



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE  
MÉXICO  
CENTRO UNIVERSITARIO UAEM TEXCOCO**

**“SISTEMA DE INFORMACIÓN PARA LA VENTA  
DE REFACCIONES AUTOMOTRICES”**

**TESIS**

QUE PARA OBTENER EL TITULO DE  
LICENCIADO EN INFORMÁTICA  
ADMINISTRATIVA

PRESENTA:

ALMARAZ RAMÍREZ ÓSCAR

*DIRECTOR DE TESIS:  
DR. ADRIAN TRUEBA ESPINOSA*

*REVISORES:  
LIC. FABIOLA MARTÍNEZ MEJÍA  
MC. JOSE SERGIO RUÍZ CASTILLA  
MC. ÁNGEL RAFAEL QUINTOS RAMÍREZ*

TEXCOCO, ESTADO DE MÉXICO

septiembre 2012.

Texcoco, México a 22 agosto de 2012.

M. EN C. JUAN MANUEL MUÑOZ ARAUJO  
SUBDIRECTOR ACADEMICO DEL  
CENTRO UNIVERSITARIO UAEM  
TEXCOCO  
P R E S E N T E:

AT'N M. EN F. GUADALUPE LIZETH ARCE CHÁV  
RESPONSABLE DEL DEPARTAMENTO DE TITULACIÓN

Con base en las revisiones efectuadas al trabajo escrito titulado "SISTEMA DE INFORMACIÓN PARA LA VENTA DE REFACCIONES AUTOMOTRICES" que para obtener el título de licenciado en INFORMATICA ADMINISTRATIVA presenta el sustentante ALMARAZ RAMIREZ OSCAR, con numero de cuenta 0524709 respectivamente, se concluye que cumple con los requisitos teórico- metodológicos necesarios para su aprobación, pudiendo continuar con la etapa de impresión del trabajo escrito.

ATENTAMENTE



M. EN C.A. JOSÉ SERGIO  
RUÍZ CASTILLA  
REVISOR



M. EN C.C. ÁNGEL RAFAEL  
QUINTOS RAMIREZ REVISOR

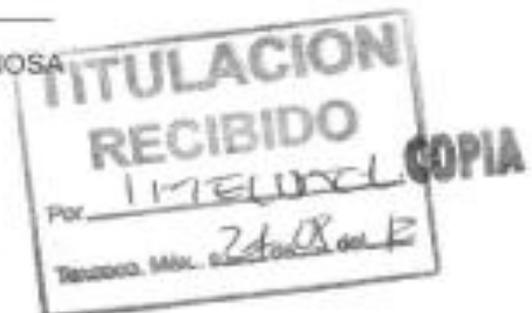


L. EN I.A. FABIOLA MARTÍNEZ  
MEJÍA REVISOR



DR. ADRIÁN TRUEBA ESPINOSA  
DIRECTOR

c.c.p. Sustentante -ALMARAZ RAMIREZ OSCAR  
c.c.p. Director -DR. ADRIAN TRUEBA ESPINOSA  
c.c.p. Titulación -



## CONTENIDO

<b>I. INTRODUCCIÓN</b> .....	12
<b>II. PLANTEAMIENTO DEL PROBLEMA</b> .....	14
<b>III. JUSTIFICACIÓN</b> .....	17
<b>IV. OBJETIVOS</b> .....	19
4.1.- OBJETIVO GENERAL.....	19
4.2.- OBJETIVOS PARTICULARES .....	19
<b>V. MARCO TEÓRICO</b> .....	20
5.1.- MODELO UML.....	20
5.1.1.- Los casos de uso.....	21
5.1.2.- Diagrama de casos de uso .....	23
5.1.3.- Elementos del diagrama de casos de uso .....	24
5.1.3.1.- Actor .....	24
5.1.3.2.- Relaciones.....	26
5.1.3.2.1.- Asociación .....	26
5.1.3.2.2.- Extensión.....	27
5.1.3.2.3.- Usos .....	27
5.1.3.2.4.- Dependencia .....	28
5.1.3.2.5.- Herencia .....	29
5.2.- SISTEMAS DE INFORMACIÓN .....	29
5.2.1.- Funciones principales de los sistemas de información (SI) .....	32

5.2.2.- Clasificación de los SI .....	32
5.2.2.1.- Sistemas de procesamiento de transacciones (TPS) .....	33
5.2.2.2.- Sistemas de automatización de oficinas (OAS) y sistemas de manejo de conocimiento (KWS) .....	33
5.2.2.3.- Sistemas de información gerencial (SIG o MIS) .....	34
5.2.2.4.- Sistemas de apoyo a la toma de decisiones (DSS) .....	34
5.2.2.5.- Sistemas expertos e inteligencia artificial (AI) .....	35
5.2.2.6.- Sistemas de soporte a las decisiones en grupo (DGSS) .....	36
5.2.2.7.- Sistemas de información ejecutiva (EIS) .....	36
5.2.3.- Análisis de sistemas .....	36
5.3.- BASE DE DATOS .....	37
5.3.1.- Antecedentes .....	37
5.3.2.- Definición de base de datos .....	38
5.3.3.- Estructura de base de datos .....	38
5.3.4.- Modelado de base de datos .....	39
5.3.5.- Estructura del Modelo Relacional .....	41
5.4.- BASES DE DATOS RELACIONALES .....	43
5.4.1.- Modelo entidad relación .....	44
5.4.1.1.- Entidades .....	44
5.4.1.2.- Atributos .....	45
5.4.1.3.- Claves primarias .....	48
5.4.1.4.- Relaciones .....	50

5.4.1.5.- Conectividad y Cardinalidad .....	51
5.5.- SQL .....	53
5.5.1.- Lenguaje de definición de datos (DDL) .....	54
5.5.1.1.- Definición de una base de datos a través MySQL .....	55
5.5.1.1.1.- Como crear una base de datos .....	55
5.5.1.1.2. - Como crear una tabla .....	56
5.5.1.1.3.- Definición de valores nulos.....	59
5.5.1.1.4.- Definición de valores por defecto .....	59
5.5.1.1.5.- Definición de claves primarias .....	60
5.5.1.1.6.- Definición de columnas autoincrementadas .....	61
5.5.1.1.7.- Definición de índices .....	61
5.5.1.1.8.- Definición de claves foráneas.....	63
5.5.1.1.9.- Como eliminar una tabla.....	65
5.5.1.1.10.- Como eliminar una base de datos .....	66
5.5.2.- Lenguaje de manipulación de datos (DML) .....	66
5.5.2.1.- Manipulación de datos a través de MySQL .....	67
5.5.2.1.1.- Consultar datos .....	68
5.5.2.1.2.- Agrupar datos .....	69
5.5.2.1.3.- Insertar datos.....	71
5.5.2.1.4.- Reemplazar datos .....	75
5.5.2.1.5.- Eliminar registros.....	77
5.5.2.1.6.- Consultas multitable o consultas anidadas.....	77

5.6.- FUNCIONAMIENTO DE UN NEGOCIO DE VENTA DE REFACCIONES AUTOMOTRICES .....	79
<b>VI. METODOLOGÍA .....</b>	<b>82</b>
6.1.- PASOS METODOLÓGICOS .....	82
6.2 DESARROLLO DE METODOLOGIA .....	84
6.2.1.- Investigación de requerimientos .....	84
6.2.1.1.- Entrevistas .....	85
6.2.1.2.- Revisión de registros .....	88
6.2.1.3.- Observación .....	88
6.2.2.- Especificación de requerimientos .....	89
6.2.2.1.- Identificación de requerimientos básicos .....	89
6.2.2.2.- Requerimientos de transacciones de los usuarios .....	91
6.2.2.3.- Requerimientos de decisión de los usuarios .....	92
6.2.2.4.- Requerimientos de toda la organización .....	93
6.2.3.- Diseño del SI con UML (casos de uso) .....	94
6.2.3.1.- Identificación de los actores .....	94
6.2.3.2.- Identificación de los casos de uso para los requerimientos de negocios .....	95
6.2.3.3.- Construcción del diagrama de modelos de casos de uso .....	96
6.2.3.4.- Narración de los casos de uso .....	97
6.2.4.- Definición de datos .....	98
6.2.5.- Análisis y diseño de la base de datos.....	100
6.2.7.- Desarrollo de interfaces de usuario del sistema .....	106

6.2.8.- Programación de las interfaces del sistema .....	108
6.2.9.- Pruebas de validación y verificación en el sistema .....	108
6.2.10.- Desarrollo de un manual de usuario para el sistema.....	108
6.2.11.- Implementación del sistema .....	109
<b>VII. RESULTADOS</b> .....	110
<b>VIII. DISCUSIÓN</b> .....	125
<b>IX. CONCLUSIONES</b> .....	130
X. Bibliografía.....	131
<b>X. ANEXOS</b> .....	133
ANEXO I .....	133
ANEXO II.....	134

## Índice de figuras

Figura 1. Un proceso de desarrollo de software. Fuente [2] .....	21
Figura 2. Representación de un caso de uso. Fuente elaboración propia .....	23
Figura 3. Diagrama de casos de uso de un sistema de información para venta de refacciones. Fuente elaboración propia .....	24
Figura 4. Ejemplo de una relación de asociación. Fuente elaboración propia .....	26
Figura 5. Ejemplo de una relación de extensión. Fuente elaboración propia .....	27
Figura 6. Ejemplo de una relación de usos. Fuente elaboración propia.....	28
Figura 7. Ejemplo de una relación de dependencia. Fuente elaboración propia...	28
Figura 8. Ejemplo de una relación de herencia. Fuente elaboración propia .....	29
Figura 9. Panorama de un sistema de información para venta de refacciones. Fuente elaboración propia .....	31
Figura 10. Tipos de modelos de datos que corresponden a cada nivel de abstracción de una arquitectura a tres niveles. Fuente [11]. .....	40
Figura 11. Clasificación de modelos de datos globales. Fuente [11]. .....	41
Figura 12. Representación de una relación en forma de tabla. Fuente [11].....	41
Figura 13. Atributos de la entidad CLIENTE. Fuente elaboración propia.....	46
Figura 14. Clave primaria para la entidad PROVEEDOR. Fuente elaboración propia .....	49
Figura 15. Claves primarias. Fuente elaboración propia .....	50
Figura 16. Una relación de uno a muchos. Fuente elaboración propia .....	51
Figura 17. Una relación uno a uno. Fuente elaboración propia .....	52
Figura 18. Una relación muchos a muchos. Fuente elaboración propia .....	52

Figura 19. Diagrama de casos de uso del negocio de venta de refacciones automotrices. Fuente elaboración propia .....	97
Figura 20. Base de datos del sistema para venta de refacciones .....	101
Figura 21. Ventana principal del manual de ayuda del sistema para la refaccionaria .....	109
Figura 22. Presentación del sistema .....	110
Figura 23. Seguridad del sistema.....	110
Figura 24. Menú principal del sistema.....	111
Figura 25. Ventana para ventas de mostrador .....	111
Figura 26. Menú agregar refacción .....	112
Figura 27. Ventana agregar refacción.....	112
Figura 28. Impresión de consulta de refacciones.....	113
Figura 29. Venta y presupuesto de refacciones .....	114
Figura 30. Edición de información de refacciones.....	115
Figura 31. Administración de facturas de compra .....	116
Figura 32. Ventana compras ingresar al inventario.....	117
Figura 33. Ventana proveedores.....	118
Figura 34. Ventana consultar ventas.....	119
Figura 35. Ventana vendedores .....	119
Figura 36. Ventana clientes .....	120
Figura 37. Ventana facturas de venta .....	121
Figura 38. Ventana detalle de factura de venta.....	122
Figura 39. Bosquejo de informe fiscal de ventas.....	123

Figura 40. Ventana de faltantes de refacciones.....	124
Figura 41. Ventana de catálogo de autos .....	124

## Índice de cuadros

Cuadro 1. Dominios lógicos representativos para tipos de datos lógicos. Fuente [4] .....	47
Cuadro 2. Componentes y contenido de la tabla VENTA. Fuente elaboración propia .....	50
Cuadro 3. Tipos de datos predefinidos. Fuente [23] .....	58
Cuadro 4. Operaciones de condición where. Fuente [23] .....	69
Cuadro 5. Resultado de una consulta Multitabla. Fuente elaboración propia .....	79
Cuadro 6. Glosario de actores. Fuente elaboración propia .....	94
Cuadro 7. Glosario de casos de uso. Fuente elaboración propia.....	96
Cuadro 8. Narración de los casos de uso. Fuente elaboración propia.....	98
Cuadro 9. Datos utilizados dentro del negocio de venta de refacciones. Fuente elaboración propia.....	100
Cuadro 10. Comparación de los sistemas utilizados por el negocio de refaccionaria.....	127

## **I. INTRODUCCIÓN**

El manejo de la información es parte fundamental de cualquier empresa u organización para la toma de decisiones en forma rápida y eficiente, además se requiere que esta información sea confiable y oportuna para facilitar una toma de decisiones precisa. Para que este proceso sea adecuado se han desarrollado los sistemas de información. Esta herramienta se ha convertido en una parte funcional y crítica de una organización, ya que aprovecha los beneficios que otorgan en el procesamiento de datos en forma rápida y confiable, consecuentemente los sistemas de información desempeñan un rol esencial en una organización además de que constituyen una ventaja competitiva para la organización.

Las empresas dedicadas a la compra venta de refacciones automotrices manejan grandes cantidades de datos, ya que existe gran cantidad de piezas que se almacenan para su venta y el movimiento de ellas es muy dinámico, otro aspecto importante a considerar es el registro de compras que se hacen a diferentes distribuidores y después el control del stock para que siempre haya refacciones en existencia. Más aún si la empresa recibe crédito las actividades del personal que se dedica a la administración de la empresa se duplican y en muchas ocasiones se puede llegar a perder el control del manejo de la información e incluso de la misma empresa y llegar a la declaración de quiebra por un mal manejo administrativo.

Considerando los aspectos ya comentados el presente trabajo tuvo la encomienda de diseñar y programar un sistema de información para la empresa Adonay dedicada al giro de la compra venta de refacciones automotrices. Los resultados obtenidos permitieron que la empresa tenga un manejo adecuado del inventario, lo cual se ha reflejado en primera instancia en un mejor servicio al cliente y comunicación con el proveedor para surtir en tiempo y forma las refacciones. La administración general de la refaccionaria mejoró sustancialmente, los errores se disminuyeron y el manejo del personal ha sido más eficiente, ya que los empleados dedican más tiempo a otras actividades que permiten mejorar el

servicio al cliente y el administrador no pierde el control de los recursos de la empresa.

## II. PLANTEAMIENTO DEL PROBLEMA

La Refaccionaria Adonay, es una organización con fin de lucro dedicada a la comercialización de refacciones automotrices y productos en general para el mantenimiento de un Automóvil.

El sistema actual es de forma manual y había estado funcionando correctamente; sin embargo, el incremento en las ventas ha ocasionado que el negocio comience a tener problemáticas debido a que no cuenta con una infraestructura necesaria que le permita atender más clientes en un periodo de tiempo más corto y a su vez poder documentar la información esto conlleva a que la atención al cliente sea tardada, lo cual trae como consecuencia descontento de los consumidores ya que en ocasiones esperan un tiempo considerable en ser atendidos, aunado a esto, se tiene que los procesos de facturación o emisión de notas de remisión, elaboración de presupuestos y pedidos a proveedores se llevan mucho tiempo de elaboración debido a que se elaboran manualmente.

También se ha detectado que el control del inventario de este negocio presenta algunas deficiencias por que no se tiene una relación de los productos almacenados, no se sabe con rapidez si un producto se tiene o no, además que si algunos productos se extraen sin autorización no podría notarse fácilmente; como consecuencia al momento de requerir un producto en algunas ocasiones no se encuentra en existencia debido a la deficiencia en el control de inventarios, lo cual se traduce en pérdidas económicas por falta de producto para su venta.

Las notas de remisión y facturación se realizan manualmente, y se hace al momento de la venta por lo que aumentan las posibilidades de errores matemáticos, ocasionando pérdidas al negocio. La facturación lleva aún más tiempo pues hay que desglosar el IVA a cada producto, además de copiar los datos del cliente en cada factura que requiera y pedirle su RFC en cada factura pues no se tiene una relación de clientes.

Se ha empleado una técnica de recolección de información llamada observación, realizada durante dos semanas, se concluyó lo siguiente:

La identificación de una refacción se realiza mediante un catálogo, el cliente proporciona el nombre de su carro, modelo, número de cilindros y la refacción que requiere, el vendedor busca en los catálogos la pieza que corresponde al auto, se dirige al almacén, verifica si existe en el inventario, muestra al cliente la refacción, se realiza la venta con remisión o factura ; esto se tarda en promedio un total de cuatro minutos, tratándose solo de la venta de una refacción pero en la mayoría de los casos un cliente compra todo lo necesario para el mantenimiento del auto, esto es de cinco a diez refacciones lo que se lleva un total de veinte minutos el atender a un sólo cliente, cabe mencionar que cada refacción se busca en un catálogo distinto; y en ocasiones algunos clientes tienen que esperar demasiado en ser atendidos.

El control de ventas, es también de forma manual, al término de cada jornada de trabajo se realiza un corte de caja, implica sumar todo lo vendido y realizar el corte de caja. Resulta muy notable que ésta tarea es un tanto monótona y tardada aunado a esto resulta interesante imaginar cómo se llevaría a cabo un informe de ventas semanal, mensual, anual, etc.

En este negocio se lleva a cabo gran cantidad de compras a crédito por cierto periodo de tiempo, en ocasiones se vence el pazo de alguna factura y existe el riesgo de perder los descuentos otorgados por pronto pago, no se sabe con certeza cuanto se debe en total, cuanto se debe a cada proveedor; obtener dicha información implica realizar un cálculo manual en cada momento que se requiere, no obstante el crecimiento del negocio implica más facturas y más créditos que en un momento dado resulta una tarea sumamente difícil de realizar. Esta información es de vital importancia para el administrador ya que al saber la situación de los créditos permite tomar decisiones acertadas y evitar pérdidas económicas.

La actualización de precios, es un problema recurrente en este negocio, la rotación de los productos puede durar desde un día hasta cinco años o más, entonces si un producto tiene más de cinco años puede suceder que se venda a un precio con esa antigüedad y al momento de resurtirlo sea más alto el costo, es por eso que se requiere la información de la fecha en que acceso un producto al inventario para poder en un momento dado actualizar su precio y la inversión este protegida en ese sentido.

### **III. JUSTIFICACIÓN**

La creación de un nuevo sistema de información para venta de refacciones automotrices adecuado a las funciones de negocio de la refaccionaria Adonay permitiría atender mayor cantidad de clientes en un periodo de tiempo más corto.

Un sistema de información computarizado, permite acceso a datos en segundos, con el nuevo sistema se podrá integrar la información de los catálogos de refacciones en una base de datos. El proceso de búsqueda de una refacción se haría mediante filtros que permitan seleccionar una refacción que corresponda a una marca y modelo de automóvil en específico o bien, obtener en una misma consulta todo el conjunto de refacciones que se tienen disponibles para un auto modelo y marca específicos; cabe mencionar que manualmente buscar un conjunto de refacciones implicaría buscar cada refacción en un catálogo distinto. El usuario podría consultar, seleccionar e incluso ver la imagen de las refacciones que requiere vender, posteriormente terminar el proceso de venta imprimiendo una nota de remisión o una factura según sea el caso y finalmente documentar la venta con los datos necesarios para consultas o reportes necesarios para la correcta toma de decisiones de esta organización.

El sistema maneja un control de facturas de compra donde el usuario podrá obtener información precisa del monto total en créditos, el monto que se debe a cada proveedor, o bien la información de facturas próximas a vencer para que no se generen intereses o se pierda el descuento por pronto pago.

Un sistema de información computarizado, tiene la capacidad de recordar que refacciones están pendientes de resurtir en el inventario, el tiempo que llevan almacenados, existencias, costos, precios de venta, que proveedor surte cada refacción y demás información que el negocio requiere para el control de su inventario.

Este sistema contará con una relación de clientes y proveedores registrados ante la Secretaria de Hacienda y Crédito Público; la relación de clientes con el fin de

poder realizar facturas sin tener que pedir los datos necesarios cada vez que se realiza una factura. Además los datos de cada factura se guardarán en la base de datos para que el administrador pueda generar en un momento dado un bosquejo del reporte fiscal de ventas con respecto a algún periodo y a su vez el reporte de facturas de compra con el fin de que el contador haga las operaciones pertinentes para el reporte fiscal del mes ante Secretaría de Hacienda y Crédito Público. Sin tener que realizar todo el conjunto de operaciones monótonas de cada reporte.

Mientras tanto un sistema comercial de control de inventario no soluciona por completo la problemática de este negocio, debido a que no cuenta con las especificaciones antes descritas que identifique una refacción para un automóvil en particular. Un sistema de información comercial, hace la búsqueda por medio de una clave o código que no serviría de mucho a este negocio, ya que conocer la clave de una refacción requiere de realizar la búsqueda en el catálogo, como es notable no resulta viable esta solución.

Con este nuevo sistema de información, se obtendría la infraestructura para atender a los clientes en un periodo de tiempo más corto, por lo tanto, el negocio tendría la capacidad de atender mayor cantidad de clientes al día, a su vez mayores ventas y como resultado mayores utilidades; con un inventario disponible y controlado con la posibilidad de mandar los pedidos por correo, para evitar tener que dictarlos por teléfono.

## **IV. OBJETIVOS**

### **4.1.- OBJETIVO GENERAL**

- ✓ Implementar un sistema de información para la refaccionaria Adonay, que permita tener un control sobre el inventario, además de tener de forma ordenada la información necesaria para llevar a cabo los procesos de negocio y administración de forma precisa y controlada.

### **4.2.- OBJETIVOS PARTICULARES**

- ✓ Conocer el negocio de venta de refacciones automotrices a fin de determinar las necesidades del nuevo sistema de información.
- ✓ Diseñar y desarrollar un sistema de información que cumpla con las necesidades del negocio de venta de refacciones.
- ✓ Capacitar al personal del negocio, acerca del funcionamiento del nuevo sistema y la forma de operarlo.

## **V. MARCO TEÓRICO**

### **5.1.- MODELO UML**

Desde tiempo atrás el hombre siempre ha tratado de representar y expresar sus ideas y objetos con el fin de entender y manipular su medio. En el desarrollo de sistemas, un requerimiento básico para cualquier modelo es describir al sistema con suficiente detalle para predecir el comportamiento del sistema.

Un modelo se utiliza como ayuda al pensamiento al organizar y clasificar conceptos confusos e inconsistentes. Al realizar un análisis se crea un modelo del sistema que muestre las entidades, las interrelaciones, etc. Se enfoca en ir detallando parte por parte del sistema hasta su totalidad.

Un modelo, es una representación de la realidad [1]. La adecuada construcción de un modelo ayuda a organizar, evaluar y examinar la validez de pensamientos. Asimismo, representa un medio de comunicación más eficiente y efectivo para transmitir las ideas a las diferentes partes involucradas en la elaboración del sistema.

En el desarrollo de sistemas de software el modelado constituye una parte fundamental, ya que explica en forma detallada el comportamiento y la estructura que va a tener el sistema, se analiza desde distintas perspectivas para entender de mejor manera posibles situaciones y problemas en el desarrollo del proyecto.

UML, es un lenguaje de modelado de sistemas de software por sus siglas (Unified Modeling Language) que quiere decir lenguaje unificado de modelado. Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software.

