



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO UAEM TEXCOCO

**“DATAWAREHOUSE PARA CONTROL DE STOCK
EN FORD ZAPATA S.A. DE C.V.”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
Licenciado en Informática Administrativa**

PRESENTA

Cristian Ruiz Hernández

DIRECTOR

M. en C.C. Juan Carlos Moreno Sánchez

REVISORES

Dr. Adrián Trueba Espinosa

Dr. en I. de S. José Sergio Ruiz Castilla

TEXCOCO, ESTADO DE MÉXICO, ENERO DE 2017.

Texcoco, México a 14 de Noviembre de 2016.


M. EN C. ED. VIRIDIANA BANDA ARZATE
SUBDIRECTORA ACADEMICA DEL
CENTRO UNIVERSITARIO UAEM TEXCOCO.
PRESENTE:


AT'N L. EN D. MARCO RODRIGO LÓPEZ GONZÁLEZ
RESPONSABLE DEL DEPARTAMENTO DE TITULACIÓN.

Con base en las revisiones efectuadas al trabajo escrito titulado "DATAWAREHOUSE PARA CONTROL DE STOCK EN FORD ZAPATA S.A. DE C.V." que para obtener el título de **Licenciado en Informática Administrativa** presenta el sustentante **Cristian Ruiz Hernández** con número de cuenta **0814953** respectivamente, se concluye que cumple con los requisitos teórico-metodológicos por lo que se le otorga el voto aprobatorio para su sustentación, pudiendo **continuar con la etapa de digitalización** del trabajo escrito.

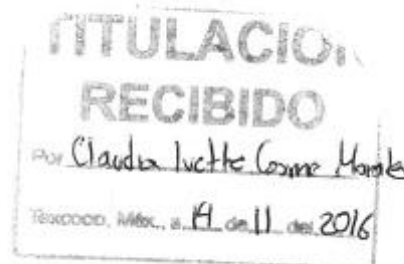
ATENTAMENTE


DR. EN C. ADRIÁN TRUEBA ESPINOSA
REVISOR


M. EN C. JOSÉ SERGIO RUIZ CASTILLA
REVISOR


M. EN C.C. JUAN CARLOS MORENO SÁNCHEZ
DIRECTOR

C.e.p. Sustentante.- CRISTIAN RUIZ HERNÁNDEZ
C.e.p. Director.- M. EN C.C. JUAN CARLOS MORENO SÁNCHEZ
C.e.p.-Titulación.- L. EN D. MARCO RODRIGO LÓPEZ GONZÁLEZ



AGRADECIMIENTOS

Le agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en momentos de debilidad, por brindarme una vida llena de aprendizajes y experiencias, por protegerme durante todo mi camino, además de darme fuerzas para superar los obstáculos y dificultades a lo largo de mi vida, sobre todo por darme felicidad.

Gracias a Dios por la vida de mis padres, también porque cada día bendice mi vida con la hermosa oportunidad de estar y disfrutar al lado de las personas que sé que más me aman, y a las que yo sé que más amo en mi vida, gracias por permitirme amar a mis padres, gracias a mis padres por permitirme conocer de Dios y de su infinito amor.

Gracias a mis padres por ser los principales promotores de mis sueños, gracias a ellos por cada día confiar y creer en mí y en mis expectativas,

Gracias a la vida por este nuevo triunfo, gracias a todas las personas que me apoyaron y creyeron en la realización de esta tesis.

Agradezco al M. en C.C. Juan Carlos Moreno Sánchez por su apoyo en la dirección y realización de esta tesis, la oportunidad de trabajar con él durante el desarrollo de este proyecto, que en los momentos difíciles estuvo ahí para apoyarme, gracias por sus consejos y a su experiencia he aprendido demasiado.

Gracias Dr. Adrián Trueba Espinosa y Dr. en I. de S. José Sergio Ruiz Castilla por darme la oportunidad y por el tiempo que me han dedicado para leer este trabajo y por ese conocimiento transmitido durante el desarrollo de este proyecto.

DEDICATORIAS

El amor recibido, la dedicación y la paciencia con la que cada día se preocupaban mis padres por mi avance y desarrollo de esta tesis, es simplemente único y se refleja en la vida de un hijo.

Gracias a mi madre por estar dispuesta a acompañarme cada larga y agotadora noche de estudio, agotadoras noches en las que su compañía era para mí como agua en el desierto, mamá tú eres la persona que siempre me ha levantado los ánimos tanto en los momentos difíciles de mi vida estudiantil como personal. Gracias por tu paciencia y esas palabras sabias que siempre tienes para mis enojos, mis tristezas y mis momentos felices, por ser mi amiga y ayudarme a cumplir mis sueños.

A mi padre por desear y anhelar siempre lo mejor para mi vida, gracias por cada consejo y por cada una de sus palabras que me guiaron durante mi vida, gracias por tu apoyo, la orientación que me has dado, por iluminar mi camino y darme la pauta de poder realizarme en mis estudios y mi vida. Agradezco los consejos sabios que en el momento exacto ha sabido darme para no dejarme caer y enfrentar los momentos difíciles, por ayudarme a tomar decisiones que me ayuden a balancear mi vida.

A mis hermanos Suanny y Jhonatan, que con su cariño hacia a mí me ha enseñado a salir adelante. Gracias por su paciencia y preocuparse por su hermano mayor, por compartir sus vidas conmigo, pero sobre todo por estar en otro momento tan importante de mi vida, espero esto sea un motivo para seguir superándose y realizar sus sueños.

A Nancy por formar parte de este momento tan importante en mi vida, gracias por tu amor, por ser como eres, por presionarme para terminar este trabajo, gracias por aguantarme, por apoyarme, pero sobre todo gracias por enseñarme a creer en mí y motivarme hacer las cosas de mejor manera, gracias por todo.

A todos aquellos que no están aquí, pero me hicieron creer en mí para concluir mis estudios profesionales para mi futuro laboral, gracias a mi abuelita Natividad que siempre quiso que su nieto llegara a ser una persona exitosa y con un futuro mejor, gracias.

CONTENIDO

I. Resumen	1
II. Antecedentes de la temática.....	3
III. Importancia del problema	5
IV. Pregunta de investigación	6
V. Marco conceptual.....	7
5.1 Sistema de información.....	7
5.1.1 Componentes de los sistemas de información.	9
5.1.2 Clasificación de los sistemas de información.	9
5.1.3 Sistemas de información en áreas funcionales de empresas.	12
5.1.4 Operaciones de un sistema de información.	14
5.1.5 Ventajas de los sistemas de información.	15
5.1.6 Tecnologías de información (TI).	16
5.2 Modelo de desarrollo del software	17
5.2.1 Ingeniería de software.	17
5.3 Modelos del ciclo de vida del software	19
5.3.1 Modelo incremental.	20
5.3.2 Modelo en espiral.....	21
5.3.3 Modelo diente de sierra.....	22

5.3.4	Modelo en cascada.....	23
5.4	Lenguaje de modelado UML (Unified Modeling Language)	28
5.4.1	Casos de Uso.....	28
5.4.2	Diagramas de casos de uso.....	29
5.5	Bases de datos	34
5.5.1	Niveles de abstracción de una base de datos.....	34
5.5.2	Características de las bases de datos.....	36
5.5.3	Tipos de bases de datos.....	37
5.6	Modelado de base de datos.....	38
5.6.1	Características.....	39
5.6.2	Objetos de la base de datos relacional.....	43
5.6.3	Tipos de datos.....	44
5.7	Sistemas gestores de bases de datos.....	46
5.7.1	Manejador de base de datos.....	46
5.7.2	Función de los sistemas de gestión de bases de datos.....	49
5.7.3	Esquema.....	50
5.7.4	Diccionario de datos.....	50
5.8	Normalización.....	51
5.8.1	Concepto de normalización.....	51
5.8.2	Grados de normalización.....	52
5.9	Datawarehouse.....	54

5.9.1	Tipos.....	55
5.9.2	Procesos.	56
5.9.3	Características.	57
5.9.4	Aportaciones de un datawarehouse.....	58
5.10	Java	59
5.10.1	Objetivos.	59
5.10.2	Características.	60
5.11	Patrón MVC (Modelo - Vista - Controlador)	63
5.11.1	Interacción con un MVC.....	64
5.12	MySQL	66
5.12.1	Características.	66
5.13	SQL.....	68
5.13.1	Lenguaje de definición de datos (DDL).....	69
5.13.2	Lenguaje de manipulación de datos (DML).	70
5.14	Marco referencial.....	72
5.14.1	Empresas y software referenciales	72
5.14.2	EASY-WMS (Sistema de fácil gestión de inventario)	72
5.14.3	Inventoria	73
VI.	Métodos y técnicas de investigación empleadas	74
6.1	Desarrolló de la metodología	76
6.1.1	Investigación preliminar.	76

6.1.2	Análisis de los requerimientos.	79
6.1.3	Modelado y diseño del SI con UML (Casos de uso).	81
6.1.4	Desarrolló de la base de datos.	84
6.1.5	Desarrolló del modelo relacional.	94
6.1.6	Planteamiento de algoritmos.	99
6.1.7	Implementación de algoritmos.	100
6.1.8	Pruebas de validación y verificación del sistema.	114
6.1.9	Implementación del sistema.	115
VII.	Presentación y discusión de resultados.	116
VIII.	Conclusiones y sugerencias.	133
IX.	Referencias de consulta.	134
X.	Anexos.	138

ÍNDICE DE FIGURAS

Figura 1 Sistemas de información (Kenneth C. L., 2008).	8
Figura 2 Diagrama de operaciones de un sistema de información (Kenneth C. L., 2008).	14
Figura 3 Fases del ciclo de vida del desarrollo de sistemas según (Pressman, 2005).	19
Figura 4 Esquema del modelo incremental (Weitzenfeld A. , 2005).	20
Figura 5 Diagrama del modelo en espiral (Weitzenfeld A. , 2005).	21
Figura 6 Modelo diente de sierra (Weitzenfeld A. , 2005).	22
Figura 7 Esquematización del modelo en cascada (Pressman, 2005).	24
Figura 8 Proceso de desarrollo de software (Elaboración propia).	28
Figura 9 Representación de un caso de uso (Elaboración propia).	29
Figura 10 Diagrama de casos de uso para control de stock (Elaboración propia).	30
Figura 11 Relación de asociación (Elaboración propia).	32
Figura 12 Relación de herencia (Elaboración propia).	33
Figura 13 Tipos de bases de datos según (Hansen, 2002).	37
Figura 14 Partes del modelo relacional (Raghu, 2007).	38
Figura 15 Relación uno a muchos (Elaboración propia).	41

Figura 16 Relación muchos a muchos (Elaboración propia).	41
Figura 17 Relación uno a uno (Elaboración propia).	41
Figura 18 Modelo jerárquico (Bertino, 2001).	42
Figura 19 Modelo de red (Bertino, 2001).	42
Figura 20 Funcionamiento del SGBD (Bertino, 2001).	47
Figura 21 Mecanismos datawarehouse con datos de producción (Fon, 2008).	55
Figura 22 Formato de solicitud de almacenista (Elaboración propia).	77
Figura 23 Formato de búsqueda de piezas (Elaboración propia).	78
Figura 24 Diagrama de casos de uso (Elaboración propia).	82
Figura 25 Recolección de datos de SAP (Elaboración propia).	84
Figura 26 Registros obtenidos por SAP (Elaboración propia).	85
Figura 27 Separación de tablas MC9 (Elaboración propia).	85
Figura 28 Presentación de tablas obtenidas en 1ra normalización (Elaboración propia).	85
Figura 29 Primera forma normal (Elaboración propia).	86
Figura 30 Segunda forma normal (Elaboración propia).	87
Figura 31 Tablas con asignación de nombres y atributos (Elaboración propia).	88

Figura 32 Diagrama de clases del sistema (Elaboración propia).	90
Figura 33 Creación de id en la tabla pieza (Elaboración propia).	94
Figura 34 Asignación de atributos y tipos de datos (Elaboración propia).	95
Figura 35 Asignación de la llave primaria (Elaboración propia).	96
Figura 36 Tablas obtenidas de la normalización (Elaboración propia).	97
Figura 37 Modelo relacional del sistema implementado (Elaboración propia).	98
Figura 38 Menú de archivo (Elaboración propia).	101
Figura 39 Menú de mantenimiento (Elaboración propia).	102
Figura 40 Menú de consultas (Elaboración propia).	102
Figura 41 Menú de reportes (Elaboración propia).	104
Figura 42 Menú de administrador (Elaboración propia).	104
Figura 43 Menú de sistema (Elaboración propia).	105
Figura 44 Interfaz de invitado (Elaboración propia).	105
Figura 45 Interfaz de administrador (Elaboración propia).	105
Figura 46 Código del paquete DAO (Elaboración propia).	107
Figura 47 Código de la clase DAO (Elaboración propia).	107

Figura 48 Código de creación de almacén (Elaboración propia).	109
Figura 49 Código de búsqueda de almacén (Elaboración propia).	109
Figura 50 Código de actualización de almacén (Elaboración propia).	109
Figura 51 Código de eliminación de almacén (Elaboración propia).	110
Figura 52 Código del paquete Delegate (Elaboración propia).	110
Figura 53 Código para el acceso a la base de datos (Elaboración propia).	111
Figura 54 Código para realizar movimientos de almacén (Elaboración propia).	111
Figura 55 Código del paquete Facade (Elaboración propia).	112
Figura 56 Código de conexión a la base de datos (Elaboración propia).	112
Figura 57 Código de asignación de variables (Elaboración propia).	113
Figura 58 Interfaz de acceso al sistema (Elaboración propia).	116
Figura 59 Interfaz de bienvenida al usuario (Elaboración propia).	117
Figura 60 Interfaz de administrador (Elaboración propia).	118
Figura 61 Interfaz de registro de pieza del invitado (Elaboración propia).	119
Figura 62 Interfaz de registro de pieza del administrador (Elaboración propia).	120
Figura 63 Interfaz de consulta por clave (Elaboración propia).	122

Figura 64 Interfaz de reporte de existencias (Elaboración propia).	125
Figura 65 Resultado de búsqueda en formato de Excel (elaboración propia).	128
Figura 66 Registro de solicitud de pedido (Elaboración propia).	129
Figura 67 Exportación de material existente (Elaboración propia).	130

ÍNDICE DE TABLAS

Tabla 1 Errores de Excel (Elaboración propia).....	4
Tabla 2 Componentes de Sistemas de información (Gómez, 2010).	9
Tabla 3 Características de una base de datos propuestas por (Raghu, 2007).	36
Tabla 4 Características de los cinco modelos de bases de datos (Hansen, 2002).	40
Tabla 5 Terminología estructural (Ramez, 2007).	43
Tabla 6 Tipos de datos en Java (Pérez, 2003).....	45
Tabla 7 Niveles de normalización (Kcoenke, 2003).....	53
Tabla 8 Narración de casos de uso (Elaboración propia).....	82
Tabla 9 Descripción de los atributos agregados en las tablas (Elaboración propia).	90
Tabla 10 Descripción del menú de mantenimiento de SICSARF (Elaboración propia).	120
Tabla 11 Descripción del menú de consultas de SICSARF (Elaboración propia).	123
Tabla 12 Descripción del menú de reportes de SICSARF (Elaboración propia).	125

I. Resumen

Ford Motor Company, más conocida como Ford, es una empresa multinacional fabricante de automóviles, camiones y microbuses. La sucursal ubicada en Tlalnepantla, está organizada en departamentos: Recursos Humanos, Administración, Mercadotecnia, Producción, Contabilidad, Sistemas, Distribución, Refacciones, Compras, Ventas y Almacén.

En el departamento de compras de refacciones automotrices, manejan grandes cantidades de datos, llegando a tener en el inventario hasta 15,000 piezas. Aquí existen 2 tipos de comportamiento, las piezas de movimiento lento y las de movimiento rápido. Las piezas de movimiento rápido, se venden y se usan de manera constante lo cual obliga al usuario a tener un control referente a, si se cuenta con existencias de estas piezas. Por el contrario, las piezas de movimiento lento, no son usadas con frecuencia, por lo cual estas pueden ser descontinuadas, además de representar un gasto para la empresa. Por estos problemas, es necesario contar con un sistema que permita conocer la existencia de un material para poder darle un tratamiento adecuado a este.

Al realizar el pedido de piezas en el área de compras de refacciones, el tiempo de solicitud de material es un problema, ya que las piezas que se encuentran en cada almacén, hacen un total de 12,000 a 15,0000 piezas, siendo esta cantidad de datos representada en un listado dentro de Excel muy amplia, para poder realizar la búsqueda de piezas que son solicitadas.

En el presente proyecto se propone un sistema de información para una empresa de compra de refacciones, SICSARF (Sistema Informático para Control de Stock en el Área de Refacciones de Ford). Cuyo objetivo es mejorar el flujo de información, así como de, facilitar la búsqueda de piezas dentro de cada almacén realizando el pedido en tiempo y forma. Es por ello que se

implementó un sistema de información que permita la toma de decisiones de forma rápida, eficiente, confiable y oportuna.

Considerando que no existe un sistema en el mercado, que realice una comparación de material de inventario con material que se solicita. Se tuvo la encomienda en diseñar y programar un sistema de información en Ford, para el área de compras de refacciones. Los resultados obtenidos permitieron que la empresa tenga un procedimiento de operación más rápida, ya que, se automatizó el proceso para saber la localización de la pieza solicitada de cada almacén.

Uno de los resultados es que el sistema de información (SICSARF), es un almacén de datos con la información histórica del comportamiento de cada uno de los inventarios y material automotriz que posee la empresa Ford, actualmente solo se obtiene una lista de estas piezas, para evitar la duplicidad de material y permitir un ahorro a Ford, pero el almacén de datos permite consultas futuras, referentes a las piezas más usadas en un almacén, con el fin de pedir las directamente al proveedor de estos almacenes y así evitar gastos relacionados con envíos entre almacenes.

De esta manera el administrador y el encargado, puedan tomar la decisión adecuada en comprar o no piezas que se solicitan. Además se pudo concluir que con la implementación de SICSARF se genera un mejor desempeño de las labores del área de compras de refacciones en cuanto a la realización de los procesos en forma automatizada.

II. Antecedentes de la temática

Ford es una empresa multinacional fabricante de automóviles, camiones y microbuses, su actividad primordial a la que se dedica, es a la venta de autos nuevos y seminuevos. En esta sucursal se estima que se venden 250 unidades nuevas. Además se le ofrece al cliente el mantenimiento de autos nuevos, por lo tanto, a seminuevos servicio y compostura a las unidades de Ford.

La sucursal de Tlalnepantla cuenta con departamentos: Recursos humanos, Administración, Contabilidad, Mercadotecnia, Producción, Contabilidad, Sistemas, Distribución, Refacciones, Compras, Ventas y Almacén. En del área de compras de refacciones, son realizados procesos para llevar el cumplimiento de compras de material de acuerdo a lo solicitado en mostrador y taller automotriz.

Este proceso se realiza en Excel para obtener la búsqueda del material que se encuentran en cada sucursal. El encargado del almacén envía a planta (Tlalnepantla) un formato llamado PedidoDiario, este es obtenido de cada asesor ya que es la única persona encargada de levantar el pedido al cliente. Una vez que el encargado haya verificado el material a solicitar este es enviado a planta. La persona encargada de realizar las compras recibe el formato del material que está solicitando el almacenista de cada sucursal, realiza el cotejo de los pedidos para tener todo el material en un formato, además obtiene un listado de todo el material que se encuentra en el inventario de cada sucursal mediante el sistema SAP que maneja Ford, llamado MC9.

Teniendo ambos formatos se inicia la búsqueda de material aplicando formulas en Excel, esto conlleva a una complejidad elevada debido a que dichas formulas pueden presentar errores o los resultados de los filtro tengan que ser depurados de forma manual, además de estos problemas

la solicitud de pedido tiene un cierto límite para poder ser elaborado, esto ocasiona que trabajen en marchas forzadas para poder generar este documento de búsqueda, ya que dentro de este documento se verifican las piezas que hay en existencia o inexistencias. Este proceso es de suma importancia, ya que existen actividades que necesitan ser ejecutadas, por ejemplo, la revisión de *BackOrder* (Búsquedas de piezas que no fueron surtidas un día anterior).

Microsoft Excel es sin duda uno de los programas más potentes y flexibles en las empresas, sin embargo, su mayor virtud se puede convertir en una fuente de errores comunes para el encargado del área de compras de refacciones a la hora de trabajar con demasiada información, a continuación, se muestra en la Tabla 1 dichos errores que se pueden presentar.

Tabla 1 Errores de Excel (Elaboración propia).

Errores de Excel	
Error	Descripción
Error #1: Usar una versión obsoleta.	Al usar el encargado una versión anterior, puede modificar la cantidad de los datos ya que esta no soporta aun demasiada información, dificultándose y realizar doble trabajo de búsqueda.
Error #2: No dedicar tiempo a aprender Excel.	Debido a que Excel maneja gran variedad de fórmulas y algoritmos, al no estar actualizado el encargado este no podrá realizar las operaciones correspondientes para llevar a cabo la solicitud de pedido de forma adecuada.
Error #3: Utilizar demasiadas hojas	A la hora de realizar la búsqueda de material dentro del inventario, la persona encargada de compras tarda en buscar el pedido diario, la sucursal que lo solicita, el inventario o si la pieza solicitada se encuentra en existencia o no.
Error #4: Repetir la misma tarea	Debido a que el encargado realiza el mismo proceso diariamente, se pueden tener errores humanos por factores de cansancio, información compleja, formulas mal planteadas, considerando esto en área de compras de refacciones puede existir una duplicación de material llevando a que realice Ford un gasto innecesario.
Error #5: Aplicar excesivos formatos y formulas	Debido a que la realización de pedido debe estar en un tiempo determinado, los datos excesivos y las formulas no permiten al usuario poder realizar la búsqueda de material en un menor tiempo ya que con esta saturación dentro del libro de Excel se vuelve más lento y pesado recalcular.

III. Importancia del problema

En el área de compras de refacciones, para obtener la búsqueda de material que son solicitadas en mostrador y en taller automotriz, se realizan operaciones en Excel de forma manual, obteniendo de esta manera la ubicación del material, este proceso consume demasiado tiempo para el encargado de compras, al implementar el sistema de SICSARF se reduce el tiempo de búsqueda.

El proceso para poder obtener el listado de búsqueda de refacciones es complejo, debido a que la cantidad de refacciones automotrices que se tiene en inventario, es aproximadamente de 15,000 piezas, al ser una cantidad muy alta existe una gran probabilidad de que el encargado del área de compras cometa errores, debido a factores como cansancio, exceso de trabajo y el trabajo bajo presión, al implementar el sistema SICSARF, este garantiza que la información requerida en el área de compras no presente errores por factores humanos.

Excel es una herramienta en la cual se puede escribir, almacenar, manipular, calcular y organizar todo tipo de información ya sea numérico o texto, sin embargo, también puede convertirse en una fuente de errores, tales como usar una versión anterior, ir a atajos no solicitados, los límites de los listados son limitados dependiendo de la versión, funciones de fórmulas muy lentas por rangos de listados muy amplios, los números son convertidos a texto al ser importados de fuentes externas (Avalos, 2010). Con la implementación de SICSARF se garantiza que no contendrá ningún error al realizar el proceso de búsqueda.

Con el uso del sistema SICSARF el encargado del área de compras puede dedicar su tiempo en otras actividades, siendo así que al implementarse el sistema este estará absorbiendo el tiempo de realización de búsqueda, mientras que el encargado podrá realizar otras actividades para tener una mejor coordinación, visualización y claridad.

IV. Pregunta de investigación

¿Cuál es el ahorro de tiempo, al implementar SICSARF para control de stock de refacciones en Ford Zapata S.A. de C.V.?

V. Marco conceptual

5.1 Sistema de información

Para definir un sistema de información es necesario conocer lo que es un dato, información, y un sistema. A continuación se define cada una:

Dato es, una representación simbólica (numérica, alfanumérica, algorítmica, etc.) de un atributo o variable cuantitativa o cualitativa. Un dato por sí mismo no constituye información, es el procesamiento de los datos que la proporciona (Senn, 2008).

Información es, un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que percibe dicho mensaje. A diferencia de los datos, la información tiene significado. No solo se puede formar potencialmente al que la percibe, sino que está organizada para algún propósito (Turban, 2001).

Sistema se define como un conjunto de elementos o individuos que forman un todo organizado que interactúan entre sí y tienen una conducta coherente. Los sistemas pueden ser cerrados o abiertos, un sistema abierto es cuando tiene acción recíproca o interdependencia con elementos de su ambiente. Cuando no existe tal acción con el medio, se trata de un sistema cerrado (Edwards, 2001).

De acuerdo a esto, un sistema de información es, un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. De tal manera que un sistema de información tiene una determinada eficiencia que se refiere a hacer las cosas bien, para obtener el máximo rendimiento, utilizando un mínimo de recursos (Velthuis, Calidad de sistemas

de información, 2012). El sistema de información trabaja en conjunto con la información, determinando una solución a una problemática, obteniendo beneficios para la toma de decisiones.

Esta información, es compuesta por un conjunto ordenado de datos, teniendo esto en cuenta, un dato, es un conjunto de caracteres con algún significado, pueden ser numéricos, alfabéticos, o alfanuméricos. Los datos pueden ser manejados según las necesidades del usuario, siendo un conjunto de datos que pueden ser procesados eficientemente y pueda dar lugar a la información. Mientras tanto los elementos mencionados por (Kenneth C. L., 2008) son los que se muestran en la Figura 1.

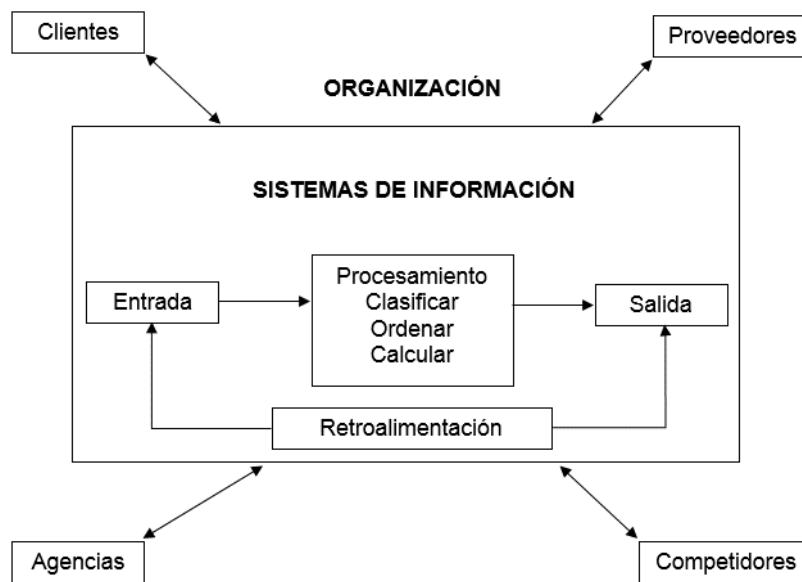


Figura 1 Sistemas de información (Kenneth C. L., 2008).

Un sistema de información, es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad o un objetivo. Para formar un sistema de información, es necesario que

diferentes componentes interactúen entre sí, dentro de estos encontramos los propuestos por (Gómez, 2010), (Tabla 2).

5.1.1 Componentes de los sistemas de información.

Tabla 2 Componentes de Sistemas de información (Gómez, 2010).

Componentes de sistemas de información	
Componentes	Definición
Datos	Son hechos concretos, que el sistema toma para producir información.
Hardware	Una computadora y su equipo periférico: Dispositivos de entrada, salida y almacenamiento. También incluye equipo de comunicación de datos.
Software	Son conjunto de instrucciones, que indican a la computadora como obtener datos, procesarlos, desplegar información y almacenar datos e información.
Telecomunicaciones	Es el hardware y el software los que facilitan la transmisión y recepción rápida de textos, imágenes, sonidos y animación en forma de datos electrónicos.
Personal	Son los profesionales y usuarios que analizan las necesidades de la información de una organización, diseñan y construyen sistemas de información, escriben programas de computadora, operan el hardware y el software.
Procedimientos	Son las reglas para lograr operaciones óptimas y seguras en el procesamiento de datos, estos incluyen prioridades al ejecutar diferentes aplicaciones en la computadora, además de medidas de seguridad.

5.1.2 Clasificación de los sistemas de información.

Los sistemas de información, se llevan a cabo en el área de la empresa de acuerdo a las necesidades del cliente, recabando toda la información necesaria para la manipulación de los datos, llegando al objetivo, ayudando a los encargados a establecer la toma de decisiones, generalmente estas son clasificadas de acuerdo a su nivel.

5.1.2.1 Sistemas de procesamiento de transacciones o PTS.

Son los sistemas de información más utilizados. La función predominante de los PTS consiste en registrar datos reunidos dentro de los límites de las organizaciones; es decir, en el punto en que la organización interactúa con las partes de todas las áreas, con las cuales hace negocios en cualquier empresa. Los PTS incluyen cajas registradoras, que registran ventas; cajeros automáticos, que registran retiros, depósitos y transferencia de efectivo; y sistemas de pedidos de compras, que registran compras. Después de reunir estos datos, los sistemas de información los procesan automáticamente, para convertirlos en información o los almacena para su procesamiento posterior (Senn, 2008).

5.1.2.2 Sistemas de información para la administración o SIA.

Es el sistema de información, que ayuda a los administradores y otros profesionales a planear, controlar, al mismo tiempo obteniendo estrategias para la toma de decisiones. Los niveles de complejidad de los SIA son variados. Hasta cierto grado este tipo de sistemas de información, es utilizado para la administración dependiendo del propio nivel (Gómez, 2010).

5.1.2.3 Sistemas de apoyo para la toma de decisiones o SATD.

Ayudan a encontrar el curso de acción óptimo, respondiendo a cualquier tipo de preguntas que se realizan los encargados, para la toma de decisiones. Los SATD están programados para procesar datos en bruto, hacer comparaciones y generar información que le ayude a los administradores a obtener las mejores alternativas para una inversión financiera, estrategia de mercadotecnia, aprobación de crédito y similares (Kenneth C. L., 2008).

5.1.2.4 Sistemas de información ejecutiva o EIS.

Ayudan a los funcionarios de alto nivel a dirigir una organización, al ser programados para integrar miles de datos, produciendo informes resumidos; estos sistemas son una ayuda para que lo ejecutivos puedan concentrarse en la planeación estratégica a largo plazo en una organización (Senn, 2008).

5.1.2.5 Sistemas expertos o SE.

Estos sistemas de información, se desarrollan incorporando la experiencia humana, se utilizan cuando la toma de decisiones no está estructurada. Ahorrando a la compañía el alto costo de emplear humanos expertos. Después de reunir el conocimiento de los especialistas diseñan un programa, este puede distribuirse o utilizarse varias veces. La experiencia reside en el programa, en forma de una base de datos que incluye hechos y relaciones entre ellos (Kenneth E. K., 2005).

5.1.2.6 Sistemas de información geográfica o SIG.

Es una aplicación vinculada con una base de datos, que contienen datos sobre toda una ciudad, una población, un municipio, una región, un país e incluso de todo el mundo. Al representar los datos de un mapa en diferentes formas geográficas, el usuario puede comprender rápidamente una situación que tiene lugar en esa parte del mundo, actuando de manera más sencilla, ya habiendo tenido los datos requeridos se actúa con rapidez (Senn, 2008).

Este proyecto es adaptado a los Sistemas de Información para la Administración o SIA y los Sistemas de Apoyo para la Toma de Decisiones o SATD, ya que en estos sistemas se lleva a cabo la recopilación de datos e información procesada, haciendo que a los administradores les facilite la planeación, el control y la toma de decisiones, procesando datos complejos para lograr

hacer comparaciones, dándoles alternativas para su inversión financiera con el objeto de reducción de tiempo y costo dentro del área que se desee implementar.

5.1.3 Sistemas de información en áreas funcionales de empresas.

La información es muy importante para la gestión de la empresa. Por lo tanto, los gerentes intentan interactuar y formalizar la información para un buen resultado de datos, que servirán para observar las posibilidades de que la empresa pueda cumplir con sus objetivos. Las áreas funcionales de las empresas son: contabilidad, finanzas, mercadotecnia, recursos humanos, producción entre otros, (Kenneth E. K., 2005):

5.1.3.1 Contabilidad.

Ayudan a registrar transacciones de negocios, producir informes financieros, como balances generales, estados de pérdidas y ganancias. También ayudan a elaborar informes que permiten a los administradores, a entender los cambios en las finanzas de una organización (Gómez, 2010).

