interfaz de la tabla ofertas.



REGISTRO DE GERENTES

Para el control de gerentes se desarrollo la interfaz gerente, se utiliza para dar de alta, buscar, cambiar o eliminar los datos de las gerentes ver figura 61.

The screenshot shows a software window titled "SELLSYS1.0 GERENTE". The main heading is "GERENTE". The form contains the following fields and values:

- ID Gerente: 1
- Nombre: Susana Navarro Sanchez
- Direccion: Aldama #115 texcoco Edo de Mexico
- Telefono: 59554898
- Division: DALIAS
- Estimado de Ventas: 100000

At the bottom of the form are five buttons: Alta (with a green checkmark), Buscar (with a magnifying glass), Cambios (with a star), Eliminar (with a red X), and Limpiar (with a trash can).

Figura 61 Interfaz de la tabla gerente.

REGISTRO DE PAPELERIA

Para el registro de memorándums es utilizada por la gerente para describir a detalle de los memorándums que son enviados a la empresa, es decir, especificar si en cierta campaña hubo anomalías en los pedidos, o en productos de toda la campaña, figura 62.

The screenshot shows a software window titled "SELLSYS1.0 PAPELERIA". The main heading is "PAPELERIA". The form contains the following fields and values:

- ID Papeleria: 1
- ID Coordinadora: 1 (Silvia Barba Villalpando)
- No. de Memorandum: 1
- Descripcion de Memorandum: anomalias en 10 productos
- No. de Informe: 1
- Descripcion de Informe: sin problemas en campanna 5
- Fecha de Envio: 01/11/2008

At the bottom of the form are five buttons: Alta (with a green checkmark), Buscar (with a magnifying glass), Cambios (with a star), Eliminar (with a red X), and Limpiar (with a trash can).

Figura 62 Interfaz que muestra cómo funciona la tabla de papelería.



REGISTRO DE PRODUCTOS

Aquí se dan de alta, búsquedas, cambios o eliminaciones de productos que existen en la empresa ver figura 63.

Figura 63 Interfaz de la tabla producto.

REGISTRO DE PUNTOS FULLER

Es donde se dan de alta, búsquedas, cambios o eliminaciones de premios que son otorgados a las fullerettes ya sea por venta o para que la coordinadora realice una rifa entre las fullerettes, ver figura 64.

Figura 64 Interfaz de la tabla puntos Fuller.



REGISTRO DE PEDIDOS

Aquí se capturan los pedidos con su descripción en general, en algunas casillas los datos se exponen de manera automática, por ejemplo cuando presionamos el botón GENERAR TOTAL DE PEDIDO, se realiza una operación matemática para evitar realizarla a mano, dando el total de la cantidad de cada pedido, ver figura 65.

PEDIDO				
ID Pedido	1	Zona	1942	
ID Fullerette	1	Monserrat Maldonado Velazqu	Cantidad de Productos	3
ID Distribuidor	1	Fernando Rodriguez Barba	Campaña	5
ID Producto	1	crema de glicerina y limon	Total Pedido	75
GENERAR TOTAL DE PEDIDO				
Fecha de Entrega de Pedido	01/11/2008	Fecha de Pedido	07/11/2008	
Precio de producto	25			

Figura 65 Interfaz de la tabla pedido.

INTERFAZ DE REPORTE

Esta interfaz es una de las más importantes ya que con ella la gerente puede realizar sus actividades de reportar a la empresa de sus actividades, teniendo como resultado un ahorro de tiempo y control preciso sobre la situación que guarda cada vendedora.