El lenguaje unificado de modelado (UML), es una notación gráfica para dibujar diagramas de conceptos de software. Como un diseño de software propuesto o una implementación de software ya completado. [2]

El proceso Unificado, es un proceso de desarrollo de software. Un proceso de desarrollo de software, es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software (véase la figura 1). Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones. Diferentes niveles de aptitud y diferentes tamaños de proyecto. [2]

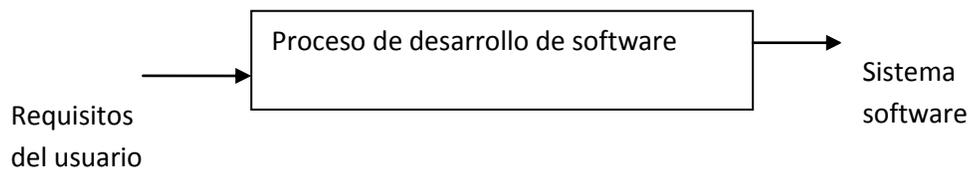


Figura 1. Un proceso de desarrollo de software. Fuente [2]

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. [2]

El Proceso Unificado, utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso Unificado sus desarrollos fueron paralelos. [3]

No obstante los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres frases clave dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Esto es lo que hace único al Proceso Unificado. [2]

### **5.1.1.- Los casos de uso**

Durante mucho tiempo las personas se auxiliaban de escenarios típicos para poder comprender los requerimientos de un sistema, pero estos se trataban de forma muy informal, ya que se escribían pero muy pocas veces se documentaban,

fue hasta entonces que el DR. Ivar Jacobson con el método objectory y su primer libro acerca del tema en donde elevó la visibilidad del caso de uso (nombre para un escenario) a tal punto que lo convirtió en elemento primario para la planificación y el desarrollo de proyectos.

En la actualidad se le reconoce ampliamente como la mejor práctica para la definición, la documentación y la comprensión de los requerimientos funcionales de un sistema de información. [4]

Uno de los retos fundamentales de importancia vital para cualquier equipo de desarrollo de sistemas de información, y especialmente para el analista de sistemas, es la capacidad para obtener de los involucrados, los requerimientos correctos y necesarios del sistema, y especificarlos de manera inteligible para los involucrados de modo que estos requerimientos se verifiquen y se validen. [4] 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 el fracaso de todo el proyecto, es por eso que se requiere un alto nivel de participación por parte del usuario.

Un caso de uso es, en esencia, una interacción típica entre un usuario y un sistema de cómputo. [5] Un caso de uso se obtiene a partir de la comunicación que se establece con la persona que va a interactuar con el sistema y es precisamente las funciones que desea realizar mediante el sistema de información. A este tipo de interacciones se les asigna un nombre breve que describa la función a realizar.

Un caso de uso, es una sintonía instantánea de algún aspecto de su sistema. La suma de todos los casos de uso constituyen la vista externa del sistema, que es un gran avance hacia la explicación de lo que hará el sistema. [5] Un caso de uso, tiene como fin describir una funcionalidad que deberá permitir realizar el sistema.

Un caso de uso, es una descripción de la secuencia de interacciones que se producen entre un actor (este término se estudia en este capítulo) y el sistema, es decir, un caso de uso representa una funcionalidad del sistema que se realizará

tras la orden de un actor o bien desde la invocación de otro caso de uso, expresando una unidad coherente de funcionalidad; un caso de uso se representa en el diagrama de casos de uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe ser la función específica que el usuario desea realizar utilizando el sistema.



Figura 2. Representación de un caso de uso. Fuente elaboración propia

### 5.1.2.- Diagrama de casos de uso

Jacobson (1994), además de introducir los casos de uso como elementos primarios del desarrollo de software, también diseñó un diagrama para la representación gráfica de los casos de uso. **El diagrama de casos de uso** es ya también parte del UML. [5]

En ocasiones suele confundirse el término diagrama de casos de uso con los casos de uso, resulta útil aclarar como ya se ha mencionado un caso de uso representa una funcionalidad que debe permitir realizar el sistema, de modo que un sistema contiene diversas funcionalidades, se deben escribir el mismo número de casos de uso que representen este conjunto de funcionalidades. Los casos de uso se agrupan de manera ordenada dentro de un diagrama de casos de uso, con el fin de abarcar en un principio las diversas funcionalidades o interacciones que el sistema permitirá que el usuario realice.

Diagrama, ilustra las interacciones entre el sistema y los sistemas y usuarios externos. En otras palabras describe gráficamente quien va a usar el sistema y de que manera el usuario espera interactuar con el sistema. [4]

Dentro del diagrama de casos de uso, se representa al sistema como un recuadro con el nombre en su interior, los casos de uso se encuentran en el interior de la caja del sistema mientras que los actores se encuentran fuera, ya que estos

representan un agente externo que va a interactuar con el sistema pero no está dentro de él (Véase la figura 3).

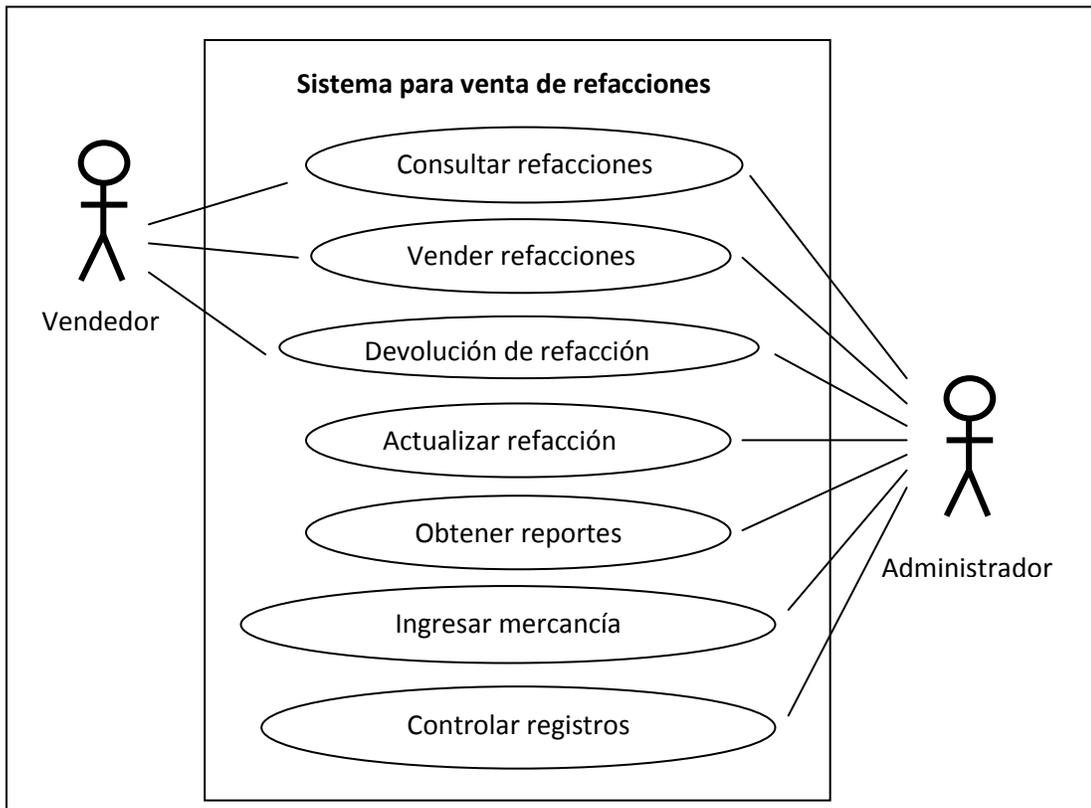


Figura 3. Diagrama de casos de uso de un sistema de información para venta de refacciones.  
Fuente elaboración propia

### 5.1.3.- Elementos del diagrama de casos de uso

Los elementos que suelen aparecer dentro de un diagrama de casos de uso son los actores, los casos de uso y sus relaciones.

#### 5.1.3.1.- Actor

Los casos de uso se inician o son generados por los usuarios externos llamados actores. [4]

Un actor, es un agente externo al sistema. El actor se define como un rol que el usuario desempeña con respecto al sistema, cabe mencionar que un actor no representa una persona sino la labor que desempeña frente al sistema; es por eso que un actor puede representar la labor de un usuario o bien de un sistema

externo con el cual se comunica nuestro sistema. El actor se representa mediante una figura humana dibujada con líneas, ya que el actor es un agente externo al sistema se dibuja fuera de él.

Es importante observar que hay principalmente cuatro tipos de actores. [4]

**Actor primario de negocios.** Se beneficia de la ejecución del caso de uso al recibir algo de valor medible u observable, este actor puede o no iniciar el evento de negocios. Ejemplo: el cliente pide una refacción al vendedor, el vendedor ejecuta una consulta y posteriormente hace la venta de la refacción, el cliente recibe la refacción (algo con valor medible), el vendedor hace esto con cada cliente, los clientes no inician el evento pero son el receptor primario de algo con valor medible.

**Actor primario del sistema.** Interactúa directamente con la interfaz del sistema por lo tanto, este actor inicia el evento de negocios, los actores primarios del sistema pueden interactuar con el actor primario de negocios con el propósito de usar el sistema real. Estos actores facilitan el inicio del evento en el sistema con el fin de beneficiar al actor primario de negocios. Citando el ejemplo anterior el actor primario del sistema sería el vendedor, que ejecuta funciones del sistema en beneficio del cliente. Cabe mencionar, que en algunos casos el actor primario de negocio y el actor primario del sistema pueden ser el mismo, un ejemplo sería una persona que hace la reservación de un hotel desde un sitio web.

**Actor externo servidor.** Involucrado que responde a una solicitud desde el caso de uso. Un ejemplo de esto sería un buro de crédito que autoriza el pago mediante tarjeta de crédito

**Actor externo receptor.** Involucrado que no es el actor primario pero recibe algo de valor medible u observable (salida) proviene del caso de uso. Un ejemplo de esto sería que se generara un reporte de faltantes de refacciones el cual se debe hacer llegar a los proveedores para que surtan la mercancía.

En ocasiones dentro de los sistemas de información suele haber eventos de negocios ocasionados por el calendario del reloj, a este tipo de eventos se les llama eventos temporales que se ejecutan automáticamente en una hora y fecha determinada. Aplicando dicho ejemplo al sistema de información para venta de refacciones automotrices, se propone una funcionalidad que el sistema debe contener la cual debe avisar al usuario que una factura de crédito está a punto de vencer entonces esto deberá ejecutarse en determinada fecha antes que la factura se venza y ocasione intereses.

### 5.1.3.2.- Relaciones

Las relaciones describen la interacción que se da entre los actores y los casos de uso o bien entre dos o más casos de uso, un caso de uso en un principio debería describir una tarea que tiene un sentido completo para el usuario, sin embargo, en ocasiones resulta útil describir una interacción más detalladamente sobre un caso de uso.

#### 5.1.3.2.1.- Asociación

Este tipo de relación se da entre un caso de uso y un actor, dicha relación se representa mediante una línea continua que conecta al caso de uso y al actor. Nótese en la figura 4, la relación que existe entre el administrador y el caso de uso tiene una línea continua con una cabeza de flecha hacia el caso de uso, esto indica que el caso fue iniciado por el actor; por otro lado la asociación que existe entre el caso de uso y el proveedor no tiene punta de flecha esto indica una relación entre un caso de uso y un actor externo servidor o receptor.

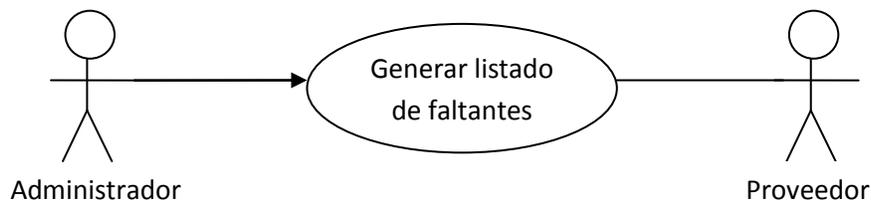


Figura 4. Ejemplo de una relación de asociación. Fuente elaboración propia

### 5.1.3.2.2.- Extensión

Este tipo de relación se da entre los casos de uso, en ocasiones los casos de uso describen una funcionalidad compleja que consiste de varios pasos que hacen difícil entender la lógica del caso, con el fin de poder entender este caso de uso se extraen los pasos más complejos para formar su propio caso.

Se usa la relación **extends** cuando se tiene un caso de uso que es similar a otro, pero que hace un poco más. [5]

Dicha relación se da cuando no se efectúa el comportamiento habitual de un caso de uso, es decir se efectúa una variación del caso de uso. En otras palabras extiende un comportamiento adicional en un caso de uso base sin que este tenga conocimiento. Véase el ejemplo de la figura 5, en donde el actor inicia el caso de uso vender refacción si la refacción no está agotada el caso de uso se efectúa sin variaciones pero por otro lado, si la refacción está agotada se efectúa la variación del caso de uso refacción agotada.

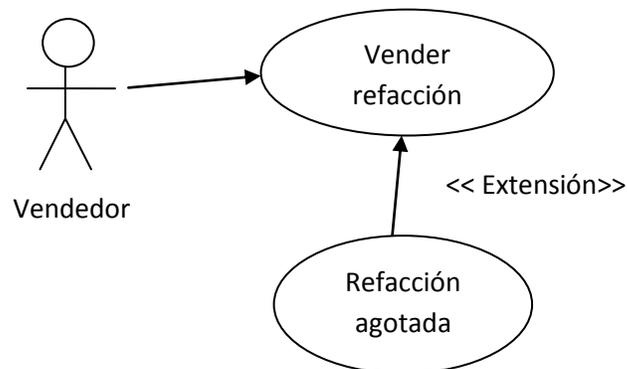


Figura 5. Ejemplo de una relación de extensión. Fuente elaboración propia

### 5.1.3.2.3.- Usos

Indica que un caso de uso es incluido en otro, esto ocurre normalmente cuando algunos casos de uso comparten pasos en común, lo que se hace es extraer los pasos comunes y crear un propio caso de uso que sirva como caso de uso resumen, un caso de uso resumen se encuentra disponible para cualquier otro caso de uso que necesite de su funcionalidad. Como se muestra en la figura 6, la

relación comienza en el caso de uso oficial y apunta al caso de uso que está usando, esto implica que si el actor en este caso el administrador requiere vender una refacción antes debe consultar que refacción es la que desea vender o bien si desea realizar algún cambio sobre la información de una refacción también debe consultar sobre que refacción desea hacer los cambios.

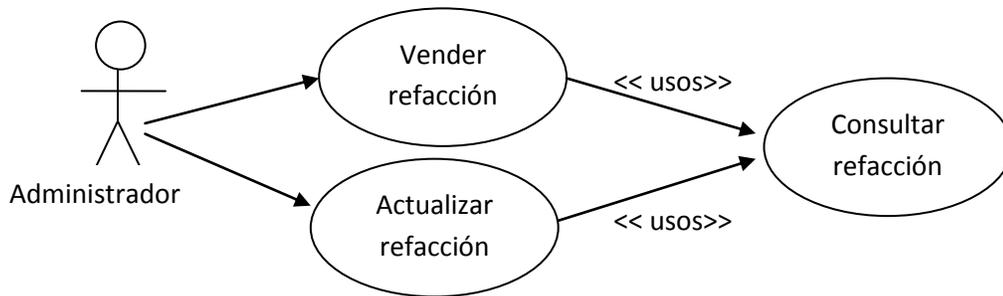


Figura 6. Ejemplo de una relación de usos. Fuente elaboración propia

#### 5.1.3.2.4.- Dependencia

Relación entre casos de uso que identifica que casos de uso dependen de otros con el objeto de determinar una secuencia entre los casos de uso para los propósitos de condiciones de uso y pruebas. En la figura 7, se ilustra la relación de dependencia entre los casos de uso, conectados por una línea continua y en uno de los extremos una punta de flecha que apunta al caso de uso del cual depende.

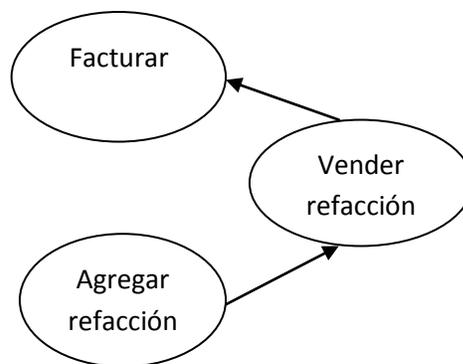


Figura 7. Ejemplo de una relación de dependencia. Fuente elaboración propia

### 5.1.3.2.5.- Herencia

Este tipo de relación se da entre los actores, cuando dos o más actores comparten un comportamiento en común, se extrae el comportamiento en común y se asigna a un actor resumen, con el fin de evitar la repetición en el sistema. En la figura 8, se muestra la relación de herencia entre los actores administrador y vendedor representada por una línea continua y un extremo una punta de flecha que representa que hereda las funcionalidades que desempeña el actor señalado, donde el administrador hereda las funcionalidades del vendedor y añade otras, ya que el administrador de un negocio tiene la facultad de ejecutar funciones de un vendedor y además desempeña otras funciones que el vendedor no puede llevar a cabo. Comparece la figura 1.9 con la figura 3 (contenida en este capítulo) con el fin de ver la diferencia.

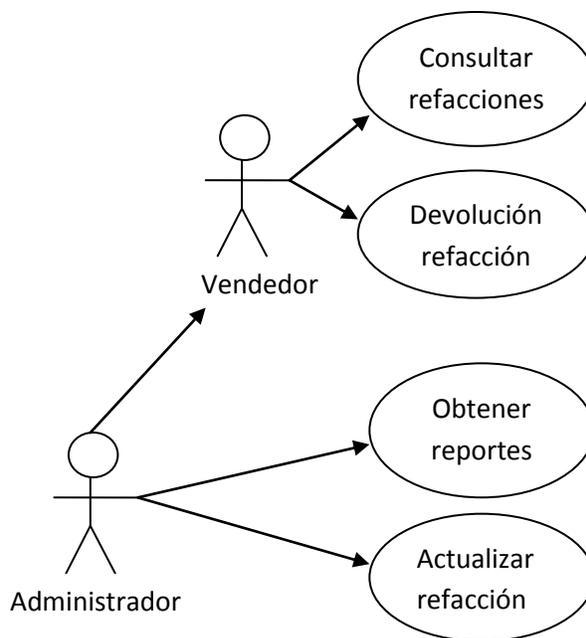


Figura 8. Ejemplo de una relación de herencia. Fuente elaboración propia

## 5.2.- SISTEMAS DE INFORMACIÓN

Para poder entender mejor una definición de un sistema de información antes se deben repasar tres conceptos importantes que tienen una relación con lo que se está estudiando; el primer concepto se refiere a un dato, un dato es un conjunto de

caracteres con algún significado los cuales pueden ser numéricos, alfabéticos o alfanuméricos; el segundo concepto es la información, que se refiere a el conjunto de datos que tienen un determinado significado y por último el concepto de Sistema que se refiere al conjunto de partes que interactúan entre sí con el fin de alcanzar un objetivo común.

Se tienen diversas analogías con respecto al funcionamiento de un sistema, una de ellas es el sistema circulatorio de los seres humanos el cual está formado por la sangre las venas y el corazón, dicho sistema hace que la sangre circule hacia las distintas partes del cuerpo además interactúa con otros sistemas del cuerpo para asegurar que la cantidad y la composición correcta de la sangre llegue a tiempo a las distintas partes del cuerpo; mientras tanto un sistema de información maneja los datos provenientes de su entorno, los procesa y genera datos e información para la toma de decisiones.

Aplicando dichos conceptos a un sistema de información para la venta de refacciones automotrices ayuda al vendedor a identificar la refacción que el cliente le está solicitando, realiza la venta entregando un comprobante de compra al cliente remisión o factura según sea el caso, a su vez el cliente realiza el pago de la mercancía, dicha mercancía se solicita a los proveedores por medio de un reporte de faltantes que genera el sistema en un momento dado; con la llegada de la mercancía se tiene que dar de alta las nuevas existencias que se tienen así como actualizaciones en cuanto a cambio de precios, especificaciones o en su defecto agregar un nuevo producto o la baja de alguno de ellos. Véase la figura 9.

Un componente muy primordial dentro de los sistemas de información son las bases de datos, cuyo papel principal es la capacidad de almacenar información permanentemente de tal forma que se puede tener acceso a dicha información; cabe mencionar que las bases de datos no son el único componente de un sistema de información ya que también contiene personas, procedimientos, datos de entrada, datos de salida, software y hardware.

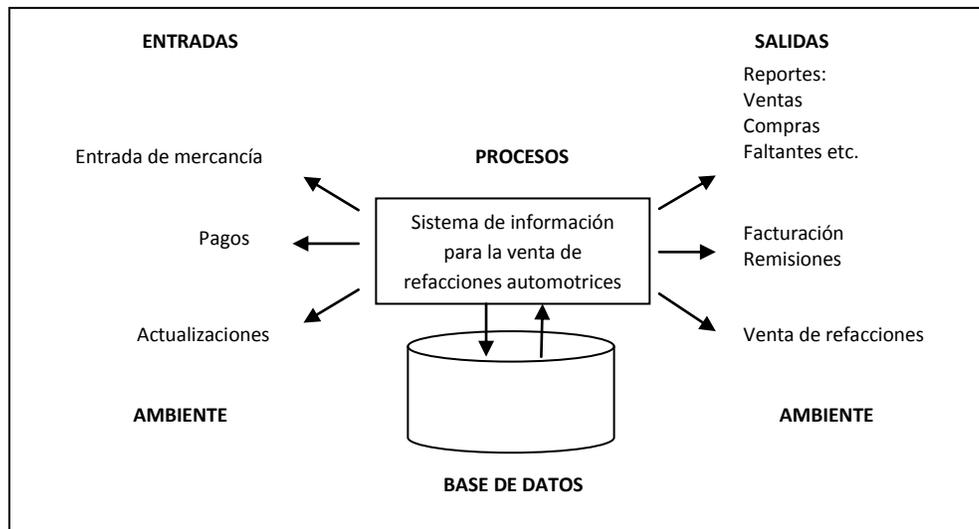


Figura 9. Panorama de un sistema de información para venta de refacciones.  
Fuente elaboración propia

Un sistema de información (SI), está orientado al tratamiento y administración de datos e información con el fin de apoyar actividades de una empresa o negocio.

Toda organización necesita para su funcionamiento un conjunto de informaciones que han de transmitir entre sus distintos elementos y generalmente también, desde y hacia el exterior del sistema. Una parte de esta comunicación se realiza por medio de contactos interpersonales entre los empleados, **es el sistema de información informal**. Pero este tipo de flujo de información, cuando se trata de organismos complejos, se muestra insuficiente y costoso, siendo preciso disponer de un **sistema de información formal**, también llamado *organizacional* que, integrado en el sistema de orden superior que es el organismo, aporte a este la información necesaria de forma eficaz y eficiente.[6]

Todo sistema de información se diseña a fin de satisfacer las necesidades de información de una organización (empresa o cualquier otro tipo de institución pública o privada) y está inmerso en ella. El SI ha de tomar los datos del entorno (la propia organización así como fuentes externas) y sus resultados han de ser la información que dicha organización necesita para su gestión y toma de decisiones; por otra parte, los directivos de la organización tendrán que marcar los objetivos y directrices por los que se regule el SI. [6]

### **5.2.1.- Funciones principales de los sistemas de información (SI)**

Las funciones principales de un sistema de información son las siguientes:

**Entrada:** es el proceso mediante el cual el sistema obtiene datos para posteriormente ser procesados y producir información; la entrada de datos puede ser manual o automática, manual es proporcionada directamente por el usuario mientras que la automática se refiere a datos e información que provienen de otro sistema o modulo.

**Almacenamiento de información:** almacena información para ser procesada por el sistema posteriormente, la información es guardada dentro de una base de datos en un orden específico.

**Procesamiento de información:** es la capacidad que tiene el sistema de efectuar cálculos mediante una secuencia de operaciones preestablecida, este proceso se efectúa mediante datos introducidos recientemente o datos almacenados anteriormente con el fin de generar información útil para la correcta toma de decisiones dentro de una organización.