5.1.3.2 Finanzas.

Su propósito es facilitar la planeación financiera, así mismo, poder realizar las transacciones de negocios necesarias para un mejor resultado. También ayudan a organizar presupuestos, manejar flujos de caja, analizar inversiones llegando a la toma de decisiones, que podrían reducir los pagos de interés aumentando las ganancias de transacciones financieras (Turban, 2001).

5.1.3.3 Mercadotecnia.

Ayudan a analizar la demanda de productos en diferentes regiones, comercializando con mayor exactitud el producto correcto con los clientes, su propósito es detectar a las personas que poseen mayor posibilidad de comprar lo que vende la organización, promoviendo los productos a través de los servicios apropiados para esas personas que interactúan con la empresa (Kenneth E. K., 2005).

5.1.3.4 Recursos humanos.

Ayudan principalmente a mantener registros, así realizando evaluaciones de sus empleados para saber sus objetivos de ellos con la empresa en la que se encuentran, estos incluyen todos sus datos que pueden utilizar otros sistemas, como una nómina, antigüedad de personal, entre otros (Kenneth C. L., 2008).

5.1.3.5 Producción.

Ayudan a asignar recursos, como personal, materias primas, el tiempo, para optimizar la productividad. Los sistemas con los que se cuentan dependiendo en el departamento que se encuentre, pueden realizar algunos procedimientos como: procesar los pedidos de los clientes, programar la producción, realizan el control de calidad y preparación de documentación de embargo (Kenneth E. K., 2005).

Tomando en cuenta las áreas funcionales anteriores el presente proyecto, está enfocado dentro del área de producción, dado que dentro del área de refacciones se asignan los recursos necesarios, para no tener material fuera de inventario, haciendo que estos incrementen sus piezas obsoletas, agilizando el tiempo para la realización de pedidos y así el usuario tenga mayor tiempo

disponible para realizar otras operaciones que no se hacen diariamente, en este caso una visualización de las piezas en existencia, cumpliendo con un control de stock satisfactorio.

5.1.4 Operaciones de un sistema de información.

Las operaciones principales que se realizan dentro de una empresa, es el tratamiento de la información, tomando en cuenta la entrada de datos, el tratamiento de los mismos, la salida de la información obtenida y el almacenamiento, las cuales (Kenneth C. L., 2008), las define de la siguiente manera (Figura 2):

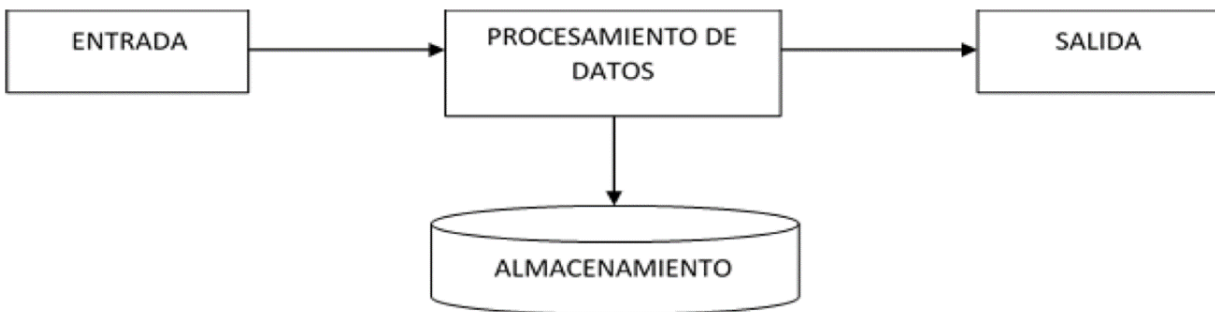


Figura 2 Diagrama de operaciones de un sistema de información (Kenneth C. L., 2008).

5.1.4.1 Entrada.

Aquí se va a recabar e introducir los datos en el “Sistema de Información”, a través de dispositivos de entrada (Kenneth C. L., 2008).

5.1.4.2 Procesamiento de datos.

Es la mayor utilización de la computadora para los sistemas de información, haciendo el procesamiento de datos eficiente, resultando la parte esencial para estos. La velocidad y precisión de la computadora, permiten a las organizaciones procesar millones de componentes de datos por

segundo. Las computadoras también agregan un componente a los sistemas de información, como la necesidad de traducir instrucciones de lenguajes naturales a lenguajes de computadora (Kenneth C. L., 2008).

5.1.4.3 Salida.

Es la información producida y desplegada por un sistema de información, en un dispositivo de salida en el formato más útil para una organización (Kenneth C. L., 2008).

5.1.4.4 Almacenamiento.

La computadora utiliza diversos medios de almacenamiento para datos e información, existen dispositivos internos y externos (Kenneth C. L., 2008).

5.1.5 Ventajas de los sistemas de información.

Las ventajas de los Sistemas de información que menciona (Edwards, 2001) son las siguientes:

- Disponibilidad inmediata de la información, mediante un sistema de información está disponible para todos los usuarios.
- Ayudan al manejo rápido de las cuentas por cobrar.
- Ayuda a reducir los ciclos mensuales del cierre del libro mayor. Esto permite que los administradores, tomen mejores decisiones de acuerdo con el análisis de información financiera más oportuna.

- Reduce el tiempo y el costo de preparar los presupuestos, planes de negocios y propuestas, al hacer disponibles los datos de negocios en tiempo real.
- Intercambio instantáneo de los resultados.

5.1.6 Tecnologías de información (TI).

Los sistemas de información, son la aplicación de ordenadores, así como equipos de telecomunicación para almacenar, recuperar, transmitir y manipular datos. Para que el sistema opere, existen diversos tipos de tecnologías de la información (Turban, 2001) como lo son:

- TI informática de hardware: Incluyen microcomputadoras, servidores de tamaño medio, grandes sistemas centrales y los dispositivos de entrada, salida y procesamiento.
- TI informática de software: Destaca el sistema operativo, navegadores de web, paquetes de software de productividad, al mismo tiempo software para aplicaciones de negocios empresariales.
- TI de redes de telecomunicaciones: Están en los medios de telecomunicación, procesadores y software necesarios para proporcionar acceso e incluso apoyo mediante la red.
- TI de administración de recursos informáticos: El software de sistemas de administración de bases de datos de la empresa.

5.2 Modelo de desarrollo del software

La ingeniería de software, con el fin de ordenar el caos que era anteriormente el desarrollo de software, dispone de varios modelos, paradigmas y filosofías de desarrollo, estos los conocemos principalmente, como modelos o ciclos de vida del desarrollo de software, esto incluye el proceso que se sigue para construir, entregar y hacer evolucionar el software, desde la concepción de una idea hasta la entrega o retiro del sistema, para ello, se es necesario representar todas las actividades y artefactos (productos intermedios) necesarios para desarrollar una aplicación (Schach, 2006).

5.2.1 Ingeniería de software.

El término de ingeniería de software, fue acuñado en 1968 como una respuesta a nivel de progreso, con el objetivo de desarrollar el software de calidad a tiempo y dentro del presupuesto. El énfasis de la ingeniería de software, está en las dos palabras según (Weitzenfeld A. , 2005) en ingeniería y software. Donde un ingeniero, es capaz de construir un producto de alta calidad, usando componentes ya elaborados e integrándolos bajo restricciones de tiempo y presupuesto. El ingeniero, a través de la experiencia, soluciona problemas mal definidos o planteados. La ingeniería de software, es una actividad según (Pressman, 2005) de:

5.2.1.1 Modelado.

Es una actividad de modelado, debido a que el ingeniero de software, necesita primero comprender el ambiente en el que va a operar el sistema, necesita aprender los conceptos del dominio de problema que son relevantes. La mayoría de los sistemas son demasiados complejos para que sean comprendidos por una sola persona y cuesta mucho construirlos e implementarlos (Pressman, 2005).

5.2.1.2 Solución de problemas.

Los ingenieros buscan una solución adecuada, a menudo mediante un ensayo, evaluando alternativas en forma empírica con recursos limitados y con conocimientos incompletos. Según (Pressman, 2005) el método de ingeniería incluye cinco pasos:

- Formular el problema.
- Analizar el problema.
- Buscar soluciones.
- Decidir cuál es la solución más adecuada.
- Especificar la solución.

5.2.1.3 Adquisición de conocimiento.

En el modelado de los dominios de la aplicación y la solución, el ingeniero de software recopila los datos, los organiza en información y los formaliza en conocimiento (Pressman, 2005).

5.2.1.4 Dirigida por recomendación.

Cuando se adquiere conocimiento tomando decisiones acerca del sistema o sus dominios de aplicación, los ingenieros también necesitan captar el contexto en el que se tomaron las decisiones y las razones que hay tras las mismas (Pressman, 2005).

5.3 Modelos del ciclo de vida del software

Un modelo de ciclo de vida del software, representa todas las actividades y productos de trabajos necesarios para desarrollar un sistema. Estos modelos permiten que los gerentes y desarrolladores manejen la complejidad del proceso de desarrollo de software. Según (Pressman, 2005), un modelo de ciclo de vida del software, es un marco de referencia que contienen los procesos, las actividades, las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software.

Hace referencia comúnmente como SDLC (Ciclo de vida del desarrollo de sistemas), como un enfoque por las fases del análisis y diseño de los sistemas, (Pressman, 2005) las divide en 7 fases, ver Figura 3.

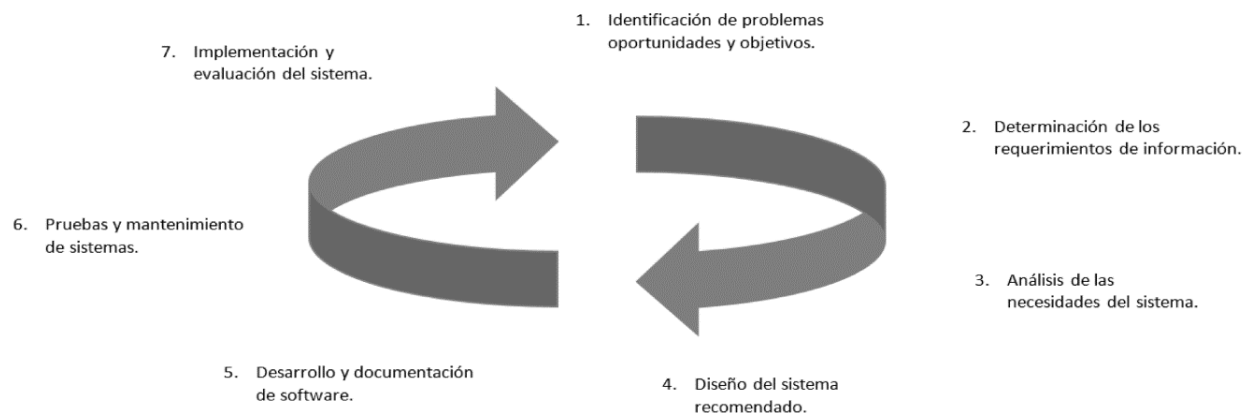


Figura 3 Fases del ciclo de vida del desarrollo de sistemas según (Pressman, 2005).

Estos determinan el orden, las actividades y las tareas que se han de realizar para desarrollar el software. Además, pueden describirse mediante modelos diferentes como el modelo incremental, modelo en espiral, diente de sierra, modelo de cascada entre otros.

5.3.1 Modelo incremental.

Este modelo fue propuesto por Lehman en 1984, corrige la necesidad de una secuencia no lineal de pasos de desarrollo, en la cual se va desarrollando el sistema de software añadiendo componentes funcionales al sistema, ir a Figura 4 para observar el modelo. Algunas características principales que menciona (Weitzenfeld A. , 2005) de este modelo son:

- En cada paso sucesivo, se actualiza el sistema con nuevas funcionalidades o requisitos.
- El sistema de software, ya no se ve como una única entidad monolítica con una fecha fija de entrega, si no como una integración de resultados sucesivos obtenidos después de cada interacción.
- Se ajusta a entornos de alta incertidumbre, por no tener la necesidad de poseer un conjunto exhaustivo de requisitos.

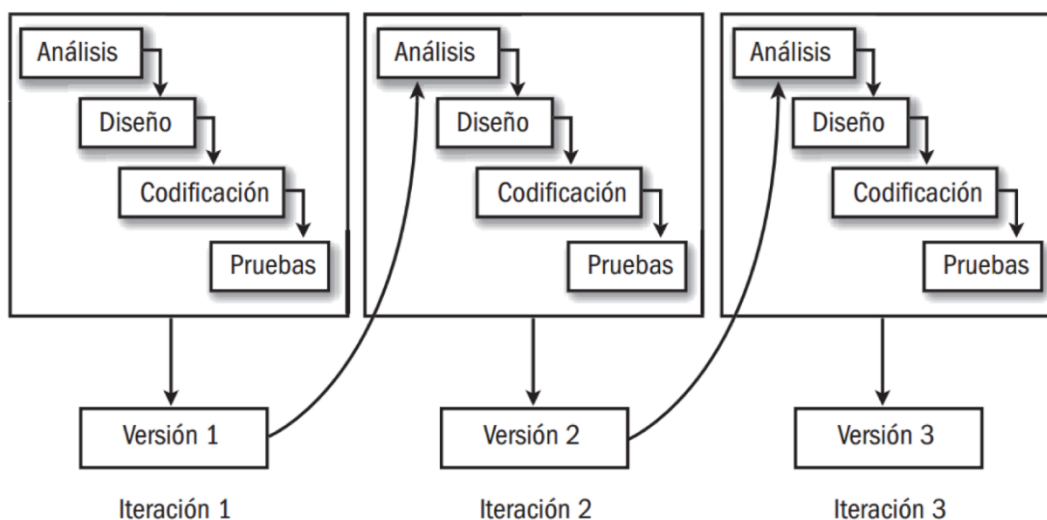


Figura 4 Esquema del modelo incremental (Weitzenfeld A. , 2005).

5.3.2 Modelo en espiral.

Este modelo fue propuesto en 1988 por Boehm, consta de una serie de ciclos, cada uno empieza identificando los objetivos, las alternativas y las restricciones del ciclo (Figura 5), (Weitzenfeld A. , 2005), hace mención de las siguientes características:

- Una vez realizado el primer ciclo se volverá a realizar.
- Cada ciclo se completa con una revisión en la que la participación las principales personas u organizaciones que tienen relación con el producto.
- Los planes para fases sucesivas pueden incluir una participación del producto en incrementos, para desarrollos sucesivos.

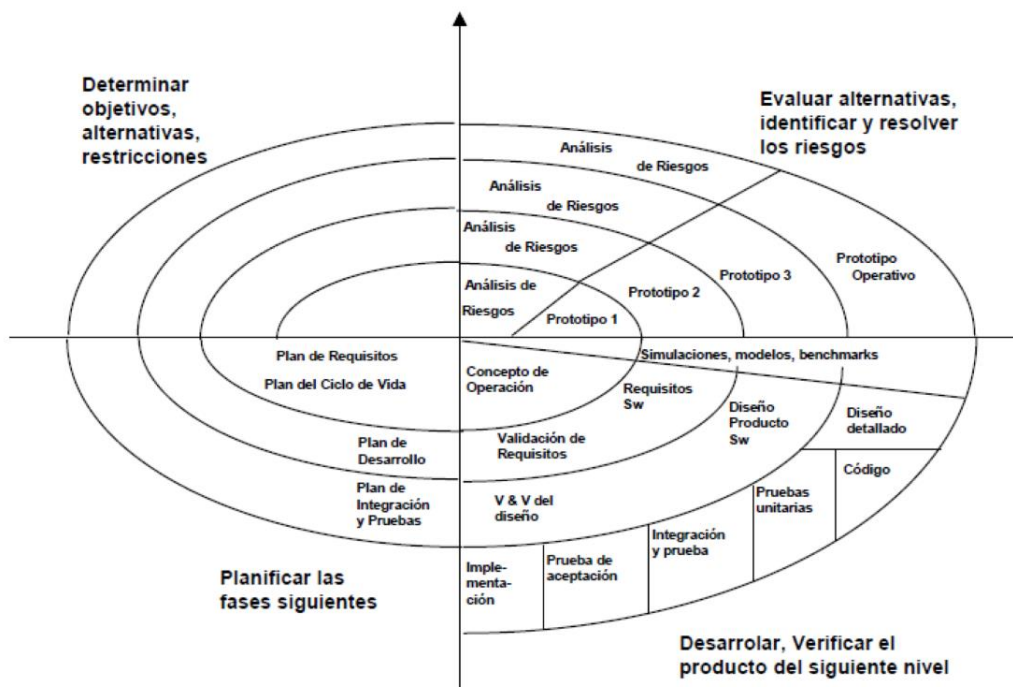


Figura 5 Diagrama del modelo en espiral (Weitzenfeld A. , 2005).

5.3.3 Modelo diente de sierra.

Este modelo fue propuesto por Rowen en 1990, es llamado así, porque cada demostración del prototipo da como resultado un diente. La punta de cada diente, es una intersección con el nivel de abstracciones del cliente. Trata de resolver todas las diferentes necesidades que tienen los usuarios y los implementadores, mostrando las percepciones del sistema por parte del usuario y el desarrollador del software, en diferentes niveles de abstracción a lo largo del tiempo (Figura 6), (Weitzenfeld A. , 2005).

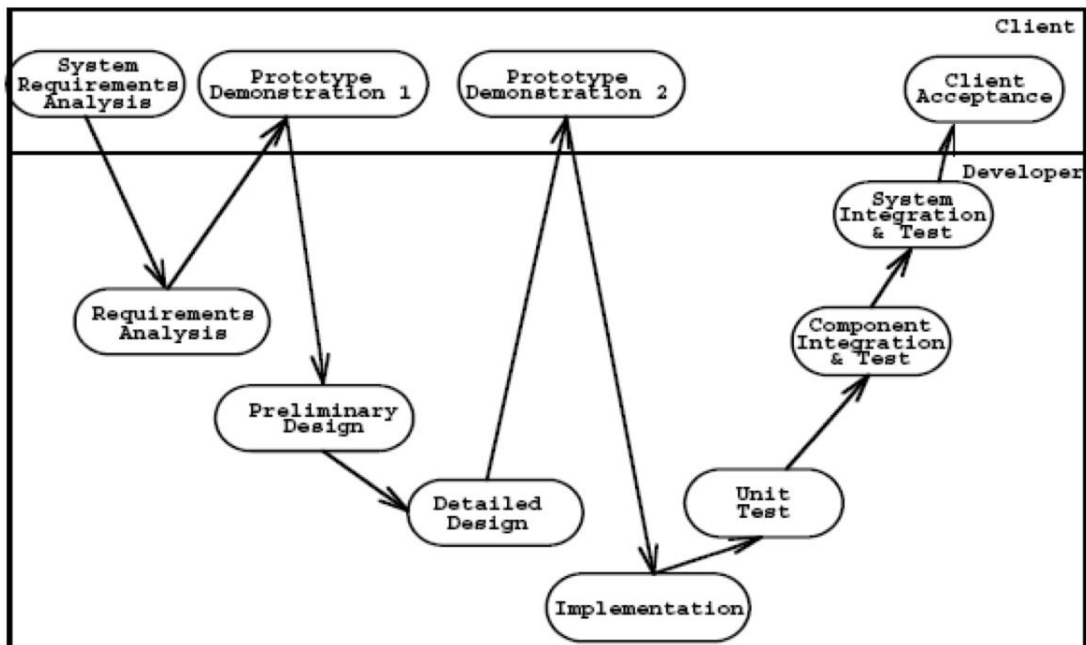


Figura 6 Modelo diente de sierra (Weitzenfeld A. , 2005).

Por ultimo mencionamos el modelo en cascada, que se dará información detallada a continuación. Este será el modelo a utilizar para el desarrollo de este sistema, el cual se implementará más adelante.

5.3.4 Modelo en cascada.

Este modelo también es conocido como el ciclo de vida clásico o lineal secuencial. Fue propuesto por primera vez en 1970 por Royce, desde entonces han aparecido diversas variaciones de dicho modelo (Pressman, 2005). Este modelo exige un enfoque sistemático y secuencial del desarrollo de software, que comienza en el nivel sistema y progresa a través del análisis, diseño, codificación, prueba, implementación y mantenimiento, ver Figura 7. Algunas características que encontramos de este modelo son:

- Todas las actividades de requerimientos, se deben de terminar antes de que comience el diseño del sistema.
- Puede calendarizarse como un proceso gradual, que transforma en códigos las necesidades del usuario.
- Cada fase comienza siempre y cuando la anterior haya sido terminada.
- Para pasar de una fase a otra, es necesario conseguir todos los objetivos de la etapa previa. Para esto, se realiza una revisión final de la base de datos.
- Ayuda a que no se sobrepasen las fechas de entrega y los costos esperados.

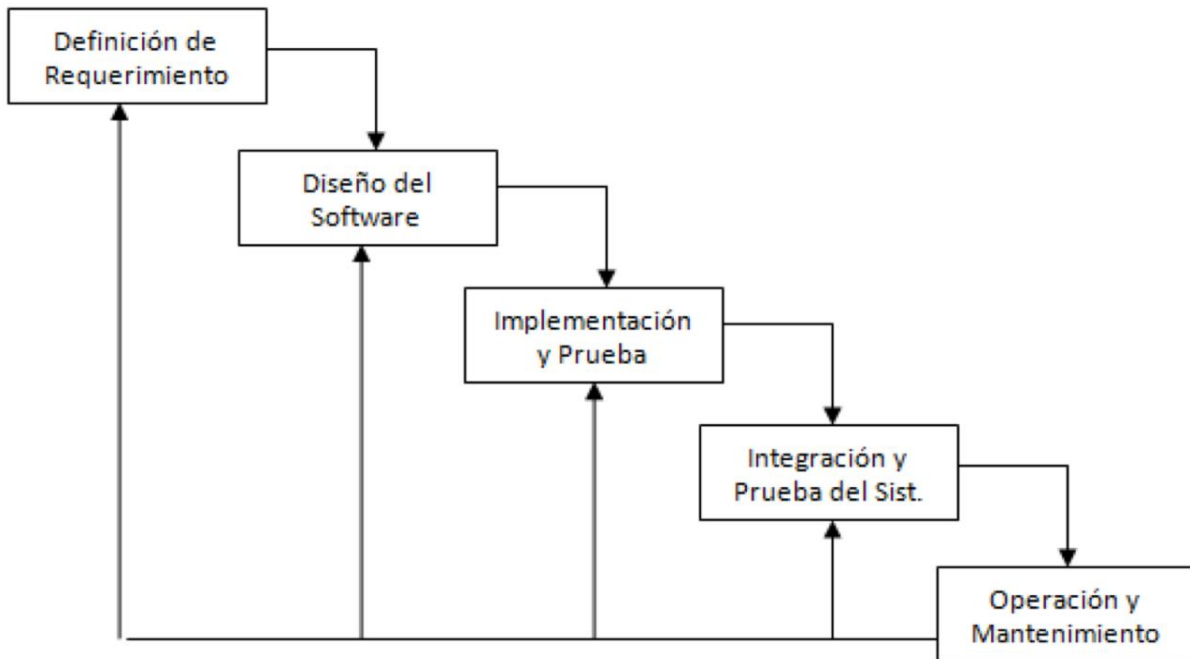


Figura 7 Esquemización del modelo en cascada (Pressman, 2005).

5.3.4.1 Etapas.

Las etapas que conforman este modelo según (Pressman, 2005) son, la recopilación de la información, análisis, diseño, codificación, pruebas, implementación y mantenimiento, las cuales definió así:

5.3.4.1.1 Recopilación de la información.

Esta etapa consiste en reunir toda la información necesaria para efectuar el análisis del sistema, a través de entrevistas preliminares, con la persona que lo solicita, examinar las diferentes fuentes de datos, que disponen del análisis dentro y alrededor de la organización que provienen de dos fuentes (Pressman, 2005):

- Estudio de las fuentes internas: Son las entrevistas que tienen que ver con el sistema que se va a realizar y la documentación con que cuenta la organización.
- Estudio de las fuentes externas: En el estudio de otro sistema de información, que fue una nueva fuente de datos o ideas para el procesamiento; presentación de reportes o aplicación de técnicas, también pueden ser los libros, otras aplicaciones especializadas en el tema.

5.3.4.1.2 Análisis.

En esta fase, se define el proceso de reunión de requisitos, clasificación e interpretación de hechos, diagnósticos de problemas y empleo de la información para recomendar mejoras a los sistemas; recopilación de la información del ambiente del sistema, de la entrada, salida y reportes (Pressman, 2005).

5.3.4.1.3 Diseño.

El diseño de software, es un proceso de muchos pasos que se centra en cuatro atributos distintos de programa: estructura de datos, arquitectura de software, presentaciones de interfaz y detalle procedimental (Algoritmo). Para diseñar el sistema, se debe tener en cuenta los recursos de la institución, las necesidades de la información de los usuarios, los métodos de procesamiento de datos, las operaciones, herramientas de diseño, puntos de control y protección (Pressman, 2005).

5.3.4.1.4 Codificación.

El diseño se debe de traducir en una forma legible por la computadora. Si se lleva acabo el diseño de una forma detallada, la generación de código se realiza mecánicamente. En esta fase se debe seleccionar un lenguaje específico para desarrollar el sistema, justificando cuales son las

ventajas y desventajas con otros lenguajes, para que el programador siga las instrucciones que se siguen en el diseño lógico, que dan como resultado la ejecución de instrucciones para la computadora (Pressman, 2005).

5.3.4.1.5 Prueba.

Una vez generado el código, comienzan las pruebas del programa, este se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores y asegura que la entrada definida produzca los resultados requeridos, dentro de estos tipos de pruebas de validación (Pressman, 2005) menciona las siguientes:

- Pruebas de aceptación: Estas son desarrolladas por el cliente.
- Pruebas alfa: Estas son realizadas con el usuario, de igual manera, con el programador como observadores en un entorno controlado.

5.3.4.1.6 Implementación.

La implementación incluye todas aquellas actividades que se realizan para convertir el anterior sistema al nuevo. El sistema puede ser totalmente nuevo o bien puede ser una modificación o mejora del sistema anterior. Se hace una revisión del sistema por parte del analista, debe ser un proceso formal para determinar que también está funcionando el sistema, incluso cuáles son los ajustes necesarios, ya que es importante para recabar información y realizar el mantenimiento (Pressman, 2005).

5.3.4.1.7 Mantenimiento.

El software indudablemente sufrirá cambios después de ser entregado. Se producirán cambios, porque se han encontrado errores, porque el software debe de adaptarse a los cambios de su entorno, o tal vez porque el cliente requiere mejoras funcionales o de rendimiento. El mantenimiento del software vuelve a aplicar cada una de las fases precedentes a un programa y no a uno nuevo (Pressman, 2005).

5.4 Lenguaje de modelado UML (Unified Modeling Language)

En el desarrollo de sistemas, un requerimiento básico para cualquier modelo, es describir al sistema a detalle, para predecir su comportamiento. Un modelo UML (Lenguaje Unificado de Modelado), se utiliza como ayuda al pensamiento de organizar y clasificar conceptos confusos. Al realizar el análisis, se crea un modelo del sistema que muestre las entidades, las interrelaciones, detallando parte por parte el sistema hasta su totalidad (Booch, 2005).

UML es una anotación grafica para dibujar diagramas de conceptos de software. Como un diseño de software propuesto o una implementación. El proceso unificado, es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema (Figura 8). Sin embargo este es más que un simple proceso, siendo un marco de trabajo genérico que puede especializarse para gran variedad de sistemas de software para diferentes áreas de aplicación, diferentes organizaciones, niveles de aptitud y diferentes tamaños de proyectos (Jacobson, 2000).

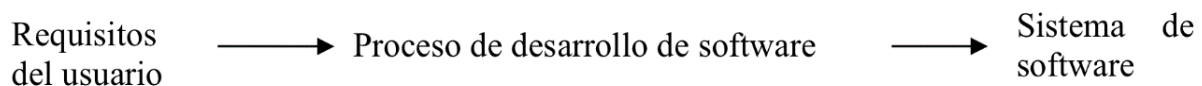


Figura 8 Proceso de desarrollo de software (Elaboración propia).

5.4.1 Casos de Uso.

En mucho tiempo las personas se auxiliaban de escenarios típicos para poder comprender los requerimientos del sistema, pero estos eran de forma informal ya que se escribía, pero muy pocas veces se documentaba. Uno de los retos fundamentales para el desarrollo del sistema de

información, y especialmente para el analista de sistemas, es la capacidad de tener involucrados los requerimientos correctos, necesarios especificándolos de manera que se verifiquen y se validen.

La parte más crucial en el desarrollo de un sistema de información, es especificar lo que va a construirse, de esto depende el éxito o fracaso del proyecto, es por eso que se requiere un alto nivel de participación por parte del usuario (Whitten, 2008). Un caso de uso, es una sintonía instantánea de algún aspecto del sistema. La suma de todos los casos de uso constituye a la vista externa del sistema, siendo un gran avance a la explicación de lo que realiza el sistema.

Es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, es decir, que presenta la funcionalidad del sistema tras la orden del actor. El nombre del caso de uso debe ser la función específica que el usuario desea realizar utilizando el sistema, ver Figura 9 (Kendall, 2006).



Figura 9 Representación de un caso de uso (Elaboración propia).

5.4.2 Diagramas de casos de uso.

El diagrama de casos de usos, es también parte del UML. Siendo este un diagrama que ilustra las interacciones entre sistemas y usuarios. En otras palabras, describe gráficamente quien va a usar el sistema y de qué manera el usuario espera interactuar con él. Dentro del diagrama de casos de usos se representa al sistema, como un recuadro con el nombre en su interior, estos casos

de uso se encuentran fuera ya que estos representan un agente externo que va a interactuar, pero no está dentro de él (Figura 10), (Kendall, 2006).

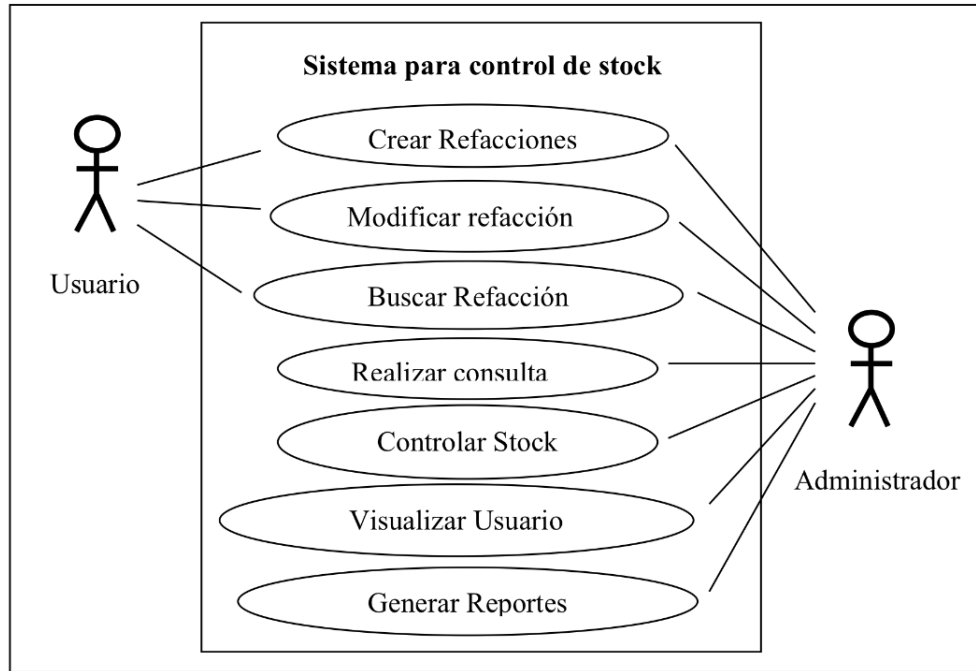


Figura 10 Diagrama de casos de uso para control de stock (Elaboración propia).