En la figura 66 se muestra una serie de botones, de color amarillo son los botones que logran hacer consultas sobre todo lo que hay en la base de datos, como los datos de las gerentes, distribuidores, coordinadoras, fullerettes, estados de cuenta, productos, premios, etc., son los que se muestran señalados, lo que hace al sistema SELLSYS1.0 más amigable, interactivo, divertido y eficiente para el usuario.

id_coordinadora	gerente_id_gerente	nombre_coordinadora	direccion_coordinadora	telefono_coordinadora	zona_coordinadora	estimado_venta_coordinadora
1	1	Silvia Barba Villalpando	Guztavo Baz #12	50444776	1942	50000
2	2	Ma de Jesus Barrio sanch	fresnos #56san martin de	56874585	1942	50000
3	5	Juanita hernandez mende	calle el prado #12chicolo	59878364	1942	55000
4	3	yesica montes perez	progreso #14 chicoapan	265458546	1942	67000
5	2	Andrea Velazquez Rosale	cabasolo #113 texcoco me	59557482	1942	98000
6	4	Fernanda de Leon Marq	prolongacion#100 int 9 ch	58495452	1942	102000
7	3	Laura Marban Perez	valle se santa cruz# 118	59587146	1942	136000
8	2	Teresa Rodriguez Ortiz	Guztavo baz #9 col ejerc	47585168	1942	48000
9	4	Estefania Snches san Ro	Piril #98 chimalhuacan Ed	65658149	1942	48000
10	4	socorro rodriguez duvon	calle revolucion #16 netz	25469157	1942	350000
11	1	X Cruz Chavez	calle alatrste # 35 San M	659851225	1942	56600
12	3	Remedios Vazquez Ramo	prolongacion de aldama #	54785254	1942	680000
13	1	Domitila Hernandez Juare	los robles #45 chicoapal	78541544	1942	62000
14	4	Rosario Quintos Velazque	Moctezuma #112 int 34 T	59589874	1942	68000

Estos botones son los que te mostraran los datos que contiene la base de datos del sistema que no requieren de fecha especifica para consultar

Figura 66 Interfaz que hace referencia a los reportes sin fechas definidas.

Un poco más a la derecha veremos las consultas que se pueden visualizar entre lapsos de tiempo, es decir, entre una fecha y la otra, como lo son:

- El reporte de cambios entregados
- El reporte de devoluciones
- El reporte de ofertas
- El saldo de alguna fullerette en especifico
- Las promociones de Puntos Fuller
- Ventas totales de cierre de campaña
- Las fullerettes con pedidos vencidos, etc.

Tales consultas en periodos específicos de tiempo hacen de Sellsys1.0 una herramienta de trabajo muy confiable para el usuario, así como, eficaz al momento de querer consultar un dato ya sea específico o general de las ventas realizadas en una campaña, permitiendo al usuario tener más tiempo para dedicarlo en conseguir nuevos clientes o realizar la cobranza de los clientes



moratorios. En otras palabras con sellsys1.0 se obtiene mayor productividad que genera más ganancias para la empresa y para los usuarios ver figura 67.

CONSULTAS GENERALES

id coordinadora	gerente id gerente	nombre coordinadora	direccion coordinadora	telefono coordinadora	zona coordinadora	estimado venta coordinadora	
1	1	Silvia Barba Villalpando	Guztavo Baz #12	50444776	1942	50000	
2	2	Ma de Jesus Barrio sanchifresnos #56	san martin de	56874585	1942	50000	
3	5	Juanita herrandez mendecalle el prado #12	chicoloaz	59878364	1942	55000	
4	3	yesica montes perez	progreso #14	chicoloapan	265458546	1942	67000
5	2	Andrea Velazquez Rosaleabasolo #113	texcoco me	59557482	1942	98000	
6	4	Fernanda de Leon Marq	prolongacion#100 int 9 ch	58495452	1942	102000	
7	3	Laura Marban Perez	valle se santa cruz# 118	59587146	1942	136000	
8	2	Teresa Rodriguez Ortiz	Guztavo baz #9 col ejerc	47585168	1942	48000	
9	4	Estefania Snches san Ro Piril #98	chamalhuacan Ed	65658149	1942	48000	
10	4	socorro rodriguez duvon	calle revolucion #16 netz	25469157	1942	350000	
11	1	X Cruz Chavez	calle alatraste # 35 San M	659851225	1942	56600	
12	3	Remedios Vazquez Ramo	prolongacion de aldama #54	785254	1942	680000	
13	1	Domitila Hernandez Juare	los robles #45	chicoloapan	78541544	1942	62000
14	4	Rosario Quintos Velazque	Moctezuma #112 int 34 T	59589874	1942	68000	