**Salida de información:** proporciona información hacia el exterior a través de dispositivos de salida como son impresoras, graficadores, terminales entre otros, cabe mencionar que una salida de información puede constituir la entrada hacia otro sistema o modulo.

Las finalidades de un sistema de información, como la de cualquier otro sistema dentro de una organización, son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas.[7]

### **5.2.2.- Clasificación de los SI**

Los sistemas de información, son desarrollados con propósitos diferentes dependiendo de las necesidades del negocio. [8] Según la función a la que vayan destinados o el tipo de usuario final del mismo. [9]

### **5.2.2.1.- Sistemas de procesamiento de transacciones (TPS)**

Son sistemas de información computarizados desarrollados para procesar gran cantidad de datos para transacciones rutinarias de los negocios, tales como nomina e inventario. [8] Estos sistemas reducen el tiempo que lleva ejecutar transacciones que de forma manual resultarían sumamente tediosas, sin olvidar la rapidez con que se puede obtener información de un TPS acerca de lo que pasa dentro de la organización.

Una transacción, es un suceso en el cual se crean o modifican datos. El procesamiento de transacciones consiste en obtener, manipular y almacenar los datos importantes dentro del proceso de negocio de una organización, además se preparan los reportes que se desea obtener a partir de los datos utilizados dentro del sistema; lo importante en una transacción son los datos que se desean obtener como se obtienen, cuales y en qué momento se modifican.

### **5.2.2.2.- Sistemas de automatización de oficinas (OAS) y sistemas de manejo de conocimiento (KWS)**

Al nivel de conocimiento de la organización hay dos clases de sistemas. [8] Los sistemas de automatización de oficinas, se refiere a todas las aplicaciones destinadas a ayudar en el trabajo diario del administrativo de una empresa u organización, estos sistemas no ayudan a crear un nuevo conocimiento sino usan la información para analizarla, transformarla y compartirla formalmente a toda la organización; tales aplicaciones incluyen procesadores de texto, hojas de cálculo, agendas electrónicas, correo electrónico, videoconferencias.

Los sistemas de manejo de conocimiento dan soporte a trabajadores profesionales, tales como científicos, ingenieros y doctores, a diferencia de los OAS este tipo de sistemas les ayuda a los profesionales a crear un nuevo conocimiento que contribuya a la organización o la sociedad en general.

### **5.2.2.3.- Sistemas de información gerencial (SIG o MIS)**

Estos sistemas están orientados a la solución de problemas empresariales en general, dicho sistema prevé información que apoya las operaciones la administración y las funciones de toma de decisiones de una empresa. El sistema utiliza equipos de cómputo, software especializado, procedimientos, manuales, modelos para el análisis, la planificación, el control y la toma de decisiones, además de las bases de datos. Los sistemas de información dan soporte a un espectro más amplio de tareas organizacionales que los sistemas de procesamiento de transacciones, incluyendo el análisis de decisiones y la toma de decisiones.

Las funciones gerenciales planificación, organización, dirección y control son necesarias para un buen desempeño organizacional; este tipo de sistemas de información apoyan estas funciones principalmente la planificación y el control, es por eso la importancia del valor de la información proporcionada, la cual debe cumplir con los cuatro supuestos básicos calidad, oportunidad, cantidad y relevancia; calidad es decir deben ser un fiel reflejo de la realidad planteada, oportunidad se refiere a que la información debe tenerse a tiempo pues en caso de ser necesario se puedan tomar medidas correctivas a tiempo, la cantidad de información debe ser la suficiente pero tampoco debe verse desbordado por información irrelevante e inútil y por último la relevancia nos dice que la información debe estar relacionada con las tareas y actividades que el gerente desempeña.

### **5.2.2.4.- Sistemas de apoyo a la toma de decisiones (DSS)**

Tienen como propósito fundamental apoyar y facilitar el proceso de toma de decisiones a través de la obtención oportuna y confiable de información relevante para dicho suceso.

El DSS, es similar al sistema de información gerencial tradicional en que ambos dependen de una base de datos como fuente [8]. Para poder implantar estos tipos de sistema resulta necesario haber implantado ya un sistema transaccional en la

organización, ya que la mayoría de la información proviene de la base de datos del sistema transaccional, para finalmente poder apoyar en la toma de decisiones a los directivos.

Un sistema de apoyo a decisiones se aparta del sistema de información gerencial tradicional en que enfatiza el apoyo a la toma de decisiones en todas sus fases, aunque la decisión actual todavía es del dominio del tomador de decisiones. [8]

Cabe mencionar que estos sistemas suelen ser interactivos y amigables con altos estándares de diseño gráfico y visual ya que están dirigidos al usuario final.

#### **5.2.2.5.- Sistemas expertos e inteligencia artificial (AI)**

La inteligencia artificial puede ser considerada la meta de los sistemas expertos. [11]. La inteligencia artificial, tiene como objetivo desarrollar máquinas que se comporten de forma inteligente. Los enfoques de la inteligencia artificial, es la comprensión del lenguaje natural y otro la capacidad o habilidad para razonar un problema y llegar a una conclusión lógica; un sistema experto emula el comportamiento de un experto en un dominio en concreto, y en ocasiones son usados por estos. Con los sistemas expertos se busca una mejor calidad y rapidez en las respuestas dando así lugar a una mejora de la productividad del experto.

Estos sistemas imitan las actividades de un humano para resolver problemas de distinta índole usando enfoques de razonamiento para resolver un problema.

Un sistema experto selecciona la mejor solución a un problema. [8]

Los componentes básicos de un sistema experto son la base del conocimiento, los mecanismos de inferencia y la interfaz de usuario. Los ingenieros de conocimiento capturan el conocimiento del experto en lo que se llama base de conocimiento con una estructura bien definida para que los motores de inferencia puedan acceder a la base de conocimiento a fin de proporcionar al usuario la mejor solución a un problema planteado.

#### **5.2.2.6.- Sistemas de soporte a las decisiones en grupo (DGSS)**

Estos sistemas son usados en salones equipados con configuraciones que permiten que un equipo interactúe de forma electrónica, brinda soporte en el proceso de toma de decisiones simultáneas, es decir, que requieren ser tomadas por un grupo de personas, uno de los principales objetivos de estos sistemas es lograr la mayor productividad en el desarrollo de las juntas de trabajo e implica aumentar la velocidad y calidad con que se toman decisiones; en dichas discusiones las opiniones personales son desplegadas en pantalla, permitiendo la discusión abierta y anónima de temas difíciles, promoviendo la creatividad y participación sana de cada uno de los participantes.

#### **5.2.2.7.- Sistemas de información ejecutiva (EIS)**

Herramienta orientada a usuarios de nivel gerencial, que permite monitorizar el estado de las variables de un área o unidad de la empresa a partir de información interna o externa a la misma, que le permita tomar decisiones a nivel estratégico, estos sistemas se apoyan en la información generada por los TPS y los MIS, los sistemas de apoyo a ejecutivos ayudan a los usuarios a que ataquen problemas de decisión sin estructura; además les ayudan a organizar sus interacciones con el ambiente externo, proporcionando apoyo de gráficos y comunicaciones en lugares accesibles tales como salas de juntas u oficinas personales corporativas.

#### **5.2.3.- Análisis de sistemas**

**Análisis de sistemas.** Técnica de solución de problemas que descompone el sistema en sus componentes para estudiar el grado en que éstos funcionan e interactúan para lograr su propósito. [4]

**Diseño de sistemas.** Técnica complementaria (de la de análisis de sistemas) de solución de problemas que reensambla los componentes de un sistema en el sistema completo, con la esperanza de mejorarlo. Ello puede abarcar la adición, la eliminación y el cambio de componentes en relación con el sistema original. [4]

**Análisis de sistemas de información.** Las fases de desarrollo de un proyecto de desarrollo de sistemas de información que se centran principalmente en los problemas y requerimientos de negocios, con independencia de la tecnología que pueda usarse o se use para implantar una solución al problema. [4]

El análisis de sistemas, es un requerimiento previo al diseño de sistemas, la especificación de un sistema nuevo y mejorado.

Un análisis de sistemas, es impulsado por los asuntos de negocios de los propietarios de sistemas y los usuarios de sistemas. Por tanto aborda los bloques de construcción de conocimiento, proceso y comunicaciones desde la perspectiva de propietarios del sistema y los usuarios del sistema. [4]

### **5.3.- BASE DE DATOS**

#### **5.3.1.- Antecedentes**

Toda organización reúne a diario gran cantidad de hechos sobre personas, cosas o eventos como estados de cuenta, montos de compras, de ventas, etc. Así como hechos no convencionales como son imágenes, huellas digitales entre otros. Dichos hechos son contenidos dentro de una base de datos.

Las bases de datos, constituyen una parte fundamental de los sistemas de información, actualmente existe una gran necesidad de información dentro de las organizaciones, el uso de la información está directamente relacionada con el desarrollo económico y social de las mismas, ya que el uso de información en forma correcta y oportuna representa mayor eficiencia en el desempeño de sus actividades. La investigación, la planificación y la toma de decisiones exigen información precisa, oportuna, completa, coherente y adaptada a las necesidades específicas de cada usuario y de cada circunstancia [10].

El uso de los sistemas de bases de datos automatizadas, se desarrolló a partir de la necesidad de almacenar grandes cantidades de información, para su posterior

consulta, producidas por las nuevas industrias que creaban gran cantidad de información [11].

La aparición de la expresión de *base de datos*, se produce a comienzos de los años sesenta. En 1963 tuvo lugar en Santa Mónica (EEUU), un simposio en cuyo título se encontraba la expresión Data Base. En una de sus sesiones, se propuso una definición de base de datos que, según las actas del simposio, no fue universalmente aceptada. Posteriormente, en 1970, el grupo de estandarización Codasyl decidió cambiar su primitiva denominación en la que no aparecía base de datos por el de Data Base Task Group. Poco a poco, el concepto y la expresión base de datos iba imponiéndose. [10]

### **5.3.2.- Definición de base de datos**

Una base de datos, es una colección de datos que puede estar representada por la información acerca de cualquier tema. Un número telefónico es un dato, un nombre es un dato, etc. [12]

Una base de datos es un conjunto, colección o depósito de datos almacenados en un soporte informático no volátil. [11]

Una base de datos, es una colección de datos clasificados y estructurados que son guardados en uno o varios ficheros pero referenciados como si de un único fichero se tratara. [13]

Con las definiciones anteriores se determina que una base de datos, es una colección de registros del mismo tipo almacenados en tablas con una estructura en específico. La base de datos, es almacenada como un archivo de forma permanente dentro de la computadora.

### **5.3.3.- Estructura de base de datos**

Las bases de datos, están formadas por uno o varios bloques de información llamados *tablas* (inicialmente denominado ficheros o archivos) que normalmente tendrán una característica en común.

Una *tabla* o archivo de datos, es un conjunto conexo de información del mismo tipo. Cada tabla está formada por *registros*; el cual es una unidad elemental de información de la tabla o fichero. Cada registro está formado por uno o más elementos llamados *campos*. Un campo, es cada una de las informaciones que interesa almacenar en cada registro y es, por tanto, la unidad elemental de información del registro [14].

Los sistemas de gestión de base de datos (SGBD o DBMS), organizan y estructuran los datos de tal modo que puedan ser recuperados y manipulados por usuarios y programas de aplicación el modelo de datos determina la personalidad de un DBMS y las aplicaciones para las cuales está particularmente bien vinculado.

El Sistema de Gestión de Bases de Datos (SGBD o DBMS), es el conjunto de programas que permiten la implantación, acceso y mantenimiento de la base de datos. El SGBD junto con la base de datos y con los usuarios, constituyen el Sistema de Bases de Datos [11].

Los datos están interrelacionados y estructurados de acuerdo con un modelo capaz de recoger el máximo contenido semántico. La redundancia de los datos debe ser controlada, de forma que no existan duplicidades perjudiciales ni innecesarias, y que las redundancias físicas sean tratadas por el mismo sistema, de modo que no puedan producirse inconsistencias [10].

La definición o descripción del conjunto de datos contenidos en la base (lo que se denomina estructura o esquema de la base de datos) debe ser únicas y estar integradas con los mismos datos [14].

#### **5.3.4.- Modelado de base de datos**

Los *Modelos de Datos*, son una colección de herramientas conceptuales para describir datos, relacionados entre ellos, semántica asociada a los datos y restricciones de consistencia [10].

Un *Modelo de Datos* (MD), es un conjunto de conceptos que permiten describir, a distintos niveles de abstracción, la estructura de una base de datos, a la cual denominamos *esquema*. Según el nivel de abstracción de la arquitectura ANSI el modelo que permite su descripción será un modelo *externo, global o interno* [11] (Figura 10).

Los modelos externos permiten representar los datos que necesita cada usuario en particular. Los modelos globales ayudan a describir los datos para el conjunto de usuarios, se puede decir que es la información a nivel empresa; y por último los modelos internos, están orientados a la máquina, siendo sus elementos de descripción, punteros, índices, agrupamientos, etc.

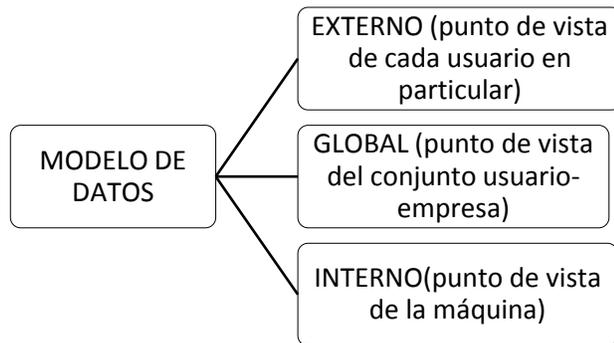


Figura 10. Tipos de modelos de datos que corresponden a cada nivel de abstracción de una arquitectura a tres niveles. Fuente [11].

Los Modelos Globales se clasifican, a su vez, en conceptuales y convencionales. Los *modelos Conceptuales* (también llamados de alto nivel), facilitan la descripción global del conjunto de información de la empresa con independencia de la máquina porque sus conceptos son cercanos al mundo real (entidades, atributos, etc.); son modelos de análisis no de implementación. Los *modelos Convencionales*, están orientados a describir los datos a nivel lógico para que sus conceptos sean propios de cada SGBD (Figura 11).

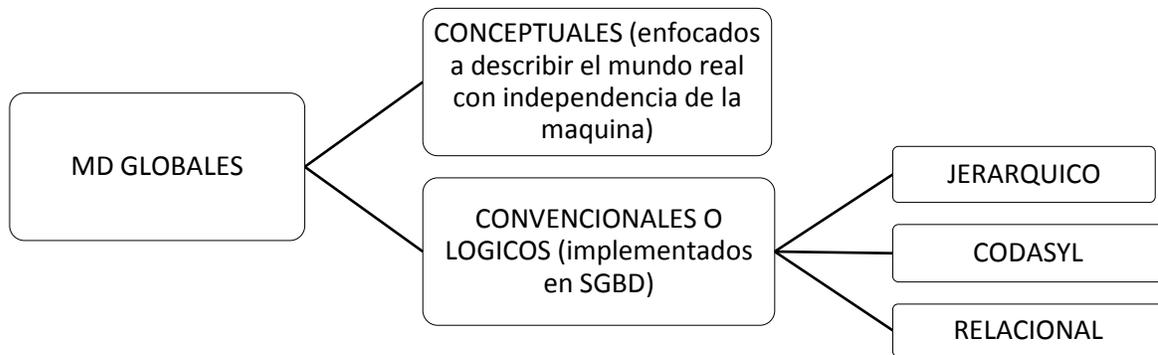


Figura 11. Clasificación de modelos de datos Globales. Fuente [11].

### 5.3.5.- Estructura del Modelo Relacional

Una base de datos relacional, consiste en una colección de tablas, a cada una de las cuales se asigna un nombre único. Una fila de una tabla representa una relación entre un conjunto de valores, por lo tanto una tabla es una colección de dichas relaciones [11].

Atributo 1	Atributo 2	.....	Atributo n	Nombre
XXXXXX	XXXXXXXXXX	.....	XXXXXXXXXX	→ Tupla 1
XXXXXX	XXXXXX	.....	XXXXXXXXXX	→ Tupla 2
.....	.....	.....	.....	
XXXXXX	XXXXXX	.....	XXXXXXXXXX	→ Tupla n

Figura 12. Representación de una relación en forma de tabla. Fuente [11]

En la figura 12, podemos apreciar el nombre (el nombre que se va asignar a la tabla), a cada columna de la tabla se le denomina atributo y cada fila de los atributos son llamadas tuplas, las cuales contienen los valores de cada una de los atributos para cada relación.

El objetivo principal de las bases de datos relacionales, es evitar la repetición de datos, es decir, columnas repetidas en diferentes tablas (redundancia de datos); la redundancia en las bases de datos implica ocupar más espacio de almacenamiento, además se dificulta la gestión de registros.

El problema principal que viene a solucionar la base de datos es precisamente evitando la redundancia de datos guardando los datos en distintas tablas, pero con una relación en específico y al momento de gestionar los datos actúa como si fuera una misma tabla.

De acuerdo con (Piattini, 1999) se deben considerar los siguientes aspectos:

### **Ventajas**

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

### **Características**

Para que la estructura de las tablas cumpla las leyes de la teoría relacional deben satisfacerse de las siguientes condiciones:

- ✓ Todos los registros de la tabla 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 una tabla cada nombre de campo debe ser distinto.
- ✓ Los registros de una misma tabla tienen que diferenciarse, al menos, en el contenido de alguno de sus campos no puede a ver dos registros idénticos.

- ✓ Los registros de una tabla pueden estar dispuestos en cualquier orden.
- ✓ El contenido de cada campo está delimitado por un rango de valores posibles.
- ✓ Permite la creación de nuevas tablas a partir de las existentes 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).

#### **5.4.- BASES DE DATOS RELACIONALES**

El modelo de datos relacional, fue introducido por Tedd Codd de IBM research en 1970 y pronto acaparo gran atención debido a su simplicidad y a sus fundamentos matemáticos.

Una base de datos relacional, está compuesta de tablas que están relacionadas entre sí de alguna manera, directa o indirectamente. Si pensamos en términos de objetos, las tablas representan los objetos, y estos objetos/tablas están conectados utilizando lo que se denomina relaciones. Un ejemplo, podría ser un sistema de pedidos, en donde tenemos productos, clientes, pedidos, etc. En un sistema de pedidos simple tendremos tablas para almacenar pedidos, clientes y productos. Un cliente tiene cero o más pedidos, un pedido está compuesto de uno o más productos, por lo tanto, podemos ver lo sencillo que resulta crear relaciones entre las tablas. [12]

Base de datos, es una colección de datos persistentes que pueden compartirse e interrelacionarse [15]

Persistente: los datos residen en un disco magnético ya que algunos datos suelen usarse de manera continua, la persistencia de estos datos depende de la importancia del uso deseado, es decir que los datos no deben almacenarse eternamente cuando estos dejan de ser relevantes.

Compartir: significa que una base de datos puede tener múltiples usos y usuarios, por lo tanto proporciona diversas funciones así como puede ser utilizada de forma simultánea por dos o más usuarios.

Interrelación: significa que los datos almacenados como unidades separadas se pueden conectar para mostrar un cuadro completo. La información se almacena en diferentes tablas relacionadas por un campo en común llamados llaves, los cuales permiten realizar consultas de forma más completa.

#### **5.4.1.- Modelo entidad relación**

Las bases de datos relacionales se basan en el modelo relacional y usan un conjunto de tablas para representar los datos como las relaciones entre ellos. [16]

**Modelado de datos** técnica para organizar y documentar los datos de un sistema. Alguna veces llamada *modelo de bases de datos*.

El modelado de datos de un sistema de información se utiliza para definir los requerimientos de negocio de una base de datos. Un modelo de datos al final se implementa como una base de datos. En la actualidad existe software que permite realizar modelos de bases de datos y posteriormente implementar la base de datos diseñada previamente.

Hay varias notaciones para el modelado de datos. El modelo real se designa como un diagrama de entidad relación (***entity relationship diagram, ERD***), por que diseña los datos en términos de las entidades y las relaciones descritas por los datos. [4]

##### **5.4.1.1.- Entidades**

Una entidad es algo acerca de lo cual el negocio necesita almacenar datos. De esta forma se asigna un concepto abstracto a una forma. Por ejemplo, la entidad refacción representa a todos los productos de la refaccionaria que son capturados en la base de datos. Así, una identidad identifica las clases específicas de entidades y es distinguible de las otras entidades.

Las categorías de entidades (y ejemplos por mencionar algunos) incluyen:

Personas: CLIENTE, EMPLEADO, VENDEDOR, PROVEEDOR, cabe mencionar que la entidad persona puede estar representada por un individuo, organización o grupo.

Lugares: EDIFICIO, CUARTO, REGIÓN.

Objetos: PRODUCTO, LIBRO, VEHÍCULO, MODELO DE VEHÍCULO.

Eventos: VENTA, DEVOLUCIÓN, FACTURA, ACTUALIZACIÓN.

Conceptos: CUENTA, CAPACITACIÓN.

La ocurrencia individual de una entidad se le llama instancia de entidad, es decir que la entidad empleado puede tener múltiples instancias: Juan, Pedro, etc. Sin embargo, el modelado de datos no se centra en las instancias si no en las entidades ya que cada empleado se describe por piezas similares de datos.

En otras palabras una entidad corresponde a una tabla y no a una fila en el ambiente relacional. Sin embargo, cuando el modelo entidad relación se refiere a una fila en específico se dice que es una estancia de entidad u ocurrencia de entidad. En el modelo Pata de Gallo, una entidad está representada por un rectángulo que contiene el nombre de la entidad. El nombre de la entidad, un sustantivo, casi siempre se escribe con mayúsculas. El modelo ERD Pata de Gallo, se ilustrará a lo largo del capítulo mediante MySQL Workbench 5.1 OSS. Este software permite un diseño ilustrativo de una base de datos y a su vez transforma el diseño en la base de datos final del sistema.

#### **5.4.1.2.- Atributos**

Son piezas de datos que describen a una entidad, es decir, dentro del sistema de venta de refacciones se tiene la entidad CLIENTE, de esta entidad Cliente se requiere guardar datos como su clave RFC (Registro federal de contribuyente), Nombre, Dirección etc.

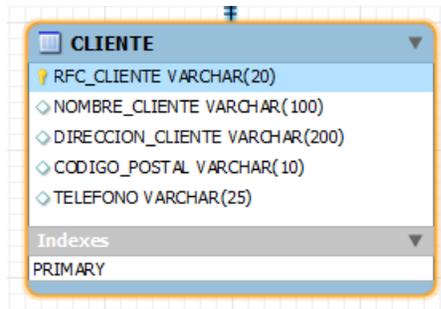


Figura 13. Atributos de la entidad CLIENTE. Fuente elaboración propia

Tipo de datos: define la clase de datos que puede guardarse en ese atributo. La escritura de datos resulta muy familiar para los que escriben programas de computadora ya que la declaración de variables es muy común en los lenguajes de programación. El cuadro 1 muestra los diferentes tipos de datos.

Dominio: el dominio de un atributo define que valores puede adoptar en realidad un atributo. Especifica como debe estar representado el tipo de dato. El cuadro 1 muestra como pueden expresarse los dominios lógicos para cada tipo de datos.

La figura 13, define el tipo de dato para el atributo RFC\_CLIENTE como VARCHAR, que significa por sus siglas en Ingles (variable de tipo carácter); donde su dominio debe ser menor o igual a veinte caracteres. VARCHAR puede también interpretarse como tipo de dato TEXTO.