5.4.2.1 Actor.

Los casos de usos se inician o son generados por los usuarios externos llamados actores, un actor es definido como un rol que el usuario desempeña con respecto al sistema, cabe mencionar que un actor no representa a una persona, sino la labor que desempeña frente a un sistema, esta es representada mediante una figura humana con líneas, siendo que este es un agente externo se dibuja fuera de él según (Whitten, 2008).

Es importante observar que hay cuatro tipos de actores según (Kendall, 2006), los cuales se mencionan a continuación:

- Actor primario de negocios: Se beneficia en la ejecución del caso de uso, al recibir algo de valor medible u observable. Este actor puede o no iniciar un evento referente al negocio.
- Actor primario de sistemas: Interactúa directamente con la interfaz del sistema, por lo tanto este actor inicia el evento de negociación.
- Actor interno servidor: Involucrado que responde a una solicitud desde el caso de uso.
- Actor externo servidor: Involucrado que no es el actor primario, pero recibe un valor medible u observable.

5.4.2.2 Relaciones.

Las relaciones describen la interacción entre los actores y los casos de uso. En ocasiones resulta útil escribir más a detalle sobre un caso de uso completo para el usuario (Kendall, 2006).

5.4.2.3 Asociación.

Este tipo de relación, se da entre un caso de uso y un actor. Dicha relación, es representada respectivamente por una línea continua (Figura 11), (Kendall, 2006).

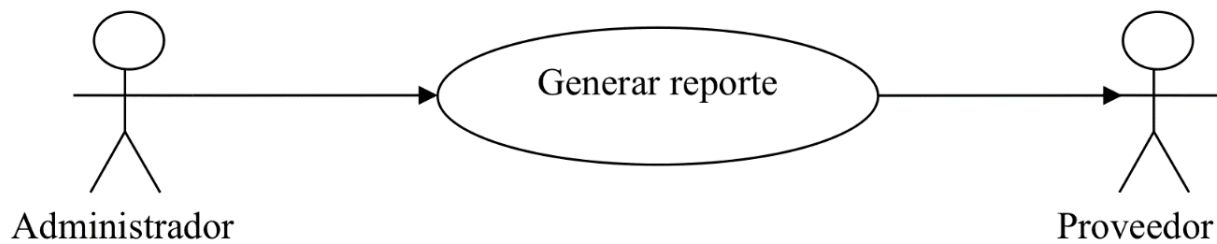


Figura 11 Relación de asociación (Elaboración propia).

5.4.2.4 Extensión.

Este tipo de relación, se dan entre dos casos de uso, en ocasiones estos describen una funcionalidad compleja que consisten en varios pasos, que hacen difícil entender la lógica del caso de uso; con el fin de entenderlos se hacen extracciones de pasos más complejos para formar su propio caso (Kendall, 2006).

5.4.2.5 Usos.

Indica que un caso de uso, es incluido en otro, esto ocurre normalmente cuando estos comparten pasos en común, lo que hace es extraer los pasos comunes y crear su propio caso, en el cual puedan ser llamados ambos casos (Kendall, 2006).

5.4.2.6 Dependencia.

Es la relación entre casos que identifican que procesos llevan una secuencia, esto para los propósitos de condiciones de uso y pruebas (Kendall, 2006).

5.4.2.7 Herencia.

Este tipo de relaciones, es común entre los actores, cuando dos o más actores comparten un comportamiento en común, se extraen para asignar un resumen entre ambos, esto para evitar la repetición en el sistema (Kendall, 2006).

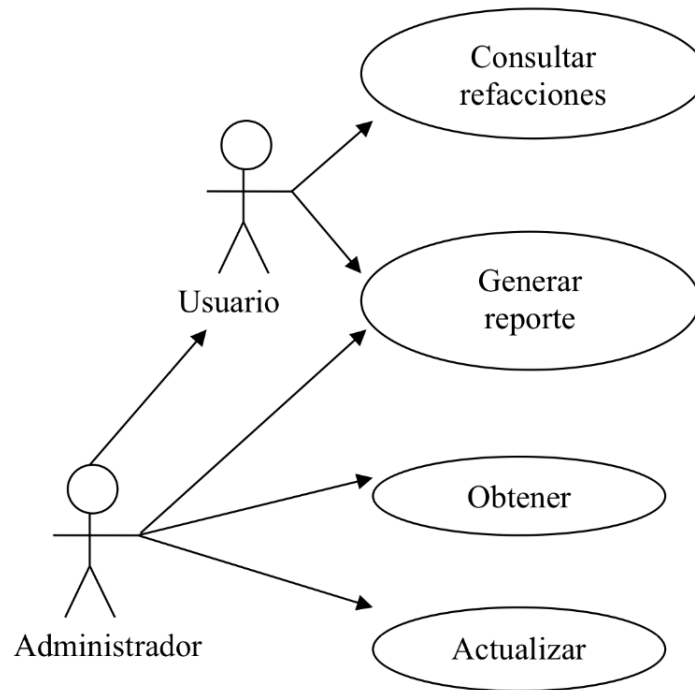


Figura 12 Relación de herencia (Elaboración propia).

5.5 Bases de datos

La necesidad de organizar y recuperar información, ha llevado a crear diversos métodos, para los cuales, son utilizadas herramientas que permitan realizar esta labor más rápida y fácil. Una de ellas son las bases de datos, siendo estas que permiten agrupar una serie de información que puede ser registrada, ordenada, analizada, clasificada y consultada por uno o varios usuarios. Se puede definir una base de datos (BD), como una colección o depósito de datos integrados, con redundancia ordenada, por tanto, debe contar con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real (Castaño, 2006).

Según (Velthuis, Diseño de bases de datos relacionales, 2000), define base de datos como información almacenada en archivos de computadora, siendo accesibles vía terminal remota y liga de telecomunicaciones.

Una Base de Datos, es una colección generalizada integrada de datos, que estructura según la relación natural de los mismos, de modo que proporcione todos los caminos de acceso a la unidad de información, satisfaciendo las diferentes necesidades de todos sus usuarios (Mackin, 2007).

De acuerdo a las definiciones anteriores una Base de datos, es aquella colección de datos almacenados de manera estructurada, que pueden ser registrados, analizados, clasificados, ordenados y consultados.

5.5.1 Niveles de abstracción de una base de datos.

Las estructuras de base de datos, hacen referencia a los tipos de datos, los vínculos o relaciones y las restricciones que deben de cumplir esos datos. Esta estructura es diseñada o

descrita empleando algún tipo de modelo de datos. En las bases de datos existen tres estructuras lógicas según (Hansen, 2002):

5.5.1.1 Estructura lógica de usuario.

Ya que este esquema, es la visión global de los datos según (Hansen, 2002), deberán encontrarse reflejados:

- Solo aquellos datos e interrelaciones que necesite el usuario.
- Especificaciones de las restricciones de uso (Puede ser el derecho a insertar o borrar determinados datos).

5.5.1.2 Estructura lógica global.

(Hansen, 2002), menciona que, en este esquema por ser la visión global de los datos, deberá incluirse:

- La descripción de todos los datos e interrelaciones entre estos.
- Las restricciones de integridad y confidencialidad.

5.5.1.3 Estructura física.

El contenido de este esquema, es muy independiente según (Hansen, 2002), de cada SGBD se debe distinguir: En la estructura del almacenamiento, ya que, es en la que se incluye la asignación de los espacios de almacenamiento, para el conjunto de datos, así como, las relaciones que existan entre los distintos espacios de almacenamiento.

5.5.2 Características de las bases de datos.

Para crear una base de datos, es necesario que diferentes elementos interactúen, dentro de estos encontramos los propuestos por (Raghu, 2007), (Tabla 3):

Tabla 3 Características de una base de datos propuestas por (Raghu, 2007).

Características de una base de datos	
Características	Definición
Interrelación	Permite tomar cualquier dato de información y cruzarlo con otra parte de la base de datos.
Evolución	Las características de las bases de datos pueden ser modificadas, ya sea en su estructura o agregando nuevas piezas de información, lo que da lugar a su actualización. Entre menos sea el plazo de actualización de la base de datos, mayor será su valor como fuente de información.
Accesibilidad	La información contenida en la base de datos, puede estar disponible para varios usuarios de acuerdo con sus necesidades. Si una base de datos ofrece varias opciones de acceso y precio, mayores serán las posibilidades de ser consultadas.
Mínima redundancia	La redundancia se refiere a la repetición de una misma pieza de información. Generalmente una pieza se almacena en la base de datos una sola vez.

5.5.3 Tipos de bases de datos.

Se puede dividir a las bases de datos de dos tipos según (Hansen, 2002), ver Figura 13:

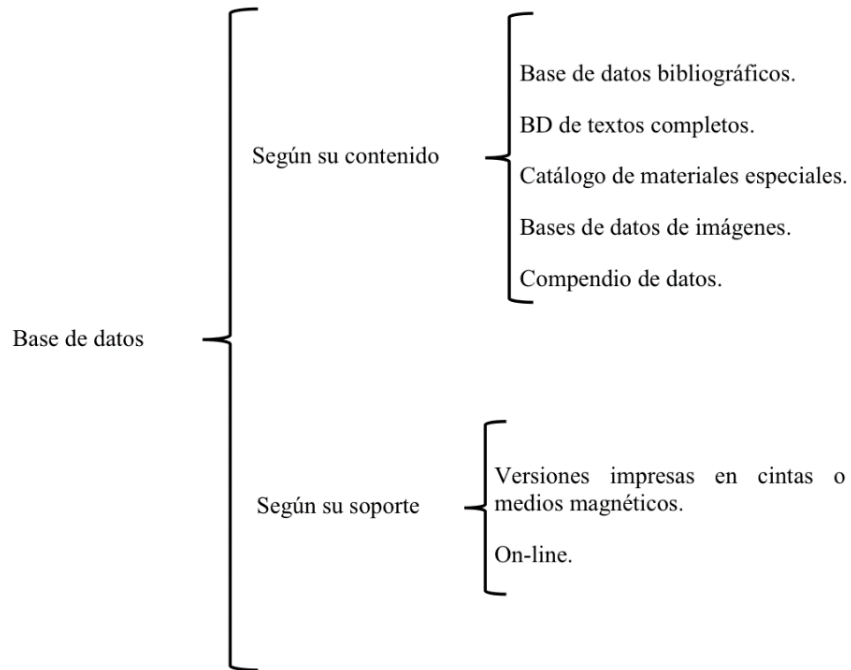


Figura 13 Tipos de bases de datos según (Hansen, 2002).

5.6 Modelado de base de datos

Un modelado de base de datos determina la estructura lógica de una base de datos y de manera fundamental determina el modo de almacenar, organizar y manipular los datos. Típicamente un modelo de datos permite describir, las estructuras de datos de la base de datos, las restricciones de integridad y las operaciones de manipulación de datos (Bertino, 2001).

De acuerdo a esto, se toma en cuenta el modelo relacional, en la cual cada relación consiste en un esquema y un ejemplar de relación, donde un ejemplar es una tabla, y un esquema describe las cabeceras de las columnas de esa tabla. En este esquema de relación, cada ejemplar, es un conjunto de tuplas también denominados como registros según (Raghu, 2007). Ver la Figura 14.

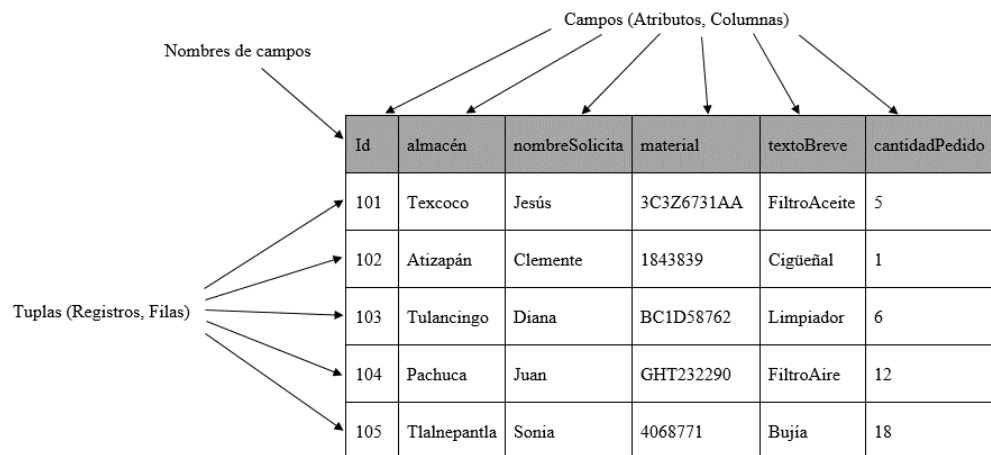


Figura 14 Partes del modelo relacional (Raghu, 2007).

En conclusión este modelo (Relacional), para el modelado y gestión de bases de datos, es un modelo de datos basado en lógica de predicados (Ramez, 2007). Algunas ventajas que se mencionan son las siguientes:

- Actúan sobre las tablas en su conjunto.
- Se pueden realizar consultas complejas que utilizan varias tablas de forma simple.
- Son fáciles de utilizar.

5.6.1 Características.

Para que la estructura de las tablas cumpla con las leyes de la teoría relacional, deben satisfacerse de las siguientes condiciones según (Spona, 2010):

- Todos los registros de las tablas deben tener el mismo número de campos, aunque alguno de ellos este vacío.
- Cada campo tiene un nombre o etiqueta que hay que definir previamente a su utilización.
- La base de datos estará formada por muchas tablas, una por cada tipo de registro.
- Dentro de cada tabla, el nombre del campo debe ser distinto entre ellos.
- Los registros de una tabla deben de diferenciarse entre sí, siendo que dentro de su contenido, no debe de haber dos registros idénticos.
- Los registros de una tabla pueden estar disponibles en cualquier orden.

- El contenido de cada campo, está delimitado por un rango de valores.
- Permite la creación de nuevas tablas a partir de las existencias, relacionando campos de distintas tablas anteriores. Esta condición es la esencia de la base de datos relacional, formando lo que se llama un fichero virtual (temporalmente en memoria).

Se mencionan algunas características de los cinco modelos en el Tabla 4:

Tabla 4 Características de los cinco modelos de bases de datos (Hansen, 2002).

Características de los cinco modelos de bases de datos		
Modelo	Organización de los elementos de datos	Identidad
Relacional	Los identificadores para filas de una tabla, se insertan como valores de atributo en otra.	Con base en valor.
Orientado a objetos	Los objetos relacionados de contención lógica, se encuentran dentro de un objeto determinado, por examen recursivo de atributos de un objeto que son los objetos mismos.	Con base en registro.
Deductivo	Reglas que permiten generar a petición, hechos relacionados.	Con base en valor.
Jerárquico	Proximidad lógica en un árbol estructurado por la linealización.	Con base en registro.
Red	Cadenas que se intersectan.	Con base en registro.

El diseño de las estructuras de datos y las relaciones entre ellas dentro de la base de datos, es una actividad crucial. Su diseño se realiza de una forma más sencilla cuando se utilizan modelos. Un modelo de base de datos son construcciones lógicas usadas para representar las estructuras de datos y las relaciones entre tablas dentro de la base de datos, (Cairó, 2010). Los modelos de base de datos pueden ser agrupados en dos categorías: modelo conceptual y modelo de implementación.

El modelo conceptual se enfoca en la naturaleza lógica de la representación de datos de información, es concernido con lo que está representado en la base de datos, en vez de como este se representa. Este modelo incluye el modelo entidad relación y el modelo orientado a objetos. Además, tres tipos de relaciones para describir la asociación entre los datos. Estos tres tipos de relaciones son uno a muchos, muchos a muchos y uno a uno.

Relación uno a muchos (Figura 15).

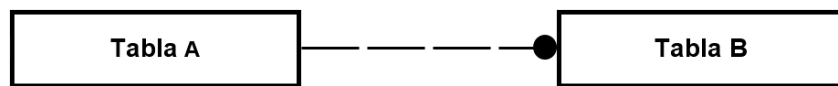


Figura 15 Relación uno a muchos (Elaboración propia).

Relación donde, cada elemento de la tabla A puede poseer muchos atributos en la tabla B.

Relación muchos a muchos (Figura 16).

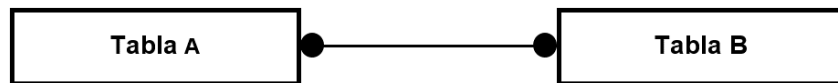


Figura 16 Relación muchos a muchos (Elaboración propia).

Cada elemento de la tabla A puede tener dos o más atributos en la tabla B.

Relación uno a uno (Figura 17).

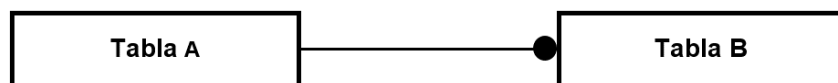


Figura 17 Relación uno a uno (Elaboración propia).

Cada atributo dentro de la tabla A esta relacionado con un solo atributo de la tabla B.

Por lo tanto, el modelo de implementación pone especial énfasis en como los datos son representados en la base de datos, o en cómo estas estructuras de datos son implementadas para representar lo que está modelado. Incluye los modelos de bases de datos jerárquicos (Figura 18), el modelo de base de datos de red (Figura 19), y el modelo de base de datos relacional (Bertino, 2001).

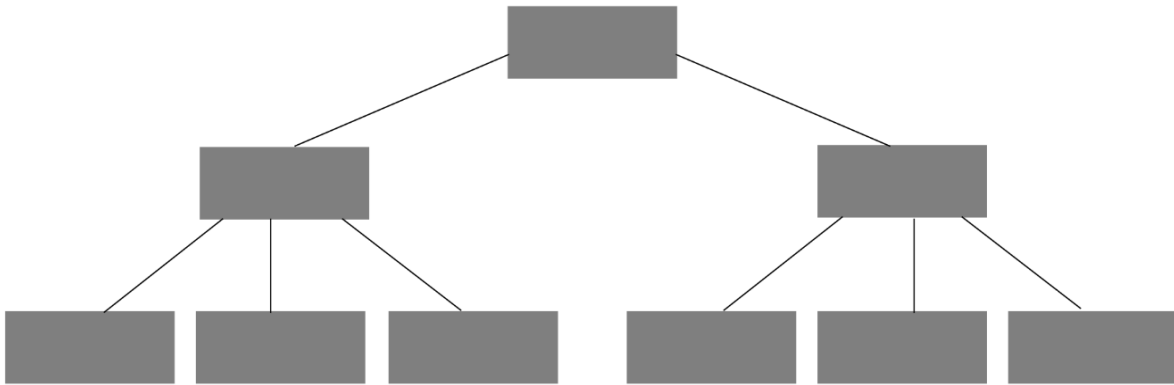


Figura 18 Modelo jerárquico (Bertino, 2001).

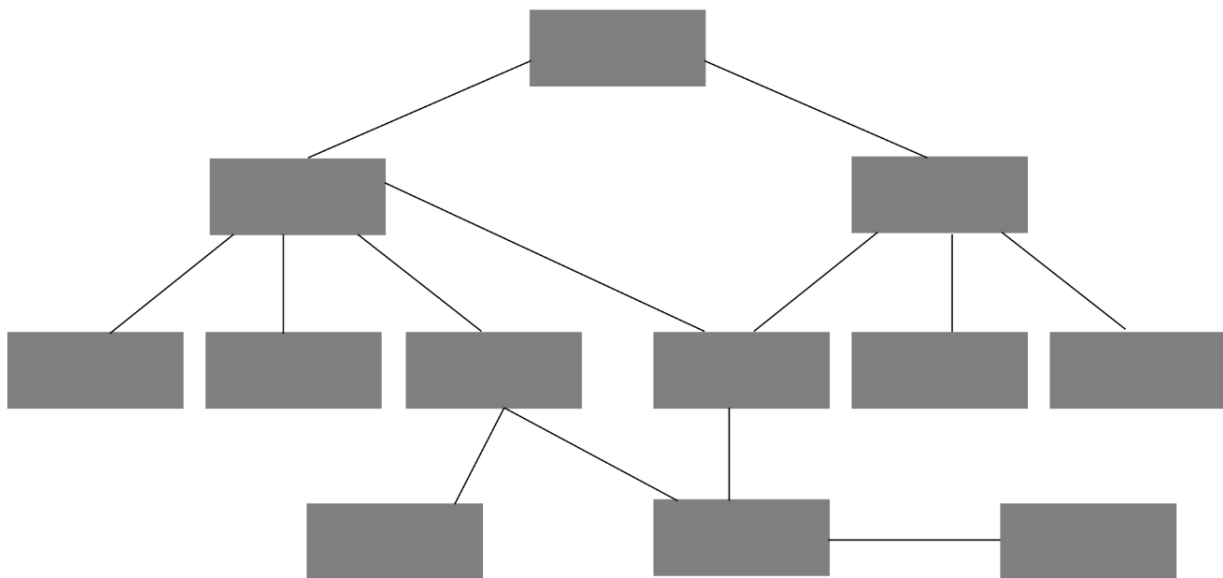


Figura 19 Modelo de red (Bertino, 2001).

5.6.2 Objetos de la base de datos relacional.

Una base de datos relacional, se integra por colecciones de objetos, donde la tabla constituye el objeto básico. Una tabla contiene información referente a una entidad, un renglón o registro, describe una instancia de esa entidad; una columna o campo describe un atributo de esta entidad. Las llaves primarias identifican a un renglón único dentro de la tabla. Las llaves foráneas relacionan los renglones de una tabla en otra.

Finalmente, los índices, son objetos que mejoran el proceso de localización de datos en una tabla. El índice relaciona cada valor de la columna indexada a la página o posición física sobre el renglón que contiene el dato, mejorando el desempeño. Ir al Tabla 5 (Ramez, 2007).

Tabla 5 Terminología estructural (Ramez, 2007).

Terminología estructural	
Termino relación formal	Equivalente informal
Relación	Tabla.
Tupla	Fila o registro.
Cardinalidad	Numero de filas.
Atributo	Columna o campo.
Grado	Numero de columnas.
Dominio	Conjunto de valores válidos.
Llave primaria	Identificador único.

5.6.3 Tipos de datos.

Para la asignación de valores de los tipos de datos, se debe tener conocimiento para poder dar un valor, ya sea un carácter o un valor numérico, como lo menciona (Wu, 2005). C# proporciona todos los tipos de datos que están disponibles en Java, agregan compatibilidad para los números sin signo y un nuevo tipo de punto flotante de 128 bits de alta precisión.

Se menciona que para cada tipo de dato primitivo en Java, la biblioteca de clases principal proporciona una clase contenedora, que lo representa como un objeto Java, ver Tabla 6. A continuación se muestran los tipos de datos mencionados por (Pérez, 2003) más comunes:

Tabla 6 Tipos de datos en Java (Pérez, 2003).

Tipos de datos		
Nombre de tipo de dato	Tipo	Tamaño / Formato (bits)
byte	Entero sin signo.	8
int	Entero con signo.	32
short	Entero con signo.	16
long	Entero con signo.	64
float	Tipo de flotante de precisión simple.	32
double	Tipo de flotante de precisión doble.	64
char	Un carácter Unicode.	16
boolean	Tipo boolean lógico.	True / False
object	Tipo base de datos los otros tipos.	
decimal	Tipo preciso fraccionario o integral, que puede representar números decimales.	128
date	Tipo fecha.	

En relación con el cuadro anterior se asignarán los tipos de datos a los atributos que se insertarán en cada tabla en el modelo relacional.

5.7 Sistemas gestores de bases de datos

Un sistema de gestión de bases de datos (SGBD), es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información de la base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. En el cual los usuarios pueden acceder a la información, usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto. También pueden incluir un módulo gráfico que permita presentar la información en forma gráfica o de tablas. Permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. (Rod, 2009).

5.7.1 Manejador de base de datos.

Un sistema manejador de bases de datos (SGBD, por sus siglas en inglés). Son una colección de software específico, cuya función es de servir de interfaz entre la base de datos, el usuario y distintas aplicaciones utilizadas, en la Figura 20 se muestra un diagrama que ejemplifica el funcionamiento de un SGBD (Bertino, 2001).

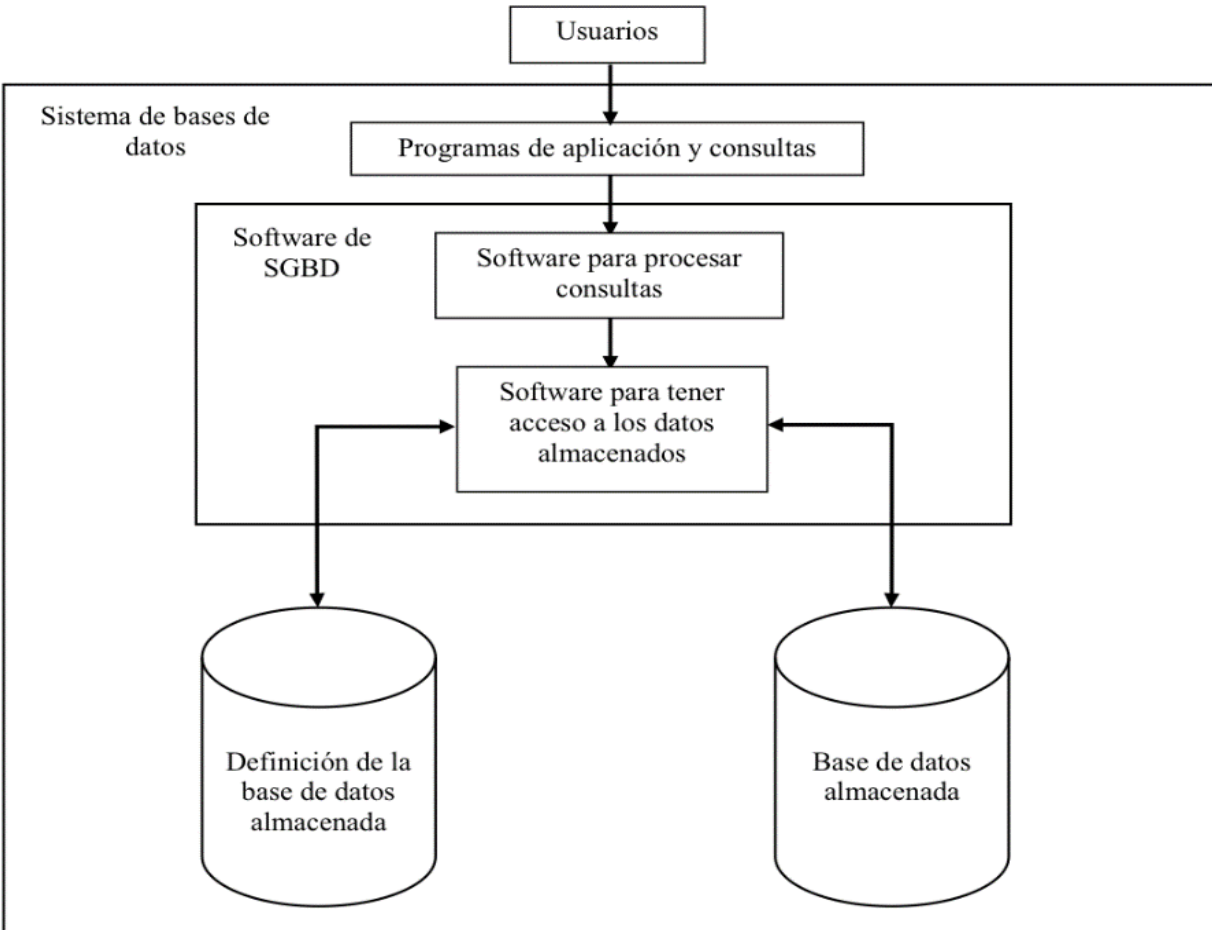


Figura 20 Funcionamiento del SGBD (Bertino, 2001).

El objetivo de los sistemas manejadores de bases de datos, es precisamente el manejar un conjunto de datos para convertirlos en información relevante para la organización, siendo nivel operativo o estratégico. Un SGBD, es sinónimo de independencia, una redundancia mínima, consistencia de la información, así como un acceso seguro y la adopción de las medidas necesarias para garantizar la integridad de los datos. Los procesos de un SGBD esenciales, son la manipulación y construcción de los datos, facilitando el cumplimiento de una serie de funciones, como una rápida gestión, representar relaciones complejas entre datos, aspectos relacionados entre la seguridad y la validación (Rod, 2009).

Un SGBD realiza las funciones de procesar la información, garantizando la integridad y consistencia de los datos dentro del sistema. Muchas de estas funciones son transparentes a los usuarios finales siendo realizadas solo a través del SGBD. Estas funciones descritas por (Kourchenko, 2001) incluyen:

- **Administración de diccionarios de datos:** El SGBD requiere que la definición de los elementos de datos y sus relaciones sea almacenada en un diccionario de datos. Además, utiliza el diccionario para buscar las estructuras requeridas, de igual forma, sus relaciones, facilitando las tareas de consulta y modificación.
- **Administración de datos almacenados:** El SGBD crea complejas estructuras requeridas para el almacenamiento de datos. También algunos integran reportes de la definición de las estructuras, reglas de validación y procedimientos.
- **Transformación y prestación de datos:** Transforma las solicitudes lógicas en comandos que encuentran la ubicación física de los datos, los recupera y muestra de manera coherente al usuario.
- **Administración de seguridad:** El SGBD crea un sistema que asegura la seguridad y control en la base de datos. Las reglas de seguridad determinan cuales usuarios pueden tener acceso, cual conjunto de datos puede consultar y que operaciones de estos puede realizar.
- **Control de acceso multiusuario:** Este utiliza algoritmos que aseguran que múltiples usuarios pueden ingresar a la base de datos, concurrentemente garantizando la integridad de estos.

- **Respaldo y recuperación:** Proporciona procedimientos para respaldo y recuperación de la información.
- **Integridad de datos:** Utiliza las reglas y relaciones almacenadas dentro del diccionario de datos para asegurar la consistencia de los datos, así se minimiza la redundancia de estos.
- **Interfaces de comunicación de las bases de datos:** Los actuales SGBD proporcionan características especiales de comunicación, diseñadas para permitir a la base de datos, aceptar transacciones dentro de una red de computadoras. Con estas funciones de acceso, es posible ingresar a la base de datos desde el internet.

5.7.2 Función de los sistemas de gestión de bases de datos.

La ejecución de operaciones sobre la base de datos, se proporcionan al usuario en función del requerimiento, se realiza de manera eficiente y segura. Sus características de un SGBD, posibilitan el cumplimiento de una serie de funciones mencionadas por (Rod, 2009), principalmente son:

5.7.2.1 Función de descripción.

Esta permite al administrador especificar los elementos de datos que la integran, por ejemplo, los tipos de datos, las relaciones que existen entre ellos, controles de acceso, las reglas de integridad semántica, entre otros (Rod, 2009).

5.7.2.2 Función de manipulación.

Esta función permite a los usuarios agregar, eliminar, modificar o buscar dentro de la base de datos, respetando las normas de seguridad establecidas por el administrador (Rod, 2009).

5.7.2.3 Función de utilización.

En este se reúnen todas las interfaces que necesitan los diferentes usuarios para comunicarse con la base de datos (Rod, 2009).

5.7.3 Esquema.

Nos describe la estructura de base de datos. Los tipos de campo, nombre y las relaciones, pertenecen a conjuntos de registros diferentes, es decir, como está realizada la base de datos, cual es la llave principal, nombre de cada campo, así como tipo y tamaño de cada mismo, permitido para cada uno de ellos el acceso e interacción con el sistema (Bertino, 2001).

5.7.4 Diccionario de datos.

Toda la información proporcionada en el esquema, se mantiene en el diccionario de datos, esta información incluye nombre de archivo, tipo de registro, campos y las relaciones entre los tipos de registro. Los diccionarios de datos, son conocidos como metadatos, es decir, datos acerca de los datos que sirven para que un usuario comprenda el diseño de la base de datos que realizó otra persona (Bertino, 2001). El lenguaje de manipulación de datos DML (*Data Manipulation Language*), es el software que permite al usuario a consultar la base de datos de los demás, es el encargado de permitir la colocación y ejecución de una consulta, también con este lenguaje el usuario manipula los datos.

5.8 Normalización

Es una serie de reglas que sirven para ayudar a los diseñadores de base de datos, a desarrollar un esquema que minimice los problemas de lógica. Se encarga de obtener los datos agrupados en distintas tablas siguiendo una serie de pasos, de tal manera que los datos obtenidos tengan una estructura óptima para su implementación, gestión y explotación desde distintas aplicaciones futuras. Una de las ventajas principales que se obtiene al realizar la normalización, es que la información no estará duplicada innecesariamente dentro de las estructuras, además de haber mínima redundancia (Rod, 2009).