CONSULTAS QUE NO REQUIEREN FECHA DEFINIDA

GERENTES DADAS DE ALTA, ESTADO DE CUENTA PARTICULAR, PUNTOS FULLER, COORDINADORAS DADAS DE ALTA, OFERTAS, FACTURACION, DATOS DE CAMBIOS, PAPELERIA, PRODUCTOS, FULLERETTES DADAS DE ALTA, PREMIOS, DEVOLUCIONES REALIZADAS, DISTRIBUIDORES, ESTADO DE CUENTA GENERAL, DATOS DE PEDIDOS

FECHA 1, FECHA 2, REPORTES: VENTAS TOTALES DE CIERRE, SALDO DE PEDIDOS, PEDIDOS REALIZADOS, REPORTE DE CAMBIOS ENTREGADOS, COORDINADORAS QUE ENVIARON INFORMES, REPORTE DE CAMBIOS PEDIDOS, REPORTE DE ESTADO DE CUENTA DE FULLERETTES, FULLERETTES CON FACTURA VENCIDA, PROMOCIONES PUNTOS FULLER, REPORTE DE DEVOLUCIONES, PREMIOS ENTREGADOS, ESTADOS DE CUENTA, OFERTAS EN FECHAS ESPECIFICAS, FULLERETTES CON PREMIOS, VENTAS DE COORDINADORAS, REPORTE INDIVIDUAL DE FULLERETTES

NOMBRE DE LA FULLERETTE

Limpier

Figura 67 Interfaz que muestra la tabla de reportes con consultas en fechas determinadas.

Al dar click en alguno de los botones de consulta, ya sea por periodos de tiempo o por reporte general en la pantalla de arriba se podrán visualizar los registros que se desean visualizar que contiene el sistema figura 68.

CONSULTAS GENERALES

id coordinadora	gerente id gerente	nombre coordinadora	direccion coordinadora	telefono coordinadora	zona coordinadora	estimado venta coordinadora	
1	1	Silvia Barba Villalpando	Guztavo Baz #12	50444776	1942	50000	
2	2	Ma de Jesus Barrio sanchifresnos #56	san martin de	56874585	1942	50000	
3	5	Juanita herrandez mendecalle el prado #12	chicoloaz	59878364	1942	55000	
4	3	yesica montes perez	progreso #14	chicoloapan	265458546	1942	67000
5	2	Andrea Velazquez Rosaleabasolo #113	texcoco me	59557482	1942	98000	
6	4	Fernanda de Leon Marq	prolongacion#100 int 9 ch	58495452	1942	102000	
7	3	Laura Marban Perez	valle se santa cruz# 118	59587146	1942	136000	
8	2	Teresa Rodriguez Ortiz	Guztavo baz #9 col ejerc	47585168	1942	48000	
9	4	Estefania Snches san Ro Piril #98	chamalhuacan Ed	65658149	1942	48000	
10	4	socorro rodriguez duvon	calle revolucion #16 netz	25469157	1942	350000	
11	1	X Cruz Chavez	calle alatraste # 35 San M	659851225	1942	56600	
12	3	Remedios Vazquez Ramo	prolongacion de aldama #54	785254	1942	680000	
13	1	Domitila Hernandez Juare	los robles #45	chicoloapan	78541544	1942	62000
14	4	Rosario Quintos Velazque	Moctezuma #112 int 34 T	59589874	1942	68000	

CONSULTAS QUE NO REQUIEREN FECHA DEFINIDA

FECHA 1, FECHA 2, REPORTES: VENTAS TOTALES DE CIERRE, SALDO DE PEDIDOS, PEDIDOS REALIZADOS

Área de visualización de datos

Figura 68 Interfaz que muestra los datos del DATAGRID



En la figura 69 se observa una muestra de la pantalla cuando se hace un reporte específico.

Reporte de productos

Figura 69 Interfaz que muestra el reporte de productos.

En la figura 70, se da una muestra de los reportes que requieren de la especificación de tiempo.

Se trata de un reporte de ventas totales de cierre de campaña

Figura 70 Interfaz que muestra la utilidad de las fechas en los reportes.



Toda función de estos reportes es disminuir considerablemente el tiempo que las gerentes y coordinadoras le dedican a hacer reportes y manejar la información que necesitan realizar su trabajo eficientemente.