<b>Dominios lógicos representativos para tipos de datos lógicos.</b>		
<b>Tipo de datos</b>	<b>Dominio</b>	<b>Ejemplos</b>
NÚMERO	Para los enteros, especifique el rango:  {mínimo - máximo}	{10-99}
	Para los números reales, especifique el rango ya la precisión: {mínimo. precisión-máximo. Precisión	{1.00-799.999}
TEXTO	TEXTO (tamaño máximo del atributo)	

	Generalmente los valores reales son infinitos; sin embargo, los usuarios pueden identificar ciertas restricciones en la narración.	TEXTO(30)
MEMO	No es aplicable. No hay restricciones lógicas de tamaño o contenido.	No es aplicable
FECHA	Variación del formato MMDDYYYY. Para anotar al 2000, no abrevie el año como YY. Los caracteres de formateado rara vez se almacenan; por lo tanto, no incluyan guiones ni diagonales.	MMDDYYYY MMYYYY YYYY
TIEMPO	Para los horarios AM/PM: HHMMT  o  Para los horarios militares HHMM	HHMMT   HHMM
SI/NO	{SÍ, NO}	{SÍ, NO}  {ENCENDIDO, APAGADO}
CONJUNTO DE VALORES	{Valor#1, valor#2, ..., valor#n}  o  {tabla de códigos y significados}	{ALUMNO DE PRIMER INGRESO, ALUMNO DE SEGUNDO AÑO, ALUMNO DE TERCER AÑO, ALUMNO DE ULTIMO AÑO}  { PI= ALUMNO DE PRIMER INGRESO  SA= ALUMNO DE SEGUNDO AÑO  TA= ALUMNO DE TERCER AÑO  UA= ALUMNO DE ULTIMO AÑO}
IMAGEN	No es aplicable; sin embargo, cualquiera de las características conocidas de las imágenes finalmente resultaran útiles para los diseñadores.	No es aplicable

Cuadro 1. Dominios lógicos representativos para tipos de datos lógicos. Fuente [4]

**Atributos compuestos.** Un atributo compuesto es aquel que puede subdividirse en atributos adicionales. Por ejemplo, el atributo dirección puede subdividirse en calle, municipio, estado y código postal. Es decir, para detallar un atributo compuesto se utiliza una serie de atributos simples.

**Atributos simples.** Son aquellos atributos que se pueden subdividir como son edad, sexo, estado civil etc.

**Atributos de un solo valor.** Es aquel que puede tener solamente un valor. Por ejemplo una persona puede tener solamente un número de seguro social.

**Atributos de valores múltiples.** Son aquellos que pueden tener muchos valores. Por ejemplo una persona puede tener varios números telefónicos, cada uno con su propio número.

**Atributos derivados.** Un atributo derivado no se tiene que guardar físicamente en la base de datos, en su lugar este atributo se deriva por medio de un algoritmo. Por ejemplo la edad de una persona, se determina mediante la diferencia del año en curso menos el año de nacimiento.

#### **5.4.1.3.- Claves primarias**

Las claves primarias (atributos de clave), están representadas dentro de las ilustraciones con una llave al lado izquierdo del atributo.

Idealmente una clave primaria está formada por un sólo atributo, el cual sirve para identificar como única cada instancia de la entidad; significa que el atributo seleccionado como clave primaria no puede contener dos registros idénticos.

La figura 14, se define como clave primaria el atributo ID\_PROVEEDOR dentro de la entidad PROVEEDOR; el diseñador de esta base de datos podría haber utilizado el RFC como clave primaria para esta entidad, pues esta nunca se repite, sin embargo, en ocasiones no toda empresa cuenta con un RFC; en este caso, se

desarrolla un nuevo atributo llamado ID\_PROVEEDOR que identifica como único a cada proveedor dentro del sistema sin necesidad de contar con el RFC.

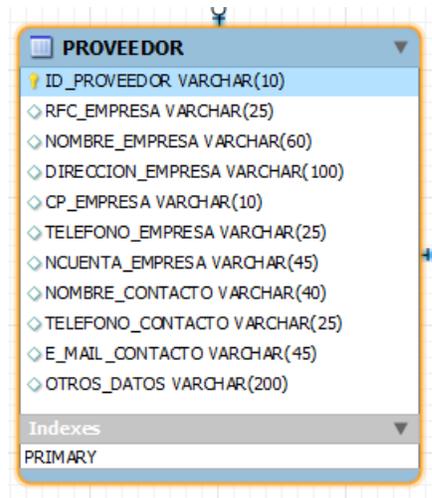


Figura 14. Clave primaria para la entidad PROVEEDOR. Fuente elaboración propia

Algunas veces se utiliza una **Clave compuesta** es decir, una clave primaria formada por más de un atributo. El administrador de una base de datos de venta de refacciones para automóviles puede que decida identificar cada instancia de entidad VENTA. Mediante una clave primaria compuesta de N\_REMISON y LINEA, en lugar de utilizar ID\_VENTA. Uno u otro enfoque identifica de manera única cada instancia de entidad. Dada la estructura actual de la tabla VENTA, mostrada en el cuadro 2, ID\_VENTA es la clave primaria y la combinación de N\_REMISON y LINEA es una clave candidato apropiada. Si el atributo ID\_VENTA se elimina de la entidad VENTA, la clave candidato se convierte en una clave primaria compuesta aceptable.

Tabla venta			
ID_venta	N_remision	Linea	Clave_refaccion
129	1	1	BKR5EGP
130	1	2	RC12YC
131	2	1	BKR5EGP
132	2	2	BM6A
133	3	1	N12Y

Cuadro 2. Componentes y contenido de la tabla VENTA. Fuente elaboración propia

La figura 15, muestra como representa la entidad VENTA en el lado izquierdo utilizando como identificador una clave primaria formada por un solo atributo y en el lado opuesto la misma entidad VENTA identificada por una clave compuesta.

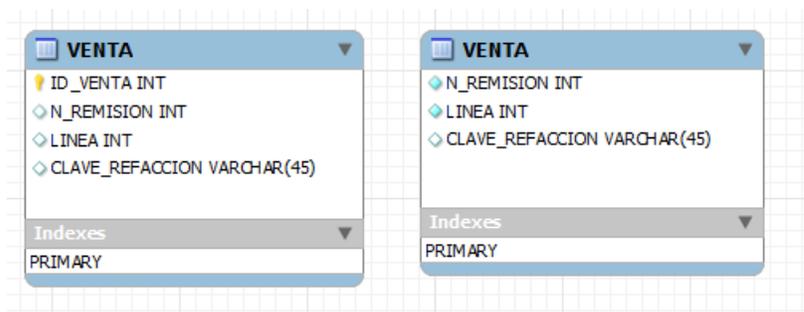


Figura 15. Claves primarias. Fuente elaboración propia

#### 5.4.1.4.- Relaciones

La **relación**, es una asociación entre entidades. [17]

Las entidades y los atributos, no existen en forma aislada si no que se relacionan entre sí para lograr los objetivos de negocio.

Una **relación**, es una asociación natural de negocios que existe entre una o más entidades. La relación puede representar un evento que enlaza a las entidades o meramente una afinidad lógica que existe entre las entidades. [4]

Cada relación se identifica mediante un nombre en particular. El nombre de la relación es un verbo activo o pasivo.

Las relaciones son bidireccionales; es decir, operan en ambas direcciones. Considerando por ejemplo, las entidades CLIENTE Y FACTURA\_VENTA. Podemos hacer las siguientes afirmaciones de negocios que enlazan a las refacciones y las ventas.

- ✓ Un CLIENTE puede generar muchas FACTURAS.
- ✓ Cada FACTURA es generada por un solo CLIENTE.

Gráficamente podemos ilustrar a esta asociación entre un CLIENTE y una FACTURA\_VENTA. La línea que conecta las dos entidades representa una relación. El nombre que describe la relación se encuentra en medio de las dos entidades. Es fácil ver que esta relación puede clasificarse como 1:M. Véase la figura 16.

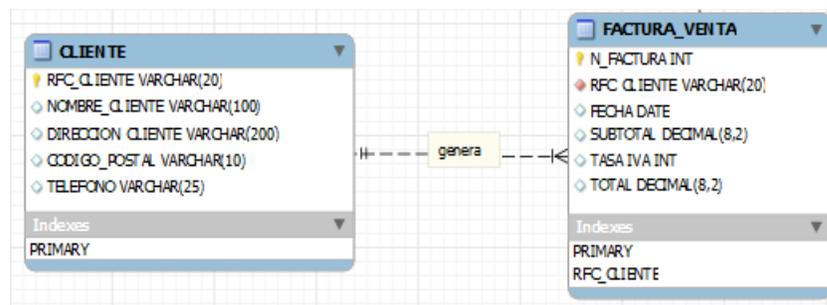


Figura 16. Una relación de uno a muchos. Fuente elaboración propia

#### 5.4.1.5.- Conectividad y Cardinalidad

El término **conectividad**, se utiliza para describir la clasificación de las relaciones. Las relaciones entre entidades se clasifican como uno a uno, uno a muchos y muchos a muchos

**Relación uno a uno**, se representa cuando existe una relación como su nombre lo indica uno a uno. Por ejemplo, la asignación de un automóvil a un empleado, sucede que un empleado no puede tener más de un auto asignado, así como un

auto está asignado a un solo empleado. En la figura 17 se muestra la relación entre EMPLEADO y AUTO.

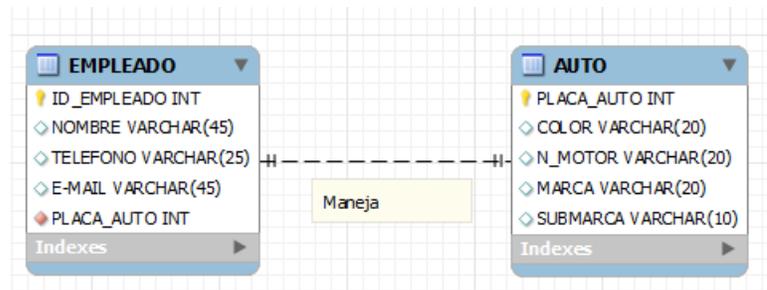


Figura 17. Una relación uno a uno. Fuente elaboración propia

**Relación uno a muchos**, significa que una entidad puede relacionarse con cualquier cantidad de registros de otra entidad, tomando como ejemplo la figura 16 se afirma que la entidad cliente puede generar muchas facturas, por otro lado muchas facturas pueden ser generados por un solo cliente por consiguiente una factura pertenece a un solo cliente.

**Relación muchos a muchos**, tomando como ejemplo la relación ALUMNO-MATERIA, se afirma que cualquier cantidad de alumnos pueden inscribirse en cualquier cantidad de materias, en otras palabras un alumno se inscribe en una o más materias y por consiguiente una materia puede tener inscritos cualquier cantidad de alumnos. Este tipo de relaciones en la práctica suelen simplificarse tal como se ilustra en la figura 18.

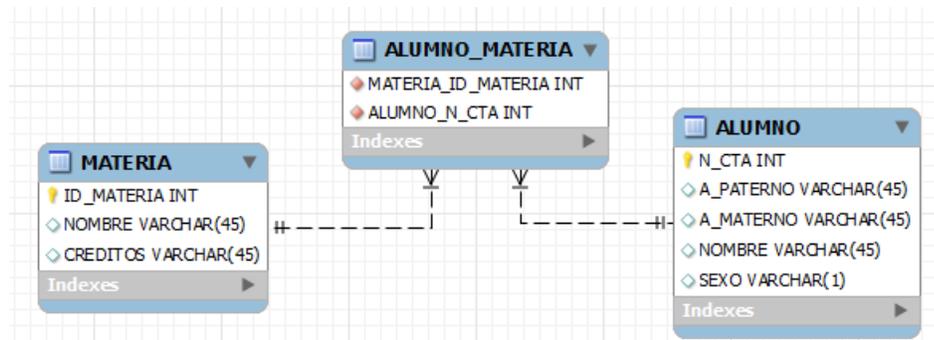


Figura 18. Una relación muchos a muchos. Fuente elaboración propia

La **cardinalidad**, expresa el número específico de ocurrencias de entidad asociadas con una ocurrencia de la entidad relacionada. Conocer el número específico de ocurrencias de entidad relacionada. En el modelo de pata de gallo no suele ilustrarse la cardinalidad, ya que el DBMS no puede manejar las cardinalidades a nivel tabla; sin embargo, conocer el número de ocurrencias de entidad mínimo y máximo es muy útil a nivel de software de aplicación.

## 5.5.- SQL

SQL, es el lenguaje estándar para trabajar con bases de datos relacionales y es soportado prácticamente por todos los productos en el mercado. Originalmente, SQL fue desarrollado en IBM Research a principios de los años setenta, fue implementado por primera vez a gran escala en un prototipo de IBM llamado System R, y posteriormente en numerosos productos comerciales de IBM y de muchos otros fabricantes, el nombre oficial es Estándar Internacional del Lenguaje de Bases de Datos SQL en 1992. [18]

Lenguaje estándar de bases de datos que incluye sentencias para la definición, la manipulación y el control de la base de datos. [15]

El lenguaje de consulta estructurada (SQL, por sus siglas en inglés), es un lenguaje estándar de la industria soportable por la mayoría de los sistemas gestores de bases de datos, se puede usar para definir tablas, relaciones entre ellas, restricciones de integridad (reglas que definen datos permitidos), autorización de privilegios (reglas que restringen el acceso a datos) [15].

En una base de datos todo acaba en SQL. Independientemente de 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. [19]

Debido a la diversidad de lenguajes y de bases de datos existentes, la manera de comunicar sería complicado 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, que permite trabajar con cualquier tipo de lenguaje en combinación con cualquier tipo de base 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 un lenguaje de cuarta generación. Es decir, en este lenguaje se indica que información se desea obtener o procesar, pero no como se debe hacer. Es labor interna del sistema elegir la forma más eficiente de llevar a cabo la operación ordenada por el usuario.

Aunque SQL es considerado como un sublenguaje por su naturaleza no procesa datos, no obstante es un lenguaje completo que permite crear, mantener, obtener y manipular objetos en BD. Un método común utilizado para clasificar las sentencias SQL es dividirlos de acuerdo a las funciones que realizan. Con base en este método, SQL se pueden separar en tres tipos de declaraciones: Lenguaje de definición de datos, Lenguaje de manipulación de datos... [20].

El lenguaje SQL esta subdividido en dos el DDL y el DML.

### **5.5.1.- Lenguaje de definición de datos (DDL)**

SQL emplea los términos tabla, fila y columna en vez de relación, tupla y atributo, respectivamente. Nosotros usaremos de manera indistinta los términos correspondientes. Las instrucciones SQL2 para la definición de datos son CREATE; ALTER y DROP; [10]

(DDL) data definición language, en español lenguaje de definición de datos, que se encarga de la creación o modificación de la estructura de los objetos de una base de datos. En otras palabras el DDL, es el lenguaje que se utiliza para crear bases de datos y tablas, para modificar sus estructuras y establecer permisos y privilegios. Permite trabajar con las estructuras de datos en vez de trabajar con los datos en sí mismos. Se manejan las instrucciones: CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX. DROP INDEX.

### 5.5.1.1.- Definición de una base de datos a través MYSQL

MYSQL, es un gestor de bases de datos. Una aplicación capaz de manejar los datos de forma veloz y eficiente a través de bases de datos relacionales. Para poder utilizar dicha aplicación se utiliza el lenguaje estándar de programación SQL.

Para fines ilustrativos se utilizaran tablas sencillas a fin de comprender la implementación de una base de datos, algunos ejemplos de cómo crear una base de datos así como realizar consultas, sin embargo, no es posible abarcar todas las funciones de MYSQL, ya que resultaría demasiado extenso y este no es el fin de esta tesis. Para mayor información se recomienda consultar el manual de MYSQL.

Una vez que hemos ingresado a la consola de mysql y accedido por medio de una contraseña (según sea el caso) podemos proseguir a crear nuestra base de datos ejemplo.

#### 5.5.1.1.1.- Como crear una base de datos

Para crear una base de datos se usa la sentencia CREATE DATABASE:

```
mysql> CREATE DATABASE REFACCIONARIA;
Query OK, 1 row affected (0.03 sec)
mysql>
```

Mediante la sentencia SHOW, se encuentran las bases de datos que existen actualmente en el servidor: [21]

```
mysql> SHOW DATABASES;
+-----+
| Database          |
+-----+
| mysql             |
| REFACCIONARIA    |
| test              |
+-----+
```

```
3 rows in set (0.00 sec)
```

Es posible que la lista de bases de datos que se muestren sean distintas en cada caso, sin embargo, la base de datos “mysql” y “test” estarán entre ellas.

Para seleccionar una base de datos se utiliza el comando USE, que no es exactamente una sentencia SQL, sino más bien de una opción de MySQL:

```
mysql> USE REFACCIONARIA;  
Database changed  
mysql>
```

#### **5.5.1.1.2. - Como crear una tabla**

La instrucción CREATE TABLE, sirve para especificar una nueva relación dándole un nombre y especificando sus atributos y restricciones. [10]

La sentencia CREATE TABLE, nos permite crear tablas nuevas dentro de la base de datos que hemos seleccionado mediante el comando USE, en este caso la base de datos de la REFACCIONARIA. La sintaxis de esta sentencia resulta muy compleja, ya que existen variedad de opciones y se tienen muchas posibilidades diferentes a la hora de crear una tabla.

En su forma más simple, la sentencia CREATE TABLE creará una tabla con las columnas que indiquemos. Crearemos, como ejemplo, una tabla que nos permita almacenar los nombres de los empleados y sus fechas de nacimiento. Para esto se debe indicar el nombre de la tabla, nombre de cada columna y tipos de datos para cada columna.

La forma general de la instrucción CREATE es la siguiente [22]:

```
CREATE TABLE base-table-name  
(columna1- definición,  
 [columna2- definición],  
 [columna-n- definición],  
 [[lave-primaria- definición]]);
```

Donde la columna - definición toma la forma de:

Columna- nombre data-type [NULL | NOT NULL [WITH DEFAULT | UNIQUE]]

La sintaxis anterior permitirá crear una tabla nueva, cabe mencionar que en el siguiente ejemplo sólo se define una tabla en su forma simple, sin precisar aun alguna columna como clave primaria, sin embargo, más adelante en este mismo apartado se verán ejemplos de asignación de claves primarias al crear una tabla.

```
mysql> USE REFACCIONARIA
Database changed
mysql> CREATE TABLE EMPLEADO
->(ID_Empelado INT,
->Nombre VARCHAR(40),
->Fecha DATE);
Query OK, 0 rows affected (0.53 sec)
mysql>
```

Hemos creado una tabla llamada "EMPLEADO" con tres columnas: "ID\_Empelado" que permite valores enteros, "Nombre" que puede contener cadenas de hasta 40 caracteres y "Fecha" de tipo fecha.

Para poder especificar el tipo de dato que va a contener cada columna dentro de una tabla, se utilizan los tipos de datos predefinidos por el lenguaje SQL. En el cuadro 3 se muestran tipos de datos predefinidos.

<b>Tipos de datos predefinidos.</b>	
<b>Tipo de datos</b>	<b>Descripción</b>
CHAR (longitud)	Cadena de caracteres de longitud fija.
CHAR VARYING (longitud)	Cadena de caracteres de longitud variable.
BIT (longitud)	Cadena de bits de longitud fija.
BIT BARYING (longitud)	Cadena de bits de longitud variables.

NUMERIC (precisión escala)	Números decimales con tantos dígitos como indique la precisión y tantos decimales como indique la escala
DECIMAL (precisión escala)	Números decimales con tantos dígitos como indique la precisión y tantos decimales como indique la escala
INTEGER	Números enteros
SMALLINT	Números enteros pequeños
REAL	Números con coma flotante con precisión predefinida.
FLOAT (precisión)	Números con coma flotante con la precisión específica.
DOUBLE PRECISION	Números con coma flotante con más precisión predefinida que la del tipo REAL
DATE	Fechas, están compuestas de YEAR año, MONTH mes, DAY día.
TIME	Horas, están compuestas de HOUR hora, MINUT minutos, SECOND segundos.
TIMESTAMP	Fechas y horas están compuestas de YEAR año, MONTH mes, DAY día, HOUR hora, MINUT minutos, SECOND segundos.

Cuadro 3. Tipos de datos predefinidos. Fuente [23]

Para conocer las tablas contenidas en la base de datos actual (por ejemplo, si no se está seguro del nombre de una tabla) se usa el siguiente comando: [21]

```
mysql> SHOW TABLES;
+-----+
| Tables_in_REFACCIONARIA |
+-----+
| EMPLEADO |
+-----+
1 row in set (0.01 sec)
```

Como se ha mencionado anteriormente, se tienen mucho más opciones en el momento de definir las columnas de las tablas. Además de seleccionar el tipo de variable que contendrá la columna también es posible definir valores por defecto para una columna, permitir que contengan o no valores nulos, además de crear una clave primaria para una tabla.

#### 5.5.1.1.3.- Definición de valores nulos

Al definir cada columna es posible decidir si podrá o no contener valores nulos.

Dado que SQL permite NULL como un valor para un atributo, se puede especificar la *restricción* NOT NULL si es que no se permite el valor nulo para un determinado atributo. Esta restricción se debe especificar siempre para los atributos de clave primaria de toda relación, así como para cualquier otro atributo cuyos valores no deban ser nulos. [10]

De esta forma aquellas columnas que sirven como clave primaria no deben permitir valores nulos, por lo tanto, si se define una columna como clave primaria automáticamente se impide que pueda contener valores nulos; sin embargo, no es el único caso en que se impide la asignación de valores nulos para una columna.

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

```
mysql> CREATE TABLE PROVEEDOR
->(ID_Proveedor INT NOT NULL,
->Nombre CHAR(40) , Telefono INT NULL);
Query OK, 0 rows affected (0.98 sec)
```

#### 5.5.1.1.4.- Definición de valores por defecto

También es posible definir un *valor por omisión* para un atributo añadiendo la cláusula DEFAULT <valor> a la definición de un atributo [10]

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

Si una columna puede tener un valor nulo, y no se especifica un valor por defecto, se usará NULL como valor por defecto. En el ejemplo anterior, el valor por defecto para ID\_Proveedor es NULL.

Por ejemplo, si se requiere para el negocio de refacciones, al momento de dar de alta una refacción no se especifica la existencias en el inventario, por defecto sea existencias=1.

```
mysql> CREATE TABLE REFACCION
->(Clave_Refaccion CHAR(20) NOT NULL,
->Nombre CHAR(20) ,
->Existencias INT NULL DEFAULT 1);
Query OK, 0 rows affected (0.09 sec)
```

#### 5.5.1.1.5.- Definición de claves primarias

La cláusula PRIMARY KEY, especifica uno o más atributos que constituyen la clave primaria de una relación. [10]

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

La sintaxis para definir claves primarias es:

definición\_columnas

| PRIMARY KEY (index\_nombre\_col,...)

Por ejemplo, si se requiere definir como clave primaria la columna Clave\_Refaccion de la tabla de REFACCION, crearemos la tabla así:

```
mysql> CREATE TABLE REFACCION
->(Clave_Refaccion CHAR(20) NOT NULL PRIMARY KEY,
->Nombre CHAR(20) ,
->Existencias INT NULL DEFAULT 1);
Query OK, 0 rows affected (0.20 sec)
```

Usar *NOT NULL PRIMARY KEY* equivale a *PRIMARY KEY, NOT NULL KEY* o sencillamente *KEY*.

Existe una sintaxis alternativa para crear claves primarias, que en general es preferible, ya que es más potente. De hecho, la que se ha explicado es un alias para la forma general, que no admite todas las funciones (como por ejemplo, crear claves primarias sobre varias columnas).

#### **5.5.1.1.6.- Definición de columnas autoincrementadas**

Al insertar una nueva fila, su valor se calcula automáticamente, tomando el valor más alto de la columna y sumándole una unidad. Esto permite crear, de una forma sencilla, una columna con un valor único para cada fila de la tabla.