5.8.1 Concepto de normalización.

Conjunto de reglas, que nos permiten asegurar que un esquema relacional cumpla con unas ciertas propiedades, cuando no se aplica la metodología de diseño, la teoría de normalización resulta imprescindible. También se entiende como proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y estables, son más fáciles de mantener. Entre los problemas que puede presentar un esquema relacional cuando el diseño es inadecuado cabe destacar según (Kcoenke, 2003):

- Incapacidad para almacenar ciertos hechos.
- Redundancias y, por tanto, posibilidad de inconsistencias.
- Ambigüedades.
- Pérdida de información.

- Pérdida de ciertas restricciones de integridad que dan lugar a interdependencias entre los datos.
- Aparición en la base de datos, de estados que no son válidos como: anomalías de inserción, borrado y modificación.

La teoría de normalización, se basa en restricciones definidas sobre los atributos de una relación, las cuales son conocidas con el nombre de dependencias. Las dependencias son propiedades inherentes al contenido semántico de los datos que se han de cumplir para cualquier extensión del esquema de relación y forman parte de las restricciones de usuario del modelo relacional. (Kcoenke, 2003).

5.8.2 Grados de normalización.

Existen básicamente tres niveles de normalización: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada uno de estos niveles tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada. No siempre es buena idea una base de datos conformada en el nivel más alto de normalización, ya que, puede llevar a un nivel de complejidad que pudiera ser evitado en un nivel más bajo. En la Tabla 7, se describe brevemente en que consiste cada una de las reglas según (Kcoenke, 2003).

Tabla 7 Niveles de normalización (Kcoenke, 2003).

Normalización	
Regla	Descripción
Primera Forma Normal (1FN)	Incluye la eliminación de todos los grupos repetidos.
Segunda Forma Normal (2FN)	Asegura que todas las columnas que no son llave, son dependientes de la llave primaria.
Segunda Forma Normal (3FN)	Elimina cualquier dependencia transitiva. Una dependencia transitiva, es aquella en la cual, las columnas que no son llave, son dependientes de otras columnas que tampoco son llave.

5.9 Datawarehouse

Un datawarehouse, es una colección de datos orientados a un determinado ámbito (Empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en que se utiliza. Se puede caracterizar un datawarehouse haciendo un contraste de como los datos de un negocio almacenados, difieren de los datos operacionales usados por las aplicaciones de producción. Un datawarehouse se caracteriza por integrar y depurar información de una o más fuentes distintas, para luego procesarla, permitiendo su análisis desde una infinidad de perspectivas y grandes velocidades de respuesta (Fon, 2008).

(Kcoenke, 2003), hace mención que los sistemas de apoyo para la toma de decisiones, son sistemas que ayudan en el análisis de información de negocios. El propósito principal de estos sistemas, es ayudar a la administración para que marque tendencias, señale problemas y tome decisiones inteligentes. Este autor define que la base de datos, es principalmente de solo lectura. La actualización de estos datos, es limitada a operaciones de carga o actualizaciones periódicas (Estas operaciones están dominadas, a su vez, por INSERTs – los DELETEDs se realizan muy ocasionalmente y los UPDATEs casi nunca).

(Velthuis, Calidad de sistemas de información, 2012), define datawarehouse como: Un elemento funcional de un sistema de información de apoyo para la toma de decisiones. Por consiguiente, un almacén de datos orientados a un tema, integrado no volátil y variable en el tiempo que soporta decisiones de administración (Figura 21).

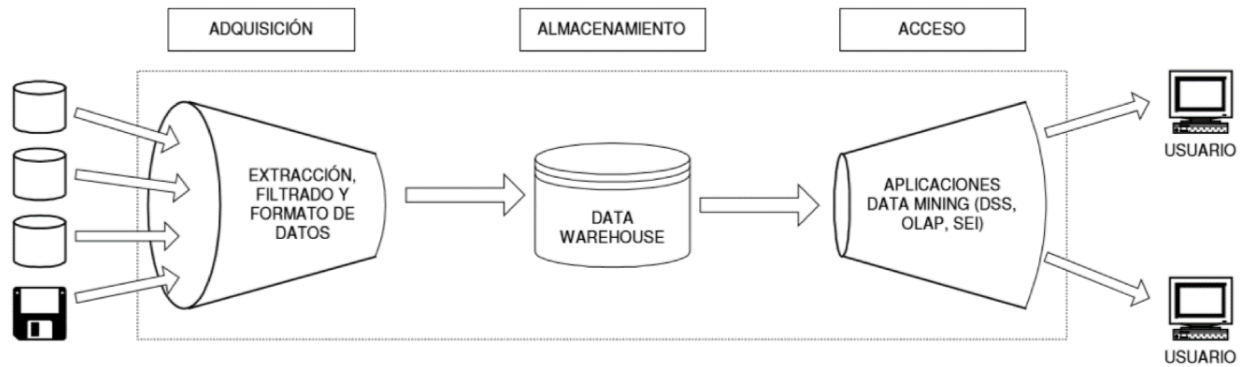


Figura 21 Mecanismos datawarehouse con datos de producción (Fon, 2008).

5.9.1 Tipos.

Últimamente se manejan los appliances data warehouse, tanto en los movimientos de Ingres, HP, como en la inclusión de las tendencias en *Business Intelligence* que han sido identificados. El autor (Richard, 2007), menciona distintos tipos de data warehouse que se identifican de la siguiente manera:

- *Native data warehouse appliance:*

Donde tanto el hardware como el software están estrechamente integrados en una sola plataforma. No se pueden licenciar por separado y tampoco se puede utilizar de manera individualizada.

- *Software data warehouse appliance:*

En este caso, bases de datos relacionales, tanto de Open Source como comerciales son optimizadas para su uso de entornos DataWareHouse. Aquí, se puede utilizar en diferentes configuraciones de hardware.

- *Packaged data warehouse appliance:*

Donde el software y hardware comercial son configurados para funcionar como una única plataforma y es comercializado por un único vendedor. Además se instala y mantiene en un único sistema.

- *Data management appliance:*

Obtiene los datos de forma intensiva desde un servidor. Estos sistemas pueden implicar procesos operacionales, analíticos o de almacenamiento.

5.9.2 Procesos.

(Harold, 2010) Menciona que los procesos que se realizan en data warehouse son la extracción, transformación y carga, a continuación se definirán cada uno:

- **Extracción:**

Es el primer paso para la obtención de la información hacia el ambiente datawarehouse.

- **Transformación:**

Una vez que la información es extraída hacia el área de tráfico de datos, hay posibles pasos de transformación como, limpieza de información, eliminar los datos que no sirven, seleccionar los campos que son necesarios, combinar fuentes de datos, haciéndolas coincidir por los valores de las llaves, creando nuevas llaves para cada registro de una dimensión.

- **Carga:**

Al final del proceso de transformación, los datos son cargados.

5.9.3 Características.

Según (Fon, 2008), un datawarehouse se caracteriza por ser integrado, temático, histórico y no volátil, a continuación se describe cada uno de ellos:

- Integrado: Los datos almacenados en el datawarehouse deben integrarse en una estructura consistente. La información suele estructurarse también en distintos niveles, para adecuarse a las necesidades del usuario.
- Temático: Solo los datos necesarios en el proceso de generación de conocimiento del negocio, se integran desde el entorno operacional. Los datos se organizan para facilitar su acceso y entendimiento por los usuarios finales.
- Histórico: El tiempo es parte implícita de la información contenida en un datawarehouse. En los sistemas, los datos siempre reflejan la actividad del negocio, encargándose de los valores que toma una variable en el tiempo, permitiendo comparaciones.
- No volátil: La información existe para ser leída, pero no modificada.

Es importante conocer cuál es el proceso de construcción de un datawarehouse, denominado ETC (Extracción, Transformación y Carga) a partir de los sistemas operacionales de una compañía (Fon, 2008):

- Extracción: Obtención de información de distintas fuentes internas como externas.
- Transformación: Filtrado, limpieza, depuración y agrupación de la información.
- Carga: Organización y actualización de los datos.

5.9.4 Aportaciones de un datawarehouse

Proporciona una herramienta para la toma de decisiones en cualquier área funcional, basándose en información integrada y global del negocio según (Harold, 2010):

- Facilita la aplicación de técnicas de análisis y modelización para encontrar relaciones ocultas entre los datos de almacén, obteniendo un valor de dicha información necesitada.
- Proporciona la capacidad de aprender de los datos del pasado y predecir situaciones futuras dentro del área funcional.
- Supone una optimización tecnológica y económica en entornos de centro de información, estadística o generación de informes con retornos de la inversión.

5.10 Java

Java es un lenguaje de programación que recibe mucha atención por todo el mundo industrial y académico. Fue desarrollado por James Gosling con su equipo de trabajo en Sun Microsystems en California. El lenguaje se basa en C y C++. El propósito original era escribir programas que controlaran los electrodomésticos de consumo como: tostadores, hornos de microondas entre otros (Sánchez, 2009).

Al inicio este lenguaje de programación se llamó Oak. Una aplicación de Java, es un programa completo y autónomo que no requiere de un navegador web, es parecido a un programa escrito en otros lenguajes de programación. (Wu, 2005).

Sun Microsystems, es el líder en servidores para internet, quien invento el lenguaje Java, en un intento de resolver simultáneamente todos los problemas que se planteaban a los desarrolladores de software, por la proliferación de arquitecturas incompatibles, tanto entre las diferentes máquinas como entre los diversos sistemas operativos y sistemas de ventanas que funcionaban sobre una misma máquina, añadiendo la dificultad de crear aplicaciones distribuidas en una red de internet, (Froufe, 2006). El lenguaje Java y los navegadores con soporte Java, proporcionan una forma diferente de hacer que ese navegador sea capaz de ejecutar programas, con Java se puede reproducir sonido directamente desde el navegador.

5.10.1 Objetivos.

Los objetivos principales por lo cual se creó el lenguaje Java (Dean, 2009) son:

- Usar el paradigma de la programación orientada a objetos.
- Permitir la ejecución de un mismo programa en múltiples sistemas operativos.

- Incluir por defecto soporte para trabajo en red.
- Diseñarse para ejecutar código en sistemas remotos de manera segura.
- Ser fácil de usar y tomar lo mejor de otros lenguajes de programación, como C++.

5.10.2 Características.

Las características principales que ofrece Java, respecto a cualquier otro lenguaje, teniendo presentes los lenguajes C y C++, se resumen en las siguientes según (Ceballos, 2007):

- Simple

Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de estos. C++ no es un lenguaje conveniente por razones de seguridad, pero C y C++ eran los lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ facilitando un rápido y fácil aprendizaje.

Java elimina muchas características de los otros lenguajes, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles. Además, reduce el 50% de los errores más comunes de programación.

- Orientado a objetos

Java implementa tecnología básica de C++ con algunas mejoras y elimina algunas cosas, para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de los objetos, son

llamadas clases y sus copias instancias. Estas instancias necesitan ser construidas y destruidas en espacios de memoria.

- Distribuido

Java se ha construido con extensas capacidades de interconexión TCP/IP. Ya que existen librerías de rutinas para acceder e interactuar con protocolos, como http y ftp. Java en si no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se ejecuten en varias máquinas.

- Robusto

Realiza verificaciones en busca de problemas, tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos de datos en Java ayuda a detectar errores lo antes posible en el ciclo de desarrollo. Java obliga la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador, de la liberación o corrupción de memoria.

- Portable

Java implementa otros estándares de portabilidad, para facilitar el desarrollo. Además, Java construye sus interfaces de usuario, a través de un sistema abstracto de ventanas, de forma que estas pueden ser implantadas en otros entornos.

- Interpretado

El intérprete Java (sistema *run-time*), puede ejecutar directamente el código objeto. Enlazar (*linkar*) un programa, normalmente consume menos recursos que compilarlo, el compilador es

JDK. Aunque Java sigue siendo un lenguaje interpretado, la situación se acerca mucho a la de los programas compilados, sobre todo en la rapidez de ejecución del código.

- Multitarea

Java permite realizar muchas actividades simultáneas en un programa. Las tareas a veces llamadas, procesos ligeros o hilos de ejecución, son básicamente partes independientes de un gran proceso. Al estar las tareas construidas en el mismo lenguaje, son más fáciles de usar de igual manera siendo más robustas.

El beneficio de ser multitarea consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real, aunque es limitado a las capacidades del sistema operativo subyacente (Unix, Windows etc.) de la plataforma, aun supera a los entornos de flujo único de programa, tanto en facilidad de desarrollo como en rendimiento.

- Dinámico

Java se beneficia de todo lo posible de la tecnología orientada a objetos, no conectando todos los módulos que comprenden una aplicación hasta el mismo tiempo de ejecución. Las librerías nuevas o actualizadas no paralizarán la ejecución de las aplicaciones actuales, siempre que estén manteniendo lo anterior.

5.11 Patrón MVC (Modelo - Vista - Controlador)

El patrón de arquitectura MVC, es un patrón que define la organización independiente del modelo (objetos de negocio), la vista (interfaz con el usuario u otro sistema) y el controlador (controlador de *workflow* de la aplicación). De manera genérica, los componentes de MVC se definen a continuación (Weaver, s.f.):

- Modelo

Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Envía la vista de aquella parte de la información, que a cada momento se solicita, para que sea mostrada (típicamente a un usuario). Las peticiones de acceso por manipulación de información llegan al modelo a través del controlador.

- Vista

Presenta el modelo en un formato adecuado para interactuar, por tanto, requiere de dicho modelo, la información que debe representar como salida.

- Controlador

Responde a eventos (acciones de usuarios) e invoca peticiones al modelo, cuando se hace una solicitud sobre la información. También puede enviar comandos a su vista, solicitando un cambio en la forma en que se presenta el modelo, por tanto, se podría decir que el controlador hace el intermediario entre la vista y el modelo.

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos (SGBD), para gestionar los datos que debe utilizar la aplicación, dicha gestión corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido como el paradigma de la programación, representa la integración entre la vista correspondiente, controlador de eventos y acceso a datos (Bernal & director de tesis, 2011).

El MVC no pretende discriminar entre capa de negocio y capa de presentación, pero si separar la capa visual grafica de su correspondiente programación y acceso a datos, mejorando el desarrollo en cuanto al mantenimiento de la vista, de igual manera el controlador en paralelo, ya que ambos cumplen con el ciclo de vida muy distintos entre sí (Bernal & director de tesis, 2011).

5.11.1 Interacción con un MVC.

La interacción con un Modelo Vista Controlador, es mencionada de la siguiente manera, siendo mencionados por (Bahit, 2011):

- Modelo:
 - Contiene el núcleo de la funcionalidad (dominio de la aplicación).
 - Encapsula el estado de la aplicación.
 - No sabe nada independiente del controlador y la vista.

- Vista:
 - Es la representación del modelo.
 - Puede acceder al modelo, pero nunca cambiar su estado.

➤ Puede ser notificada cuando hay un cambio de estado en el modelo.

- Controlador:

➤ Reacciona a la petición del cliente, ejecutando la acción adecuada y creando el modelo pertinente.

5.12 MySQL

MySQL es un sistema gestor de bases de datos de forma cliente-servidor, de coste mínimo que incluye un servidor SQL, programa al cliente para acceder al servidor; herramientas administrativas y una interfaz de programación. MySQL es portable ya que es ejecutable en sistemas operativos comerciales como Linux y Windows.

Determinadas sentencias de este SGBD pueden ser compatibles con PHP y HTML para diseñar aplicaciones web dinámicas, que incorporan la información de las tablas a páginas web. (Pérez, 2003).

Nuestro sistema será complementado con MySQL, para la elaboración de las tablas y obteniendo así un modelo de base de datos relacional. Así llegando a la elaboración del sistema.

5.12.1 Características.

Anteriormente MySQL, carecía de elementos considerados esenciales, tales como: integridad, referencial y transacciones. Entre las últimas versiones destacan:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones.
- Transacciones y claves foráneas.

- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración relacional de base de datos, el cual archiva datos en tablas. Esta puede ser desde una simple lista de compras o un volumen de información saturada. Dado que los computadores son buenos manejando grandes cantidades de información, los administradores de base de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones. Esto permite velocidad y flexibilidad de la información. (Raghu, 2007).

5.13 SQL

SQL, es el lenguaje estándar para trabajar con base de datos relacionales, siendo soportado prácticamente por todos los productos en el mercado. Originalmente, SQL fue desarrollado en IBM. El lenguaje de consulta estructurada (SQL), es un lenguaje en la industria soportable por la mayoría de los sistemas gestores de bases de datos, se puede usar para definir tablas, relaciones entre ellas, restricciones, autorización de privilegios entre otros (Post, 2008).

En una base de datos todo acaba en SQL. Independientemente si una consulta, se hace por medio de un programa cliente o por medio de una interfaz gráfica, en el fondo, fue un comando SQL el responsable del resultado. Conocer este lenguaje es esencial para quien trabaje con una base de datos (Post, 2008). Debido a la diversidad de lenguajes y de base de datos existentes, la manera de comunicar sería complicada, a no ser por la existencia de estándares que nos permiten realizar las operaciones básicas de una forma universal, es por eso que SQL (*Structured Query Language*), no es más que un lenguaje estándar de comunicación con bases de datos. SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad de los sistemas relacionales, permitiendo gran variedad de operaciones.

El lenguaje SQL, es considerado de cuarta generación, es decir, que indica que información desea obtener o procesar, pero no como se deberá hacer. Es labor del sistema elegir la forma más eficiente para llevar a cabo la operación ordenada por el usuario (Pérez, 2003).

Aunque SQL es considerado como un sublenguaje por naturaleza, no procesa datos; no obstante, es un lenguaje completo que permite crear, borrar, visualizar, obtener y manipular objetos en BD. Un método común utilizado para clasificar las sentencias SQL, es dividirlos de acuerdo a las operaciones que realizan. Con base en este método, SQL se puede separar en los

siguientes tipos de declaraciones: Lenguaje de definición de datos, lenguaje de manipulación de datos y lenguaje de control de datos. El lenguaje según (Pérez, 2003) es subdividido en dos, el DDL y el DML:

5.13.1 Lenguaje de definición de datos (DDL).

El DDL es el que se encarga de la modificación de la estructura de los objetos de la base de datos, incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos, existiendo las siguientes operaciones (Pérez, 2003):

- ***Create* – Crear**

Este comando permite crear objetos de datos, como nuevas bases de datos, tablas, vistas y procedimientos almacenados.

- ***Alter* – Modificar**

Este comando permite modificar la estructura de una tabla u objeto. Se pueden agregar o quitar campos de una tabla, modificar el tipo de campo, agregar o quitar índices de una tabla, modificar un trigger, entre otros.

- ***Drop* – Borrar**

Este comando elimina un objeto de la base de datos, puede ser un, índice, *trigger*, función, procedimiento, o cualquier objeto que el motor de la base de datos soporte, se puede combinar con la sentencia *ALTER*.

- ***Truncate* – Borrar tabla**

Este comando trunca todo el contenido de una tabla, las ventajas sobre el comando *DROP* (Borrar), es que el contenido de la tabla, es eliminado de manera más rápida, especialmente si la tabla es muy grande. La desventaja es que solo sirve cuando se quieren borrar absolutamente todos los registros.

5.13.2 Lenguaje de manipulación de datos (DML).

Es un lenguaje proporcionado por el sistema de gestión de base de datos, que permite a los usuarios llevar a cabo tareas de consulta o manipulación, organizados por el modelo de datos adecuado. El lenguaje de manipulación de datos, es el SQL, usado para recuperar y manipular datos en una base de datos relacional (Pérez, 2003). Las operaciones utilizadas son:

- ***Select* – Seleccionar**

Nos permite realizar consultas sobre los datos ya almacenados en una base de datos realizando alguna búsqueda que se desea hacer.

- ***Insert* – Insertar**

Una sentencia *INSERT* de SQL, agrega uno o más registros a una tabla en una base de datos relacional.

- ***Update* – Modificar**

Esta sentencia, es utilizada para modificar los valores de un conjunto de registros ya existentes en una tabla.

- **Delete – Borrar**

Borra uno o más registros existentes en una tabla dentro de la base de datos que se encuentra dentro de algún área empresarial.

5.14 Marco referencial

5.14.1 Empresas y software referenciales

Dentro del tema relacionado con el desarrollo de software, las bases de datos han tenido mucha utilidad dentro de áreas empresariales, en las cuales la información, es indispensable para realizar operaciones, siendo las bases de datos utilizadas para la automatización de oficinas, por lo cual facilitara al personal, llegar a la toma de decisiones. Estos sistemas son utilizados en diferentes tipos de empresas tanto administrativas como de servicio (Weitzenfeld & Guardati, 2008).

5.14.2 EASY-WMS (Sistema de fácil gestión de inventario)

Es el proveedor de servicio de inventario más importante del mundo, WMS (*Warehouse Management System*) lleva más allá del conteo con servicios de inventarios que incluyen: conteos físicos, totales, cíclicos, acomodo, preparación de inventario y emplacado de activo fijo, inventario de códigos, etiquetado, generación de catálogos de activos y productos, auditorías operativas, smart space, carga garantizada (auditoría de origen y destino de mercancía). Con éxitos probados en todo tipo de formatos de inventario (venta minorista, mayorista, salud, manufactura, etc.), no es de extrañar que cada vez más organizaciones, en más lugares, confíen sus inventarios a EASY-WMS (Astasio, 2005).

EASY-WMS tiene una fuerte presencia internacional, incluyendo más de 400 oficinas y 25.000 colaboradores, en México contamos con 20 oficinas, con más de 1,500 colaboradores y clientes que van desde empresas independientes, hasta las grandes cadenas de distribución en el mundo, abarcando todos los sectores de la industria. Esta empresa puede hacer comparaciones de todo lo relacionado con el almacén (Astasio, 2005).

5.14.3 Inventoria

Es un software especializado en datawarehouse, el cual tiene la facilidad de sacar datos de una base de datos, sin tener que hacer una mejora de la misma, sino que solo manda el stock de los centros de almacén, el cual es una herramienta para la toma de decisiones dentro del área de compras de refacciones (Village, 2011). Algunas ventajas que puede ofrecer Inventoria dentro de un almacén son las siguientes:

- Da una mejora en cuanto a la utilización de espacio.
- Reducción de elementos de manutención.
- Mejora y disminuye la productividad en la preparación de pedidos.
- Reducción en devoluciones de los clientes.
- Disminución de operaciones de manipulación.
- Reducción de costos de stock.

VI. Métodos y técnicas de investigación empleadas

El sistema se desarrolló bajo los siguientes pasos metodológicos:

Investigación preliminar: Se definen cómo son realizados los procesos antes de la implementación del sistema, extrayendo los datos necesarios, obteniendo y depurando los datos para desarrollar y programar el sistema.

Análisis de requerimientos: Son obtenidos mediante entrevistas realizadas a encargados del área de compras, de observar los procesos que se realizan y se obtienen los registros que se generan.

Modelado y diseño del SI con UML (Casos de uso): Se construyen los casos de uso de las operaciones que va a realizar el usuario, para definir lo que va a contener el sistema, a fin de modelar, documentar y especificar lo que va a hacer el sistema de software.

Desarrollo de la base de datos: Se definen los tipos de datos para construir el diccionario de datos y después de modelar la base de datos a partir del modelo relacional y obtener el modelo lógico.

Desarrollo del modelo relacional: Utilizando MySQL Workbench 6.2 CE, se realizó la implementación física de la base de datos.

Planteamiento de algoritmos: A partir de los diagramas UML, se plantean los algoritmos para el funcionamiento del sistema.

Implementación de algoritmos: Se implementan los algoritmos con el lenguaje de programación Java usando con el editor netBeans IDE 8.0.1, utilizando el patrón MVC.

Pruebas de validación y verificación en el sistema: Una vez terminado el prototipo, se comprueba que la implementación sea correcta, descubriendo errores, evaluándose el producto desarrollado por el usuario, verificando que cumpla con las necesidades que originaron su desarrollo. Para ello se utiliza el modelo en cascada.

Implementación del sistema: Implica la puesta en marcha del sistema dentro del área de compras de refacciones, así como, la capacitación de usuarios.

6.1 Desarrolló de la metodología

6.1.1 Investigación preliminar.

En el área de compra de refacciones de Ford, son aplicadas formulas en Excel de forma manual, para realizar la búsqueda de piezas de cada una de las sucursales. Dentro del departamento, se trabajan las piezas de pedido diario de cada una de las siguientes sucursales:

- Atizapán.
- Tlalnepantla (Planta).
- Pachuca.
- Tula.
- Tulancingo.
- Texcoco.

En el almacén de cada sucursal, el encargado realiza la observación de su inventario, para ver si alguna pieza que se solicita, se encuentra dentro o en alguna de las otras sucursales. Terminando este proceso se manda un documento en Excel por correo (Figura 22), de las piezas a solicitar, en este caso es Tlalnepantla, quien recibe toda la documentación de cada sucursal.

1	Solicitud de Compra de Refacciones						
2	Almacen	Nombre	Material	Texto Breve	Nota de caja	Observaciones	Ctd. Pedido
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							

Figura 22 Formato de solicitud de almacenista (Elaboración propia).

En planta se realiza una recolección de datos, de los almacenes sobre las piezas que se van a solicitar. Para ello solo cuentan con un tiempo en el cual, si no es mandado el pedido a Ford no se surtirá la pieza, teniendo en cuenta que, en la página web de Ford, ya no aceptan pedidos después de las 3:00 pm., el encargado realiza este proceso bajo presión, ya que cada día se tienen que surtir las piezas en las sucursales mencionadas anteriormente, de acuerdo a esto, en el área de compras de refacciones, no existe un sistema el cual realice la búsqueda.

El encargado de realizar el proceso de búsqueda, copiando y revisando pieza por pieza en el documento de Excel (Figura 23), aplica las fórmulas; independientemente de que pueda existir un error, podría hacer un duplicado de la pieza o no pedir la pieza que está siendo solicitada.

	A	B	C	D	E	F	G	H	K	L	M	N
1	FECHA	ALMACEN	ORDEN/VENDEDOR	NO. DE PARTE	DESCRIPCION	FECHA ORDEN		Ctd.dif.				
2												
3	06.01.2015	R121	7628562	1769065	PARTE NO RECONOCIDA	TLFMFLOR	7628562	1	#N/A	#N/A	#N/A	#N/A
4	07.01.2015	R120	7636575	M554510849PA	CUADRO DE INSTRUMENTOS	TLFJCAST	7636575	1	#N/A	#N/A	#N/A	#N/A
5	14.01.2015	R121	7635543	1790945	COSTADO	TLFISANM	7635543	1	#N/A	#N/A	#N/A	#N/A
6	14.01.2015	R121	7635543	1807687	FASCIA TRASERA LATERAL	TLFISANM	7635543	1	#N/A	#N/A	#N/A	#N/A
7	19.01.2015	R120	LUIS G.	MF2721061726A	VARILLA DE MANDO	LUIS G.	LUIS G.	1	#N/A	#N/A	#N/A	#N/A
8	26.01.2015	R128	ACCESORIOS	DS7254133001A	TAPETES DE FUSION	ACCESORIOS	STOCK	2	#N/A	#N/A	#N/A	#N/A
9	03.02.2015	R120	gregorio rodriguez	M7N1518045AC	REGISTRAR B.O. N. DE SERIE 9BFUT35FX88	gregorio rodriguez		2	#N/A	#N/A	#N/A	#N/A
10	06.02.2015	r120	Carlos Olguin	M6L3218124AH	DEJAR EN BO VIN 1FTRX12W58KE14160	Carlos Olguin	RAFAELO	2	#N/A	#N/A	#N/A	#N/A
11	13.02.2015	R121	7645080	1842117	PARTE NO RECONOCIDA	JESUS	7645080	1	#N/A	#N/A	#N/A	#N/A
12	13.02.2015	R121	7645080	1771938	PARTE NO RECONOCIDA	JESUS	7645080	1	#N/A	#N/A	#N/A	#N/A
13	13.02.2015	R121	7645080	1770453	PARTE NO RECONOCIDA	JESUS	7645080	1	#N/A	#N/A	#N/A	#N/A
14	13.02.2015	R121	7645080	1790962	PARTE NO RECONOCIDA	JESUS	7645080	1	#N/A	#N/A	#N/A	#N/A
15	14.02.2015	R120	LUIS	1371391	DEJAR EN BACKORDER	LUIS	WFORS4G50BJA8543	6	#N/A	#N/A	#N/A	#N/A
16	14.02.2015	R120	ALFREDO	MBL322200C	BALATAS	ALFREDO		4	#N/A	#N/A	#N/A	#N/A
17	14.02.2015	R120	gregorio rodriguez	AA527G391A	GENERAR B.O. N. DE SERIE1LNHM93R99G6	gregorio rodriguez		1	#N/A	#N/A	#N/A	#N/A
18	18.02.2015	R120	STOCK	CV622001A	BALATAS	STOCK	STOCK	12	#N/A	#N/A	#N/A	#N/A
19	20.02.2015	R121	7648869	F7529927840DA	PANEL EXTERIOR - ALETA TRASERA	TLFJCABA	7648869	1	#N/A	#N/A	#N/A	#N/A
20	20.02.2015	R120	LUIS	1795006	FEDEX TIJUANA FRANCISCO APOLO	LUIS	WFORT47H78IA0536	1	#N/A	#N/A	#N/A	#N/A
21	20.02.2015	R120	LUIS	1795006	CONCEPCION	LUIS	WFORT47H78IA0620	1	#N/A	#N/A	#N/A	#N/A
22	20.02.2015	R120	LUIS	1795006	FEDEX TIJUANA FRANCISCO APOLO	LUIS	WFORT47H68IA8435	1	#N/A	#N/A	#N/A	#N/A
23	21.02.2015	R120	PGARCIA	2L7216490AAA	ALFOMBRA ESTRI	PGARCIA	5LMFU28R83L02008	1	#N/A	#N/A	#N/A	#N/A
24	23.02.2015	R120	LUIS	1732801	DEJAR EN BO	LUIS	WFORS4G59BJA8380	1	#N/A	#N/A	#N/A	#N/A
25	24.02.2015	R121	7650628	MCN15N40010AA	PUERTA TRASERA	TLFJCABA	7650628	1	#N/A	#N/A	#N/A	#N/A
26	24.02.2015	R121	7639994	1769764	ESPEJO RETROVISOR	TLFMFLOR	7639994	1	#N/A	#N/A	#N/A	#N/A
27	24.02.2015	R120	LUIS	1732801	DEJAR EN BO	LUIS	WFORS4G53BJA8344	1	#N/A	#N/A	#N/A	#N/A
28	25.02.2015	r120	luis	1732801	DEJAR EN BO CONCEPCION tepic	luis	WFORS4G50AJA7620	1	#N/A	#N/A	#N/A	#N/A
29	25.02.2015	R121	7650503	1797477	CAPO SIN BISAGRAS	TLFMFLOR	7650503	1	#N/A	#N/A	#N/A	#N/A

Figura 23 Formato de búsqueda de piezas (Elaboración propia).

6.1.2 Análisis de los requerimientos.

Lo que el sistema debe realizar, se llaman también requerimientos funcionales, el análisis de los requerimientos, es la actividad más importante dentro del desarrollo de sistemas, ya que la calidad del trabajo realizado en esta etapa, se reflejara más adelante. Existen diversas técnicas de recolección de requerimientos, para este proyecto se utilizó lo más conveniente.

Las entrevistas fueron aplicadas al usuario que utilizara el sistema propuesto, en este caso, al personal del área de compra de refacciones (Para más información de la entrevista ver Anexo I y II). Los siguientes comentarios se derivan de las entrevistas realizadas.

Obteniendo la necesidad para el desarrollo de este sistema, los usuarios que realizan la recolección de solicitudes de cada almacén, están insatisfechos con el tiempo, ya que se debe mandar el pedido antes de las 3:00 p.m., siendo que es el tiempo estimado para el acceso a la página web de Ford.

Además de que la búsqueda es realizada manualmente en Excel, son aplicadas fórmulas para buscar las piezas solicitadas; obteniendo como resultado el material a solicitar, en consecuencia, al no ingresar los datos de manera correcta, ocasionaría un error, al mismo tiempo no realizar el pedido en tiempo y forma.