Como se muestran en las figuras es sencillo y amigable de manejar, además cuenta con elementos importantes para hacer más ágil el proceso de ventas en esta empresa como lo es imprimir un reporte ya que en el momento de presionar el botón imprimir, Sellsys1.0 manda una impresión de los datos seleccionados a la impresora. Otro elemento importante son los accesos rápidos a la calculadora o los enlaces de correo electrónico.

En estos reportes es muy importante remarcar que los datos en los reportes son personalizados y generalizados para cada Fullerette lo que da como resultado que para el usuario sea más fácil detectar que clientes requieren de más atención, especialmente los que tienen deudas con la empresa.

VALIDACION

El sistema de información fue instalado en una computadora de un gerente de la empresa para su validación. Se realizaron pruebas y se dejó que se probara en todas sus funciones, de acuerdo a los comentarios hechos por el usuario se realizaron mejoras de acuerdo al método del modelo en espiral. Hasta finalmente quedó a satisfacción del cliente.

VIII. DISCUSION

Con la integración de conocimientos de diseño y análisis de sistemas de información, bases de datos y programación, fue posible desarrollar un conjunto de rutinas que permitió desarrollar un sistema que ayuda las actividades de control y manejo de personal de una gerente dedicada a las ventas de cosméticos. Todo ello tiene como consecuencia que el tiempo que se dedicaba a la supervisión de cuentas, asignación de premios, control de vendedoras y actividades de conciliación de cuentas se disminuyó considerablemente, además de que no se tienen errores, todo esto, trae como consecuencia en un aumento en la productividad de las vendedoras y gerentes, ya que, el tiempo que se dedicaba a estas actividades actualmente se invierten en la atención a clientes, cobrar cuentas de morosos y buscar más clientes para aumentar los ingresos de cada vendedora, cliente y la misma empresa.



Sin olvidar que la automatización de los procesos de una empresa también conlleva a disminuir tiempos en la creación de reportes ya que se hacían de forma manual, actualmente se hacen automáticamente en una impresora. Es preciso destacar que las empresas muchas veces no quieren invertir en pagar software especializado ya que no lo consideran como una inversión y para ellos es un gasto innecesario. Sin embargo después de probar los beneficios que este sistema pudo aportar a esta empresa se pudo probar en la práctica que en realidad es una inversión, solo que se reflejara con el tiempo y el buen uso de un sistema automatizado.

Se detecto que el factor humano es un inconveniente para la utilización de tecnologías de automatización de procesos en pequeñas empresas, por lo que hay que crear y fomentar una cultura informática en la sociedad.

IX. CONCLUSIONES

El factor humano puede ser un factor que no permite que se empleen las tecnologías de automatización de los procesos de una empresa.

Se disminuyen los tiempos en control de personal de la empresa.

Se pudieron disminuir los morosos en la empresa.

Se aumentaron los clientes al tener más tiempo para ofrecer el producto.

Se aumento la productividad de las vendedoras y gerentes al dedicar menos tiempo en actividades manuales rutinarias.

La inversión en automatización de procesos administrativos en una empresa se refleja en la eficiencia de los trabajadores.

El modelo de espiral permite que el cliente participe y quede satisfecho con el sistema de información.

La opinión de los usuarios es primordial en el desarrollo de sistemas.