Generalmente, estas columnas se usan como claves primarias.

```
mysql> CREATE TABLE CLIENTE
->(ID_Cliente INT AUTO_INCREMENT PRIMARY KEY,
-> Nombre CHAR(40) NOT NULL);
Query OK, 0 rows affected (0.9 sec)
```

#### **5.5.1.1.7.- Definición de índices**

Se tienen tres tipos de índices. El primero corresponde a las claves primarias, que como se trató con anterioridad, también se pueden crear en la parte de definición de columnas.

El ejemplo anterior para crear claves primarias, utilizando índice

```
mysql> CREATE TABLE CLIENTE
->(ID_Cliente INT NOT NULL,
-> Nombre CHAR(40),
-> PRIMARY KEY (ID_Cliente));
Query OK, 0 rows affected (0.12 sec)
```

Pero esta forma tiene más opciones, por ejemplo, entre los paréntesis podemos especificar varios nombres de columnas, para construir claves primarias compuestas por varias columnas:

```
mysql> CREATE TABLE REMISION (  
    -> Remision_Numero INT NOT NULL,  
    -> Linea INT NOT NULL,  
    -> Articulo CHAR(30),  
    -> PRIMARY KEY (Remision_Numero, Linea));  
Query OK, 0 rows affected (0.09 sec)
```

El segundo tipo de índice permite definir índices sobre una columna, o sobre partes de columnas. Para definir estos índices se usan indistintamente las opciones *KEY* o *INDEX*.

```
mysql> CREATE TABLE FACTURA(  
    -> Numero INT,  
    -> RFC_Cliente CHAR(20),  
    -> INDEX (Numero));  
Query OK, 0 rows affected (0.09 sec)
```

O su equivalente:

```
mysql> CREATE TABLE FACTURA (  
    -> Numero INT,  
    -> RFC_Cliente CHAR(20),  
    -> KEY (Numero));  
Query OK, 0 rows affected (0.09 sec)
```

También podemos crear un índice sobre parte de una columna:

```
mysql> CREATE TABLE FACTURA (  
    -> Numero INT,  
    -> RFC_Cliente CHAR(20),  
    -> INDEX (RFC_Cliente(4)));  
Query OK, 0 rows affected (0.09 sec)
```

Este ejemplo usará sólo los cuatro primeros caracteres de la columna "RFC\_Cliente" para crear el índice.

La tercera forma de definir índices es con la cláusula UNIQUE

La cláusula UNIQUE especifica claves alternativas (secundarias). [10]

La diferencia entre un índice único y uno normal es que en los únicos no se permite la inserción de filas con claves repetidas. La excepción es el valor *NULL*, que sí se puede repetir.

```
mysql> CREATE TABLE VENDEDOR(  
    -> ID_Vendedor INT,  
    -> Nombre CHAR(19),  
    -> UNIQUE (nombre));  
Query OK, 0 rows affected (0.09 sec)
```

Una clave primaria equivale a un índice de clave única, en la que el valor de la clave no puede tomar valores *NULL*. Tanto los índices normales como los de claves únicas sí pueden tomar valores *NULL*.

Los índices sirven para optimizar las consultas y las búsquedas de datos. Mediante su uso es mucho más rápido localizar filas con determinados valores de columnas, o seguir un determinado orden. La alternativa es hacer búsquedas secuenciales, que en tablas grandes requieren mucho tiempo.

#### **5.5.1.1.8.- Definición de claves foráneas**

La integridad referencial se especifica mediante la clave FOREIGN KEY. [10]

En MySQL es posible definir claves foráneas. Las claves foráneas relacionan las tablas, las cuales hacen referencia de una clave foránea de una tabla que también existe dentro de otra tabla como clave principal. Ejemplo:

```
mysql> CREATE TABLE CLIENTE (  
    -> RFC_Cliente VARCHAR(20) PRIMARY KEY,
```

```

-> Nombre VARCHAR(100));
Query OK, 0 rows affected (0.13 sec)

mysql> CREATE TABLE FACTURA(
-> Numero_Factura INT KEY,
-> RFC_Cliente VARCHAR(20) NOT NULL REFERENCES CLIENTE
(RFC_Cliente)
-> ON DELETE CASCADE ON UPDATE CASCADE); (1)
Query OK, 0 rows affected (0.13 sec)

```

Con el ejemplo anterior se asume que existe para cada factura un Numero\_factura, y esta factura corresponde a un cliente, cuyos datos del cliente son posibles de localizar a través del RFC\_Cliente que existe en la tabla FACTURA como una clave foránea y Existe como clave primaria dentro de la tabla CLIENTE.

Con dicho ejemplo se pueden hacer las siguientes aseveraciones lógicas de la base de datos:

Dentro de la tabla CLIENTE, no puede existir un registro con el RFC\_Cliente duplicado, sin embargo; en la tabla facturas Puede aparecer el RFC\_Cliente n cantidad de veces ya que es una característica de las claves foráneas.

Se pueden violar una restricción de integridad referencial cuando se insertan o eliminan tuplas o cuando se modifica un valor de atributo de clave externa. [10] Las opciones incluyen SET NULL; CASCADE y SET DEFAULT. Una opción debe calificarse bien con ON DELETE u ON UPDATE. [10]

ON DELETE <opción>, indica que acciones se deben realizar en la tabla actual si se borra una fila en la tabla referenciada.

ON UPDATE <opción>, indica que acciones se deben realizar en la tabla actual si existen modificaciones de claves.

Existen cinco opciones diferentes para las dos opciones anteriores, de modo que se explica lo que hace cada una de ellas:

**RESTRICT:** esta opción impide eliminar o modificar filas en la tabla referenciada si existen filas con el mismo valor de clave foránea.

**CASCADE:** borrar o modificar una clave en una fila en la tabla referenciada con un valor determinado de clave, implica borrar las filas con el mismo valor de clave foránea o modificar los valores de esas claves foráneas.

**SET NULL:** borrar o modificar una clave en una fila en la tabla referenciada con un valor determinado de clave, implica asignar el valor *NULL* a las claves foráneas con el mismo valor.

**NO ACTION:** las claves foráneas no se modifican, ni se eliminan filas en la tabla que las contiene.

**SET DEFAULT:** borrar o modificar una clave en una fila en la tabla referenciada con un valor determinado implica asignar el valor por defecto a las claves foráneas con el mismo valor.

#### **5.5.1.1.9.- Como eliminar una tabla**

Si ya no se necesita una relación de base de un esquema, podemos eliminarla utilizando la opción **DROP TABLE**. [10]

Ejemplo, si ya no se requiere información acerca de la tabla empleado, o bien se desea volver a definir la tabla, se elimina de la siguiente manera

```
mysql> DROP TABLE Empleado;  
Query OK, 0 rows affected (0.75 sec)
```

Se pueden añadir las palabras *IF EXISTS* para evitar errores si la tabla a eliminar no existe.

```
mysql> DROP TABLE Empleado;  
ERROR 1051 (42S02): Unknown table 'Empleado'
```

```
mysql> DROP TABLE IF EXISTS Empleado;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

#### **5.5.1.1.10.- Como eliminar una base de datos**

De modo parecido, se pueden eliminar bases de datos completas, usando la sentencia DROP DATABASE.

La sintaxis es simple:

```
DROP TABLE [IF EXISTS] tbl_name [, tbl_name] ...
```

Hay que tener sumo cuidado, ya que al borrar cualquier base de datos se elimina también cualquier tabla que contenga. Ejemplo:

```
mysql> DROP DATABASE IF EXISTS REFACCIONARIA;
Query OK, 1 row affected (0.11 sec)
```

#### **5.5.2.- Lenguaje de manipulación de datos (DML)**

Lenguaje de manipulación de datos (en inglés *Data Manipulation Language* o *DML*). Incluye aquellas sentencias que sirven para manipular o procesar los datos, como por ejemplo la inserción, borrado, modificación o actualización de datos en las tablas. [24].

(DML) data manipulación language, es un lenguaje proporcionado por el sistema de gestión de bases de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de datos. Se utiliza para manipular los datos existentes dentro de una base de datos existente. El lenguaje de manipulación se concentra en las instrucciones SELECT, INSERT, DELETE y UPDATE, permiten trabajar con los datos de las tablas.

En otras palabras el DDL, es el lenguaje que permite la creación de una base de datos, creación de tablas dentro de la base de datos, definir la estructura de las tablas, así como los permisos y privilegios para la base de datos; mientras que el

DML permite ingresar datos en las tablas que se crearon así como realizar consultas y cambios en dichos datos.

### 5.5.2.1.- Manipulación de datos a través de MYSQL

Disponemos de varias sentencias para verificar o consultar tablas y sus características. Podemos ver las tablas que existen dentro de una base de datos seleccionada mediante la sentencia SHOW TABLES

```
mysql> use refaccionaria;
```

```
Database changed
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_refaccionaria |
+-----+
| cliente                  |
| proveedor                |
| refaccion                |
| vendedor                 |
+-----+
4 rows in set (0.05 sec)
```

Mediante la sentencia DESCRIBE es posible ver la estructura de una tabla

```
mysql> DESCRIBE CLIENTE;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| RFC_CLIENTE   | varchar(20)   | NO   | PRI |          |       |
| NOMBRE_CLIENTE | varchar(100)  | YES  |     | NULL    |       |
|               |               |      |     |         |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.19 sec)
```

### 5.5.2.1.1.- Consultar datos

Una de las operaciones básicas del lenguaje SQL consiste en seleccionar filas de una o más tablas que cumplan con ciertas condiciones de selección.

La forma básica de la sentencia SELECT, en ocasiones denominada correspondencia o **bloque select-from-where**, consta de las tres cláusulas SELECT; FROM; WHERE y tienen la siguiente forma. [10]

```
SELECT <lista de atributos>
FROM <lista de tablas>
WHERE <condicion>
```

```
mysql> SELECT * FROM CLIENTES WHERE NOMBRE_CLIENTE="RUBEN";
+-----+-----+
| FECHA   | RFC       |
+-----+-----+
| RUBEN   | AARR671010 |
+-----+-----+
1 row in set (0.03 sec)
```

```
mysql> SELECT * FROM VENTA WHERE fecha>="1986-01-01";
+-----+-----+
| FOLIO   | FECHA     |
+-----+-----+
| 001     | 2001-12-02 |
| 002     | 1993-02-10 |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM VENTA
      -> WHERE fecha>="1986-01-01" AND fecha < "2000-01-01";
+-----+-----+
| FOLIO   | FECHA     |
```

```

+-----+-----+
| 002    | 1993-02-10 |
+-----+-----+
1 row in set (0.00 sec)

```

También se puede aplicar lógica booleana para crear expresiones complejas. Disponemos de los operadores *AND*, *OR*, *XOR* y *NOT*. Véase cuadro 4.

<b>Operaciones de condición where.</b>	
<b>Operador</b>	<b>Descripción</b>
=	Igualdad
>	Estrictamente superior
>=	Superior o igual
<	Estrictamente inferior
<=	Inferior o igual.
< > o !=	Diferente
BETWEEN, min, AND, Max	Superior o igual a min o igual a máx.
IN (valor,..)	Igualdad con cualquier elemento de una lista
IS NULL, IS NOT NULL	Comprueba si una expresión es NULL o no
LIKE	Correspondencia con relación a un modelo

Cuadro 4. Operaciones de condición where. Fuente [23]

#### **5.5.2.1.2.- Agrupar datos**

SQL, permite al usuario ordenar las tuplas del resultado de una consulta según los valores de uno o más atributos, empleando la cláusula *ORDER BY*. [10]

El orden por omisión es el ascendente. Podemos incluir la palabra *DESC* si queremos que los valores queden en orden descendente. La palabra *ASC* sirve para especificar explícitamente el orden ascendente. [10]

*ORDER BY* <nombre de la columna> *DESC*

*ORDER BY* <nombre de la columna> *ASC*

También es posible agrupar filas mediante la sentencia `SELECT` según los distintos valores de una columna, utilizando la cláusula `GROUP BY`.

```
mysql> SELECT FECHA FROM VENTA GROUP BY FECHA;
```

```
+-----+
| FECHA      |
+-----+
| 1978-06-15 |
| 1985-04-12 |
| 1993-02-10 |
| 2001-12-02 |
+-----+
4 rows in set (0.00 sec)
```

Utilizando la sentencia `GROUP BY` la salida se ordena según los valores de la columna indicada.

El uso de la cláusula *GROUP BY* permite usar funciones de resumen o reunión. Por ejemplo, la función `COUNT()`, que sirve para contar las filas de cada grupo:

```
mysql> SELECT FECHA, COUNT(*) AS CUENTA FROM VENTAS GROUP BY FECHA;
```

```
+-----+-----+
| FECHA      | CUENTA |
+-----+-----+
| 1978-06-15 |      2 |
| 1985-04-12 |      2 |
| 1993-02-10 |      1 |
| 2001-12-02 |      1 |
+-----+-----+
4 rows in set (0.00 sec)
```

Esta sentencia muestra todas las fechas diferentes y el número de filas para cada fecha.

Existen otras funciones de resumen o reunión, como MAX(), MIN(), SUM(), AVG(), STD(), VARIANCE()...

Estas funciones también se pueden usar sin la cláusula *GROUP BY* siempre que no se proyecten otras columnas:

```
mysql> SELECT MAX(FOLIO) FROM VENTA;
+-----+
| max(FOLIO) |
+-----+
| 3955      |
+-----+
1 row in set (0.00 sec)
```

Esta sentencia muestra el valor más grande de FOLIO en la tabla ventas, es decir el número más grande asignado a una venta o bien la última venta realizada.

**MySQL**, dispone de multitud de operadores diferentes para cada uno de los tipos de columna. Esos operadores se utilizan para construir expresiones que se usan en cláusulas *ORDER BY* y *HAVING* de la sentencia *SELECT* y en las cláusulas *WHERE* de las sentencias *SELECT*, *DELETE* y *UPDATE*. Además se pueden emplear en sentencias *SET*.

### 5.5.2.1.3.- Insertar datos

En su forma más simple, *INSERT* sirve para añadir una sola tupla a una relación. Debemos especificar el nombre de la relación y una lista de valores para la tupla. Los valores deberán enumerarse *en el mismo orden* en que se especificaron los atributos correspondientes en la instrucción *CREATE TABLE*. [10]

La sintaxis de la instrucción *insert* es la siguiente:

```
INSERT [IGNORE]
[INTO] nombre_tabla [(nombre_columna, ...)]
VALUES ({expresión | DEFAULT},...), (...),...
[ON DUPLICATE KEY UPDATE nombre_columna= expresión,..]
```

Las columnas afectadas por la inserción se especifican, mediante una lista de nombres de columnas tras el nombre de la tabla. En esta sintaxis, si la lista de las columnas está ausente, la sentencia afecta a todas las columnas de la tabla de manera predeterminada. Hay que precisar que no es obligatorio insertar un valor en todas las columnas de la tabla. Los valores de las columnas se especifican, mediante la cláusula VALUES, debe de incluir una expresión para cada columna mencionada en la lista de columnas (en el orden correspondiente) [23].

Véase los siguientes ejemplos:

```
mysql> INSERT INTO VENTA VALUES ('1','1974-04-12');
Query OK, 1 row affected (0.05 sec)
```

```
mysql> INSERT INTO VENTA VALUES ('2','1978-06-15');
Query OK, 1 row affected (0.04 sec)
```

```
mysql> INSERT INTO VENTA VALUES
-> ('3','2000-12-02'),
-> ('4','1993-02-10');
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM VENTA;
+-----+-----+
| nombre | fecha      |
+-----+-----+
|      1 | 1974-04-12 |
|      2 | 1978-06-15 |
|      3 | 2000-12-02 |
|      4 | 1993-02-10 |
+-----+-----+
4 rows in set (0.08 sec)
```

Si no necesitamos asignar un valor concreto para alguna columna, podemos asignarle el valor por defecto indicado para esa columna cuando se creó la tabla, usando la palabra *DEFAULT*:

```
mysql> CREATE TABLE REFACCION
->(Clave_Refaccion CHAR(20) NOT NULL PRIMARY KEY
->Nombre CHAR(20) ,
-> Existencias INT NULL DEFAULT 1);
Query OK, 0 rows affected (0.20 sec)
```

```
mysql> INSERT INTO REFACCION VALUES ('bkr5egp','bujia
platino', DEFAULT);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> SELECT * FROM REFACCION;
+-----+-----+-----+
| Clave_Refaccion | Nombre          |Existencias |
+-----+-----+-----+
| bkr5egp         |Bujia platino  |1           |
+-----+-----+-----+
1 row in set (0.02 sec)
```

En este caso, como había definido un valor por defecto para Existencias de 1, se asignará ese valor para la fila correspondiente a '1'.

Si intentamos insertar dos filas con el mismo valor de la clave única se produce un error y la sentencia no se ejecuta. Pero existe una opción que podemos usar para los casos de claves duplicadas: *ON DUPLICATE KEY UPDATE*. En este caso podemos indicar a MySQL qué debe hacer si se intenta insertar una fila que ya existe en la tabla. Las opciones son limitadas: no podemos insertar la nueva fila, sino únicamente modificar la que ya existe. Por ejemplo, en la tabla vendedor podemos usar el último valor de teléfono en caso de repetición:

```
mysql> INSERT INTO VENDEDOR (Nombre, Telefono) VALUES
```

```

-> ('oscar', 9564967);
Query OK, 1 rows affected (0.02 sec)

mysql> INSERT INTO VENDEDOR (Nombre, Telefono) VALUES
-> ('Ruben', 5516422496),
-> ('Oscar', 5524177971)
-> ON DUPLICATE KEY UPDATE Telefono=VALUES(Telefono);
Query OK, 3 rows affected (0.06 sec)
Records: 2 Duplicates: 1 Warnings: 0

```

```

mysql> SELECT * FROM VENDEDOR;
+-----+-----+
| Nombre | Telefono |
+-----+-----+
| Oscar  |5524177971 |
| Ruben  |5516422496 |
+-----+-----+
2 rows in set (0.00 sec)

```

En este ejemplo, la segunda vez que intentamos insertar la fila correspondiente a 'Oscar' se usará el nuevo valor de Teléfono. Si en lugar de *VALUES (Teléfono)* usamos *Teléfono* el nuevo valor de Teléfono se ignora. También podemos usar cualquier expresión:

```

mysql> INSERT INTO VENDEDOR (Nombre, Telefono) VALUES
-> ('Oscar', 93233)
-> ON DUPLICATE KEY UPDATE Telefono=Telefono;
Query OK, 2 rows affected (0.02 sec)

```

```

mysql> SELECT * FROM VENDEDOR;
+-----+-----+
| Nombre | Telefono |
+-----+-----+

```

```

| Oscar | 93233|
| Ruben | 5516422496|
+-----+-----+
2 rows in set (0.00 sec)

```

```

mysql> INSERT INTO VENDEDOR (Nombre, Telefono) VALUES
-> ('Oscar', 16390057)
-> ON DUPLICATE KEY UPDATE Telefono=0;
Query OK, 2 rows affected (0.01 sec)

```

```

mysql> SELECT * FROM VENDEDOR;
+-----+-----+
| Nombre | Telefono |
+-----+-----+
| Oscar  | 0        |
| Ruben  | 5516422496|
+-----+-----+
2 rows in set (0.00 sec)

```

#### 5.5.2.1.4.- Reemplazar datos

La instrucción UPDATE, sirve para modificar los valores de los atributos en una o más tuplas seleccionada. Al igual que la instrucción DELETE, una cláusula WHERE en la instrucción UPDATE selecciona de una sola relación las tuplas que se van a modificar. [10]

véase el siguiente ejemplo.

```

UPDATE nombre_tabla
SET nombre_columna = {expresión| DEFAULT} [... ]
[WHERE condicion]
[ORDER BY orden]
[LIMIT numero_de_filas];

```

## Ejemplo de la sintaxis de la sentencia UPDATE:

```
mysql> UPDATE VENDEDOR SET TELEFONO='5508089046' WHERE
NOMBRE='Oscar';
Query OK, 1 row affected (0.20 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> SELECT * FROM VENDEDOR;
+-----+-----+
| Nombre | Telefono      |
+-----+-----+
| Oscar  | 5508089046   |
| Ruben  | 5528288901   |
| Yael   | 5560778999   |
+-----+-----+
3 rows in set (0.00 sec)
Mysql>
```

Existe una sentencia REPLACE, que es una alternativa para INSERT, que sólo se diferencia en que si existe algún registro anterior con el mismo valor para una clave primaria o única, se elimina el viejo y se inserta el nuevo en su lugar.