El sistema SICSARF, permite la recolección de la información sobre los stocks del sistema en SAP, este no tiene acceso directo con la BD debido a que la licencia de SAP, con la empresa Ford no permite conexiones que no sean de sus propios módulos del SAP. Por lo cual SICSARF obtiene esta información desde archivos de Excel que proporcionan un estado actual de la base de datos, estos registros son integrados y depurados para tener información histórica sobre las

refacciones dentro de la empresa Ford y en un futuro permitir nuevas consultas que generen información importante para la toma de decisiones en esta empresa.

6.1.2.1 Revisión de documento

El encargado del área, mostro los datos con los cuales trabajan para la elaboración de la solicitud de piezas diariamente, señalando que el proceso debe realizarse en un menor tiempo, para realizar el pedido a Ford antes de las 3:00 p.m.

6.1.3 Modelado y diseño del SI con UML (Casos de uso).

El modelo de desarrollo usado es el de cascada, con los pasos siguientes:

- Análisis de requisitos.
- Diseño del sistema.
- Diseño del programa.
- Codificación o programación.
- Pruebas.
- Verificación.
- Mantenimiento.

El modelo en cascada se ajustó al desarrollo del sistema, sus características se tomaron en cuenta, ya que sus requisitos fueron fijos desde un inicio. Para la obtención del sistema de base de datos, se seleccionó el modelo de caso de uso, obteniendo una visualización si el sistema cumplirá con los requisitos necesarios, siendo una técnica fácil de visualizar la manera en que la base de datos interactuara con el usuario, es de forma accesible para comprender a nivel de usuarios y permite graficar la determinación de requisitos a ser cubiertos. Ya que describe lo que hace el sistema pero sin describir como lo hace, como se observa en la Figura 24. Y observándose de igual forma el Tabla 8, el cual nos muestra la descripción de cada uno de los casos de uso.

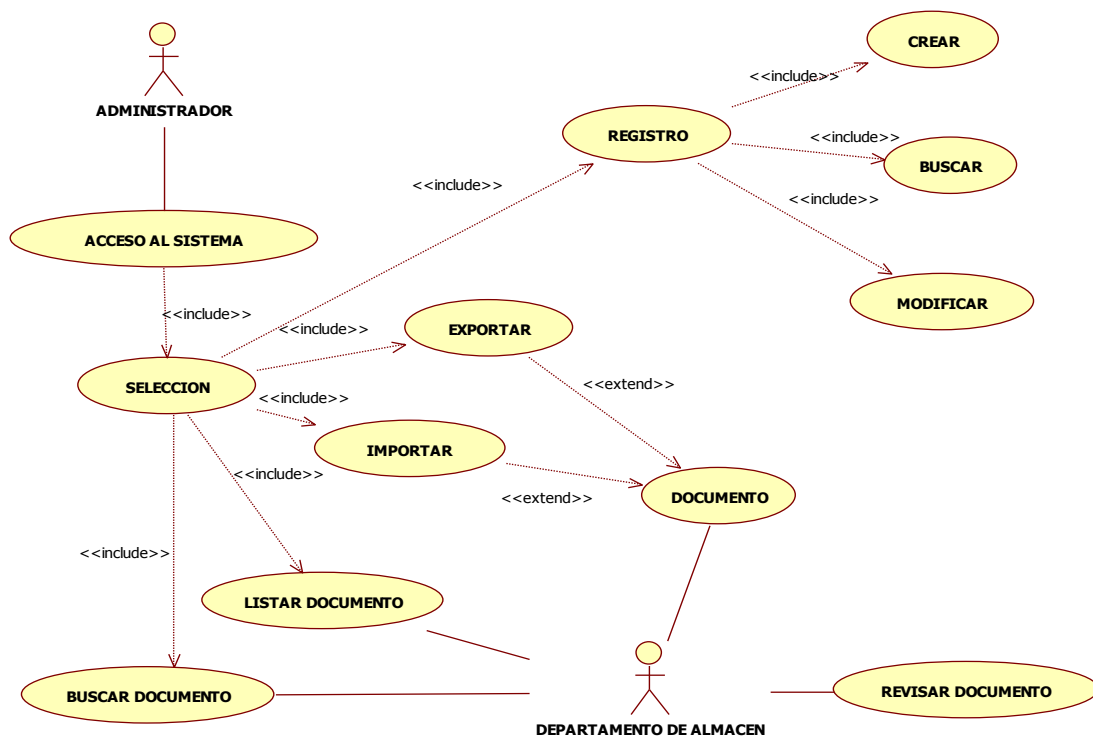


Figura 24 Diagrama de casos de uso (Elaboración propia).

Tabla 8 Narración de casos de uso (Elaboración propia).

Narración de casos de uso	
Caso de uso	Descripción
Acceso al sistema	En este caso de uso, es cuando el administrador o el encargado del área de compras deban ingresar al sistema para realizar la consulta, ingresando la contraseña para poder acceder.
Selección	Se podrá seleccionar la operación, la cual desee realizar tanto el administrador como el encargado del área de compras de refacciones.
Buscar documento	Este caso de uso describe, si se va a realizar la búsqueda del documento solicitado, el mismo día o uno anterior.
Listar documento	En este caso de uso, se describe cuando en el área de compras se quiera listar las piezas que se encuentran en cada almacén para no ingresarlas al pedido, obteniendo la lista del documento solamente las piezas faltantes, en un archivo de Excel.
Registro	Evento cuando en el área de compras de refacciones se va a dar de alta, buscar o modificar: Almacén, Centro, División, TipMaterial, GrupoArticulo, Pieza,

	Pedido y Material de inventario. Agregando los datos correctos de acuerdo a la información que se necesitan de cada uno de los mencionados anteriormente.
Exportar	El encargado de realizar los pedidos, podrá exportar un archivo Excel, el cual podrá ser revisado cuando se requiera saber de piezas existentes en el inventario.
Importar	El encargado de realizar los pedidos, podrá importar un archivo Excel a la base de datos, en la cual realizara la búsqueda de piezas existentes en el inventario.
Crear	Este describe un evento, en que el encargado deba de crear un o un(a): Almacén, Centro, División, TipMaterial, GrupoArticulo, Pieza, Pedido y Material de inventario.
Buscar	Este caso de uso describe cuando el encargado deba de buscar un o un(a): Almacén, Centro, División, TipMaterial, GrupoArticulo, Pieza, Pedido y Material de inventario. Para poder revisar si fueron registrados correctamente en la base de datos o alguna consulta que desee realizar.
Modificar	Este caso de uso describe un evento cuando el encargado deba de modificar un o un(a): Almacén, Centro, División, TipMaterial, GrupoArticulo, Pieza, Pedido y Material de inventario. Para poder corregir algún dato, ya sea por cambio de pieza o error de nombre de los registros agregados, así realizando la modificación para una consulta posterior.
Revisar documento	Caso de uso donde se obtendrá el evento de mostrar el archivo que fue obtenido en la búsqueda de las piezas que no se encontraban en el almacén y así realizar el pedido con facilidad.

Al haber obtenido el diseño del modelo de casos de uso, se implementará de acuerdo al modelo relacional. Las principales ventajas de la utilización de bases de datos relacionales son que:

- Actúan sobre las tablas en su conjunto, en lugar de hacerlo sobre registros, como ocurre en otros sistemas.
- Se pueden realizar consultas complejas que utilizan varias tablas de forma simple.
- Son fáciles de utilizar (La organización física de los datos, es independiente de su tratamiento lógico).

6.1.4 Desarrollo de la base de datos.

Para recopilar la información, se solicitó a Ford el formato de Excel, el cual es obtenido realizando una búsqueda llamada MC9 (Figura 25), que los encargados de cada almacén tienen acceso, mostrando toda la información de la pieza dentro de cada almacén. Se tomará en cuenta para realizar el sistema de información.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2	Gen	Material	Almac	Plan.n	Gara	Tpa	gpo.ar	Divisi	Sect	Mes	ValStkV	ValStkV	CtdStk	CtdStk	Stk.con	Stk.con
3	R110	1013711	A100		ND	ZREF	001	D110	03	04.2015	0,00	MXN	0,00	UN		0 UN
4	R110	1013818	A100		ND	ZREF	001	D110	03	04.2015	776,60	MXN	10,00	UN		0 UN
5	R110	1038008	A100		ND	ZR39	OM11	D110	03	04.2015	173,28	MXN	12,00	UN		0 UN
6	R110	1051773	A100		ND	ZREF	001	D110	03	04.2015	947,60	MXN	8,00	UN		0 UN
7	R110	1067029	A100		ND	ZREF	001	D110	03	04.2015	815,68	MXN	8,00	UN		0 UN
8	R110	1081173	A100		ND	ZREF	001	D110	03	04.2015	77,00	MXN	1,00	UN		0 UN
9	R110	1091566	A100		ND	ZREF	001	D110	03	04.2015	0,00	MXN	0,00	UN		0 UN
10	R110	1091567	A100		ND	ZREF	001	D110	03	04.2015	0,00	MXN	0,00	UN		0 UN
11	R110	1097077	A100		ND	ZR39	113	D110	03	04.2015	913,85	MXN	35,00	UN		0 UN
12	R110	110000026	A100		ND	ZREF	001	D110	03	04.2015	5.394,12	MXN	12,00	UN		0 UN
13	R110	1102505	A100		ND	ZREF	001	D110	03	04.2015	57,16	MXN	2,00	UN		0 UN
14	R110	1131649	A100		ND	ZR39	114	D110	03	04.2015	238,77	MXN	7,00	UN		0 UN
15	R110	1132350	A100		ND	ZREF	001	D110	03	04.2015	144,22	MXN	2,00	UN		0 UN
16	R110	1132443	A100		ND	ZREF	001	D110	03	04.2015	1.338,59	MXN	9,00	UN		0 UN
17	R110	1135010	A100		ND	ZREF	01	D110	03	04.2015	0,00	MXN	0,00	UN		0 UN
18	R110	1142140	A100		ND	ZREF	001	D110	03	04.2015	239,30	MXN	1,00	UN		0 UN
19	R110	1148288	A100		ND	ZREF	119	D110	03	04.2015	1.870,20	MXN	8,00	UN		0 UN
20	R110	1202789	A100		ND	ZREF	001	D110	03	04.2015	244,51	MXN	1,00	UN		0 UN
21	R110	1211100	A100		ND	ZREF	001	D110	03	04.2015	0,00	MXN	0,00	UN		0 UN
22	R110	1216078	A100		ND	ZREF	124	D110	03	04.2015	218,55	MXN	3,00	UN		0 UN
23	R110	1216907	A100		ND	ZREF	114	D110	03	04.2015	79,26	MXN	2,00	UN		0 UN

Figura 25 Recolección de datos de SAP (Elaboración propia).

Para realizar el modelado de datos, se emplea el modelo relacional, debido a que este, es un objeto semántico que facilita el estudio de las clases que conforman la base de datos, los cuales fueron obtenidos del documento de Excel (MC9), generado por el usuario de compras de refacciones (Figura 26).

Centro	Material	Almacén	Plan.nec.	Caract.	TpMat	Gpo.artíc.	División	Sector	Mes	ValStkVal	ValStkVal	CtdStk SV	CtdStk SV	Stk.consi.	Stk.consi.
R110	1013711	A100		ND	ZREF	1	D110	3	42.015	0,00	MXN	0,00	UN	0	UN
R110	1013818	A100		ND	ZREF	1	D110	3	42.015	776,60	MXN	10,00	UN	0	UN
R110	1038008	A100		ND	ZR39	OM11	D110	3	42.015	173,28	MXN	12,00	UN	0	UN
R110	1051773	A100		ND	ZREF	1	D110	3	42.015	947,60	MXN	8,00	UN	0	UN
R110	1067029	A100		ND	ZREF	1	D110	3	42.015	815,68	MXN	8,00	UN	0	UN
R110	1081173	A100		ND	ZREF	1	D110	3	42.015	77,00	MXN	1,00	UN	0	UN
R110	1091566	A100		ND	ZREF	1	D110	3	42.015	0,00	MXN	0,00	UN	0	UN
R110	1091567	A100		ND	ZREF	1	D110	3	42.015	0,00	MXN	0,00	UN	0	UN
R110	1097077	A100		ND	ZR39	113	D110	3	42.015	913,85	MXN	35,00	UN	0	UN
R110	11000026	A100		ND	ZREF	1	D110	3	42.015	5.394,12	MXN	12,00	UN	0	UN

Figura 26 Registros obtenidos por SAP (Elaboración propia).

A continuación, se obtendrá la normalización para llegar a la obtención del modelo relacional. Primera forma normal (1FN), para que una tabla pueda ser considerada, una relación no debe admitir grupos repetitivo. Se dice que una relación está en 1FN, si cada atributo toma un único valor del dominio subyacente. De tal forma que la separación de las tablas fue representada por un color, como se muestra a continuación en la (Figura 27 y Figura 28).

Centro	Material	Almacén	Plan.nec.	Caract.	TpMat	Gpo.artíc.	División	Sector	Mes	ValStkVal	ValStkVal	CtdStk SV	CtdStk SV	Stk.consi.	Stk.consi.
R110	1013711	A100		ND	ZREF	1	D110	3	42.015	0,00	MXN	0,00	UN	0	UN
R110	1013818	A100		ND	ZREF	1	D110	3	42.015	776,60	MXN	10,00	UN	0	UN
R110	1038008	A100		ND	ZR39	OM11	D110	3	42.015	173,28	MXN	12,00	UN	0	UN
R110	1051773	A100		ND	ZREF	1	D110	3	42.015	947,60	MXN	8,00	UN	0	UN
R110	1067029	A100		ND	ZREF	1	D110	3	42.015	815,68	MXN	8,00	UN	0	UN
R110	1081173	A100		ND	ZREF	1	D110	3	42.015	77,00	MXN	1,00	UN	0	UN
R110	1091566	A100		ND	ZREF	1	D110	3	42.015	0,00	MXN	0,00	UN	0	UN
R110	1091567	A100		ND	ZREF	1	D110	3	42.015	0,00	MXN	0,00	UN	0	UN
R110	1097077	A100		ND	ZR39	113	D110	3	42.015	913,85	MXN	35,00	UN	0	UN
R110	11000026	A100		ND	ZREF	1	D110	3	42.015	5.394,12	MXN	12,00	UN	0	UN

Figura 27 Separación de tablas MC9 (Elaboración propia).

	Tabla Centro
	Tabla Pieza
	Tabla Almacen
	Tabla TipMaterial
	Tabla GrupoArticulo
	Tabla División

Figura 28 Presentación de tablas obtenidas en 1ra normalización (Elaboración propia).

En esta primera normal formal, los datos de cada columna deben de tener un nombre único, de igual manera por tipos de datos. Se obtuvo de esta manera ya que dos filas no pueden tener el mismo valor, además no deben contener grupos repetitivos de tal forma que la tabla a modelar será la siguiente (Figura 29):

material	planNecesidad	caracteristica	sector	fecha	valStockCosto	valStockMoneda	cantStockCantidad	cantStockUnidad	stockConsiCantidad	stockConsiUnidad
1013711		ND	3	42.015	0,00	MXN	0,00	UN	0	UN
1013818		ND	3	42.015	776,60	MXN	10,00	UN	0	UN
1038008		ND	3	42.015	173,28	MXN	12,00	UN	0	UN
1051773		ND	3	42.015	947,60	MXN	8,00	UN	0	UN
1067029		ND	3	42.015	815,68	MXN	8,00	UN	0	UN
1081173		ND	3	42.015	77,00	MXN	1,00	UN	0	UN
1091566		ND	3	42.015	0,00	MXN	0,00	UN	0	UN
1091567		ND	3	42.015	0,00	MXN	0,00	UN	0	UN
1097077		ND	3	42.015	913,85	MXN	35,00	UN	0	UN
110000026		ND	3	42.015	5.394,12	MXN	12,00	UN	0	UN

Centro	Almacén	TpMat	Gpo.artíc.	División
R110	A100	ZREF	1	D110
R110	A100	ZREF	1	D110
R110	A100	ZR39	OM11	D110
R110	A100	ZREF	1	D110
R110	A100	ZREF	1	D110
R110	A100	ZREF	1	D110
R110	A100	ZREF	1	D110
R110	A100	ZREF	1	D110
R110	A100	ZR39	113	D110
R110	A100	ZREF	1	D110

Figura 29 Primera forma normal (Elaboración propia).

Los datos se distribuyeron de tal forma que no exista ninguna información duplicada en las columnas, así realizando la segunda normal formal. Segunda forma normal (2NF). Se dice que una relación está en 2NF si:

- Se encuentra en 1NF.
- Todos los campos que no son claves dependen de todos los campos claves.

Se realizó la segunda forma normal, tomando en cuenta la Figura 29, se separando por claves para cada campo o atributo, las tablas que fueron obtenidas anteriormente se les asignan las llaves primarias como se muestra a continuación (Figura 30):

id	material	planNecesidad	caracteristica	sector	fecha	valStockCosto	valStockMoneda	cantStockCantidad	cantStockUnidad	stockConsiCantidad	stockConsiUnidad
1	1013711		ND	3	42.015	0,00	MXN	0,00	UN	0	UN
2	1013818		ND	3	42.015	776,60	MXN	10,00	UN	0	UN
3	1038008		ND	3	42.015	173,28	MXN	12,00	UN	0	UN
4	1051773		ND	3	42.015	947,60	MXN	8,00	UN	0	UN
5	1067029		ND	3	42.015	815,68	MXN	8,00	UN	0	UN
6	1081173		ND	3	42.015	77,00	MXN	1,00	UN	0	UN
7	1091566		ND	3	42.015	0,00	MXN	0,00	UN	0	UN
8	1091567		ND	3	42.015	0,00	MXN	0,00	UN	0	UN
9	1097077		ND	3	42.015	913,85	MXN	35,00	UN	0	UN
10	11000026		ND	3	42.015	5.394,12	MXN	12,00	UN	0	UN

id	Centro
1	R110
1	R110
1	R110
1	R110
1	R110
1	R110
1	R110
1	R110
1	R110
1	R110
1	R110

id	Almacén
1	A100
1	A100
1	A100
1	A100
1	A100
1	A100
1	A100
1	A100
1	A100
1	A100
1	A100

id	TpMat
1	ZREF
1	ZREF
2	ZR39
1	ZREF
1	ZREF
1	ZREF
1	ZREF
1	ZREF
1	ZREF
2	ZR39
1	ZREF

id	Gpo.artic.
1	1
1	1
2	OM11
1	1
1	1
1	1
1	1
1	1
1	1
3	113
1	1

id	División
1	D110
1	D110
1	D110
1	D110
1	D110
1	D110
1	D110
1	D110
1	D110
1	D110
1	D110

Figura 30 Segunda forma normal (Elaboración propia).

De acuerdo a la relación con los datos de la tabla se obtendrá el nombre y atributos de cada una agregando la relación entre ellas (*Foreign Key*). Obtenidos de la siguiente manera (Figura 31).



Figura 31 Tablas con asignación de nombres y atributos (Elaboración propia).

En la tercera forma normal (3NF). Debe de estar en 2NF y sin contener dependencias transitivas. Esta normalización se detiene en la segunda forma normal ya que proporciona las mejores características teniendo en cuenta la inversión de tiempo y esfuerzo, pero en este caso no fue necesaria la tercera forma normal. Ya que para el desarrollo de la base de datos se debe de efectuar la normalización hasta que no haya errores de datos repetitivos o pertenezcan a un mismo campo.

Una vez obtenida la normalización de la tabla inicial, se tomará en cuenta para el desarrollo del diagrama de clases. En lenguaje unificado de modelado (UML), es un tipo de diagrama de estructura estática que describe los procesos de un sistema mostrando sus clases. De acuerdo a la normalización existe una tabla llamada Centro, la cual es una clase, debe tener una llave primaria para poder relacionarla con otras tablas, en este caso es el id, teniendo en la tabla Centro, los atributos siguientes:

- idCentro
- codigoCentro
- nombreCentro

Dentro de este diagrama de clases las operaciones, que en este caso son:

- Nuevo (Dar de alta un nuevo Centro)
- Buscar (Visualizar el Centro que haya sido de alta)
- Actualizar (Modificar un Centro o un cambio en un atributo)

A petición del encargado del área de refacciones, se realizó una interfaz de Invitado, en la cual esta cuenta con similares opciones a las del administrador, además se implementó una generación de reportes de Excel para que el administrador del área, realice ciertos procesos en cuanto al movimiento del material, en este caso si fuese lento o rápido.

Para este caso se realizaron tablas temporales en las cuales quedaran guardados los registros que cambian constantemente, siendo una para solicitudes y la otra para el inventario de piezas obtenidas del MC9 del almacén, quedando de la forma siguiente (Figura 32):

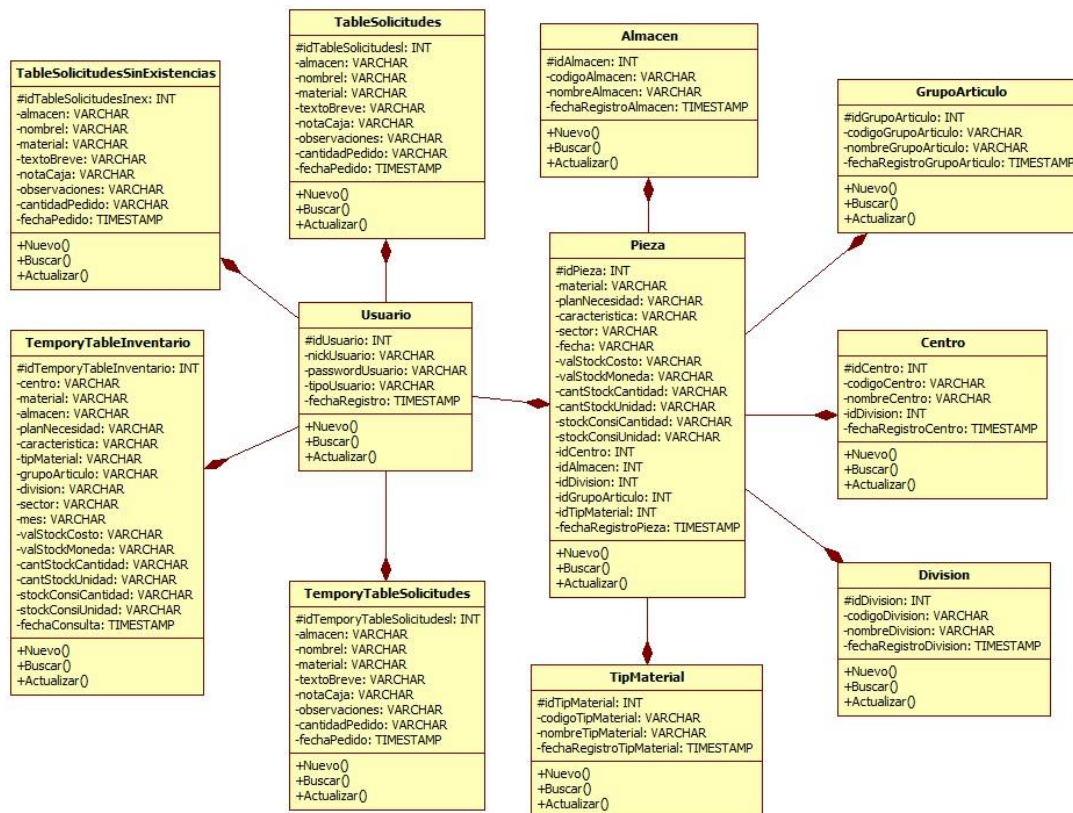


Figura 32 Diagrama de clases del sistema (Elaboración propia).

6.1.4.1 Definición de datos.

A continuación, se muestra el glosario de datos que son imprescindibles para el funcionamiento del área de compras de refacciones automotrices (Tabla 9).

Tabla 9 Descripción de los atributos agregados en las tablas (Elaboración propia).

Datos utilizados dentro del área de compra de refacciones		
Dato	Tipo de dato / Longitud	Descripción
idPieza	Int	Es la clave la cual identifica un Centro en específico. Se puede tener acceso a los datos guardados. No se repiten.

material	Varchar (45)	Se refiere al número de identificación de parte del material.
característica	Varchar (2)	Es la característica de la pieza del inventario.
sector	Varchar (2)	Se refiere al sector de la empresa en la que se encuentra la pieza.
fecha	Date	Es la fecha en la cual fue realizada la búsqueda en todos los almacenes.
valStockCosto	Decimal (18,2)	Es el costo total de todo el material.
valStockMoneda	Varchar (3)	Se refiere al tipo de moneda en que es valorado el material.
cantStockCantidad	Decimal (18,0)	Cantidad de piezas obtenidas en la búsqueda de inventario.
cantStockUnidad	Varchar (2)	Es el valor en unidades que se encuentra el material.
stockConsiCantidad	Decimal (18,0)	Se refiere si ese material ya ha sido pedido en otra sucursal.
stockConsiUnidad	Varchar (2)	Es el valor en unidades que se encuentra el material.
idCentro	Int	Es la clave con la cual se identifica un Centro en específico. No se repiten. Y se puede tener acceso con su clave.
codigoCentro	Varchar (4)	Es el código en el que se identifica por su número de sucursal.
nombreCentro	Varchar (45)	Es el nombre de la sucursal.
idDivision	Int	Es la clave con la cual se identifica. Se puede tener acceso a los datos guardados con su clave. No se repiten.
codigoDivision	Varchar (4)	Si la sucursal está dividida en dos áreas. Ejemplo: Refacciones y Accesorios.
nombreDivision	Varchar (45)	Nombre si está dividida la sucursal en 2 centros.
idGrupoArticulo	Int	Es la clave con la cual se identifica un GrupoArticulo en específico. Se puede

		tener acceso a los datos guardados con su clave. No se repiten.
codigoGrupoArticulo	Varchar (6)	Si el material proviene de algún proveedor registrado en la empresa o no.
nombreGrupoArticulo	Varchar (45)	Es el nombre del grupo en el que se encuentra la refacción.
idAlmacen	Int	Es la clave con la cual se identifica un Almacén en específico. Se puede tener acceso a los datos guardados con su clave. No se repiten.
codigoAlmacen	Varchar (4)	Es el código del Almacén del área de refacciones.
nombreAlmacen	Varchar (45)	Es el nombre del Almacén de cada sucursal.
idTipMaterial	Int	Es la clave con la cual se identifica un TipMaterial en específico. Se puede tener acceso a los datos guardados con su clave. No se repiten.
codigoTipMaterial	Varchar (4)	Se refiere al código al que pertenece la pieza.
nombreTipMaterial	Varchar (45)	Es el nombre del tipo de material en caso de contar con él.
idTemporaryTableInventario	Int	Es la clave con la cual se identifica una pieza de inventario en específico. Se puede tener acceso a los datos guardados con su clave. No se repiten.
centro	Varchar (4)	Es el centro en el cual está ubicada la pieza automotriz.
mes	Date	Es la fecha en la que se descargó el inventario del almacén en el archivo de Excel llamado MC9.
fechaConsulta	Timestamp	Es la fecha en la cual, es realizada la consulta en el inventario del sistema.
idTemporaryTableSolicitudes	Int	Es la clave con la cual se encuentra identificada la pieza de pedido

almacén	Varchar (45)	Es el nombre del almacén que está solicitando el encargado del almacén.
nombre	Varchar (45)	Es el nombre de quien solicita el material.
material	Varchar (45)	Material que es requerido por el encargado de almacén de cada una de las sucursales para su venta.
textoBreve	Varchar (45)	Es donde el usuario escribe alguna descripción, sobre el material que se está solicitando.
notaCaja	Varchar (45)	Se refiere si alguna pieza aun no llega, pero ya fue solicitada con anterioridad.
observaciones	Varchar (45)	Es alguna referencia u observación de la pieza, ya sea que este dañada, que aún no ha sido surtida o se encuentra en algún otro almacén.
cantidadPedido	Int	Se refiere a la cantidad de pedido que se está solicitando.
fechaPedido	Timestamp	Es la fecha en la cual se está solicitando el material.
idTableSolicitudes	Int	Esta es la clave con la cual el pedido se solicita, pero al mismo tiempo, es el registro que se guarda para su revisión posterior, haciendo una consulta por un rango de fechas de solicitud de pedido.
idTableSolicitudesInex	Int	Se refiere a la clave del material que no se encuentra en ningún almacén o fuera de inventario.

6.1.5 Desarrollo del modelo relacional.

Dentro de este paso metodológico, se usó la herramienta llamada MySQL Workbench 6.2 CE. Elaborándose de acuerdo a las tablas obtenidas en la normalización, haciendo uso de esta herramienta para insertar los nombres de las tablas, atributos, llaves primarias de igual forma las llaves foráneas, relacionándolas entre sí, dentro de esta herramienta se encuentran opciones específicas para poder desarrollarla, de esta manera se inicia con la tabla de pieza (Figura 33). Al haber dado el nombre de la tabla, se agregaron los atributos correspondientes empezando por su llave primaria (*Primary Key*), la cual será el id de la tabla para poder ingresar a sus datos, la tabla será llamada Pieza, tal es el caso que su id será idPieza (Figura 34).

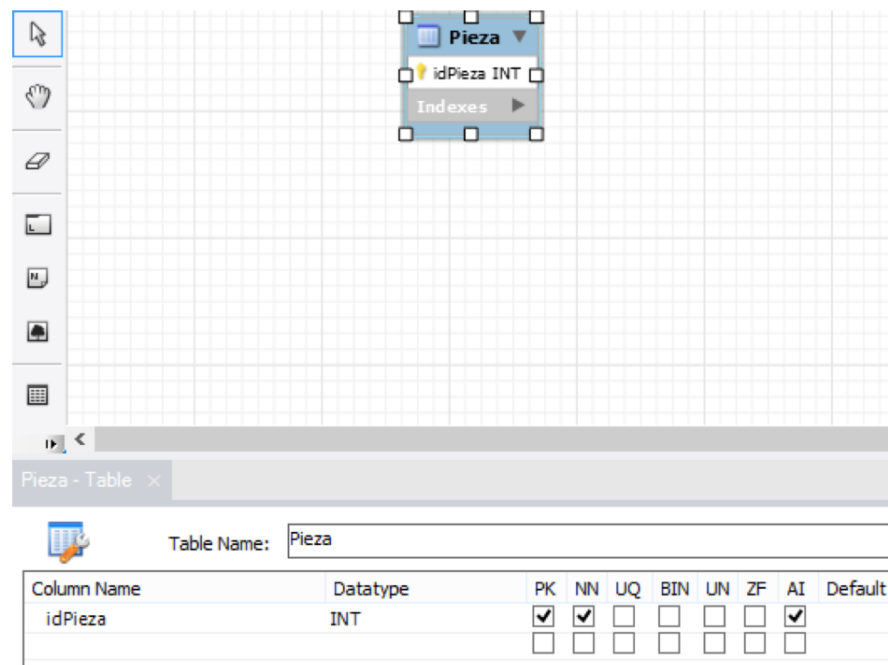


Figura 33 Creación de id en la tabla pieza (Elaboración propia).