X. BIBLIOGRAFIA

- [1] MARTIN FOWLER, KENDALL SCOTT, R. JAIME GONZALEZ V., “UML gota a gota”, 1999, ADDISON-WESLEY
- [2] CRAIG LARMAN, “UML Y PATRONES: Introducción al Análisis y Diseño Orientado a Objetos”, 1999, PRENTICE HALL
- [3] STEPHEN R. SHACH, “Análisis y Diseño orientado a objetos de sistemas usando UML and the unified process”, 2004, McGraw HILL
- [4] PEARSON, “Applying UML and Patterns, an Introduction to object-oriented analysis and design”, 2003, PRENTICE HALL
- [5] SCHMULLER JOSEPH, “UML en 24 horas”, 2007, Edit. PRENTICE HALL
- [6] SIMON BENNETT, STEVE McROBB y RAY FARMER, “Análisis y diseño orientado a objeto de sistemas usando UML”, 2007, Edit. McGRAW HILL
- [7] PAUL HARMAN Y MARK WATSON; Entendiendo UML: La guía del desarrollador, con una aplicación java basada en web, Morgan Kauffman Publishers, Inc., 1998.
- [8] DESMOND F. D’SOUZA Y ALAN C. WILLS , Objetos, componentes y Estructuras con UML, The Catalysis Aproach, , Addison Wesley Longman, 1998.
- [9] G. BOOCH. J. RUMBAUGH, I. JACOBSON The unified modeling language. Use guide. Addison-Wesley (1999). Versión castellana: El lenguaje Unificado de Modelado. Addison-Wesley (1999)
- [10] M. FOWLER WITH K. SCOTT, UML Distilled: Applying the Standard Object Modeling Language, ISBN 0-201-32563-2, Addison-Wesley, 1997.
- [11] CHECKLAND 2002, “Pensamientos de sistemas, practicas de sistemas”, Edit. WILEY
- [12] LAUDON 2000, “Administración de sistemas de información”, Edit. PRENTICE HALL
- [13] EFFY OZ 2003, “Administración de sistemas de información”, Edit. THOMSON LEARNING
- [14] LAUDON Y LAUDON, “Sistemas de información gerencial”, organización y tecnología de empresa conectada en red, 6ta. Edición, 2001, Edit. PRENTICE HALL



-
- [15] MUDRICH 2002, “Sistemas de información administrativa”, Edit. PRENTICE HALL
- [16] KENETH E. KENDAL, “Análisis y Diseño de Sistemas”, 1997, Edit. PRENTICE HALL
- [17] JAMES A. O'BRIEN, “Sistemas de Información Gerencial”, 2001, Edit. McGraw HILL
- [18] RONALD J. NORMAN, “Object – Oriented Systems Analysis and Design”, 1996, Edit. PRENTICE HALL
- [19] JAMES A. SENN, “Análisis y Diseño de Sistemas de Información”, 1992, Edit. McGRAW HILL
- [20] LAUDON Y LAUDON, “Análisis y Diseño de Sistemas de Información”, 3ra. Edición, 1996, Edit. PRENTICE HALL
- [21] RAMAKRHISHMAN GEHRKE, “Sistema de Gestión de Bases de Datos”, 2007, McGRAW HILL
- [22] I.T HAWRYSZKIEWE, “Análisis y Diseño de Bases de Datos”, 2004, Megabyte Noriega Editores.
- [23] GARY W. HANSEN & JAMES V, HANSEN, “Diseño y Administración de Bases de Datos”, Madrid 2005, Prentice Hall
- [24] C.J DALE, “Introducción a los Sistemas de Bases de Datos”, 2001 7^{ma} Edición, Pearson Prentice Hall.
- [25] SILVERSCHATZ & KORT & SUDARSHAN, “Fundamentos de Bases de Datos”, 2006, McGRAW HILL
- [26] GARCIA RINCON LUIS FRANCISCO, “Bases de Datos un Enfoque Practico”, 2007, Editorial Trillas.
- [27] ROB PETER & CORONEL CARLOS, “Sistemas de Bases de Datos Diseño, implementación y Administración”, 2005, Thompson International Editors
- [28] CAMPS PARE RAFAEL, “Introducción a las Bases de Datos”, 2008 7^{ma} Edición, Pearson Prentice Hall.
- [29] CASTAÑO MIGUEL & PRATTINI MARIO, “Diseño de Bases de Datos Relacionales”, 2007, Alfa omega.
-



[30] V. POST GERALD, “Sistemas de Administración de Bases de Datos”, 2006,
McGRAW HILL

[31] RON SUKUP/ KALEN DELANEY, “A fondo SQL Server ”,1999,
McGRAW HILL

[32] ALFONSO GONZALEZ PEREZ, “SQL Programación y
Administración”,1999, ALFAOMEGA

[33] MIKE GUNDERLOY/ MARY CHIPNAN, “SQL Server”,2000,
ANAYA MULTIMEDIA