```
mysql> REPLACE INTO VENDEDOR (Nombre, Telefono) VALUES
-> ('Oscar', 5540877017),
-> ('Ruben', 5528288901),
-> ('Yael', 5560778999);
Query OK, 5 rows affected (0.05 sec)
Records: 3  Duplicates: 2  Warnings: 0

mysql> SELECT * FROM VENDEDOR;
+-----+-----+
| Nombre | Telefono      |
```

```

+-----+-----+
| Oscar  | 5540877017 |
| Ruben  | 5528288901 |
| Yael   | 5560778999 |
+-----+-----+
3 rows in set (0.00 sec)

```

### 5.5.2.1.5.- Eliminar registros

La Instrucción DELETE. Elimina tuplas de una relación. Cuenta con una cláusula WHERE, similar a la de las consultas SQL, para seleccionar las tuplas que se van a eliminar. Las tuplas se borran explícitamente de una tabla cada vez. [10]

Cabe mencionar, que un cambio en una tupla de una tabla puede disparar actualizaciones en las demás tablas referenciadas según se hayan definido estas actualizaciones.

Condición del WHERE [22].

La sintaxis es la siguiente Sintaxis:

DELETE

FROM nombre\_tabla

WHERE nombre\_columna = dato

Ejemplo de la sentencia:

DELETE

```
mysql> FROM vendedor WHERE Nombre = 'oscar';
```

### 5.5.2.1.6.- Consultas multitabla o consultas anidadas

Hasta ahora todas las consultas que hemos usado se refieren sólo a una tabla, pero también es posible hacer consultas usando varias tablas en la misma sentencia SELECT.

**Consultas anidadas correlacionadas.** Siempre que una condición en la cláusula WHERE de una consulta anidada hace referencia a un atributo de una relación

declarada en la consulta externa, se dice que las dos consultas están **correlacionadas**. Podemos entender mejor que es una consulta correlacionada si consideramos que la *consulta anidada se evalúa una sola vez para cada tupla o combinación de tuplas) en la consulta externa*. [10] Ejemplo:

```
select  venta.FECHA_VENTA  as FECHA, venta.ID_VENDEDOR,
Remision.ID_VENTA,Remision.LINEA, Remision.CANTIDAD_VENDIDA
AS CANTIDAD, refaccion.CLAVE_REFACCION,
refaccion.NOMBRE_REFACCION, Remision.PRECIO as UNITARIO,
(Remision.precio* Remision.cantidad_vendida)as IMPORTE_VENTA,
FROM venta INNER JOIN Remision ON Venta.id_venta=
Remision.id_venta INNER JOIN refaccion on
Remision.clave_refaccion=refaccion.clave_refaccion order by
Venta.fecha_venta, Remision.id_venta, Remision.linea;
```

Esta consulta dice de la tabla venta selecciona la columna FECHA\_VENTA, pero que en el encabezado de la consulta sólo aparezca como FECHA, de la tabla venta selecciona la columna ID\_VENDEDOR, de la tabla remisión selecciona el ID\_VENTA y CANTIDAD\_VENDIDA como CANTIDAD, de la tabla REFACCION selecciona las columnas CLAVE\_REFACCION y NOMBRE\_REFACCION.

Luego, selecciona el precio de la tabla REMISION y se encabeza como UNITARIO, multiplica de la tabla REMISION el PRECIO \* CANTIDAD\_VENDIDA y el resultado se ubica en una columna con el nombre IMPORTE\_VENTA.

Las columnas utilizadas son de la tabla VENTA, REMISION y REFACCION en donde existe una relación mediante las columnas similares; para la tabla VENTA Y REMISION la relación es mediante la columna ID\_VENTA y para la tabla REFACCION Y VENTA las une la columna CLAVE\_REFACCION.

Esta consulta arrojaría un reporte como se ve en el cuadro siguiente:

Resultado de una consulta Multitabla. Fuente propia								
FECHA	ID_ VENDEDOR	ID_ VENTA	LINEA	CANTIDAD	CLAVE_ REFACCION	NOMBRE_ REFACCION	UNITARIO	IMPORTE
20/06/2012	OSCAR	1	1	1	ACML	A CASTROL 20W50LT.	65	65
20/06/2012	OSCAR	1	2	1	ACVISLT	A CASTROL VISCUS LT.	58	58
20/06/2012	OSCAR	2	1	4	AC4TPS	A CASTROL 4TPS	70	280
20/06/2012	OSCAR	2	2	1	AC40L	A CASTROL HD40	55	55

Cuadro 5. Resultado de una consulta Multitabla. Fuente elaboración propia

## 5.6.- FUNCIONAMIENTO DE UN NEGOCIO DE VENTA DE REFACCIONES AUTOMOTRICES

Dentro de un negocio de venta de refacciones las actividades principales son:

**Venta de refacciones:** es la tarea de vender al cliente la refacción adecuada al modelo y tipo de vehículo que corresponde.

**Cambios y devoluciones:** en ocasiones se requiere hacer devoluciones o cambios de refacciones, ya que en ocasiones una refacción puede no corresponder con el vehículo que se requiere y por tal razón se necesita hacer un cambio o en ocasiones una devolución.

**Listar y pedir faltantes:** realizar una lista de faltantes de mercancía de acuerdo a las existencias que se cuentan en un momento dado para posteriormente realizar el pedido al proveedor correspondiente.

**Relación de proveedores:** atención con los proveedores al momento de solicitarles un pedido ya sea por teléfono o personalmente.

**Actualización de los precios:** en el negocio de refacciones los precios de refacciones incrementan constantemente, por lo tanto se requiere fechar el día de recepción de cada refacción para poder saber en un momento determinado que

tan actual es su precio, si la recepción de esa refacción fue mayor a los tres meses atrás será necesario actualizar el precio; la actualización del precio de una refacción se hace mediante la búsqueda de la misma en un libro de Excel con los precios más actuales que proporcionan los proveedores.

**Manejo del stock:** una tarea fundamental en cualquier negocio, es cuidar veladamente las existencias de producto, ya que en ocasiones se comete el error de solicitar mercancía al proveedor hasta que las existencias son cero piezas y esto conlleva a que en ocasiones se niegue el producto al cliente, lo cual no debe suceder si se tiene un correcto control en esta parte.

**Manejo de equivalencias:** debido a la existencia de gran cantidad de marcas de refacciones y el manejo de distintos códigos para las mismas, se debe manejar equivalencias de refacciones, es decir que una refacción es exactamente la misma pero en distinta marca y distinto número, para esto es necesario conocer las equivalencias.

**Correcto manejo de los créditos:** los proveedores de refacciones en ocasiones proporcionan un crédito de “n” cantidad de días el cual debe respetarse para que no genere ningún tipo de interés o algún contratiempo en el envío de mercancía; es por eso que en el manejo de los créditos deben de tenerse bien en cuenta las fechas de pagos para no ocasionar ningún problema en este sentido.

**Relación de clientes:** debe manejarse una relación de clientes, en especial clientes con un número de contribuyente que requieren una factura para reportar ante hacienda; y a su vez como negocio registrado ante hacienda se está obligado a reportar quienes son tus clientes así como tus proveedores.

**Registro de ventas:** en este negocio se requiere un registro de ventas, con el fin de saber como se comportan las ventas y cual va siendo su tendencia, para una correcta toma de decisiones dentro de esta organización.

**Facturación:** el negocio proporciona facturas a sus clientes, y todos estos datos de facturación debe ser guardados para posteriormente en el tiempo adecuado se realice un informe fiscal ante la secretaria de hacienda y crédito público.

**Facturas de compra:** el negocio debe guardar información acerca de las facturas de compra para en tiempo adecuado se haga un reporte de pagos ante la secretaria de hacienda y crédito público.

## VI. METODOLOGÍA

### 6.1.- PASOS METODOLÓGICOS

El sistema de información se desarrollara bajo los siguientes pasos metodológicos:

**Investigación de requerimientos:** mediante entrevistas, revisión de registros y observación de procesos de negocio se va a comprender y ubicar la problemática a resolver con el desarrollo de un Sistema de Información (SI).

**Especificación de requerimientos:** una vez que se tienen los requisitos, se determinan las entradas procesos y las salidas que debe contener el sistema de información a desarrollar.

**Diseño del SI con UML:** se detalla parte por parte de lo que va a contener el sistema, a fin de definir, construir, documentar y especificar el sistema de software.

**Definición de datos:** se determinan todos los datos que requiere el negocio dentro de sus funciones para poder definir una base de datos.

**Análisis y diseño de la base de datos:** se realiza un esquema lógico de la base de datos, es decir, diseño de las tablas los datos y tipos de datos que van a conformar la base de datos, se definen llaves primarias, llaves foráneas y como va a ser su relación entre los datos de cada tabla.

**Implementación de la base de datos:** utilizando un sistema gestor de bases de datos, se implementa la base de datos que se ha venido trabajando en puntos anteriores.

**Desarrollo de interfaces de usuario del sistema:** los diagramas UML van a ser utilizados para el desarrollo de cada una de las interfaces del sistema y que procesos contendrá cada una de ellas.

**Programación de las interfaces del sistema:** se realiza la programación de las acciones que va a contener cada vista de usuario.

**Pruebas de validación y verificación en el sistema:** una vez terminado el sistema se comprueba que la implementación del sistema sea correcta, descubriendo defectos y se evalúa el producto desarrollado para verificar que cumpla con las necesidades que originaron su desarrollo.

**Desarrollo de un manual de usuario para el sistema:** de acuerdo a cada vista y a cada función del sistema se realiza un manual de ayuda que va a contener dentro de las opciones del sistema con el fin de que el usuario pueda consultar en un momento dado.

**Implementación del sistema:** implica la puesta en marcha del sistema dentro del negocio, así como la capacitación de usuarios.

## **6.2 DESARROLLO DE METODOLOGIA**

### **6.2.1.- Investigación de requerimientos**

¿Qué son los requerimientos del sistema? Los requerimientos del sistema especifican lo que el sistema de información deberá hacer o cual propiedad o cualidad debe hacer éste. [4]

Lo que el sistema debe ser capaz de hacer se llama también requerimientos funcionales, por otro lado, los requerimientos no funcionales tiene que ver con la seguridad, facilidad de uso, entre otros.

La investigación de requerimientos, es la actividad más importante dentro del análisis de sistemas, ya que la calidad del trabajo realizado en esta etapa se verá reflejada más adelante en las características del nuevo sistema.

En el desarrollo del sistema de información para venta de refacciones automotrices, el cliente solicitó los servicios para el desarrollo de un sistema de información en donde fuera posible hacer consultas de refacciones específicas para un vehículo, además que permitiera realizar la venta de refacciones con el fin de documentar datos y consultarlos en determinado momento, realización de facturas y documentarlas, control de entradas y salidas al inventario, así como una relación de clientes, proveedores y vendedores.

A grandes rasgos tenemos una idea de cómo podría verse un sistema que obedezca a estos requerimientos, sin embargo, es necesario investigar más a fondo cada uno de estos puntos para obtener la información amplia y exacta que describa que pasos se seguirían para poder realizar estos procesos, además de que datos se utilizan y cuales de ellos se documentan; como se ha mencionado esto sólo es una vaga idea e insuficiente para el desarrollo de un sistema de información.

Existen diversas técnicas de recolección para datos relacionados con los requisitos denominadas *técnicas de recolección de hechos*; para este proyecto se utilizaron los que son más adecuados y se explicaran en este mismo apartado.

#### **6.2.1.1.- Entrevistas**

Generalmente las entrevistas son aplicadas a usuarios en potencia del sistema propuesto, en este caso vendedores y administrador del negocio de venta de refacciones automotrices que interactúan directamente con el sistema, las entrevistas sirven para reunir información que son de conocimiento de personas que están involucradas con el sistema que está bajo estudio.

Las entrevistas suelen clasificarse en estructuradas y no estructuradas. El primero utiliza un conjunto anticipado de respuestas; el segundo permite que el entrevistado dé respuesta a las preguntas con sus propias palabras. Cada enfoque tiene sus ventajas y desventajas.

Una ventaja de la entrevista, es que ayuda a entender al analista áreas mal comprendidas. Descubrir como es que trabajan las cosas en realidad.

En el anexo uno y dos se podrán ver los formatos de la primera y segunda entrevista respectivamente, los siguientes comentarios se derivan de las entrevistas realizadas.

El proceso de venta de refacciones inicia cuando un cliente solicita una refacción indicando el nombre de su vehículo y modelo, el vendedor identifica el catalogo que va a utilizar para ver la refacción que corresponde a ese vehículo, una vez identificando el número de refacción el vendedor verifica que este físicamente, si el cliente requiere una refacción más, se repite proceso, una vez concluido el pedido se realiza una nota de remisión con copia o bien una factura, si así lo requiere el cliente. En caso de que no se tuviera la refacción se anota en la libreta de pedidos pendientes. Las refacciones que se vendieron se anotan de la misma manera en la libreta de pedidos para resurtirlos nuevamente. Todo el proceso se realiza de forma manual.

El vendedor da la copia al cliente para que pague en caja, en caja se sella la nota de pagado y regresa con el vendedor a recoger su mercancía y la nota original, la copia se queda con el vendedor para los cortes de caja.

Un cliente puede acumular notas de remisión y estas serán facturadas antes de finalizar el mes si se solicitan antes de fin de mes.

La información que se requiere guardar acerca de clientes es su RFC, nombre, dirección, teléfono, código postal, para los que requieren facturar, ya que las demás ventas entran como venta en general.

De los proveedores se requieren los datos de la empresa: nombre, dirección, teléfono, e-mail, así como datos del representante asignado por cada empresa.

El proceso más tardado que considera el administrador, es al identificar las refacciones que corresponden a un vehículo ya que se tienen que buscar uno a uno en distintos catálogos, aunado a esto la elaboración de notas y facturas es de forma manual susceptible a errores matemáticos.

Las copias de facturas y notas de ventas son utilizadas por el administrador para realizar un corte de caja diario y a la vez esta información se guarda en un diario de ventas. De ahí cualquier tipo de informe requerido es consultado y elaborado de forma manual.

El administrador intento utilizar un programa llamado “exion y win caja”, según su experiencia este sistema no le solucionaba todas las necesidades, además, tenía procesos que él no requería y en muchos casos no realizaba las cosas en la forma que se deseaban, por lo que, se terminó por desechar el “exion y win caja”.

Según el administrador un sistema de información a la medida tiene que adaptarse a las necesidades del negocio, sin más funciones que las que él requiere para realizar los procesos, mientras que con un sistema comercial hay que adaptarse a las funciones que contiene ese sistema.

El administrador comentó que no llevan un control de inventario como tal, solo se controla a través de las notas de venta, las cuales sugieren que el producto debe de ser resurtido nuevamente posterior a la venta efectuada, así mismo, como la mercancía que no se tiene y es solicitada. Sin embargo, no hay un control estricto de los artículos que se tienen y en ocasiones suele haber inexistencias de productos que se tenían antes, esto por que se pidió al proveedor, éste no lo tuvo y no lo envió y es difícil controlar, debido a que se pide gran cantidad de artículos diariamente. Para conocer cuando se debe de resurtir el producto, a la hora de venderlo el empleado lo registra en la libreta de pedidos.

Por lo regular las compras realizadas a los proveedores son facturas de crédito, el administrador debe agendar las fechas de pago así como los montos, devoluciones y descuentos, una vez finalizando el mes el administrador entrega facturas de venta, así como, facturas de compra para que el contador realice los reportes fiscales correspondientes cada mes.

Se detectó que no se maneja crédito de venta a ningún cliente.

Al realizar un cambio de producto se borra la refacción del listado de pedido, se hace el cambio de precio y se corrigen los documentos de control. En el caso de las devoluciones se recoge la refacción se devuelve el dinero y se borra de lista de pedidos y ventas.

Cuando un vendedor realiza una venta, en la nota se escribe la fecha, nombre de quien realizo la venta para posteriores aclaraciones.

La información requerida de cada artículo es el nombre, la clave, la descripción, el proveedor que lo surte y la fecha del precio que se tiene, ya que en ocasiones hay productos que tardan años en venderse y los precios deben verificarse y actualizarse continuamente.

### **6.2.1.2.- Revisión de registros**

El analista examina información relacionada con el sistema. Esta información puede ser proporcionada por varios tipos de registros y reportes. Es decir, notas de remisión que son expedidas, facturas de venta, orden de pedidos, reportes fiscales, reporte de ventas, facturas de crédito, catálogos utilizados al realizar consultas. Estos registros no indican la forma en que se desarrollan las actividades o como se realizan todas las tareas. Sin embargo, pueden ser de ayuda para el analista para determinar la necesidad del sistema.

El administrador del negocio de refacciones, mostro los registros con los cuales trabaja el negocio, ya sea que los elaboran diariamente o bien los reciben, en el caso de facturas de compra, además señaló reportes que deben elaborarse después de una jornada laboral y algunos otros que son de forma eventual.

Los registros que fueron revisados por el analista fueron los siguientes:

Notas de remisión, facturas elaboradas por los empleados, diario de ventas del negocio, agenda de pagos de facturas de compra a crédito, revisión de libreta de faltantes, catálogos utilizados para encontrar refacciones solicitadas por los clientes.

### **6.2.1.3.- Observación**

Por medio de esta técnica se obtuvo información de cómo se llevan a cabo los procesos de negocio de la organización. De esta forma se obtiene información de primera mano y se sabe cómo se realizan las actividades, también se detectaron los pasos específicos para realizar dicha actividad.

La tarea del observador, es recabar toda la información correspondiente acerca de cómo se lleva a cabo cada actividad y cuáles son los pasos específicos para completar cada una de ellas, de esta forma se estudia a fondo los pasos. Una de las tareas consecuentes más importantes de este método es distinguir los pasos

importantes dentro de cada actividad y tratar de eliminar pasos repetitivos o innecesarios.

El administrador permitió observar durante una semana por 1 hora diaria las diferentes tareas que se realizan donde se observaron las siguientes actividades:

- ✓ Identificación de refacciones al vender por medio de los catálogos
- ✓ Venta de refacciones y elaboración de notas y facturas
- ✓ Control de existencias de refacciones dentro del inventario
- ✓ Recepción de reabastecimiento de mercancía
- ✓ Control de agenda de pagos
- ✓ Elaboración del corte de caja al final del día

**Observación viviendo el sistema.** En este tipo observación el analista de sistemas desempeña activamente el papel del usuario por un corto periodo. [4].

El administrador del negocio permitió al analista efectuar la tarea del vendedor durante 2 días por 30 min diarios, a fin de comprender plenamente los pasos a seguir al vender una refacción e identificar los procesos mas tardados; y posteriormente idear formas para resolver la problemática, principalmente de procesos tardados que podrían realizarse de mejor forma utilizando un sistema basado en computadora.

### **6.2.2.- Especificación de requerimientos**

Se identificaron los requerimientos imprescindibles que debe satisfacer el sistema, así como, los métodos que se utilizaran para dar solución a dichos requerimientos.

#### **6.2.2.1.- Identificación de requerimientos básicos**

Con base en la investigación de requerimientos se busca respuesta a las siguientes preguntas.

¿Cuál es el proceso básico del negocio?

¿Qué datos utiliza o produce este proceso?

¿Cuáles son los límites impuestos por el tiempo y la carga de trabajo?

¿Cuál es la finalidad del reabastecimiento de inventarios?

¿Qué pasos se siguen para el reabastecimiento de inventarios?

¿Quiénes lo realizan?

El proceso básico dentro del negocio es la venta de refacciones automotrices, y su razón de ser son las utilidades que este negocio obtiene.

Los datos que se utilizan en el proceso de venta de refacciones son principalmente información contenida en catálogos de refacciones, ordenadas por alfabeto. Y finalmente el vendedor realiza un comprobante de compra para el cliente ya sea factura o nota de remisión, con una copia que sirve al administrador para hacer su corte de caja. Así mismo, el vendedor completa venta a venta la actualización de faltantes a resurtir dentro del inventario de mercancía.

Una de las principales problemáticas comenta el administrador, es que con el crecimiento del negocio y aumento subsecuente en sus ventas y compras. En ocasiones los clientes tienen que esperar de dos a veinte minutos en ser atendidos, debido a que el proceso de búsqueda de refacciones dentro de los catálogos es muy tardada y aunado a esto la elaboración de una factura manual consume tiempo que debería ser empleado en atención a más clientes con el fin de tener una atención rápida y eficiente para hacer que el negocio funcione.

La finalidad del reabastecimiento de inventarios es asegurar las cantidades de existencias adecuadas de artículos en el almacén sin que estas se vuelvan excesivas y por lo tanto costosas. Los pasos a seguir para el reabastecimiento de inventarios es mientras el vendedor hace una venta verifica las existencia de un producto y determina si debe pedirse; de esta forma se completa el listado de faltantes. Posteriormente el administrador hace el pedido de mercancía al final del

día, para que a las 12:00 del día siguiente llegue la mercancía solicitada, al recibir la mercancía se corroboran los precios, se actualizan colocando etiquetas con precio y la fecha de actualización del precio del producto y las iniciales del proveedor que surtió.

#### **6.2.2.2.- Requerimientos de transacciones de los usuarios**

El analista que trabaja con en un sistema de procesamiento de transacciones, debe conocer todo lo relacionado con la forma en que se procesan estas transacciones. Para entender los requerimientos de las transacciones en el negocio de venta de refacciones el analista se formula las siguientes preguntas:

¿Qué inicia la transacción?

¿Qué elementos se utilizan para realizar una venta de refacciones?

¿Qué información se genera al vender refacciones? ¿Qué datos se guardan?

¿Con que frecuencia se realizan pedidos de mercancía?

Primeramente, el cliente inicia una transacción, solicitando una o más refacciones indicando el nombre y modelo de su vehículo, en ocasiones con la muestra para corroborar que la refacción sea la correcta, ya que en ocasiones los catálogos no son tan exactos al indicar algunas refacciones.

El vendedor utiliza un catálogo para buscar la refacción o refacciones solicitadas, verifica si existe físicamente la refacción en el almacén, si la refacción se encuentra se le muestra al cliente, indicando precio marca y características para finalmente realizar la remisión o factura. Se efectúa el cobro en caja y se entrega la refacción al cliente. Posteriormente, el vendedor entrega la copia al administrador para corroborar las existencias del producto vendido y anotar el faltante.

Los datos que se guardan son la copia de la nota de remisión con el fin de realizar el corte al final del día, y en el caso de las facturas se guardan para que al fin de mes se haga la declaración.

Los pedidos de mercancía se realizan con distintos proveedores, en distintos días, algunos se hacen mediante la visita del proveedor, otros más por teléfono, cabe mencionar que los pedidos más frecuentes se realizan diariamente al final del día para asegurar la existencia al día siguiente.

### **6.2.2.3.- Requerimientos de decisión de los usuarios**

Este tipo de requerimientos no tiene mucho que ver con el procesamiento de transacciones si no con la toma de decisiones de los usuarios frente a tareas no rutinarias, en algunos casos se procesan los datos de transacciones a fin de generar información para la toma de decisiones.

Algunas preguntas que se hace el analista para identificar requerimientos de este tipo son:

¿Qué información se necesita para tomar decisión?

¿Cuál es la fuente de esta información? ¿Qué sistema los produce?

¿Qué otros datos son necesarios y no es posible obtenerlos del sistema actual?

¿Qué datos se originan en fuentes externas a la organización?

¿Cómo se deben procesar los datos para producir información necesaria?

¿Cómo debe presentarse la información?

En el negocio de venta de refacciones se deben guardar los datos acerca de las ventas, a fin de obtener la cantidad de ventas por día, semana, mes, etc. Con el fin de saber la situación financiera del negocio, su crecimiento, y la información acerca de productos con mayor rotación.

La fuente actual de esta información, son las notas de remisión y facturas que se efectúan al realizar una venta, estas notas generan datos que son guardados en un diario de ventas, en donde se contienen las ventas por día. Todas estas actividades realizadas en forma manual por lo que es fácil notar lo tedioso que resulta obtener dicha información.

En el sistema actual no se tienen todos los datos útiles para que el negocio de venta de refacciones funcione, sin embargo, la problemática no recae sólo en la falta de datos si no la forma en que se obtienen, y el proceso tedioso que tiene que realizarse para producir información necesaria, algunas veces inexacta. Información tan necesaria como las utilidades que se obtienen en algún periodo de tiempo, registro de existencias en el almacén sin tener que dirigirse a verificar si se encuentran, productos más vendidos, productos menos vendidos, información de cuánto costó un producto y que tiempo tiene en el almacén.

Datos que se obtienen de fuentes externas como son: información acerca de promociones, actualización de precios, nuevos productos, nuevos catálogos.

Los datos indiscutiblemente deben ser procesados y almacenados por medio de un sistema basado en computadora que permita, procesarlos y obtener información relevante en segundos; estos procesos actualmente llevan más de treinta minutos. Información acerca de utilidades en un periodo de tiempo determinado datos que actualmente son imposibles de generar. Un control estricto de entradas y salidas al inventario, con el fin de que si algún producto sigue sin resurtirse no quede en el olvido hasta que se le niegue al cliente.

#### **6.2.2.4.- Requerimientos de toda la organización**

Estos requerimientos deben especificar el orden a seguir de un proceso para que el negocio funcione de manera correcta en el conjunto de tareas que se realizan dentro de la organización. Es decir, para que el vendedor pueda realizar una factura debe tener los datos del cliente previamente guardados, si no es así se debe primero obtener los datos del cliente y posteriormente realizar facturación.

Para que el administrador pueda determinar las ventas del día debe obtener los datos de cada venta y al final obtener el resultado de las ventas totales, a su vez obtener el listado de faltantes y disponerse a realizar los pedidos correspondientes.

### 6.2.3.- Diseño del SI con UML (casos de uso)

#### 6.2.3.1.- Identificación de los actores

Al centrarse en la identificación de actores hay que centrarse en cómo se usará el sistema y no como se construirá. Además, esto ayuda a definir el alcance del proyecto, asimismo, saber si los requerimientos de sistema están completos.

Los actores son aquellas personas o sistemas externos que proporcionan entradas al sistema en observación, personas que reciben salidas del sistema, persona quien mantendrá la información en el sistema y eventos originados en un instante predeterminado.

Una vez que se identifican los actores se presentan en el siguiente glosario.

<b>Glosario de actores</b>	
<b>Termino</b>	<b>Descripción</b>
1. Cliente general	Persona que compra productos al negocio de refacciones, pero no solicita facturación.
2. Cliente con RFC	Persona que compra productos al negocio de venta de refacciones, que requiere facturación.
3. Proveedor	Persona u organización que se encarga de proveer de refacciones al negocio de refacciones.
4. Vendedor	Persona que interactúa directamente con el sistema para realizar ventas, devoluciones.
5. Administrador	Persona que interactúa directamente con el sistema para dar de baja productos, agregar nuevos productos, actualización de precios, manejo de cuentas por pagar, cortes y elaboración de reportes a través del sistema

Cuadro 6. Glosario de actores. Fuente elaboración propia

### 6.2.3.2.- Identificación de los casos de uso para los requerimientos de negocios

Durante el análisis de requerimientos se identificaron los más complejos e importantes denominados casos de uso esenciales. Se utiliza una plantilla de glosario de casos de uso para documentar los casos de uso véase el cuadro 7.

<b>Glosario de casos de uso</b>		
Nombre del caso de uso	Descripción del caso de uso	Actores y papeles participantes
Agregar una nuevo registro	Este caso de uso describe el evento de cuando el Administrador decide agregar mercancía nueva al inventario	Administrador (negocio primario)
Eliminar una refacción	Este caso de uso describe el evento de cuando el administrador decide dar de baja algún producto descontinuado.	Administrador (negocio primario)
Actualizar datos de las refacciones	Este caso de uso describe el evento de cuando administrador efectúa la actualización de datos de refacciones, como un incremento en el precio, una promoción.	Administrador (negocio primario) Proveedor (receptor primario)
Realizar un reporte de ventas	Este caso de uso describe el evento de cuando el administrador decide generar un reporte de ventas con un periodo de tiempo determinado, con el fin de obtener la cantidad de ventas, las utilidades obtenidas o bien los productos con mayor desplazamiento	Administrador (negocio primario)
Realizar una venta	Este caso de uso describe el evento de cuando un vendedor realiza la venta de una o más refacciones, a través de la solicitud que el cliente efectúa.	Vendedor (negocio primario) Cliente General (receptor primario) Cliente con RFC (receptor primario)
Realizar una devolución	Este caso de uso describe el evento de cuando un vendedor cancela la venta realizada y efectúa las actualizaciones pertinentes.	Vendedor (negocio primario) Cliente General (receptor primario) Cliente con RFC (receptor primario)

Presupuestar	Este caso de uso describe el evento de cuando el vendedor realiza un presupuesto de refacciones mediante una petición del cliente	Vendedor (negocio primario) Cliente General (receptor primario) Cliente con RFC (receptor primario)
Generar reporte de faltantes	Este caso de uso describe el evento de cuando el administrador, genera un reporte de faltantes con el fin de reabastecer el producto en calidad de faltante (este puede ser enviado por correo o bien pedido por teléfono)	Administrador (negocio primario)
Ingresar datos de un nuevo cliente	Este caso de uso describe el evento de cuando el administrador o vendedor realiza el alta un nuevo cliente que requiere facturación	Vendedor (negocio primario) Administrador (negocio primario)

Cuadro 7. Glosario de casos de uso Fuente elaboración propia

### 6.2.3.3.- Construcción del diagrama de modelos de casos de uso

Una vez identificados los actores y los principales casos de uso se elabora un diagrama de casos de uso con el fin de identificar las fronteras y alcance del sistema, se decidió poner el conjunto de casos en un solo esquema ya que todos tienen actores iguales (Figura 19).

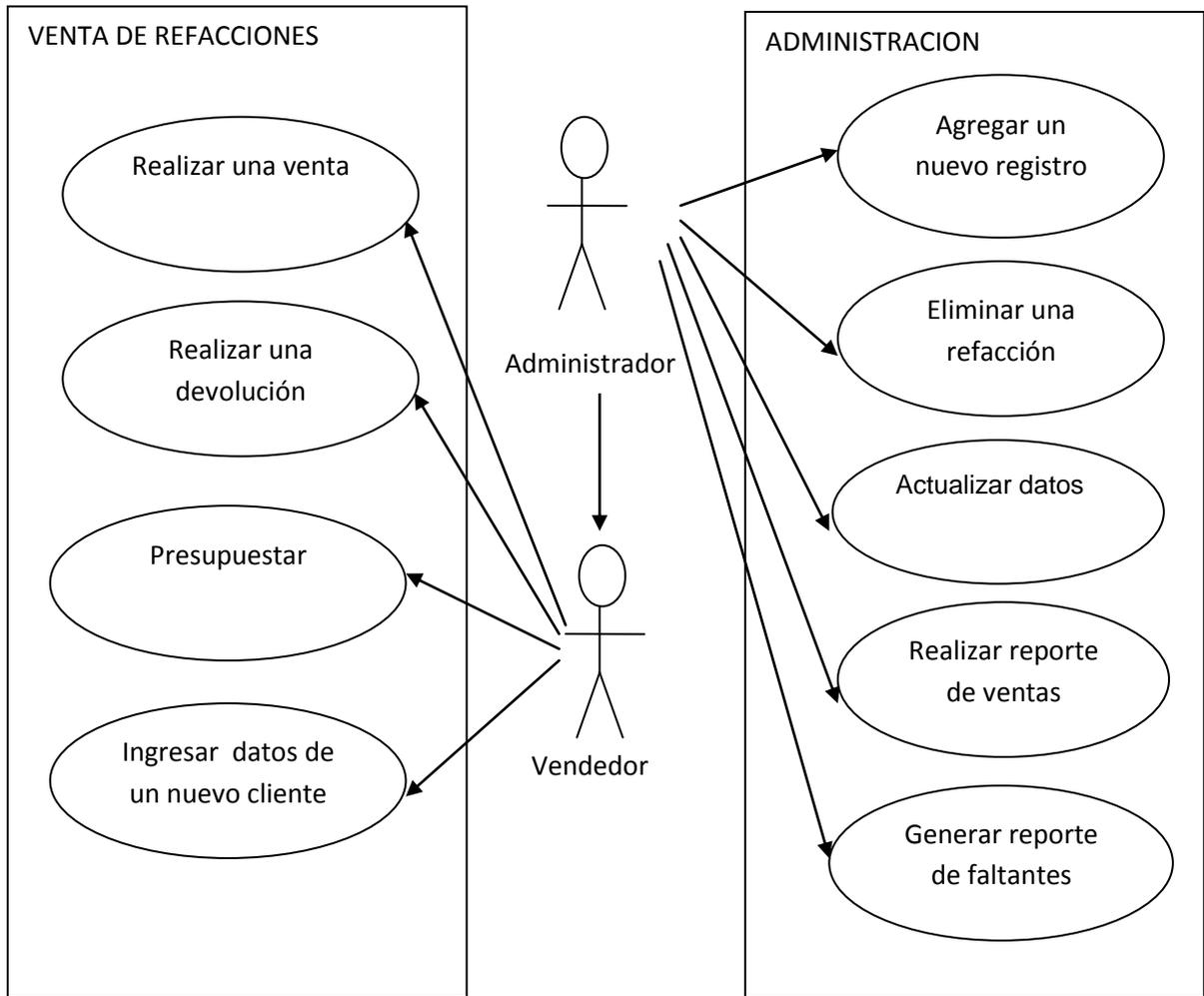


Figura 19. Diagrama de casos de uso del negocio de venta de refacciones automotrices. Fuente elaboración propia

#### 6.2.3.4.- Narración de los casos de uso

Narración de los casos de uso	
Caso de uso	Descripción
Realizar una venta	Este caso de uso se describe cuando el empleado realiza una venta por medio de una consulta de refacciones solicitadas por un cliente, una vez que se verifica que los productos están en existencia se realiza una nota de remisión o bien una factura. Se debe realizar una actualización en las existencias del producto.
Realizar una devolución	Este caso de uso se describe cuando un cliente por alguna razón hace la devolución del producto, el vendedor debe efectuar la devolución en el sistema.

Presupuestar	Este caso de uso se describe el evento cuando un cliente solicita el precio de diversas refacciones y el vendedor realiza las consultas pertinentes para finalmente entregar un presupuesto por escrito al cliente.
Ingresar datos de un cliente nuevo	Este caso de uso se describe cuando un cliente solicita la factura a un vendedor y éste aún no está dado de alta en el sistema.
Agregar un nuevo registro	Este caso de uso describe el evento de cuando un cliente solicita una refacción que aún no se maneja dentro de la gama de productos existentes y el administrador decide anexarlos al inventario para posteriormente realizar el pedido y tenerlos en existencia.
Eliminar una refacción	Este caso de uso describe un evento de cuando el administrador lleva a cabo un reporte de que refacciones tienen una antigüedad determinada y decide no volver a resurtirlas después de ser vendidas para poder darlas de baja del sistema.
Actualizar datos	Este caso de uso describe el evento de cuando el administrador realiza un reporte para actualizar los precios o datos de las refacciones.
Realizar un reporte de ventas	Este caso de uso describe el evento de cuando el administrador después de una jornada laboral lleva a cabo el reporte de ventas de un día o de cierto periodo de tiempo.
Generar reporte de faltantes	Este caso de uso describe un evento de cuando un proveedor solicita al administrador los faltantes de producto para surtirle y el administrador realiza el reporte para proporcionárselo al proveedor.

Cuadro 8. Narración de los casos de uso Fuente elaboración propia

#### 6.2.4.- Definición de datos

A continuación se muestra el glosario de datos que son imprescindibles para el funcionamiento del negocio de venta de refacciones automotrices.

<b>Datos utilizados dentro del negocio de venta de refacciones</b>	
<b>Dato</b>	<b>Descripción</b>
Clave de refacción	Es la clave con la cual se identifica una refacción en específico. No se repiten
Nombre de refacción	Es el nombre con el cual se conoce una refacción

Costo	Es la cantidad de dinero en la que el negocio compra un artículo, para posteriormente agregar la utilidad
Precio de venta	Es la cantidad de dinero en el que se oferta un artículo para su venta
Existencias	Es la cantidad que se tiene de un artículo específico
Stock	Es la cantidad mínima que indica que se ha llegado al límite de existencias mínimas y debe resurtirse más producto.
Marca	Es la marca comercial de la refacción
RFC de la empresa	El RFC de la empresa proveedora de las refacciones
Nombre de la empresa	Nombre de la empresa proveedora
Dirección de la empresa	Dirección de la empresa proveedora
CP de la empresa	Se refiere al código postal de la empresa que provee refacciones
Teléfono de la empresa	Teléfono de la empresa que sirve para realizar pedidos
N° de cuenta de la empresa	Número de cuenta de la empresa al cual se depositan pagos
Nombre del contacto	Contacto intermediario entre la empresa proveedora y el negocio de refacciones, el cual es asignado por la empresa proveedora
Teléfono del contacto	Teléfono del contacto asignado por la empresa proveedora
Grupo de refacción	Grupo general en el que están divididas las refacciones, para una rápida identificación
E-mail del contacto	E-mail del contacto asignado por la empresa proveedora, para poder enviar pedidos y recibir listas de precios actualizadas
Marca del auto	Se refiere a la armadora para la cual está fabricada una refacción es decir VW, Chrysler, etc.
Aplicación	Se refiere a los autos de las distintas armadoras, a los cuales aplica la refacción
Imagen	Imagen de la refacción, para que el vendedor pueda observarla sin tener que dirigirse al almacén
Fecha de registro	Fecha en la cual llegó el producto al negocio, con la finalidad de actualización de precios y verificación de rotación de producto.
Numero de factura de compras	El número de factura de una compra ya sea de contado o a crédito con el fin de agendar pagos y agregar mercancía al inventario
Cantidad de compra	Cantidad de artículos comprados en una factura de compra
Estado de la factura	El estado de la factura indica si la factura está en pagada, en adeudo, o vencida

Fecha de la factura	Fecha en que fue recibida la factura y el producto
Días de crédito	Plazo después de la recepción de la mercancía y factura ,para realizar el pago
Subtotal	El subtotal de la factura de compra antes de gravar el IVA
Total	El importe total de la factura de compra
Observaciones	Observaciones en una factura de compra, tales como devoluciones, o bien descuentos otorgados fuera de la factura.
Numero de remisión	El número de nota de remisión de venta
Cantidad vendida	Cantidad que se vende de un producto
Nombre del vendedor	Nombre del vendedor que realiza un presupuesto o venta
Numero de factura de venta	Número de la factura expedida en una venta
Cantidad vendida en factura	La cantidad de una refacción que fue facturada
Subtotal	El subtotal de la factura de venta antes de gravar el IVA
Total	El importe total de la factura de venta
Tasa IVA	El porcentaje que debe gravarse en una venta IVA
RFC del cliente	La clave de registro federal de contribuyente del cliente al que se le expiden facturas
Fecha de la factura	Fecha en que es expedida una factura de venta
Nombre del Cliente	Se refiere a la razón social del cliente al que se le factura
Dirección fiscal	Dirección fiscal del cliente al que se le factura
Cp. cliente	Código postal de la dirección fiscal del cliente
Teléfono cliente	Teléfono para localizar un cliente

Cuadro 9. Datos utilizados dentro del negocio de venta de refacciones. Fuente elaboración propia

### 6.2.5.- Análisis y diseño de la base de datos

Después de la definición de datos, se realizó un modelo de base de datos de acuerdo al modelo entidad relación con las tablas normalizadas quedando, como se muestra en la figura 20.

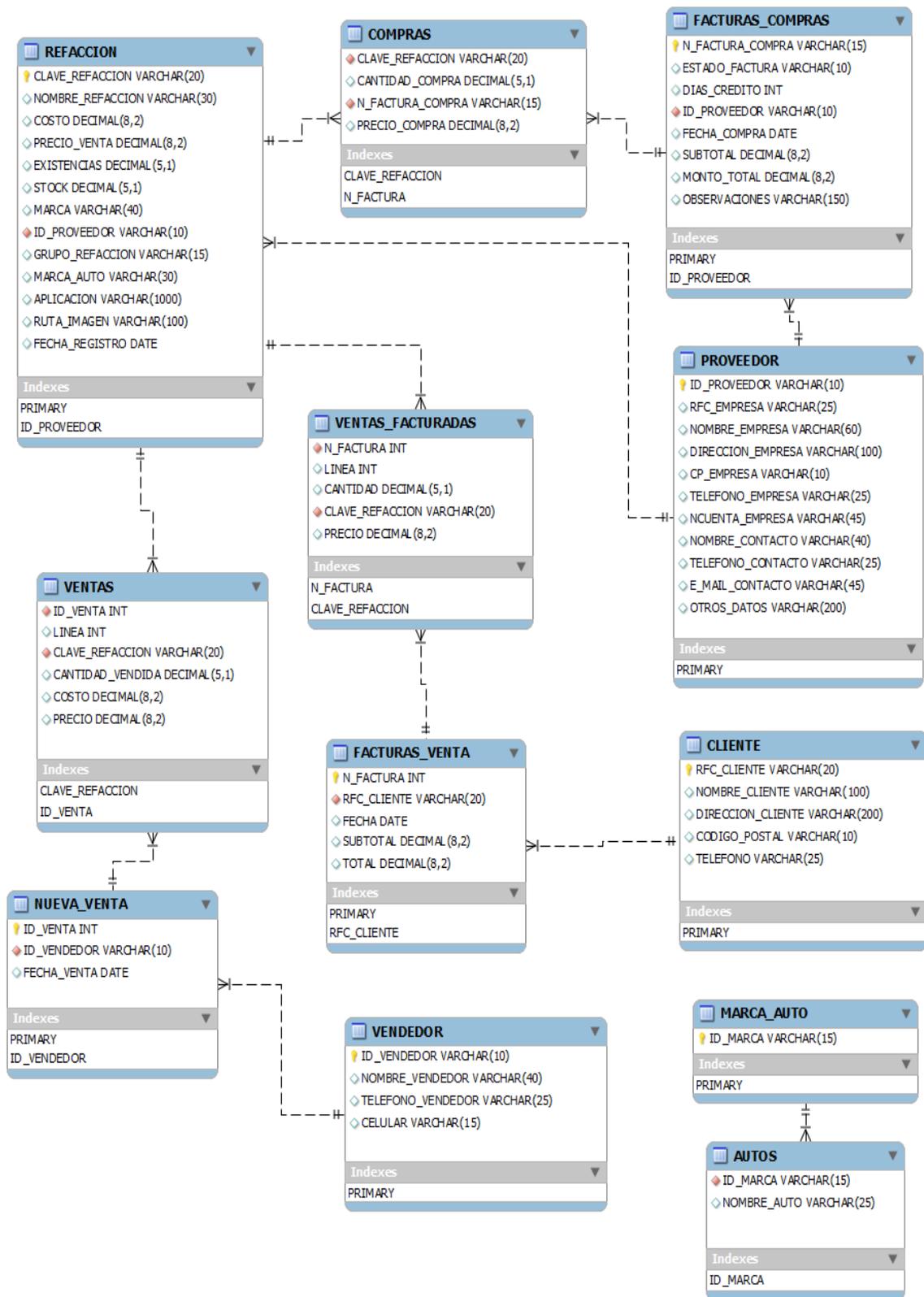


Figura 20. Base de datos del sistema para venta de refacciones

## 6.2.6.- Implementación de la base de datos

Una vez teniendo el diseño de la base de datos bien organizada y normalizada, se implementó físicamente en MYSQL.

Se ejecuta la consola mysql command client

A continuación se presenta las líneas de comandos que se ejecutaron en el editor de mysql para poder crear la base de datos diseñada previamente.

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';
```

```
DROP SCHEMA IF EXISTS `REFACCIONARIA` ;
CREATE SCHEMA IF NOT EXISTS `REFACCIONARIA` DEFAULT CHARACTER SET latin1 COLLATE
latin1_swedish_ci ;
SHOW WARNINGS;
USE `REFACCIONARIA`;
```

```
-----
-- Table `REFACCIONARIA`.`PROVEEDOR`
-----
```

```
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`PROVEEDOR` (
  `ID_PROVEEDOR` VARCHAR(10) NOT NULL ,
  `RFC_EMPRESA` VARCHAR(25) NULL ,
  `NOMBRE_EMPRESA` VARCHAR(60) NULL ,
  `DIRECCION_EMPRESA` VARCHAR(100) NULL ,
  `CP_EMPRESA` VARCHAR(10) NULL ,
  `TELEFONO_EMPRESA` VARCHAR(25) NULL ,
  `NCUENTA_EMPRESA` VARCHAR(45) NULL ,
  `NOMBRE_CONTACTO` VARCHAR(40) NULL ,
  `TELEFONO_CONTACTO` VARCHAR(25) NULL ,
  `E_MAIL_CONTACTO` VARCHAR(45) NULL ,
  `OTROS_DATOS` VARCHAR(200) NULL ,
  PRIMARY KEY (`ID_PROVEEDOR`)
ENGINE = InnoDB;
SHOW WARNINGS;
```

```
-----
-- Table `REFACCIONARIA`.`REFACCION`
-----
```

```
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`REFACCION` (
  `CLAVE_REFACCION` VARCHAR(20) NOT NULL ,
  `NOMBRE_REFACCION` VARCHAR(30) NULL ,
  `COSTO` DECIMAL(8,2) NULL ,
  `PRECIO_VENTA` DECIMAL(8,2) NULL ,
  `EXISTENCIAS` DECIMAL(5,1) ZEROFILL NULL DEFAULT 1 ,
  `STOCK` DECIMAL(5,1) ZEROFILL NULL ,
  `MARCA` VARCHAR(40) NULL ,
  `ID_PROVEEDOR` VARCHAR(10) NULL ,
```

```

`GRUPO_REFACCION` VARCHAR(15) NULL ,
`MARCA_AUTO` VARCHAR(30) NULL ,
`APLICACION` VARCHAR(1000) NULL ,
`RUTA_IMAGEN` VARCHAR(100) NULL ,
`FECHA_REGISTRO` DATE NULL ,
PRIMARY KEY (`CLAVE_REFACCION`),
INDEX `ID_PROVEEDOR` (`ID_PROVEEDOR` ASC),
CONSTRAINT `ID_PROVEEDOR`
FOREIGN KEY (`ID_PROVEEDOR` )
REFERENCES `REFACCIONARIA`.`PROVEEDOR` (`ID_PROVEEDOR` )
ON DELETE NO ACTION
ON UPDATE CASCADE)
ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`FACTURAS_COMPRAS`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`FACTURAS_COMPRAS` (
`N_FACTURA_COMPRA` VARCHAR(15) NOT NULL ,
`ESTADO_FACTURA` VARCHAR(10) NULL ,
`DIAS_CREDITO` INT NULL ,
`ID_PROVEEDOR` VARCHAR(10) NOT NULL ,
`FECHA_COMPRA` DATE NULL ,
`SUBTOTAL` DECIMAL(8,2) NULL ,
`MONTO_TOTAL` DECIMAL(8,2) NULL ,
`OBSERVACIONES` VARCHAR(150) NULL ,
PRIMARY KEY (`N_FACTURA_COMPRA`),
INDEX `ID_PROVEEDOR` (`ID_PROVEEDOR` ASC),
CONSTRAINT `ID_PROVEEDOR`
FOREIGN KEY (`ID_PROVEEDOR` )
REFERENCES `REFACCIONARIA`.`PROVEEDOR` (`ID_PROVEEDOR` )
ON DELETE NO ACTION
ON UPDATE CASCADE)
ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`COMPRAS`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`COMPRAS` (
`CLAVE_REFACCION` VARCHAR(20) NOT NULL ,
`CANTIDAD_COMPRA` DECIMAL(5,1) NULL ,
`N_FACTURA_COMPRA` VARCHAR(15) NOT NULL ,
`PRECIO_COMPRA` DECIMAL(8,2) NULL ,
INDEX `CLAVE_REFACCION` (`CLAVE_REFACCION` ASC),
INDEX `N_FACTURA` (`N_FACTURA_COMPRA` ASC),
CONSTRAINT `CLAVE_REFACCION`
FOREIGN KEY (`CLAVE_REFACCION` )
REFERENCES `REFACCIONARIA`.`REFACCION` (`CLAVE_REFACCION` )
ON DELETE NO ACTION
ON UPDATE CASCADE,
CONSTRAINT `N_FACTURA`
FOREIGN KEY (`N_FACTURA_COMPRA` )