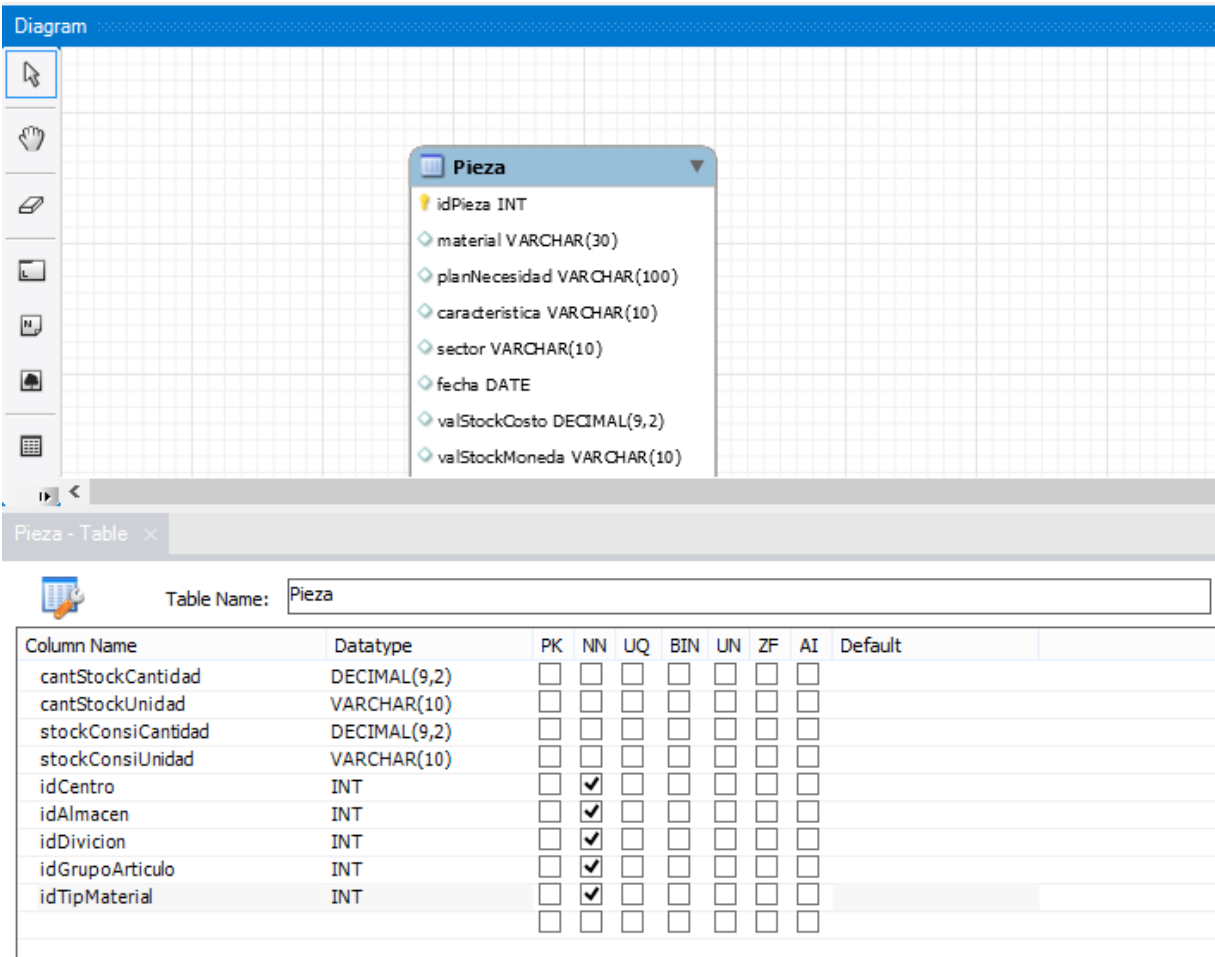


Figura 34 Asignación de atributos y tipos de datos (Elaboración propia).

Como ya sabemos que en la tabla se debe de tener un identificador, con la cual podemos realizar la relación con las demás, para poder realizar una búsqueda de sus datos, al mismo tiempo se debe de seleccionar el elemento de auto incrementable, ya que al guardar la pieza deberá obtener automáticamente un numero de id para diferenciarse entre los demás (Figura 35).

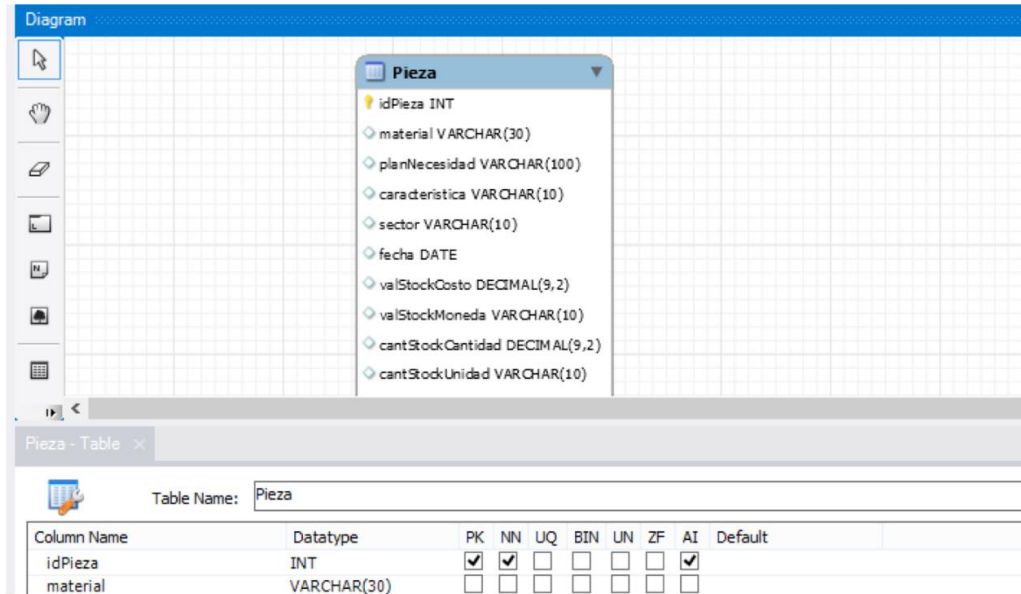


Figura 35 Asignación de la llave primaria (Elaboración propia).

Ahora bien, las tablas faltantes se realizarán de acuerdo a la primera y segunda forma normal obtenida, se les da la asignación de id para que puedan identificarse, asignando a cada atributo sus tipos de datos, quedando de la siguiente manera (Figura 36).

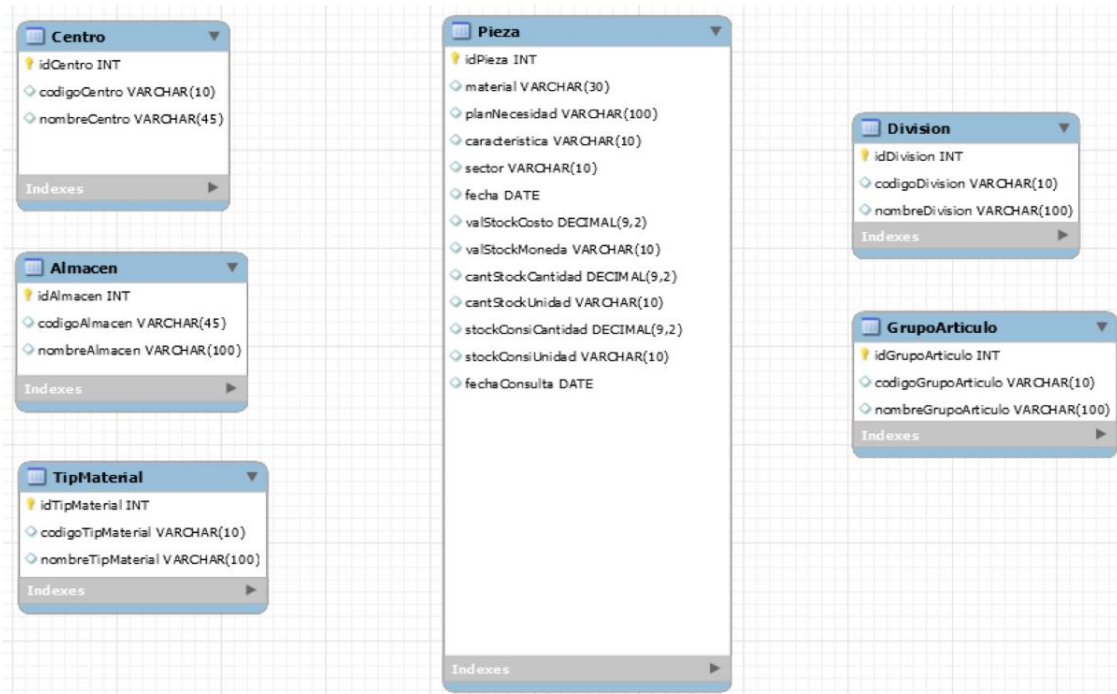


Figura 36 Tablas obtenidas de la normalización (Elaboración propia).

Ahora se hará la relación de estas, seleccionando la tabla que tiene las llaves foráneas, para obtener desde la tabla pieza los datos de las otras tablas, presentando las tablas temporales y de usuario, ver Figura 37.

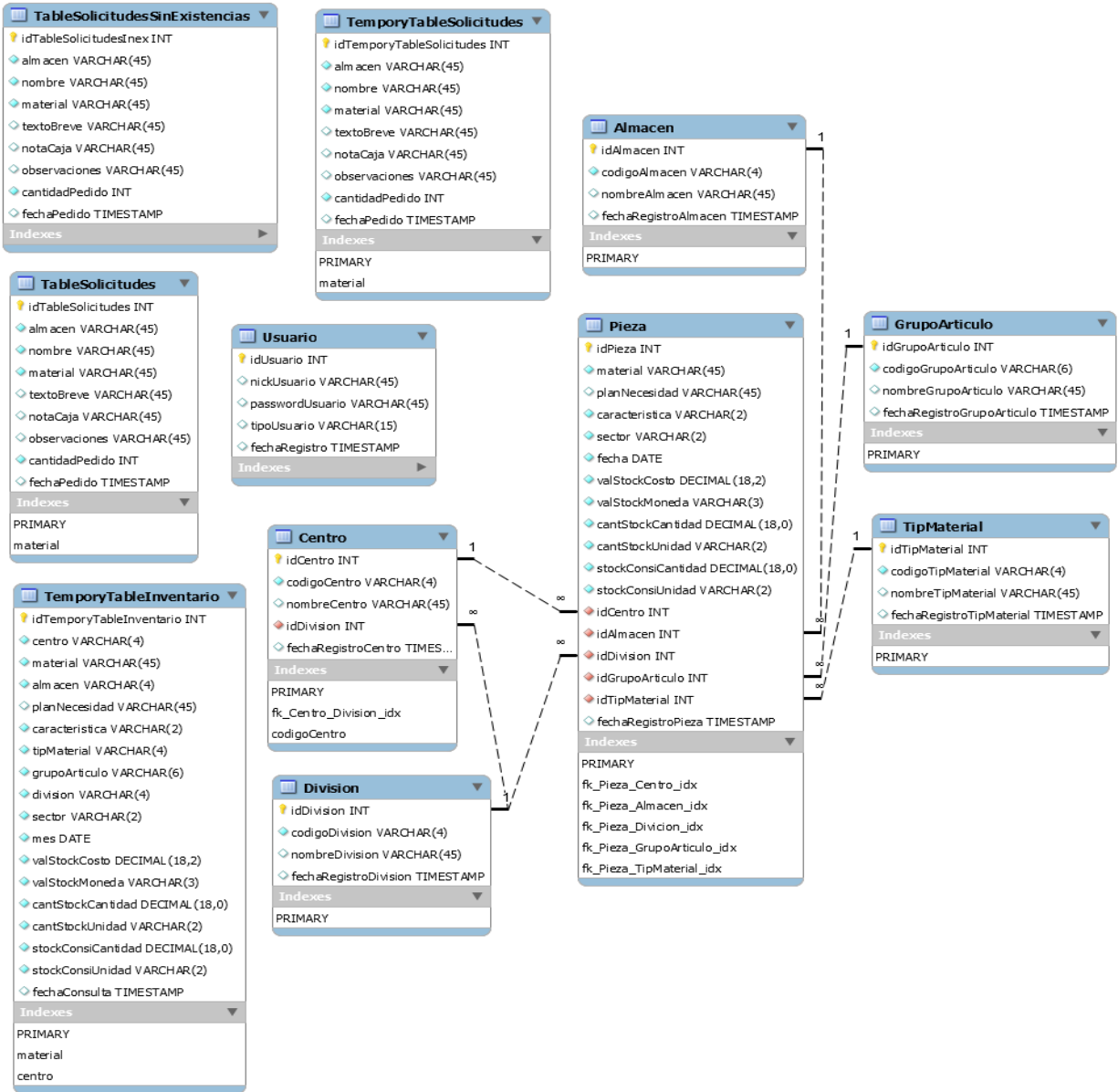


Figura 37 Modelo relacional del sistema implementado (Elaboración propia).

6.1.6 Planteamiento de algoritmos.

Una vez obtenido el diseño de la base de datos bien organizada y normalizada, se implementa físicamente en MySQL Workbench 6.2 CE, ejecutando la consola. En el Anexo III, se presentan las líneas de comando que se ejecutaron en el editor de MySQL para crear la base de datos.

El modelo relacional se toma en cuenta, para la elaboración de las tablas en la plataforma de Workbench dentro de la instancia local de MySQL, siendo que estas tablas están relacionadas entre sí, se insertaron los datos correctos para su mejor funcionalidad de la base de datos.

El código que se implementó para realizar la creación de la tabla en la base de datos llamada `bd_ford_zapata_compras`, se desarrolló de acuerdo al lenguaje de programación mencionado anteriormente con el modelado de base de datos, el cual se hizo mención a los tipos de datos que se van a utilizar en la programación.

6.1.7 Implementación de algoritmos.

En esta parte se definen cada interfaz con las que se van a trabajar, para la interacción con la base de datos, siendo en estos casos como, dar de alta un registro o varios, modificar, actualizar y buscar información perteneciente a la base de datos. De los cuales se explicarán sus interfaces a continuación:

Interfaz de acceso: Es la interfaz la cual le da permisos al usuario, para acceder al sistema, realizando las operaciones correspondientes, para el que se ha implementado.

Interfaz de registro de usuario: Es en la cual algún usuario puede crear su cuenta para poder ingresar al sistema y a la base de datos, con su respectiva información dependiendo del tipo de usuario que sea.

Interfaz de actualización de contraseña: Interfaz donde el usuario, ya sea el caso que haya olvidado su contraseña, podrá actualizarla o consultarla con el encargado o administrador.

Interfaz del sistema de invitado: Es cuando el usuario entra al sistema, se le da acceso a la interfaz del invitado, siendo este el caso, solamente se le da acceso a cierta información definida por el administrador en los requerimientos. Pero haciendo los registros, búsquedas, actualizaciones, la realización de búsqueda de pedido y generación de reportes, pero no detalladamente.

Interfaz del sistema de administrador: Es la interfaz donde el administrador puede realizar consultas, registros, dar de alta, actualizar información, realizar búsqueda de pedido, realizando de manera detallada reportes por fechas, además del control de usuarios que tienen acceso al sistema, realizando movimientos de modificación o eliminación de usuarios.

Dentro de estas interfaces está dividido en diferentes módulos para mejorar la interacción de usuario y sistema, dentro del cual se mencionan los siguientes: Archivo, Mantenimiento (Dependiendo del tipo de usuario que se encuentre en el sistema podrá tener la información permitida), Consultas, Movimiento, Reportes, Administrador (En caso de que el usuario ingresado pertenezca al Administrador), y Sistema. De los cuales se explicará cada menú a continuación:

Archivo: En este menú tenemos opciones para limpiar la pantalla, la cual si tenemos varias interfaces abiertas con solo dar un clic se limpiará el panel, la siguiente opción es para cerrar sesión, la cual es solo para poder iniciar sesión con otro usuario, y por último la opción de salir del sistema (Figura 38).

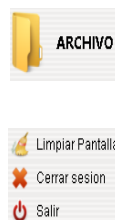


Figura 38 Menú de archivo (Elaboración propia).

Mantenimiento: Como se mencionó anteriormente, es dependiendo para cada usuario dentro del cual el Invitado podrá dar de alta, buscar y actualizar la información del Almacén, División, Centro, GrupoArticulo, TipoMaterial, Pieza, Pedido e Inventario (Figura 39). Y el administrador podrá realizar las mismas operaciones que realiza el Invitado, pero detalladamente.

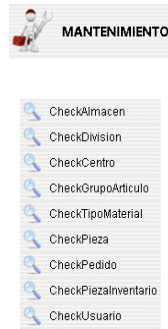


Figura 39 Menú de mantenimiento (Elaboración propia).

Consultas: Dentro de este menú se podrán realizar consultas, con las cuales se encuentran dadas de alta alguna información del Almacén, División, Centro, GrupoArticulo, TipoMaterial, Pieza, Pedido e Inventario (Figura 40), obteniendo lo que almacena cada clave con las cuales están clasificadas.



Figura 40 Menú de consultas (Elaboración propia).

Movimiento: Dentro de este menú se realiza la búsqueda de material, teniendo las siguientes opciones:

- **Importación de solicitud:** Es donde el usuario tiene acceso a realizar la importación del pedido que se está solicitando, de igual forma guardarla en la base de datos.
- **Importación de complementos:** Es la opción donde se agregan de forma más rápida desde un documento Excel, los requisitos de materiales a solicitar, para facilitarles la creación de cada una de las partes que las complementan.
- **Solicitud de material:** Dentro de esta interfaz se tiene las opciones de poder realizar la búsqueda del material que existe en el inventario, con la opción de poder realizar una exportación a un documento en Excel.
- **Búsqueda:** Es la interfaz en la cual se puede realizar la búsqueda, ya sea por material o por sucursal, contando con la opción de exportar los datos en un documento Excel.
- **Consulta de tablas:** Es donde el usuario puede tener la información de todo lo que existe en la base de datos dependiendo del tipo de usuario que sea.

Reportes: Es donde se encuentran, las interfaces de generación de reportes en documento de Excel (Figura 41), en los cuales estos pueden ser buscados por el tiempo en que fueron registrados en la base de datos, con la opción de exportar.



- FechaAlmacenSis
- FechaDivisionSis
- FechaCentroSis
- FechaGrupoArticuloSis
- FechaTipoMaterialSis
- FechaUsuarioSis
- FechaPiezaSis
- FechaSolicitudSis
- FechaInventarioSis

Figura 41 Menú de reportes (Elaboración propia).

Administrador: Son las interfaces donde solo el Administrador puede tener acceso a los usuarios, además de la generación de reportes del material que se hayan solicitado, siendo que estos se encuentren dentro o fuera del inventario, ver Figura 42. Además, ayudarle a la toma de decisiones de que material se ha estado solicitando cotidianamente.



- Nuevo usuario
- DMA-Pedido
- DMA-MaterialExist
- DMA-MaterialNoExist

Figura 42 Menú de administrador (Elaboración propia).

Sistema: Dentro de este menú se encuentra acerca de la información del sistema y del creador del mismo (Figura 43).

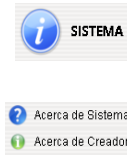


Figura 43 Menú de sistema (Elaboración propia).

Las interfaces de usuarios que se implementaron son las siguientes:

Interfaz de invitado (Figura 44), permite al invitado realizar la importación y exportación de documento de Excel, realizar la búsqueda de las piezas solicitadas que se encuentran dentro del inventario, realizando una exportación del reporte para su mejor aprovechamiento dentro del área de compras de refacciones, para el administrador.



Figura 44 Interfaz de invitado (Elaboración propia).

Interfaz donde el administrador (Figura 45), tiene acceso a más información realizando las mismas actividades del invitado, pero al mismo tiempo, poder realizar reportes, búsquedas de material solicitado en un rango de fechas, siendo otro caso la visualización de usuarios que tienen permitido el acceso al sistema y a la base de datos.



Figura 45 Interfaz de administrador (Elaboración propia).

Tomando en cuenta el apartado Modelo - Vista - Controlador (MVC), que se definió anteriormente, se desarrollaron los algoritmos necesarios para facilitar la utilización del sistema, adaptándolo de acuerdo a las necesidades del usuario. En este apartado de la metodología se desarrolló un sistema con acceso a base de datos que permite gestionar los artículos y sus respectivos movimientos, se aplicaron los patrones de diseño:

- *Data Transfer Objet (DTO).*
- *Data Access Objet (DAO).*
- *Session Facade.*
- *Bussines Delegate.*

Esto para cumplir con el patrón arquitectónico Model - View - Controller. Una vez teniendo creada la base de datos, se asignará el nombre al proyecto llamado FordZapataCompras, dentro de este proyecto se agregaron los paquetes siguientes:

- `com.fordzapatacompras.modelo.dao`
- `com.fordzapatacompras.modelo.delegate`
- `com.fordzapatacompras.modelo.dto`
- `com.fordzapatacompras.modelo.facade`

De acuerdo a los paquetes mencionados, dentro de ellos se agregaron las clases, tomando en cuenta el modelo relacional de la base de datos desarrollada. Se desarrolló la tabla almacén de la forma siguiente:

Dentro del paquete `com.fordzapatacompras.modelo.dao` (Figura 46). Se creó la clase `AlmacenDAO` (Figura 47), esta clase es la encargada de encapsular la lógica de acceso a la base de datos. Por lo tanto, la codificación dentro de la clase `AlmacenDAO` quedó de la siguiente manera:

```
package com.fordzapatacompras.modelo.dao;
import com.fordzapatacompras.modelo.dto.AlmacenDTO;
import java.sql.Connection;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
```

Figura 46 Código del paquete DAO (Elaboración propia).

Dentro del código en la parte inicial debe llevar el paquete en donde se está trabajando y creando la clase. Dentro de esta clase se hace la importación de otros paquetes, en este caso del `AlmacenDTO`, como de las importaciones necesarias, de igual manera la importación para la conexión de la base de datos con Java.

```
public class AlmacenDAO {
private static final String SQL_INSERT =
    "INSERT INTO Almacen(idAlmacen,codigoAlmacen,nombreAlmacen,"
    + "fechaRegistroAlmacen)VALUES(?,?,?,current_timestamp)";
private static final String SQL_SELECT =
    "SELECT idAlmacen,codigoAlmacen,nombreAlmacen FROM Almacen "
    + "where idAlmacen=?";
private static final String SQL_SELECT_ALL =
    "SELECT idAlmacen,codigoAlmacen,nombreAlmacen FROM Almacen";
private static final String SQL_UPDATE = "UPDATE Almacen SET codigoAlmacen=?, "
    + "nombreAlmacen=?,fechaRegistroAlmacen=current_timestamp "
    + "WHERE idAlmacen=?";
private static final String SQL_DELETE =
    "DELETE FROM Almacen WHERE idAlmacen = ?";
```

Figura 47 Código de la clase DAO (Elaboración propia).

Aquí hace referencia a las acciones que se realizan dentro de esta clase:

SQL_INSERT, es para crear un nuevo almacén con sus atributos necesarios, de tal forma que esté relacionado con las tablas de la base de datos, conforme a sus tipos de datos.

SQL_SELECT, realiza la búsqueda de los datos que se hayan guardado o creado dentro de la base de datos.

SQL_UPDATE, se refiere a que se deba de hacer una modificación de cualquier campo o atributo que se haya ingresado incorrectamente o simplemente una modificación.

SQL_DELETE, es utilizado para borrar algún dato creado por el usuario en la base de datos.

Dentro de este paquete DAO se añadió la importación del paquete DTO

```
com.fordzapatacompras.modelo.dto.AlmacenDTO
```

Se relacionó con la clase AlmacenDAO, ya que se usan ps.setString o ps.setInt dependiendo de los tipos de datos que van a ser agregados, incluso el dto.get que es para llamar los datos agregados a los campos que se desean asignar. Como se muestra a continuación (Figuras 48-49-50-51):

```

public void create(AlmacenDTO dto, Connection conn) throws
SQLException {

    PreparedStatement ps =null;
    try {
        ps = conn.prepareStatement (SQL_INSERT);
        ps.setInt (1,dto.getIdAlmacen ());
        ps.setString (2,dto.getCodigoAlmacen ());
        ps.setString (3,dto.getNombreAlmacen ());
        ps.executeUpdate ();
    } finally {
        cerrar (ps);
        cerrar (conn);
    }
}

```

Figura 48 Código de creación de almacén (Elaboración propia).

```

public AlmacenDTO load(AlmacenDTO dto,Connection conn)throws
SQLException{

    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        ps = conn.prepareStatement (SQL_SELECT);
        ps.setInt (1,dto.getIdAlmacen ());
        rs = ps.executeQuery ();
        List results = getResults (rs);
        if (results.size ()>0){
            return (AlmacenDTO) results.get (0);
        } else{
            return null;
        }
    } finally {
        cerrar (rs);
        cerrar (ps);
        cerrar (conn);
    }
}

```

Figura 49 Código de búsqueda de almacén (Elaboración propia).

```

public void update(AlmacenDTO dto, Connection conn)throws
SQLException{
    PreparedStatement ps = null;
    try{
        ps = conn.prepareStatement (SQL_UPDATE);
        ps.setString (1, dto.getCodigoAlmacen ());
        ps.setString (2,dto.getNombreAlmacen ());
        ps.setInt (3,dto.getIdAlmacen ());
        ps.executeUpdate ();
    } finally{
        cerrar (ps);
        cerrar (conn);
    }
}

```

Figura 50 Código de actualización de almacén (Elaboración propia).

```

public void delete(AlmacenDTO dto, Connection conn) throws
SQLException{
    PreparedStatement ps = null;
    try{
        ps = conn.prepareStatement (SQL_DELETE);
        ps.setInt (1,dto.getIdAlmacen ());
        ps.executeUpdate ();
    } finally{
        cerrar (ps);
        cerrar (conn);
    }
}
}

```

Figura 51 Código de eliminación de almacén (Elaboración propia).

Ahora continuando con la realización del proyecto, se realizó la implementación del paquete AlmacenDelegate, ver Figura 52. Esta es llevada de la mano con la clase AlmacenFacade, ya que tienen una relación con la conexión a la base de datos, la visualización, creación y modificación de estos, dentro del paquete que se creó, llamado com.fordzapatacompras.modelo.delegate, siendo que la clase AlmacenDelegate es la ruta de entrada a la aplicación, de igual forma es la clase que agrupa las acciones para cada módulo de la aplicación. Siendo de la misma manera con la clase anterior, se hace la importación de los paquetes quedando de la siguiente forma.

```

package com.fordzapatacompras.modelo.delegate;

import com.fordzapatacompras.modelo.dto.AlmacenDTO;
import com.fordzapatacompras.modelo.facade.AlmacenFacade;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.List;

```

Figura 52 Código del paquete Delegate (Elaboración propia).

Una vez agregando las importaciones del paquete Delegate, se dio la dirección en la cual se va a tener acceso a la base de datos, dándole el usuario de root, la contraseña para el acceso y la dirección en donde esta se encuentra, como se muestra a continuación (Figura 53).

```
public AlmacenDelegate(){
    String user ="root";
    String pwd ="cristian";
    String url ="jdbc:mysql://localhost:3306/bd_ford_zapata_compras";
    String mysqlDriver = "com.mysql.jdbc.Driver";
    try {
        Class.forName(mysqlDriver);
        cnn = DriverManager.getConnection(url, user, pwd);
    } catch (Exception e){
        e.printStackTrace();
    }
    almFacade = new AlmacenFacade(cnn);
}
```

Figura 53 Código para el acceso a la base de datos (Elaboración propia).

Se insertó el código para los datos que van a ser agregados, listados, actualizados, en caso de que se requiera dentro del almacén, agregando el paquete AlmacenDTO (Figura 54).

```
public void crearAlmacen(AlmacenDTO dto) throws SQLException{
    almFacade.crear(dto);
}
public List listarAlmacen() throws SQLException {
    return almFacade.listar();
}
public AlmacenDTO leerAlmacen (AlmacenDTO dto) throws SQLException {
    return almFacade.leer(dto);
}
public void actualizar(AlmacenDTO dto) throws SQLException {
    almFacade.actualizar(dto);
}
public void elimina(AlmacenDTO dto) throws SQLException {
    almFacade.elimina(dto);
}
```

Figura 54 Código para realizar movimientos de almacén (Elaboración propia).

En el paquete Facade, se encuentra la conexión que se genera entre la clase AlmacenDelegate - AlmacenFacade, en el código de la clase AlmacenFacade se agregó la

importación de los paquetes con los cuales se relacionan, en este caso AlmacenDAO y AlmacenDTO, como se ve a continuación en la Figura 55.

```
package com.fordzapatacompras.modelo.facade;

import com.fordzapatacompras.modelo.dao.AlmacenDAO;
import com.fordzapatacompras.modelo.dto.AlmacenDTO;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;
```

Figura 55 Código del paquete Facade (Elaboración propia).

Agregándose de la misma manera la clase AlmacenFacade, en la cual se asignó una variable llamada **cnn**, siendo la otra **dao**, para la relación con los otros paquetes que obtendrán y enviarán los datos, el código es el siguiente (Figura 56).

```
public class AlmacenFacade {
    private Connection cnn;
    private AlmacenDAO dao;

    public AlmacenFacade(Connection cnn){
        this.cnn = cnn;
        dao = new AlmacenDAO();
    }
    public void crear(AlmacenDTO dto) throws SQLException{
        dao.create(dto, cnn);
    }
    public List listar() throws SQLException{
        return dao.loadAll(cnn);
    }
    public AlmacenDTO leer(AlmacenDTO dto) throws SQLException{
        return dao.load(dto, cnn);
    }
    public void actualizar(AlmacenDTO dto) throws SQLException{
        dao.update(dto, cnn);
    }
    public void elimina(AlmacenDTO dto) throws SQLException{
        dao.delete(dto, cnn);
    }
}
```

Figura 56 Código de conexión a la base de datos (Elaboración propia).

En la clase AlmacenDTO, se hizo la asignación de variables ver Figura 57, los cuales se tomaron en cuenta del modelo de diagramas de clases, siendo que, aquí se dio la asignación de los tipos de datos, aplicando el **get** y **set**, en este caso la clase DTO es la que hace mención al nombre del almacén con un get, retornando el nombre de ese almacén que es llamado, y un set para tomar el valor donde es igual al que se está pidiendo. Por lo tanto, la codificación de la clase quedo de la siguiente forma con la importación del paquete DTO.

```
package com.fordzapatacompras.modelo.dto;

import java.io.Serializable;

public class AlmacenDTO implements Serializable {
    private int idAlmacen;
    private String codigoAlmacen;
    private String nombreAlmacen;

    public AlmacenDTO() {
    }
    public int getIdAlmacen() {
        return idAlmacen;
    }
    public void setIdAlmacen(int idAlmacen) {
        this.idAlmacen=idAlmacen;
    }
    public String getCodigoAlmacen() {
        return codigoAlmacen;
    }
    public void setCodigoAlmacen(String codigoAlmacen) {
        this.codigoAlmacen=codigoAlmacen;
    }
    public String getNombreAlmacen() {
        return nombreAlmacen;
    }
    public void setNombreAlmacen(String nombreAlmacen) {
        this.nombreAlmacen=nombreAlmacen;
    }
}
```

Figura 57 Código de asignación de variables (Elaboración propia).

Para ver el código completo de las clases que se agregaron en los paquetes ya mencionados, ir al Anexo IV.

6.1.8 Pruebas de validación y verificación del sistema.

Una vez teniendo el sistema finalizado, con las funcionalidades obtenidas desde los requerimientos, se realizaron las pruebas de validación del sistema, visualizándose de manera exhaustiva, ejecutándose cada una de las funciones, fijándose que la funcionalidad fuera la correcta.

Además, se evaluó con ambos tipos de usuarios, tomando en cuenta lo mencionado que cada uno realice las mismas funcionalidades para el que se desarrolló el sistema, pero de forma más detallada para el administrador. Llevado a cabo esto, la respuesta tanto del usuario, como la del encargado de compras de refacciones, fue positiva, mencionando que el sistema es muy amigable y contiene los términos técnicos que hacen que sea posible un manejo mejor del sistema.

6.1.9 Implementación del sistema.

La forma en que fue implementado el sistema, fue trabajando en ambas partes, con el encargado de compras de refacciones en forma manual y con SICSARF que ya ha automatizado el trabajo, obteniendo como resultado en ambas partes, los mismos resultados, siendo así que se verifico dato por dato que fueran los correctos. De esta forma SICSARF, es confiable para su uso diario; para la realización de búsqueda de material, por lo que el objetivo de realizar el pedido en un menor tiempo se cumple.

VII. Presentación y discusión de resultados

En este apartado se explican los resultados obtenidos, al haber implementado el sistema SICSARF en el área de compra de refacciones de Ford Zapata en Tlalnepantla. La pantalla inicial es la del acceso al sistema, es donde se le permite al usuario ingresar al sistema, ya sea invitado o administrador como se puede observar en la Figura 58.



Figura 58 Interfaz de acceso al sistema (Elaboración propia).

Si el usuario es nuevo en el sistema, deberá registrarse para poder acceder a él y poder realizar las operaciones que se le tienen permitidas. Además de haberse registrado con permiso del encargado del área de refacciones, si es invitado o administrador. Dentro del mismo, la actualización de contraseña del usuario es necesaria para su mayor seguridad, en caso contrario de que no la recuerde, podrá consultar con el administrador sobre su registro. Después de tener los datos cargados en la ventana se entra al sistema, de tal forma que el sistema reconoce si eres Invitado o Administrador, siendo que si no estás registrado mandara un mensaje diciendo que tus datos no son correctos, como se ve en la Figura 59.

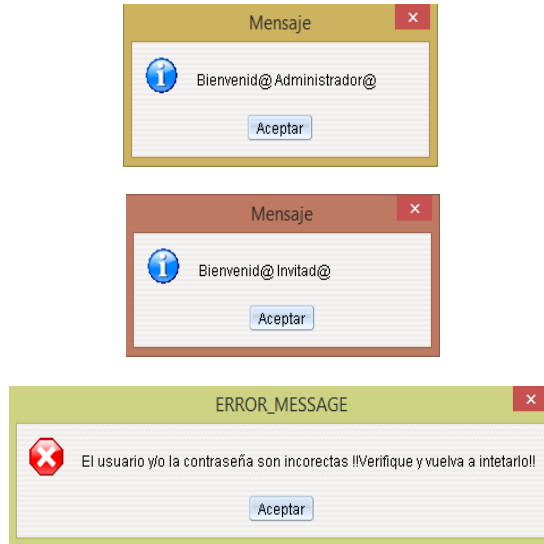


Figura 59 Interfaz de bienvenida al usuario (Elaboración propia).

Con esta acción se puede tener un nivel de seguridad deseable en cualquier sistema de datos electrónicos. Una vez ingresando al sistema, se despliega en el escritorio las opciones del sistema en el cual dependerá del tipo de usuario que ingrese, mostrándonos en la parte superior si es administrador y en la parte inferior el usuario conectado como se ve en la Figura 60.

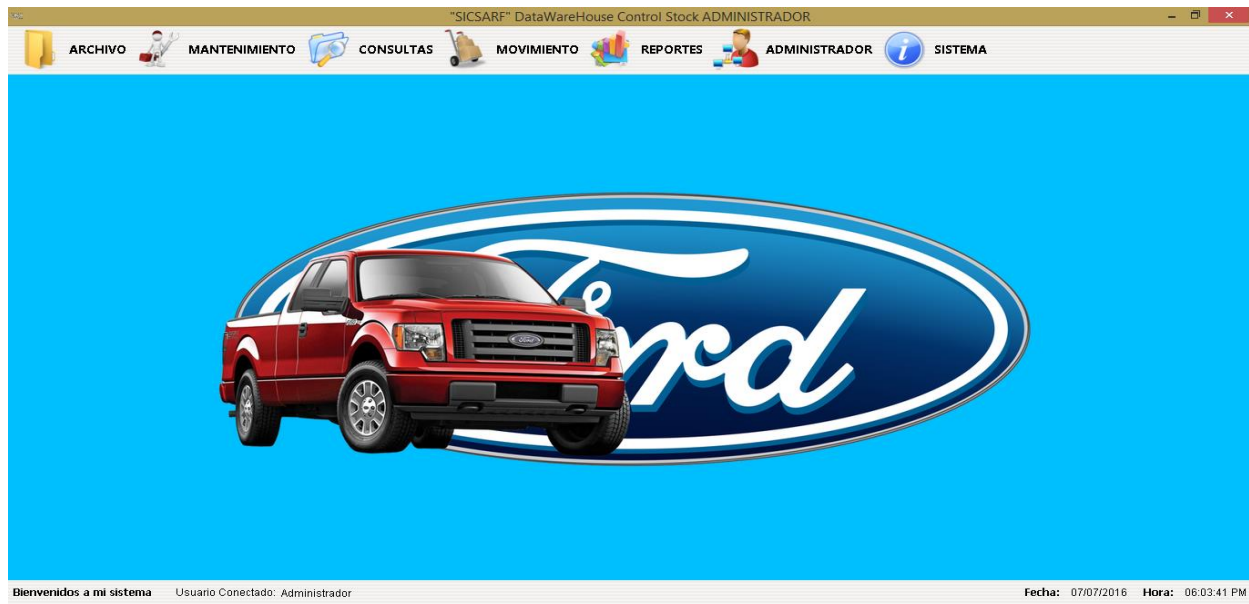


Figura 60 Interfaz de administrador (Elaboración propia).

Al seleccionar en el menú del Invitado la opción de CheckPedido, se despliega una ventana (Figura 61) en la cual se podrá dar de alta, buscar y actualizar un pedido, siempre indicando la opción de continuar operación, al confirmar, despliega un mensaje de registro guardado, como se puede ver en la figura. Con ello se logran registros de piezas.

REGISTRO PIEZA SOLICITUD	
IdTemporaryTableSolicitudes :	1
Nombre Almacen :	TEXCOCO
Nombre Solicita :	JESUS CEDILLO
Material :	MSMX05W30B
TextoBreve :	
NotaCaja :	
Observaciones :	
CantidadPedido :	5

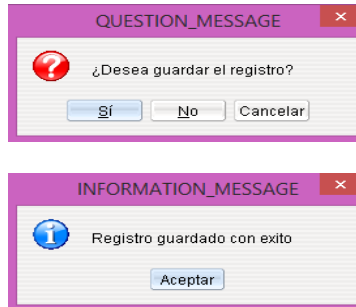


Figura 61 Interfaz de registro de pieza del invitado (Elaboración propia).

En el caso del administrador, él puede tener más información sobre lo que se está registrando, como se muestra en la Figura 62, realizando la misma operación como dar de alta, buscar y actualizar un pedido, dando la opción si deseamos continuar con la operación, mandándonos un mensaje de que el registro fue guardado. Además de realizar una búsqueda, ya sea de nombre de almacén que solicita, nombre de quien está solicitando y material.

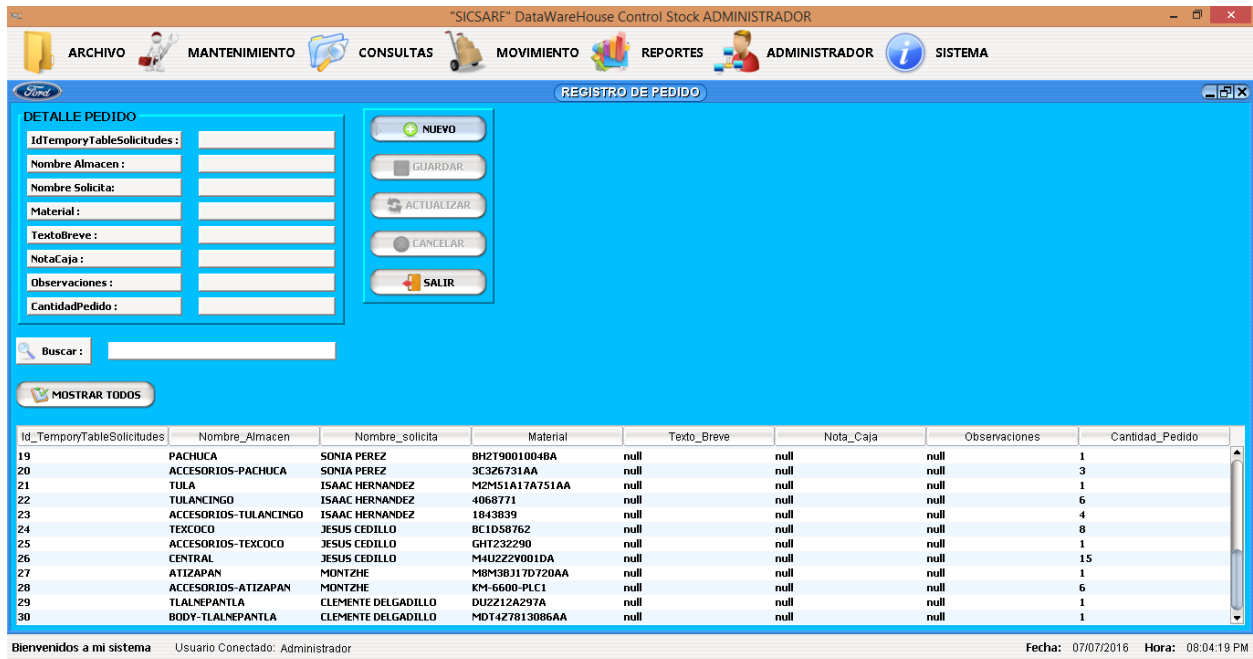


Figura 62 Interfaz de registro de pieza del administrador (Elaboración propia).

Considerando que todas las capturas de datos, son similares en todas las ventanas dentro del menú de Mantenimiento, se muestra en la Tabla 10 los atributos que la componen mostrando en la descripción lo más destacable de cada una de las pantallas.

Tabla 10 Descripción del menú de mantenimiento de SICSARF (Elaboración propia).

Descripción de menú de mantenimiento		
Interfaz	Atributos	Descripción
Registro de Almacén	<ul style="list-style-type: none"> • IdAlmacen • CodigoAlmacen • NombreAlmacen 	El IdAlmacen y el CodigoAlmacen son los más importantes de esta pantalla debido a que el id identifica al almacén y el código de almacén es el que identifica que cada almacén pertenece al mismo grupo de Ford.
Registro de División	<ul style="list-style-type: none"> • IdDivision • CodigoDivision • NombreDivision 	El IdDivision y el CodigoDivision destacan en esta pantalla debido a que el id identifica la división y el código de división es a la sucursal que pertenece dicho almacén.

Registro de Centro	<ul style="list-style-type: none"> • IdCentro • CodigoCentro • NombreCentro • IdDivision 	El IdCentro, CodigoCentro y IdDivision son los más importantes de esta pantalla debido a que el id identifica al centro, el código de centro identifica si en la misma división se encuentran varios centros y el id de la división es el atributo que identifica a que división pertenece dicho centro, por ejemplo, la división es Tlalnepantla dentro de esta sucursal existen los centros de Refacciones, Accesorios y Body.
Registro de GrupoArticulo	<ul style="list-style-type: none"> • IdGrupoArticulo • CodigoGrupoArticulo • NombreGrupoArticulo 	El IdGrupoArticulo y el CodigoGrupoArticulo destacan de esta pantalla debido a que el id identifica al grupo de articulo y el código del grupo de artículo es el atributo que describe a que pertenece cada refacción automotriz a la sucursal que pertenece dicho almacén.
Registro de TipoMaterial	<ul style="list-style-type: none"> • IdTipMaterial • CodigoTipMaterial • NombreTipMaterial 	El IdTipMaterial y el codigoTipMaterial es lo más destacable de esta pantalla debido a que el id identifica al tipo de material y el código del tipo de material identifica a qué tipo de material pertenece.
Registro de Pedido	<ul style="list-style-type: none"> • IdTemporaryTableSolicitudes • NombreAlmacen • NombreSolicita • Material • TextoBreve • NotaCaja • Observaciones • CantidadPedido 	Dentro de esta interfaz destacan el IdTemporaryTableSolicitudes, NombreAlmacen, NombreSolicita, Material y la CantidadPedido, ya que el id de la solicitud identifica a la pieza que se está siendo solicitada, el nombre del almacén hace referencia a el almacén que está solicitando ese material, el nombre que lo solicita se refiere a la persona, el material es la pieza automotriz que se necesita y la cantidad de pedido se refiere a la cantidad que necesita ser surtida en el almacén.
Registro de Inventario	<ul style="list-style-type: none"> • IdTemporaryTableInventario • Centro • Material • Almacen • PlanNecesidad • Caracteristica • TipMaterial • GrupoArticulo • Division • Sector • Fecha • ValStockCosto • ValStockMoneda 	Destacan en esta interfaz los siguientes atributos: IdTemporaryTableInventario, Centro, Material, ValStockCosto, CantStockCantidad, StockConsiCantidad, ya que el id del inventario identifica a la pieza que hay en existencia en el inventario, el centro se refiere a la ubicación a la que pertenece el material, el material es el número de parte de la pieza automotriz con la cual es identificada, el valor de stock en costo se refiere al precio en que se fue adquirida la pieza, la cantidad de stock se refiere a la cantidad de material que se tiene en el almacén y el stock que se tiene en

	<ul style="list-style-type: none"> • CantStockCantidad • CantStockUnidad • StockConsiCantidad • StockConsiUnidad 	consideración se refiere que si ese material ya fue pedido por algún centro.
Registro de Usuario	<ul style="list-style-type: none"> • IdUsuario • NickUsuario • Contraseña • TipoUsuario 	En esta interfaz todo es importante ya que el id identifica al usuario que es registrado, el nick de usuario pertenece al nombre de usuario, la contraseña da acceso al sistema y el tipo de usuario se refiere al tipo de usuario registrado, que en este caso dentro del sistema solo existe el Administrador y el Invitado.

Al seleccionar en el menú la opción de “Consultas”, se puede hacer una consulta por clave de cada uno de los registros que se dan de alta, como se muestra en la Figura 63. En la cual se observa que al seleccionar una opción, se despliega la información que se encuentra dentro de ese registro.

Figura 63 Interfaz de consulta por clave (Elaboración propia).

Considerando que cada una de las interfaces son similares y proporcionan los datos que son buscados a partir del id con el cual es identificado cualquier registro, insertando de manera automática en las cajas de texto los atributos que la conforman, esto es para adquirir una visualización por número de registro de lo que se ha registrado y tomarlo a consideración del

usuario en caso de que quisiera saber sobre algún registro en especial, se muestra a continuación cada interfaz que compone el menú de Consultas ir a la Tabla 11.

Tabla 11 Descripción del menú de consultas de SIC SARF (Elaboración propia).

Descripción de menú de consultas	
Interfaz	Descripción
Almacén	El IdAlmacen en esta pantalla identifica el almacén, al seleccionar el id de registro, este nos mostrara en la caja de texto, el código del almacén que ha sido registrado por el usuario que es el más interesante de esta interfaz.
División	Dentro de esta interfaz el IdDivision identifica la división de la sucursal, al seleccionar el id, este nos mostrara en la caja de texto, el código de la división y el nombre de la sucursal que pertenece dicho almacén.
Centro	Con el IdCentro dentro del combo en esta pantalla, al seleccionarlo nos mostrara el código de centro que ha sido registrado y de la misma manera este nos visualizara la división a la que pertenece dicho centro ya sea refacciones, accesorios y body.
GrupoArticulo	El IdGrupoArticulo dentro de esta pantalla al ser seleccionado nos muestra dentro de las cajas de texto los atributos que fueron ingresados por el usuario mostrándonos el código del grupo al que pertenece dicho artículo.
TipoMaterial	Dentro de esta interfaz el IdTipMaterial identifica al tipo de material al que pertenece, al seleccionar el id, este nos mostrara en la caja de texto, el código al que pertenece el tipo de material.
Pedido	El IdTemporaryTableSolicitudes dentro de esta pantalla al ser seleccionado nos muestra dentro de las cajas de texto los atributos que fueron ingresados por el usuario mostrándonos el nombre del almacén que está solicitando un material, el nombre de la persona que lo solicita, el material que está solicitando el encargado del almacén y la cantidad que va a ser requerida.
Inventario	Destaca en esta interfaz el IdTemporaryTableInventario, ya que al ser seleccionado, reflejara atributos en las cajas de texto como el centro donde es ubicado el material, el número de parte del material que se tiene en almacén, el valor en pesos en el que se encuentra la pieza, la cantidad que se tiene es stock y si el material ha sido reservado por alguna otra sucursal.
Usuario	En esta interfaz al seleccionar el id que identifica al usuario, nos muestra el nombre de usuario que ha sido registrado y el tipo de usuario que registró, teniendo así un acceso al sistema como administrador o como invitado.

Al seleccionar el menú de “Reportes”, la búsqueda se presentan en un cuadro donde muestra una consulta que es realizada por el usuario, por un cierto rango considerado de fechas para poder obtener el reporte de los registros que se han dado de alta en ese periodo, como material existente y no existente.

Cabe destacar, que el administrador tiene acceso a la información del pedido, siendo solo el, quien pueda observar los registros de pedido en su totalidad, mostrando el reporte como se muestra en la Figura 64. En la cual del lado derecho, se observa la suma de la cantidad de las piezas solicitadas, en el lado izquierdo el reporte a detalle de quien lo está requiriendo, con la fecha que realizo el pedido. Además de que el administrador puede realizar una búsqueda por: Nombre_Almacén, Nombre_Solicita, Material, Cantidad_Pedido, generar datos en Excel para un mejor manejo de datos.

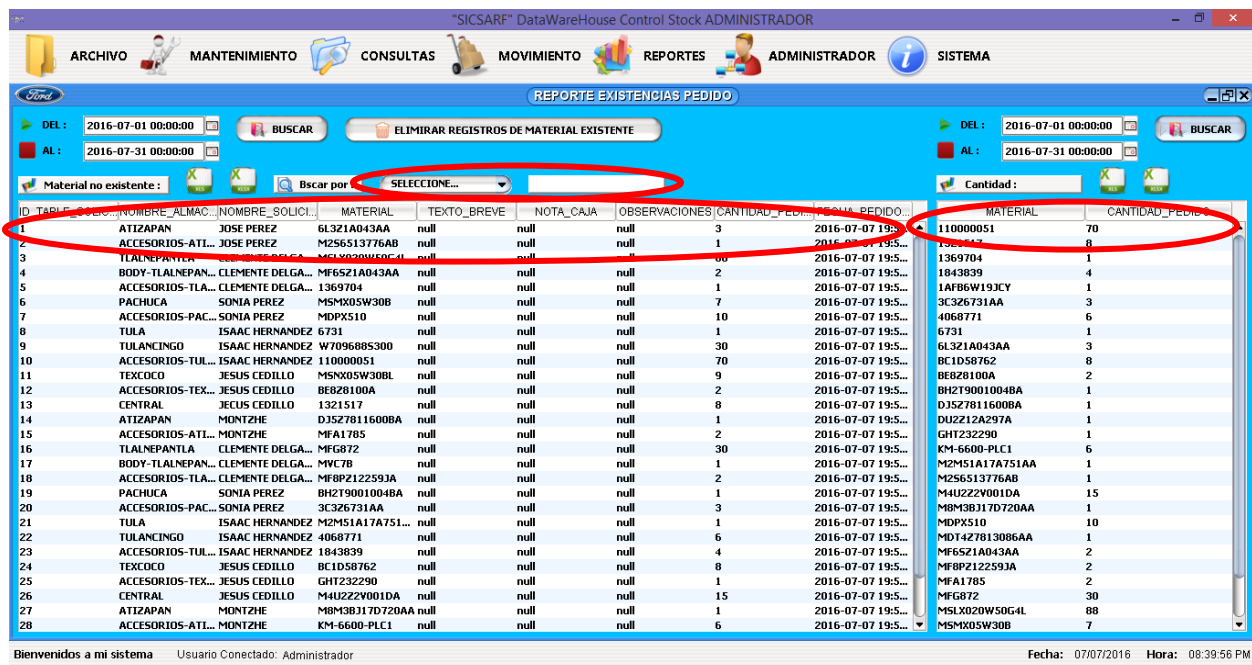


Figura 64 Interfaz de reporte de existencias (Elaboración propia).

Tomándose como consideración esta interfaz en la Tabla 12 se describe cada una que integra al menú de reportes, dependiendo del tipo de usuario que ingrese al sistema, esta información puede ser usada para poder buscar informes entre fechas de registro, haciendo que el administrador pueda realizar la toma de decisiones en cuanto al material que hay en existencia y el material que no existe.

Tabla 12 Descripción del menú de reportes de SICSARF (Elaboración propia).

Descripción de menú de reportes		
Tipo de usuario	Interfaz	Descripción
Invitado	Reporte Almacén	Dentro de esta pantalla se realiza una búsqueda entre un rango de fechas para poder revisar que almacén ha sido registrado en la base de datos.
	Reporte División	En esta interfaz se hace una búsqueda entre un rango de fechas para obtener que división ha sido dada de alta en la base de datos.

	Reporte Centro	Interfaz donde se realiza una búsqueda entre fechas que asigna el usuario, para poder obtener cada centro que ha sido registrado, además de obtener a que división es perteneciente.
	Reporte GrupoArticulo	Interfaz donde se hace una búsqueda entre un rango de fechas que desea visualizar el usuario, para obtener la información de que registros han sido realizados por los usuarios.
	Reporte TipoMaterial	En esta interfaz se hace una búsqueda entre un rango de fechas para obtener qué tipo de división ha sido dada de alta en la base de datos.
Administrador	Reporte Usuario	Dentro de esta interfaz se muestra un reporte de usuarios que han sido registrados, realizando una búsqueda por un rango de fechas para que el administrador pueda bloquear el acceso a cualquier usuario que ha sido dado de alta sin autorización del encargado del área de almacén.
	Reporte Pedido	En esta interfaz se obtiene un reporte de pedido que ha sido registrado diariamente, este reporte es buscado entre un rango de fechas, dependiendo lo que necesite el usuario para poder revisar el material que ha sido solicitado por cada almacén diariamente además de revisar los objetivos a los que se desea llegar dentro del área de compras.
	Reporte Inventario	En esta pantalla es realizada una búsqueda, ingresando la fecha inicial y final de búsqueda obteniendo como resultado un reporte de todo el inventario, obteniendo este reporte para uso del administrador para obtener cuanto ha ido disminuyendo el almacén en cada sucursal.
	Reporte Material Exist	Interfaz donde se realiza una búsqueda por un rango de fechas, mostrándonos en forma de reporte el material que existe dentro del inventario siendo este el material que no se solicita a Ford.
	Reporte Material Inexist	Dentro de esta interfaz se obtiene una búsqueda por un rango de fechas, la cual nos hace un reporte del material que no existe dentro del inventario de cada sucursal, esto es necesario ya que el administrador podrá tomar la decisión de adquirir más material que no esté registrado dentro del almacén.

A continuación, se dará mayor énfasis al menú que despliegan los informes que el usuario necesita, ya que son de mayor interés, gracias a la interfaz de búsqueda de piezas, se ha tenido un ahorro de tiempo considerable, que es una de las ventajas principales que tiene el sistema. Al seleccionar la opción del menú “Movimiento”, nos dirige a la pantalla de búsqueda de material solicitado dentro del material existente en el inventario, de acuerdo, al pedido que se está solicitando de cada sucursal, a continuación se explicara:

Dentro del menú de “Reportes”, “Consultas” y “Movimiento”, se tiene la opción de poder realizar una exportación e importación de Excel con extensión .xlsx o .xls, para que el administrador pueda tomar decisiones realizando los movimientos necesarios, de igual forma teniendo un respaldo de la información que ha obtenido SICSARF, una vez habiendo revisado el material no existente en cuanto a las compras de refacciones se podrá tener el material necesario en cada sucursal.

La exportación e importación son una de las cualidades del sistema que requirieron en el área de compras de refacciones, en la (Figura 65) se observa la obtención de la búsqueda de material existente, el cual es posteriormente utilizado para saber sobre el pedido, como de cuanto es lo que se pidió y cuál es el material que se queda estancado, esto por haber realizado una mala compra o haber realizado la búsqueda incorrecta.

	A	B	C	D	E	F	G	H	I	J	K
	SOLICITA	CENTRO	MATERIAL	CANTIDAD	COSTOUNITARIO	COSTOTOTAL	FECHA	STOCK	CODCENTROLOC	NOMBRECENTROLOC	STOCK
1	JOSE PEREZ	ATIZAPAN	6L3Z1A043AA	3	545.033333	1635.10	2016-07-07 19:59:21.0	3	R112	ACCESORIOS-ATIZAPAN	0
2	JOSE PEREZ	ATIZAPAN	6L3Z1A043AA	3	547.200000	1694.40	2016-07-07 19:59:21.0	2	R128	ACCESORIOS-TLALNEPANTLA	-1
3	JOSE PEREZ	ATIZAPAN	6L3Z1A043AA	3	545.033333	1635.10	2016-07-07 19:59:21.0	3	R213	ACCESORIOS-PACHUCA	0
4	JOSE PEREZ	ATIZAPAN	6L3Z1A043AA	3	547.200000	1641.60	2016-07-07 19:59:21.0	1	R220	TULANCINGO	-2
5	JOSE PEREZ	ATIZAPAN	6L3Z1A043AA	3	547.200000	1641.60	2016-07-07 19:59:21.0	3	R312	ACCESORIOS-TEXCOCO	0
6	JOSE PEREZ	ACCESORIOS-ATIZAPAN	M256513776AB	1	101.220000	101.22	2016-07-07 19:59:21.0	1	R310	TEXCOCO	0
7	CLEMENTE DELGADILLO	TLALNEPANTLA	MSLX020W50G4L	88	212.555069	61215.86	2016-07-07 19:59:21.0	288	R110	ATIZAPAN	200
8	CLEMENTE DELGADILLO	TLALNEPANTLA	MSLX020W50G4L	88	172.280000	10164.52	2016-07-07 19:59:21.0	59	R210	PACHUCA	-29
9	CLEMENTE DELGADILLO	TLALNEPANTLA	MSLX020W50G4L	88	172.280000	2584.20	2016-07-07 19:59:21.0	15	R220	TULANCINGO	-73
10	CLEMENTE DELGADILLO	TLALNEPANTLA	MSLX020W50G4L	88	null	0.00	2016-07-07 19:59:21.0	0	R310	TEXCOCO	-88
11	CLEMENTE DELGADILLO	BODY-TLALNEPANTLA	MF65Z1A043AA	2	null	0.00	2016-07-07 19:59:22.0	0	R110	ATIZAPAN	-2
12	CLEMENTE DELGADILLO	BODY-TLALNEPANTLA	MF65Z1A043AA	2	547.052500	2188.21	2016-07-07 19:59:22.0	4	R112	ACCESORIOS-ATIZAPAN	2
13	CLEMENTE DELGADILLO	BODY-TLALNEPANTLA	MF65Z1A043AA	2	547.073333	1641.22	2016-07-07 19:59:22.0	3	R128	ACCESORIOS-TLALNEPANTLA	1
14	CLEMENTE DELGADILLO	BODY-TLALNEPANTLA	MF65Z1A043AA	2	547.060000	547.06	2016-07-07 19:59:22.0	1	R213	ACCESORIOS-PACHUCA	-1
15	CLEMENTE DELGADILLO	BODY-TLALNEPANTLA	MF65Z1A043AA	2	null	0.00	2016-07-07 19:59:22.0	0	R310	TEXCOCO	-2
16	CLEMENTE DELGADILLO	BODY-TLALNEPANTLA	MF65Z1A043AA	2	547.050000	1094.10	2016-07-07 19:59:22.0	2	R312	ACCESORIOS-TEXCOCO	0
17	CLEMENTE DELGADILLO	ACCESORIOS-TLALNEPANTLA	1369704	1	69.050000	2831.05	2016-07-07 19:59:22.0	41	R120	TLALNEPANTLA	40
18	CLEMENTE DELGADILLO	ACCESORIOS-TLALNEPANTLA	1369704	1	69.050000	2762.00	2016-07-07 19:59:22.0	40	R210	PACHUCA	39
19	SONIA PEREZ	PACHUCA	MSMX05W30B	7	34.517498	40972.27	2016-07-07 19:59:22.0	1187	R110	ATIZAPAN	1180
20	SONIA PEREZ	PACHUCA	MSMX05W30B	7	34.387128	3232.39	2016-07-07 19:59:22.0	94	R120	TLALNEPANTLA	87
21	SONIA PEREZ	PACHUCA	MSMX05W30B	7	34.598121	53384.90	2016-07-07 19:59:22.0	1543	R210	PACHUCA	1536
22	SONIA PEREZ	PACHUCA	MSMX05W30B	7	34.570811	2558.24	2016-07-07 19:59:22.0	74	R214	TULA	67
23	SONIA PEREZ	PACHUCA	MSMX05W30B	7	34.570811	2558.24	2016-07-07 19:59:22.0	74	R214	TULA	67

Figura 65 Resultado de búsqueda en formato de Excel (elaboración propia).

Al crearse el documento de Excel con la información necesaria para que el administrador y en encargado del almacén comprueben las piezas que son solicitadas, de tal forma que el pedido a Ford sea realizado en un menor tiempo. El procedimiento para haber llegado a la exportación de este documento se menciona a continuación.

Cabe mencionar que, en el menú de movimiento, existe una interfaz de carga de datos, siendo el primer proceso, se realiza una búsqueda poder cargar los datos que se fueron importados a través de un Excel (Figura 66), donde se tiene guardado el pedido y el MC9, una vez teniendo cargados los datos dentro de la base de datos, se tiene la opción de cargarlos para poder realizar el proceso de búsqueda.

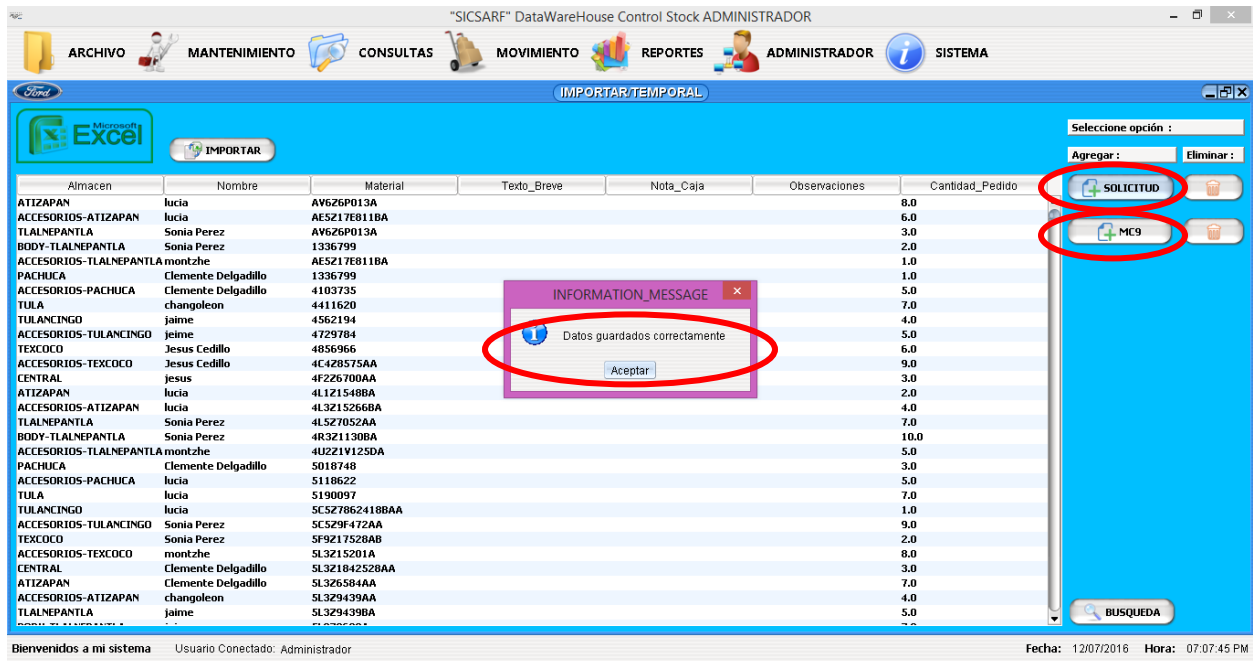


Figura 66 Registro de solicitud de pedido (Elaboración propia).

Al haber realizado el registro del pedido y del MC9, se realiza la búsqueda de material existente y no existente dentro de esta interfaz Figura 67, el cual muestra detalles específicos, realizando filtros de búsqueda de acuerdo a las necesidades del usuario, ya sea por, quien está solicitando la pieza, de que almacén se solicita, el material solicitado, el centro donde se encuentra y el nombre del mismo, siendo este un apoyo para la realización del pedido, ya que le facilita una búsqueda más detallada, tomando en cuenta que el precio unitario de cada pieza cambia en un tiempo determinado, aquí el encargado podrá obtener si esa pieza disminuyo o aumento de precio.