```

```

REFERENCES `REFACCIONARIA`.`FACTURAS_COMPRAS` (`N_FACTURA_COMPRA` )
ON DELETE CASCADE
ON UPDATE CASCADE)
ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`VENDEDOR`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`VENDEDOR` (
  `ID_VENDEDOR` VARCHAR(10) NOT NULL ,
  `NOMBRE_VENDEDOR` VARCHAR(40) NULL ,
  `TELEFONO_VENDEDOR` VARCHAR(25) NULL ,
  `CELULAR` VARCHAR(15) NULL ,
  PRIMARY KEY (`ID_VENDEDOR`))
ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`NUEVA_VENTA`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`NUEVA_VENTA` (
  `ID_VENTA` INT NOT NULL ,
  `ID_VENDEDOR` VARCHAR(10) NOT NULL ,
  `FECHA_VENTA` DATE NULL ,
  PRIMARY KEY (`ID_VENTA`),
  INDEX `ID_VENDEDOR` (`ID_VENDEDOR` ASC),
  CONSTRAINT `ID_VENDEDOR`
  FOREIGN KEY (`ID_VENDEDOR` )
  REFERENCES `REFACCIONARIA`.`VENDEDOR` (`ID_VENDEDOR` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`VENTAS`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`VENTAS` (
  `ID_VENTA` INT NOT NULL ,
  `LINEA` INT NULL ,
  `CLAVE_REFACCION` VARCHAR(20) NOT NULL ,
  `CANTIDAD_VENDIDA` DECIMAL(5,1) NULL ,
  `COSTO` DECIMAL(8,2) NULL ,
  `PRECIO` DECIMAL(8,2) NULL ,
  INDEX `CLAVE_REFACCION` (`CLAVE_REFACCION` ASC),
  INDEX `ID_VENTA` (`ID_VENTA` ASC),
  CONSTRAINT `CLAVE_REFACCION`
  FOREIGN KEY (`CLAVE_REFACCION` )
  REFERENCES `REFACCIONARIA`.`REFACCION` (`CLAVE_REFACCION` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `ID_VENTA`
  FOREIGN KEY (`ID_VENTA` )
  REFERENCES `REFACCIONARIA`.`NUEVA_VENTA` (`ID_VENTA` )
  ON DELETE NO ACTION

```

```

ON UPDATE NO ACTION)
ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`CLIENTE`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`CLIENTE` (
`RFC_CLIENTE` VARCHAR(20) NOT NULL ,
`NOMBRE_CLIENTE` VARCHAR(100) NULL ,
`DIRECCION_CLIENTE` VARCHAR(200) NULL ,
`CODIGO_POSTAL` VARCHAR(10) NULL ,
`TELEFONO` VARCHAR(25) NULL ,
PRIMARY KEY (`RFC_CLIENTE`))
ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`FACTURAS_VENTA`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`FACTURAS_VENTA` (
`N_FACTURA` INT NOT NULL ,
`RFC_CLIENTE` VARCHAR(20) NOT NULL ,
`FECHA` DATE NULL ,
`SUBTOTAL` DECIMAL(8,2) NULL ,
`TOTAL` DECIMAL(8,2) NULL ,
PRIMARY KEY (`N_FACTURA`),
INDEX `RFC_CLIENTE` (`RFC_CLIENTE` ASC),
CONSTRAINT `RFC_CLIENTE`
FOREIGN KEY (`RFC_CLIENTE` )
REFERENCES `REFACCIONARIA`.`CLIENTE` (`RFC_CLIENTE` )
ON DELETE CASCADE
ON UPDATE CASCADE)

ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`VENTAS_FACTURADAS`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`VENTAS_FACTURADAS` (
`N_FACTURA` INT NOT NULL ,
`LINEA` INT NULL ,
`CANTIDAD` DECIMAL(5,1) NULL ,
`CLAVE_REFACCION` VARCHAR(20) NOT NULL ,
`PRECIO` DECIMAL(8,2) NULL ,
INDEX `N_FACTURA` (`N_FACTURA` ASC),
INDEX `CLAVE_REFACCION` (`CLAVE_REFACCION` ASC),
CONSTRAINT `N_FACTURA`
FOREIGN KEY (`N_FACTURA` )
REFERENCES `REFACCIONARIA`.`FACTURAS_VENTA` (`N_FACTURA` )
ON DELETE CASCADE
ON UPDATE NO ACTION,
CONSTRAINT `CLAVE_REFACCION`
FOREIGN KEY (`CLAVE_REFACCION` )
REFERENCES `REFACCIONARIA`.`REFACCION` (`CLAVE_REFACCION` )

```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`MARCA_AUTO`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`MARCA_AUTO` (
`ID_MARCA` VARCHAR(15) NOT NULL,
PRIMARY KEY (`ID_MARCA`))
ENGINE = InnoDB;
SHOW WARNINGS;
-----
-- Table `REFACCIONARIA`.`AUTOS`
-----
CREATE TABLE IF NOT EXISTS `REFACCIONARIA`.`AUTOS` (
`ID_MARCA` VARCHAR(15) NOT NULL,
`NOMBRE_AUTO` VARCHAR(25) NULL,
INDEX `ID_MARCA` (`ID_MARCA` ASC),
CONSTRAINT `ID_MARCA`
FOREIGN KEY (`ID_MARCA`)
REFERENCES `REFACCIONARIA`.`MARCA_AUTO` (`ID_MARCA`)
ON DELETE NO ACTION
ON UPDATE CASCADE)

ENGINE = InnoDB;
SHOW WARNINGS;
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

### 6.2.7.- Desarrollo de interfaces de usuario del sistema

Definimos en esta parte cada una de las interfaces con las que trabajaran los usuarios para utilizar los datos de la base de datos, ya sea modificar, actualizar, eliminar o dar de alta algún registro perteneciente a la base de datos.