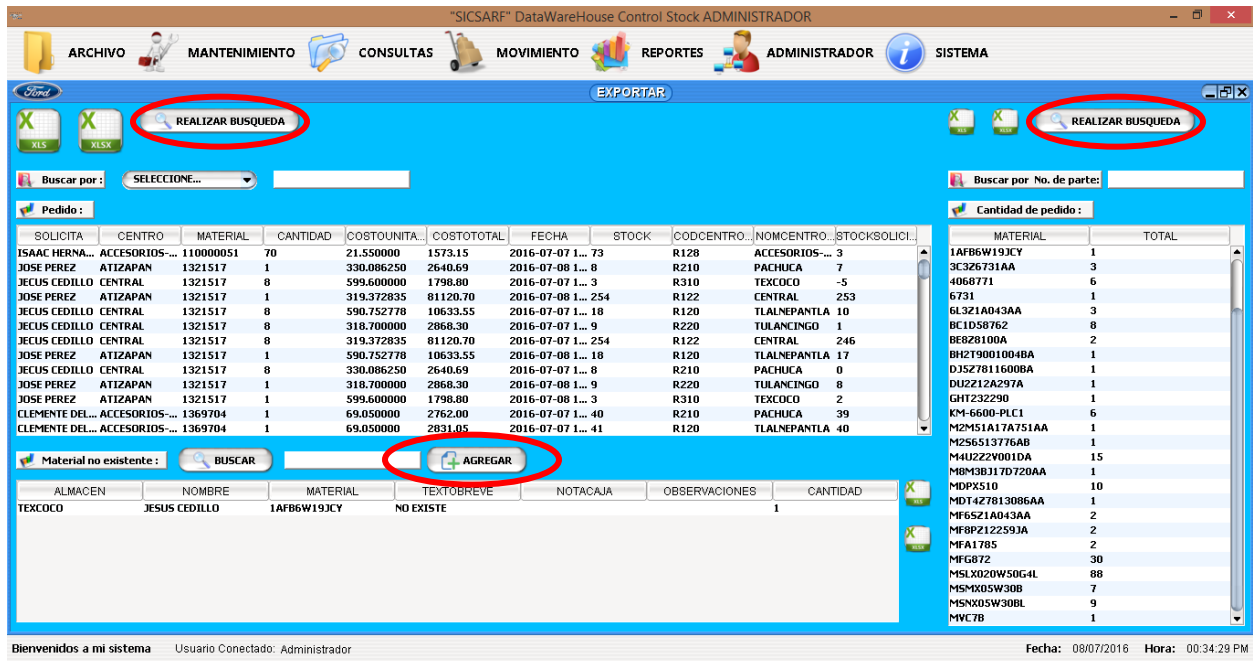


Figura 67 Exportación de material existente (Elaboración propia).

Además, con la opción de agregar las piezas fuera de inventario, el encargado de compras de refacciones, podrá tomar la decisión junto con el administrador del área de almacén, para saber si adquirir esa pieza en una cantidad mayor, siendo que algunos proveedores dan la posibilidad de adquirir más material por menor costo.

Uno de los resultados de SICSARF, es un almacén de datos (del inglés datawarehouse) con la información histórica del comportamiento de cada uno de los inventarios que posee la empresa Ford, así como la información histórica de cada una de las piezas que se han adquirido para estos almacenes, actualmente solo se obtiene una lista de estas piezas, para evitar la duplicidad y permitir un ahorro a la empresa Ford, pero el almacén de datos permite consultas futuras, referentes a las piezas más usadas en un almacén, con el fin de pedir las directamente al proveedor para estos almacenes y así evitar gastos relacionados con envíos entre estos.

El sistema SICSARF ha respondido adecuadamente a los requisitos que originaron su desarrollo. En el caso de la realización de búsqueda de refacciones específicas no mostrándole al usuario todas las partes existentes, mientras que en Excel les mostraba todas las refacciones e iban eliminando las que no eran de interés, además de poder seleccionar dentro de la búsqueda combos por el cual podría ubicar la sucursal que solicita, nombre de la persona, el material, centro donde se encuentra la pieza, el precio, y la cantidad de stock que se tiene.

En el caso de realizar el pedido a Ford, del material que se ubicó dentro de la búsqueda en SICSARF, el documento de Excel de material existente es enviado vía correo electrónico a cada encargado de almacén para que pueda visualizar adecuadamente la ubicación del material que solicito, siendo que de esta manera existe la posibilidad de que el material se tenga pero no con la cantidad adecuada, de igual forma se le agrega una observación en el email que la pieza será pedida de acuerdo a la cantidad que se necesite surtir.

Siendo el mismo caso en la realización del pedido, dentro de la página de Ford solo se realiza un proceso en el cual se debe de cargar el Excel con el material que se va a solicitar, siendo una forma mucho más sencilla, debido a que con SICSARF se puede obtener este documento de Excel con el material a solicitar ahorrándole al usuario tiempo para poder realizar el pedido de la forma adecuada y sin errores a la hora de capturar los datos en la página web.

Este sistema puede ser implementado en Mazda y Camiones, ya que pertenecen al grupo de Zapata, además de ser similares los procesos en que se realiza la búsqueda de material, por lo que le facilitaría el pedido a los encargados del área de compras de refacciones y en un menor tiempo, sin temor a errores, ya que el sistema está instalado dentro del área de refacciones de Ford Zapata S.A. de C.V., cumpliendo con las funciones requeridas.

El resultado obtenido sobre el manejo del datawarehouse, que proporciona el sistema SICSARF, es un ejemplo de utilizar un datawarehouse, pero no es el único, en un futuro se espera llegar a un convenio con Ford, para implementar nuevas consultas que le permitirán ahorros a la empresa, en el área de compra de refacciones.

VIII. Conclusiones y sugerencias

Después de analizar los datos obtenidos al hacer el proceso con el sistema SICSARF, se determina que está cumpliendo con el objetivo planteado en este proyecto, que es, disminuir de 3 horas a 15 minutos, ahorrándose un 91.7 % de tiempo en el proceso de búsqueda de material para llegar a la realización de pedido.

De acuerdo a la implementación de SICSARF, en el apartado de resultados el personal que está encargado de realizar el pedido dentro del área de refacciones, ha aceptado con agrado el sistema, adaptándose de manera sencilla, facilitándole al personal que trabajen con mayor rapidez y sin errores.

Debido a que SAP, permite solo el acceso su base de datos, es posible trabajar los datos de salida generando un nuevo uso, con SICSARF, es posible observar el comportamiento de la información histórica de cada almacén, así como las piezas que se han adquirido para cada uno de estos, evitando la duplicidad de material y permitir un ahorro para la empresa en el área de compras de refacciones. Además SICSARF con su manejo de datawarehouse en un futuro se desea implementar nuevas consultas que le permitan a Ford ahorrar tiempo y costos dentro del área de compras de refacciones en otras actividades.

Dentro del sistema SICSARF existe un repositorio (Datawarehouse), almacenando toda la información referente a la compra de refacciones, con el fin de aplicar nuevos programas para obtener información de este repositorio para brindar apoyo y ventaja estratégica hacia la empresa Ford.

IX. Referencias de consulta

- Astasio, J. C. (19 de 09 de 2005). *Mecalux*. Obtenido de <http://www.mecalux.com.mx/software-control-de-inventarios/easy-wms>
- Avalos, M. Y. (16 de 03 de 2010). *Plantilla de cálculos de Excel*. Obtenido de <http://cs.uns.edu.ar/materias/iocp/downloads/Apuntes/Unidad%204/Unidad%204%20-%20Excel.pdf>
- Bahit, E. (16 de 07 de 2011). *POO y MVC en PHP*. Obtenido de <http://eugeniabahit.blogspot.com/2011/07/poo-y-mvc-en-php.html>
- Bernal, P. M., & director de tesis, M. D. (2011). *Desarrolló de aplicaciones web con J2EE, implementando el patrón modelo vista controlador y el framework struts*. Toluca: El Autor.
- Bertino, E. (2001). *Sistemas de base de datos. Orientado a objetos: Conceptos y arquitecturas*. New York: Eddison-Wesley Iberoamericana.
- Booch, G. (2005). *The unified modeling language user guide* (Segunda ed.). USA: Addison Wesley.
- Cairó, O. (2010). *Estructuras de datos* (Tercera ed.). México: McGraw-Hill Interamericana.
- Castaño, A. d. (2006). *Concepción y diseño de base de datos : Del modelo e/r al modelo relacional*. Massachusetts: Addison-wesley iberoamericana.
- Ceballos, S. F. (2007). *Java 2 : Lenguaje y aplicaciones* (Primera ed.). México: Alfaomega.
- Dean, S. J. (2009). *Introducción a la programación con JAVA* (Primera ed.). México: McGraw-Hill.
- Edwards, C. (2001). *Fundamentos de sistemas de información*. España: Prentice-Hall.
- Fon, S. (2008). *Building and Maintaining a data warehouse* (Primera ed.). Toluca: CRC Press.
- Ford, H. (09 de Abril de 2015). *Ford*. Obtenido de http://es.wikipedia.org/wiki/Ford_Motor_Company

- Froufe, Q. A. (2006). *Java 2 Manual de usuario y tutorial* (Cuarta ed.). Madrid: Alfaomega.
- Gómez, V. A. (2010). *Sistemas de Información: Herramientas practicas para la gestión* (Tercera ed.). México: Alfaomega.
- Hansen, G. W. (2002). *Diseño y administración de bases de datos* (Décima ed.). Madrid: Prentice.
- Harold, P. J. (22 de 06 de 2010). *DataWareHouse*. Obtenido de http://Harold.udlap.mx/u_dl_a/tales/documentos/lis/ydirin_p_mm/capitulo2.pdf
- Jacobson, I. (2000). *Proceso unificado de desarrolló de software* (Primera ed.). Madrid: Addison Wesley.
- Kcoenke, M. D. (2003). *Procesamiento de bases de datos, fundamentos, diseño e implementación* (Octava ed.). México: Pearson.
- Kendall, S. A. (2006). *UML* (Primera ed.). México: Wesley Longman.
- Kenneth, C. L. (2008). *Sistemas de información gerencial* (Doceava ed.). Madrid: Prentice Hall.
- Kenneth, E. K. (2005). *Análisis y diseño de sistemas* (Sexta ed.). México: Pearson.
- Kourchenko, B. S. (2001). *Introducción a los sistemas de bases de datos* (Séptima ed.). México: Pearson Educación.
- Mackin, J. C. (2007). *Microsoft SQL: Diseño de una infraestructura de servidor de bases de datos* (Segunda ed.). Madrid: Anaya Multimedia.
- Pérez, L. C. (2003). *MySQL para Windows y Linux* (Segunda ed.). Estados Unidos: Alfaomega.
- Post, G. V. (2008). *Sistemas de administración de bases de datos* (Tercera ed.). México: McGraw-Hill.
- Potencier, Z. F. (17 de 02 de 2002). *Symfony 1.4, la guía definitiva*, 1.0. Obtenido de http://librosweb.es/libro/symfony_1_4/
- Pressman, S. R. (2005). *Ingeniería de software un enfoque práctico* (Sexta ed.). México: Thomson.

- Raghu, R. (2007). *Sistemas de gestión de bases de datos* (Tercera ed.). Madrid: McGraw-Hill.
- Ramez, E. (2007). *Fundamentos de sistemas de bases de datos* (Quinta ed.). Madrid: Pearson.
- Richard, C. W. (26 de 06 de 2007). *Data Warehouse Appliances*. Obtenido de <http://todobi.blogspot.mx/2007/10/tipos-de-data-warehouse-appliances.html>
- Rod, S. (2009). *Diseño de bases de datos* (Primera ed.). Madrid: Anaya.
- Sánchez, A. J. (2009). *Programación en Java* (Tercera ed.). Madrid: McGraw-Hill.
- Schach, S. R. (2006). *Ingeniería de software clásica y orientada a objetos* (Primera ed.). México: McGraw-Hill.
- Senn, J. (2008). *Análisis y diseño de sistemas de información* (Segunda ed.). DF, México: McGraw-Hill.
- Spona, H. (2010). *Programación de bases de datos con MySQL y PHP* (Primera ed.). México: Alfaomega.
- Turban, E. M. (2001). *Tecnologías de la información para la administración* (Primera ed.). México: Cecea.
- Velthuis, M. P. (2000). *Diseño de bases de datos relacionales* (Primera ed.). México: Alfaomega.
- Velthuis, M. P. (21 de 10 de 2012). *Calidad de sistemas de información* (Segunda ed.). México: Alfaomega.
- Village, G. C. (19 de 05 de 2011). *NCH Software*. Obtenido de <http://www.nchsoftware.com/inventory/es/>
- Weaver, R. (s.f.). *Symfony 2.4, el libro oficial*, 2.0. Recuperado el 19 de 01 de 2016, de http://librosweb.es/libro/symfony_2_4/
- Weitzenfeld, A. (2005). *Ingeniería de software orientada a objetos con UML, JAVA e INTERNET* (Primera ed.). México: Thomson.

Weitzenfeld, A. R., & Guardati, B. S. (15 de 01 de 2008). *Desarrollo de software*. Obtenido de <http://weitzenfeld.robolat.org/wp-content/uploads/2015/01/WeitzenfeldGuardatiComputacion2008.pdf>

Whitten, J. I. (2008). *Análisis de sistemas diseño y métodos* (Séptima ed.). México: McGraw-Hill.

Wu, C. T. (2005). *Programación en Java: Introducción a la programación orientada a objetos* (Primera ed.). México: McGraw-Hill.

X. Anexos

Anexo I

Primera entrevista realizada a la encargada del área de compras de refacciones.

Entrevistado(a): Claudia Álvarez Soto Cargo: Encargada del área de compras Fecha: 04 de Enero del 2016 Hora: 10:00 a.m. Lugar: Ford Zapata de Tlalnepantla (Área de refacciones). Tema: Proceso de compra de refacciones.		
Tiempo	Preguntas	Respuestas
3 min	Presentación y objetivo del tema a tratar.	
6 min	¿Cómo se lleva a cabo una o más compras de piezas automotrices?	
3 min	¿Quién hace y como se es llevado el proceso de compra de refacciones?	
3 min	¿Cuánto es el tiempo que necesita para realizar la búsqueda del material dentro del inventario?	
2 min	¿Cuál es el tiempo definido para poder realizar el pedido a Ford?	
3 min	¿El formato de solicitud de pedido cumple con los requerimientos que necesita?	
2 min	¿Cuál es el proceso que considera más tardado en realizar una compra?	
2 min	¿Qué tipo de archivo es el que necesita para poder realizar la búsqueda de material?	
3 min	¿Cómo guarda la información, acerca de las compras realizadas diariamente?	
1 min	¿Es necesario guardar la información del material solicitado?	
7 min	¿Cómo se llevan a cabo los reportes de pedido en un tiempo determinado?	
2 min	¿Ha intentado realizar el procedimiento de otra forma en la cual pueda realizar la búsqueda con más facilidad?	
4 min	¿Por qué considera que un sistema de información, facilitaría sus procesos en realizar el pedido diario?	
41 min	Tiempo de entrevista	

Anexo II

Segunda entrevista realizada a la encargada de compras de refacciones

Entrevistado(a): Claudia Álvarez Soto Cargo: Encargada del área de compras Fecha: 15 de Febrero del 2016 Hora: 10:00 a.m. Lugar: Ford Zapata de Tlalnepantla (Área de refacciones). Tema: Proceso de compra de refacciones.		
Tiempo	Preguntas	Respuestas
3 min	Presentación y objetivo del tema a tratar.	
3 min	¿Cómo se lleva a cabo la descarga de la información para realizar la búsqueda de material?	
1 min	¿Una vez añadidas las piezas solicitadas a Ford, pueden hacerse algunas modificaciones?	
7 min	¿En qué momento decide en comprar o no, el material solicitado por cada encargado de almacén?	
3 min	¿Existen documentos de compra, de los cuales desea guardar alguna información?	
2 min	¿Los proveedores manejan crédito con la empresa?	
5 min	¿Al realizar la compra, que documentación debe ser almacenada? ¿Por qué?	
6 min	¿Cómo son localizados los almacenes, sucursales, material, el precio, quien solicita el material entre otros?	
2 min	¿Qué información es la que requiere, para disminuir el tiempo en la compra de refacciones automotrices, con el proveedor?	
32 min	Tiempo de entrevista	

Anexo III

Código para la creación de la base de datos del sistema en MySQL.

```
CREATE DATABASE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`;
CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`Pieza` (
  `idPieza` INT(11) NOT NULL AUTO_INCREMENT,
  `material` VARCHAR(45) NOT NULL,
  `planNecesidad` VARCHAR(45) NULL DEFAULT NULL,
  `caracteristica` VARCHAR(2) NOT NULL,
  `sector` VARCHAR(2) NOT NULL,
  `fecha` DATE NOT NULL,
  `valStockCosto` DECIMAL(18,2) NOT NULL,
  `valStockMoneda` VARCHAR(3) NOT NULL,
  `cantStockCantidad` DECIMAL(18,0) NOT NULL,
  `cantStockUnidad` VARCHAR(2) NOT NULL,
  `stockConsiCantidad` DECIMAL(18,0) NOT NULL,
  `stockConsiUnidad` VARCHAR(2) NOT NULL,
  `idCentro` INT(11) NOT NULL,
  `idAlmacen` INT(11) NOT NULL,
  `idDivision` INT(11) NOT NULL,
  `idGrupoArticulo` INT(11) NOT NULL,
  `idTipMaterial` INT(11) NOT NULL,
  `fechaRegistroPieza` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`idPieza`),
  INDEX `fk_Pieza_Centro_idx` (`idCentro` ASC),
  INDEX `fk_Pieza_Almacen_idx` (`idAlmacen` ASC),
  INDEX `fk_Pieza_Division_idx` (`idDivision` ASC),
  INDEX `fk_Pieza_GrupoArticulo_idx` (`idGrupoArticulo` ASC),
  INDEX `fk_Pieza_TipMaterial_idx` (`idTipMaterial` ASC),
  CONSTRAINT `fk_Pieza_Centro`
    FOREIGN KEY (`idCentro`)
    REFERENCES `BD_FORD_ZAPATA_COMPRAS`.`Centro` (`idCentro`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pieza_Almacen`
    FOREIGN KEY (`idAlmacen`)
    REFERENCES `BD_FORD_ZAPATA_COMPRAS`.`Almacen` (`idAlmacen`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pieza_Division`
    FOREIGN KEY (`idDivision`)
    REFERENCES `BD_FORD_ZAPATA_COMPRAS`.`Division` (`idDivision`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pieza_GrupoArticulo`
    FOREIGN KEY (`idGrupoArticulo`)
    REFERENCES `BD_FORD_ZAPATA_COMPRAS`.`GrupoArticulo` (`idGrupoArticulo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pieza_TipMaterial`
    FOREIGN KEY (`idTipMaterial`)
    REFERENCES `BD_FORD_ZAPATA_COMPRAS`.`TipMaterial` (`idTipMaterial`)
```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`Centro` (
  `idCentro` INT(11) NOT NULL AUTO_INCREMENT,
  `codigoCentro` VARCHAR(4) NOT NULL,
  `nombreCentro` VARCHAR(45) NULL DEFAULT NULL,
  `idDivision` INT(11) NOT NULL,
  `fechaRegistroCentro` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`idCentro`),
  INDEX `fk_Centro_Division_idx` (`idDivision` ASC),
  INDEX `codigoCentro` (`codigoCentro` ASC),
  CONSTRAINT `fk_Centro_Division`
  FOREIGN KEY (`idDivision`)
  REFERENCES `BD_FORD_ZAPATA_COMPRAS`.`Division` (`idDivision`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`Almacen` (
  `idAlmacen` INT(11) NOT NULL AUTO_INCREMENT,
  `codigoAlmacen` VARCHAR(4) NOT NULL,
  `nombreAlmacen` VARCHAR(45) NULL DEFAULT NULL,
  `fechaRegistroAlmacen` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`idAlmacen`));

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`TipMaterial` (
  `idTipMaterial` INT(11) NOT NULL AUTO_INCREMENT,
  `codigoTipMaterial` VARCHAR(4) NOT NULL,
  `nombreTipMaterial` VARCHAR(45) NULL DEFAULT NULL,
  `fechaRegistroTipMaterial` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`idTipMaterial`));

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`GrupoArticulo` (
  `idGrupoArticulo` INT(11) NOT NULL AUTO_INCREMENT,
  `codigoGrupoArticulo` VARCHAR(6) NOT NULL,
  `nombreGrupoArticulo` VARCHAR(45) NULL DEFAULT NULL,
  `fechaRegistroGrupoArticulo` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`idGrupoArticulo`));

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`Division` (
  `idDivision` INT(11) NOT NULL AUTO_INCREMENT,
  `codigoDivision` VARCHAR(4) NOT NULL,
  `nombreDivision` VARCHAR(45) NULL DEFAULT NULL,
  `fechaRegistroDivision` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`idDivision`));

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`TemporaryTableSolicitudes` (
  `idTemporaryTableSolicitudes` INT(11) NOT NULL AUTO_INCREMENT,
  `Almacen` VARCHAR(45) NOT NULL,
  `nombre` VARCHAR(45) NOT NULL,
  `material` VARCHAR(45) NOT NULL,
  `textoBreve` VARCHAR(45) NULL DEFAULT NULL,

```

```

`notaCaja` VARCHAR(45) NULL DEFAULT NULL,
`observaciones` VARCHAR(45) NULL DEFAULT NULL,
`cantidadPedido` INT(11) NOT NULL,
`fechaPedido`    TIMESTAMP    NULL    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (`idTemporyTableSolicitudes`),
INDEX `material` (`material` ASC);

```

```

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`TemporyTableInventario` (
  `idTemporyTableInventario` INT(11) NOT NULL AUTO_INCREMENT,
  `centro` VARCHAR(4) NOT NULL,
  `material` VARCHAR(45) NOT NULL,
  `Almacen` VARCHAR(4) NOT NULL,
  `planNecesidad` VARCHAR(45) NULL DEFAULT NULL,
  `caracteristica` VARCHAR(2) NOT NULL,
  `tipMaterial` VARCHAR(4) NOT NULL,
  `grupoArticulo` VARCHAR(6) NOT NULL,
  `division` VARCHAR(4) NOT NULL,
  `sector` VARCHAR(2) NOT NULL,
  `mes` DATE NOT NULL,
  `valStockCosto` DECIMAL(18,2) NOT NULL,
  `valStockMoneda` VARCHAR(3) NOT NULL,
  `cantStockCantidad` DECIMAL(18,0) NOT NULL,
  `cantStockUnidad` VARCHAR(2) NOT NULL,
  `stockConsiCantidad` DECIMAL(18,0) NOT NULL,
  `stockConsiUnidad` VARCHAR(2) NOT NULL,
  `fechaConsulta`    TIMESTAMP    NULL    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (`idTemporyTableInventario`),
INDEX `material` (`material` ASC),
INDEX `centro` (`centro` ASC);

```

```

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`Usuario` (
  `idUsuario` INT(11) NOT NULL AUTO_INCREMENT,
  `nickUsuario` VARCHAR(45) NULL DEFAULT NULL,
  `passwordUsuario` VARCHAR(45) NULL DEFAULT NULL,
  `tipoUsuario` VARCHAR(15) NULL DEFAULT NULL,
  `fechaRegistro`    TIMESTAMP    NULL    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (`idUsuario`));

```

```

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`TableSolicitudes` (
  `idTableSolicitudes` INT(11) NOT NULL AUTO_INCREMENT,
  `Almacen` VARCHAR(45) NOT NULL,
  `nombre` VARCHAR(45) NOT NULL,
  `material` VARCHAR(45) NOT NULL,
  `textoBreve` VARCHAR(45) NULL DEFAULT NULL,
  `notaCaja` VARCHAR(45) NULL DEFAULT NULL,
  `observaciones` VARCHAR(45) NULL DEFAULT NULL,
  `cantidadPedido` INT(11) NOT NULL,
  `fechaPedido`    TIMESTAMP    NULL    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (`idTableSolicitudes`),
INDEX `material` (`material` ASC);

```

```

CREATE TABLE IF NOT EXISTS `BD_FORD_ZAPATA_COMPRAS`.`TableSolicitudesSinExistencias` (

```

```
`idTableSolicitudesInex` INT(11) NOT NULL AUTO_INCREMENT,  
`Almacen` VARCHAR(45) NOT NULL,  
`nombre` VARCHAR(45) NOT NULL,  
`material` VARCHAR(45) NOT NULL,  
`textoBreve` VARCHAR(45) NULL DEFAULT NULL,  
`notaCaja` VARCHAR(45) NULL DEFAULT NULL,  
`observaciones` VARCHAR(45) NULL DEFAULT NULL,  
`cantidadPedido` INT(11) NOT NULL,  
`fechaPedido`    TIMESTAMP    NULL    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE  
CURRENT_TIMESTAMP,  
PRIMARY KEY (`idTableSolicitudesInex`),  
INDEX `material` (`material` ASC));
```

Anexo IV

Códigos del sistema en NetBeans

Código del paquete: com.fordzapatacompras.modelo.dao

```
package com.fordzapatacompras.modelo.dao;
import com.fordzapatacompras.modelo.dto.AlmacenDTO;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

public class AlmacenDAO {
private static final String SQL_INSERT =
    "INSERT INTO
Almacen(idAlmacen,codigoAlmacen,nombreAlmacen,fechaRegistroAlmacen)VALUES(?,?,?,current_timestamp)";
private static final String SQL_SELECT =
    "SELECT idAlmacen,codigoAlmacen,nombreAlmacen FROM Almacen where idAlmacen=?";
private static final String SQL_SELECT_ALL =
    "SELECT idAlmacen,codigoAlmacen,nombreAlmacen FROM Almacen";
private static final String SQL_UPDATE = "UPDATE Almacen SET
codigoAlmacen=?,nombreAlmacen=?,fechaRegistroAlmacen=current_timestamp WHERE idAlmacen=?";
/*SQL to delete data*/
private static final String SQL_DELETE =
    "DELETE FROM Almacen WHERE idAlmacen = ?";
public void create(AlmacenDTO dto, Connection conn) throws
SQLException {

PreparedStatement ps =null;
try {
ps = conn.prepareStatement(SQL_INSERT);
ps.setInt(1,dto.getIdAlmacen());
ps.setString(2,dto.getCodigoAlmacen());
ps.setString(3,dto.getNombreAlmacen());
ps.executeUpdate();
} finally {
cerrar(ps);
cerrar(conn);
}
}

public AlmacenDTO load(AlmacenDTO dto,Connection conn)throws
SQLException{

PreparedStatement ps = null;
ResultSet rs = null;
try{
ps = conn.prepareStatement(SQL_SELECT);
ps.setInt(1,dto.getIdAlmacen());
rs = ps.executeQuery();
```

```

    List results = getResults(rs);
    if(results.size()>0){
        return (AlmacenDTO)results.get(0);
    } else{
        return null;
    }
} finally {
    cerrar(rs);
    cerrar(ps);
    cerrar(conn);
}
}
public List loadAll(Connection conn)throws SQLException{
    PreparedStatement ps = null;
    ResultSet rs = null;
    try {
        ps = conn.prepareStatement(SQL_SELECT_ALL);
        rs = ps.executeQuery();
        List results = getResults(rs);
        if(results.size()>0){
            return results;
        } else {
            return null;
        }
    } finally{
        cerrar(rs);
        cerrar(ps);
        cerrar(conn);
    }
}
public void update(AlmacenDTO dto, Connection conn)throws
SQLException{
    PreparedStatement ps = null;
    try{
        ps = conn.prepareStatement(SQL_UPDATE);
        ps.setString(1, dto.getCodigoAlmacen());
        ps.setString(2,dto.getNombreAlmacen());
        ps.setInt(3,dto.getIdAlmacen());
        ps.executeUpdate();
    } finally{
        cerrar(ps);
        cerrar(conn);
    }
}
public void delete(AlmacenDTO dto, Connection conn)throws
SQLException{
    PreparedStatement ps = null;
    try{
        ps = conn.prepareStatement(SQL_DELETE);
        ps.setInt(1,dto.getIdAlmacen());
        ps.executeUpdate();
    } finally{
        cerrar(ps);
        cerrar(conn);
    }
}
}

```

```

private List getResult(ResultSet rs) throws SQLException {
    List results = new ArrayList();
    while(rs.next()){
        AlmacenDTO dto = new AlmacenDTO();
        dto.setIdAlmacen(rs.getInt("idAlmacen"));
        dto.setCodigoAlmacen(rs.getString("codigoAlmacen"));
        dto.setNombreAlmacen(rs.getString("nombreAlmacen"));
        results.add(dto);
    }

    return results;
}

private void cerrar(PreparedStatement ps) throws SQLException {
    if(ps != null){
        try {
            ps.close();
        } catch(SQLException e){
        }
    }
}

private void cerrar(ResultSet rs){
    if(rs != null){
        try{
            rs.close();
        } catch (SQLException e){
        }
    }
}

private void cerrar(Connection cnn){
    if(cnn != null){
        try{
            cnn.close();
        } catch (SQLException e){
        }
    }
}

```


Código del paquete: com.fordzapatacompras.modelo.dto

```
package com.fordzapatacompras.modelo.dto;
import java.io.Serializable;

public class AlmacenDTO implements Serializable {
    private int idAlmacen;
    private String codigoAlmacen;
    private String nombreAlmacen;

    public AlmacenDTO(){
    }
    public int getIdAlmacen(){
    return idAlmacen;
    }
    public void setIdAlmacen(int idAlmacen) {
    this.idAlmacen=idAlmacen;
    }
    public String getCodigoAlmacen(){
    return codigoAlmacen;
    }
    public void setCodigoAlmacen(String codigoAlmacen) {
    this.codigoAlmacen=codigoAlmacen;
    }
    public String getNombreAlmacen(){
    return nombreAlmacen;
    }
    public void setNombreAlmacen(String nombreAlmacen) {
    this.nombreAlmacen=nombreAlmacen;
    }

    @Override
    public String toString(){
    StringBuilder sb = new StringBuilder();
    sb.append("Id Almacen ").append(getIdAlmacen()).append("\n");
    sb.append("Codigo Almacen").append(getCodigoAlmacen()).append("\n");
    sb.append("Nombre Almacen").append(getNombreAlmacen()).append("\n");
    return sb.toString();
    }
}
```

Código del paquete: com.fordzapatacompras.modelo.delegate

```
package com.fordzapatacompras.modelo.delegate;
import com.fordzapatacompras.modelo.dto.AlmacenDTO;
import com.fordzapatacompras.modelo.facade.AlmacenFacade;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.List;

public class AlmacenDelegate {
    private Connection cnn;
    private AlmacenFacade almFacade;

    public AlmacenDelegate(){
        String user ="root";
        String pwd ="crisian";
        String url ="jdbc:mysql://localhost:3306/bd_ford_zapata_compras";
        String mysqlDriver = "com.mysql.jdbc.Driver";
        try {
            Class.forName(mysqlDriver);
            cnn = DriverManager.getConnection(url, user, pwd);
        } catch (Exception e){
            e.printStackTrace();
        }
        almFacade = new AlmacenFacade(cnn);
    }
    //Codigo para los articulos
    public void crearAlmacen(AlmacenDTO dto) throws SQLException{
        almFacade.crear(dto);
    }
    public List listarAlmacen() throws SQLException {
        return almFacade.listar();
    }
    public AlmacenDTO leerAlmacen (AlmacenDTO dto) throws SQLException {
        return almFacade.leer(dto);
    }
    public void actualizar(AlmacenDTO dto) throws SQLException {
        almFacade.actualizar(dto);
    }
    public void elimina(AlmacenDTO dto) throws SQLException {
        almFacade.elimina(dto);
    }

    /* public void actualiza(ArticuloDTO dto) { }*/
}
```

Código del paquete: com.fordzapatacompras.modelo.facade

```
package com.fordzapatacompras.modelo.facade;
import com.fordzapatacompras.modelo.dao.AlmacenDAO;
import com.fordzapatacompras.modelo.dto.AlmacenDTO;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;

public class AlmacenFacade {
    private Connection cnn;
    private AlmacenDAO dao;

    public AlmacenFacade(Connection cnn){
        this.cnn = cnn;
        dao = new AlmacenDAO();
    }
    public void crear(AlmacenDTO dto) throws SQLException {
        dao.create(dto, cnn);
    }
    public List listar() throws SQLException {
        return dao.loadAll(cnn);
    }
    public AlmacenDTO leer(AlmacenDTO dto) throws SQLException {
        return dao.load(dto, cnn);
    }
    public void actualizar(AlmacenDTO dto) throws SQLException {
        dao.update(dto, cnn);
    }
    public void elimina(AlmacenDTO dto) throws SQLException {
        dao.delete(dto, cnn);
    }
}
```