Las interfaces de usuario que se deben crear son las siguientes:

Interface de usuario que permita al vendedor realizar consultas como si lo estuviera haciendo con un catálogo, pero con la diferencia de que toda la información de cada catalogo esta simplificada y se obtiene en una misma ventana; el vendedor podrá ver cada una de las refacciones que se tienen en existencia para la venta. Desde esta ventana el vendedor podrá ejecutar los

distintos tipos de consultas, por número de parte, por nombre de parte, verificar para qué tipo de autos se tienen refacciones entre otras más.

Interface de usuario donde el administrador pueda agendar las facturas de compra a crédito para saber la situación financiera de la empresa en general, incluso la situación con cada proveedor y las facturas próximas a vencer.

Interface para ingresar los productos que llegan al inventario y de esta forma estar actualizando la información de cada producto.

Interface de usuario donde se lleve el registro de todos los proveedores, con la finalidad de saber que refacciones surte cada proveedor, evaluar quien tiene mejor precio en ciertas refacciones.

Interface de usuario donde se lleve el registro de vendedores, quienes están vendiendo refacciones, la cantidad de ventas por vendedor en algún periodo de tiempo.

Interface de usuario donde se tenga el registro de ventas diarias que genera cada vendedor al realizar ventas, donde se puedan ejecutar consulta de ventas en cierto periodo de tiempo, las refacciones que se venden, imprimir copias de comprobantes de ventas y facturar ventas cuando el cliente lo requiere.

Interface donde el administrador lleve el control de registros de clientes, a los cuales se les factura.

Interface donde el administrador lleve el control de facturas de ventas y pueda imprimir la relación de notas del mes para que el contador realice el reporte fiscal correspondiente sin tener que revisar factura por factura.

Interface de faltantes que se esté actualizando constantemente, para que en un momento dado el administrador consulte los faltantes, y realice el pedido al proveedor.

Una relación de los autos existentes en el mercado para buscar la información dentro de la base de datos.

Un manual de ayuda para el usuario para que en un momento dado sea consultado por cualquiera de los usuarios.

#### **6.2.8.- Programación de las interfaces del sistema**

Para la creación de las interfaces de usuario se usó visual BASIC 6.0.

Sistema operativo Windows XP.

Gestor de bases de datos MySQL.

Se utiliza un conector ODBC que hace el puente para la comunicación de datos entre el servidor MYSQL y visual BASIC 6.0.

Se utilizó el modelo de sistemas de información en Espiral.

#### **6.2.9.- Pruebas de validación y verificación en el sistema**

Una vez que se tiene el sistema terminado, con todas las funcionalidades; se realizaron las pruebas de validación del sistema. Se utilizó el sistema de forma exhaustiva ejecutando cada funcionalidad fijándose que funcionara correctamente y corrigiendo cualquier tipo de error existente.

Se evaluó el sistema con los distintos tipos de usuarios que lo utilizaran, y la respuesta fue positiva, el sistema es muy amigable y contiene los términos técnicos que hacen que sea posible su fácil manejo para los empleados de este negocio de la venta de refacciones automotrices.

#### **6.2.10.- Desarrollo de un manual de usuario para el sistema**

El manual de usuario fue desarrollado como un documento HTML y se encuentra disponible en cada interface de usuario resolviendo las diversas dudas que el usuario pudiera tener acerca de que funciones están contenidas o pueden

realizarse desde cada interface del sistema. De esta forma cada interface tiene su opción de ayuda que describe a detalle cada parte del sistema. En la figura 21, se aprecia la ventana principal del manual de ayuda del sistema.

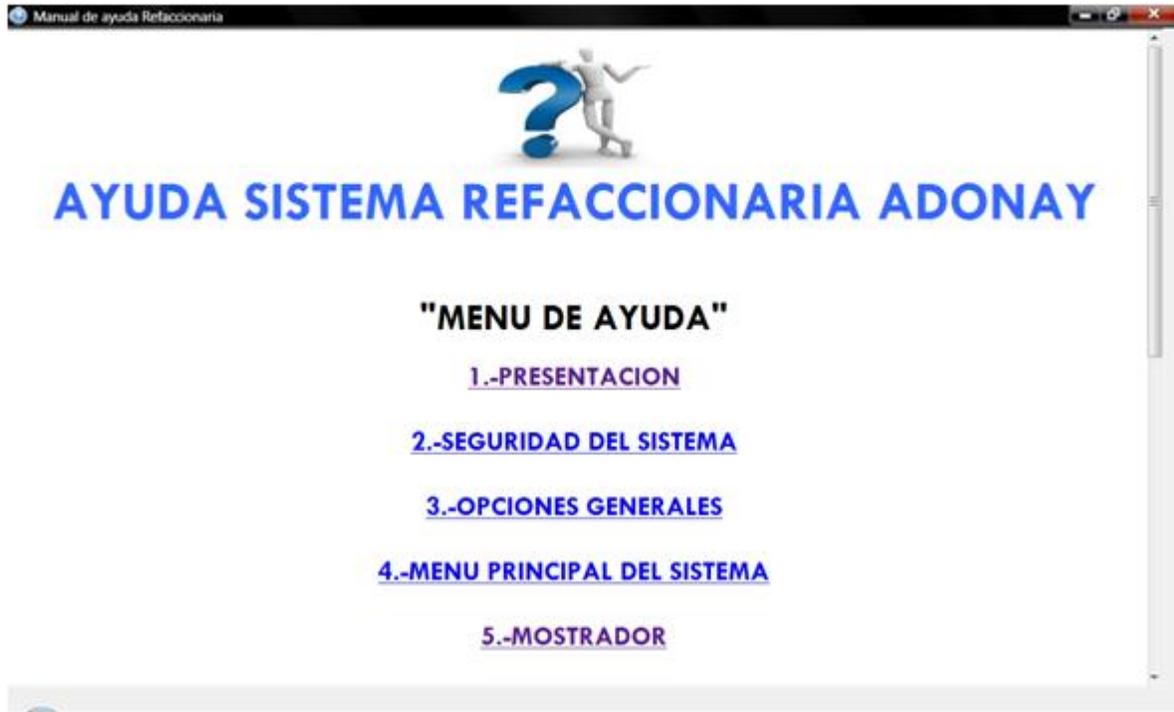


Figura 21. Ventana principal del manual de ayuda del sistema para la refaccionaria

### **6.2.11.- Implementación del sistema**

La forma en que se realizó la implementación, fue trabajando el sistema manual y el sistema basado en computadora en forma conjunta hasta que se tuvo la gran mayoría de los datos de las refacciones del inventario y se fue verificando los resultados obtenidos en ambos sistemas para que sean idénticos. De esta forma se fue fomentando la confianza y la necesidad de utilizar el nuevo sistema.

## VII. RESULTADOS

Este apartado se explica los resultados obtenidos después de implementar el sistema en la refaccionaria “Adonay”.

La entrada al sistema se presenta en la figura 22, esto mientras se cargan los datos necesarios para que funcione el sistema.



Figura 22. Presentación del sistema

Después de tener los datos cargados se entra a la ventana figura 23, que administra la seguridad del sistema, permitiendo entrar a personal autorizado para su uso

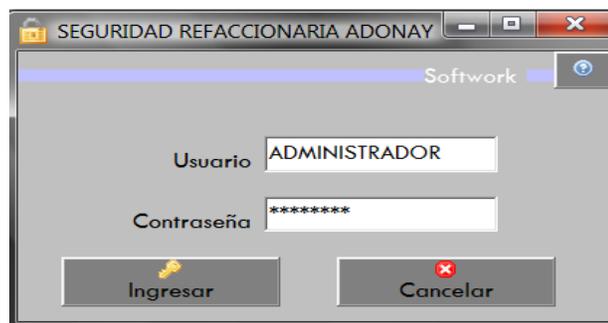


Figura 23. Seguridad del sistema.

Una vez ingresando al sistema, se despliega el siguiente menú (figura 24), que tiene acceso a las distintas partes del sistema que permiten realizar cada una de los procesos que el negocio utiliza para su funcionamiento.



Figura 24. Menú principal del sistema

Al seleccionar la opción “MOSTRADOR”, se despliega la ventana (figura 26) con la que el vendedor trabajara; en esta ventana se podrá detectar que se muestran un conjunto de celdas y cada línea corresponde a una refacción del almacén, donde se pueden visualizar algunos de los datos disponibles, la función principal de esta ventana es la de buscar las refacciones. La búsqueda se hace por clave de la refacción, nombre de la refacción, marca, por grupo de refacciones, por marca de auto, por aplicación o Incluso todas a la vez.

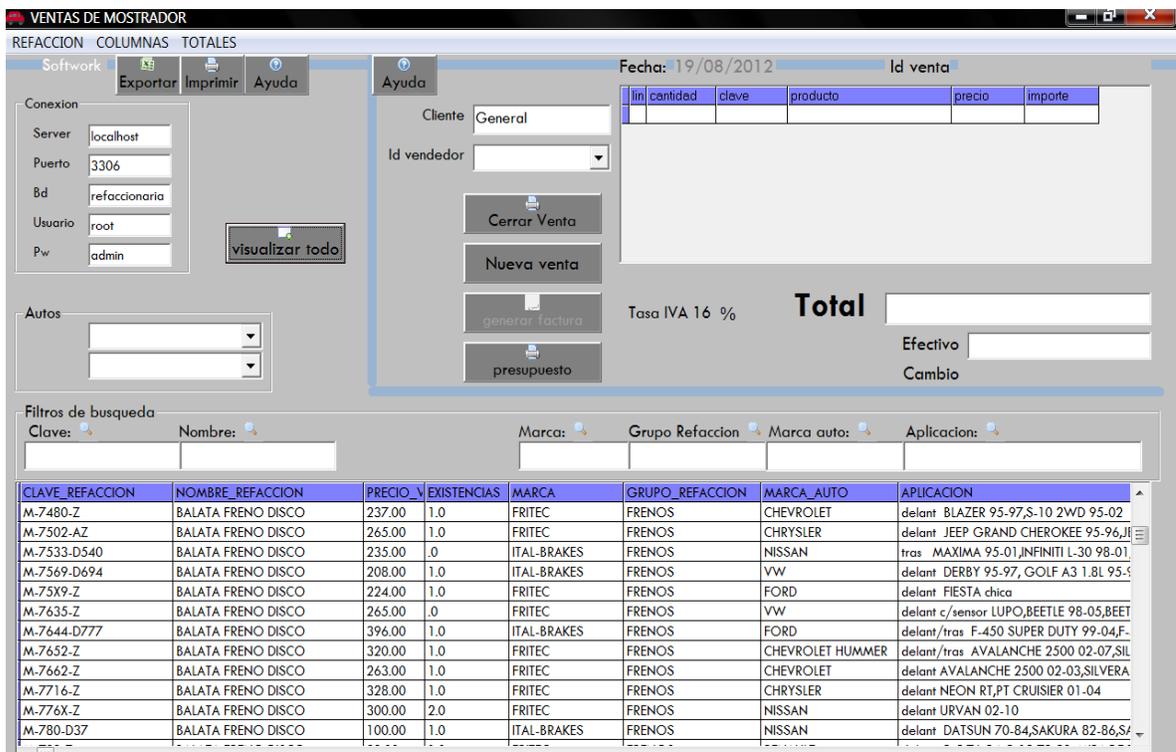


Figura 25. Ventana para ventas de mostrador

Si se selecciona del menú “REFACCIÓN” el usuario podrá agregar nuevas refacciones para ingresarlas al almacén y quedar registrada en la base de datos (Figura 26).



Figura 26. Menú Agregar refacción

De este submenú al oprimir la opción “AGREGAR”, se desplegará la ventana (figura 27) donde se podrán capturar los datos necesarios de una refacción y de esta forma queden gravados en la base de datos y se pueda consultar a futuro.

A screenshot of a software window titled "AGREGAR REFACCION". The window contains several input fields and buttons. The fields are: "Clave refaccion", "Nombre refaccion", "Costo", "Precio de venta", "Existencias", "Stock", "Marca", "Proveedor" (a dropdown menu), "Grupo de refaccion", "Marca auto", "Aplicacion" (a large text area), "Ruta imagen", and "Fecha de registro" (with the value "09/08/2012"). There are also two dropdown menus under "Consultar Autos" and a button labeled "Examinar...". At the bottom, there are two buttons: "Guardar" and "Cancelar".

Figura 27. Ventana agregar refacción.

Considerando que la captura de datos es similar en todas las ventanas de las diversas capturas, muchas de ellas no serán mencionadas, esto porque se dará mayor énfasis a las ventanas que despliegan informes que el usuario necesita, ya que son los de mayor interés para el usuario y gracias a ellos el usuario ha tenido un ahorro de tiempo considerable que es una de las ventajas principales que tiene este sistema.

Una cualidad del sistema es que se podrán imprimir las consultas que se realicen, el formato que normalmente se visualizara será semejante a l de la Figura 28.

Esta impresión normalmente es manipulada por el administrador que podrá imprimir los documentos necesarios para realizar una verificación del inventario de productos.

CLAVE	NOMBRE	PRECIO	MARCA	GRUPO	MARCA	APLICACION
7177-227-SM	BALATA FRENO	187	LUSAC	FRENOS	SEAT	IBIZA 01-01, CORDOBA 01-02,
7183-280-SM	BALATA FRENO	179	LUSAC	FRENOS	VW	del ATLANTIC GLS
7192-289-SM	BALATA FRENO	180	LUSAC	FRENOS	CHEVROLET	del CAVALIER 90-91, CAVALIER
7209-280-SM	BALATA FRENO	180	LUSAC	FRENOS	VW	del disco solido ATLANTIC GLS
7228-333SM	BALATA FRENO	216	LUSAC	FRENOS	NISSAN	DELANT. URVAN
7234-340-SM	BALATA FRENO	217	LUSAC	FRENOS	VW	tras DERBY 02-05, GOLF A3
7242-350-SM	BALATA FRENO	240	LUSAC	FRENOS	VW	del POINTER 98-99, POINTER
7242-350A-S	BALATA FRENO	220	LUSAC	FRENOS	VW	8288-350-SM NUEVO # DEL.
7259-368-SM	BALATA FRENO	223	LUSAC	FRENOS	CHRYSLER	del RAM 1500 94-98, C-15
7259-369-SM	BALATA FRENO	230	LUSAC	FRENOS	CHEVROLET	del C-15 86-91, C-15 94-99, C-20
7260-370SM	BALATA FRENO	220	LUSAC	FRENOS	CHEVROLET	C-20 91-93, C-2500 PICK UP
7264-375-SM	BALATA FRENO	265	LUSAC	FRENOS	FORD	del B.200 88-90, F-150 88-93
728-52-SM	BALATA FRENO	195	LUSAC	FRENOS	DINA	del D-3200 75-82, D-3100
7282-441-SM	BALATA FRENO	189	LUSAC	FRENOS	NISSAN	del PICK UP DOBLE CABINA
7293-440SM	BALATA FRENO	225	LUSAC	FRENOS	DODGE	VERNA 04-06 DELT.
7304-	BALATA FRENO	280	LUSAC	FRENOS	FORD	del MERCURY COUGAR
7318-430-SM	BALATA FRENO	260	LUSAC	FRENOS	NISSAN	del ALTIMA 98-01
7335-340-SM	BALATA FRENO	225	LUSAC	FRENOS	AUDI	tras todos PEUGEOT 307 CC
7338-458-SM	BALATA FRENO	230	LUSAC	FRENOS	FORD	tras todos ESCORT 94-96,
7342-462-SM	BALATA FRENO	255	LUSAC	FRENOS	NISSAN	del 240SX/ABS 98-99, SENTRA
7347-D363-S	BALATA FRENO	238	LUSAC	FRENOS	CHEVROLET	LUV 97-05
7353-473-SM	BALATA FRENO	187	LUSAC	FRENOS	FORD	del ESCORT 94-99, ESCORT GT
7358-712-SM	BALATA FRENO	260	LUSAC	FRENOS	CHRYSLER	del WRANGLER 93-96 JEEP
7385-506-SM	BALATA FRENO	275	LUSAC	FRENOS	CHEVROLET	del CAVALIER 92-94, CAVALIER
7389-509-SM	BALATA FRENO	265	LUSAC	FRENOS	NISSAN	del LUCCINO 96-00, TSURU 2000
7389-510-SM	BALATA FRENO	155	LUSAC	FRENOS	NISSAN	del SENTRA 1.6 92, TSURU GS

Figura 28. Impresión de consulta de refacciones

En la ventana “VENTAS DE MOSTRADOR”, el vendedor podrá realizar consulta de refacciones e ir las colocando en una remisión digital para posteriormente convertirla en una venta, generando los datos pertinentes en remisión, factura de venta o bien simplemente imprimir el presupuesto para el cliente. Ver figura 29.

**VENTAS DE MOSTRADOR - VISTA PREVIA TICKET**

Fecha: 09/08/2012    Id venta: 10278

Cliente: General    Vendedor: OSCAR

RFC: AARR 671010 DZ7  
Av. pasacual luna s/n, tezoayuca centro

REMISION FECHA: 09/08/2012 20:39  
FOLIO: 10278

L	CANT	CLAVE	IMPORTE
1	1	ACAMG	330.00
A CASTROL UNIT			330.00
2	4	BPR5EGP	140.00
BUJIA PLATINO UNIT			35.00
3	1	89050475	300.00
CABLES PARA BUJIAS AC/I			300.00
4	1	GP46	42.00
FILTRO ACEITE UNIT			42.00
5	1	GA-134	38.00
FILTRO AIRE UNIT			38.00
6	1	FGI12	46.00
FILTRO GASOLINA UNIT			46.00
7	1	527MX	65.00
JUNTAS TBI UNIT			65.00
TOTAL			\$961.00
EFFECTIVO			\$1,000.00
CAMBIO			\$39.00

\*\*\*NOVECIENTOS SESENTA Y UN PESOS 0/100 M.N.\*\*\*

Tasa IVA 16 %    **Total 961**

Efectivo 1000  
Cambio 39

Marca:    Grupo Refaccion:    Marca auto:    Aplicacion:

STENCIAS	MARCA	GRUPO_REFACCION	MARCA_AUTO	APLICACION
0	AC/DELCO	AFINACION	CHEVROLET,VW,CHRYS	N12Y,CHEVY,JETTA,GOLF,COMBI,TSURU
0	NGK	AFINACION	VW,CHRYSLER,NISSAN,	N12Y JETTA TSURU II CHEVY
0	NGK	AFINACION	VW,CHRYSLER,NISSAN,	JETTA TSURU II CHEVY
0	LANCER	AFINACION	CHEVROLET	CHEVY,C2,CORSA,TORNADO,MERIVA.
0	HY POWER	AFINACION	CHEVROLET	CHEVELLE, CHEVYNOVA, IMPALA OPEL, /
0	HY POWER	AFINACION	CHEVROLET	CHEVELLE, CHEVYNOVA, CONTOUR, MA
0	LANCER	AFINACION	CHEVROLET	CHEVY , MONZA 94-04 A CIL. 1.4 1.6 LTS
0	AC DELCO	AFINACION	CHEVROLET	CHEVY,CORSA,TODOS
0	AC DELCO	AFINACION	CHEVROLET	CHEVY
0	GONHER	AFINACION	CHEVROLET	CAMARO Z8, PONTIAC, FIREBIRD, TRANS
0	GONHER	AFINACION	CHEVROLET,PONTIAC	ASTRA 4L 1.8 2.0 I 02-06, CHEVY C2 4L
0	AC/DELCO	AFINACION	CHEVROLET	CHEVY,CORSA,ASTRA,OPTRA.

Figura 29. Venta y presupuesto de refacciones

Obsérvese que se imprimirá un ticket con la información necesaria para que el cliente compruebe lo que está comprando.

La información del ticket, es almacenada en la base de datos del sistema para poder realizar tareas administrativas con respecto a las ventas. También es

posible volver a imprimir una copia del mismo ticket, o en su defecto facturar los artículos correspondientes a un ticket o varios.

Otro aspecto importante que es necesario para un buen manejo en la refaccionaria es la edición de los datos de las refacciones para ser actualizados e incluso ir enriqueciendo el catalogo, para posteriores consultas (Figura 30).

The screenshot shows a software window titled "EDITAR REFACCION" with a "Softwork" logo and an "Ayuda" button. The form contains the following data:

Clave refaccion	SPC-72X1-Z
Nombre refaccion	BALATA FRENO DISCO
Costo	193.11
Precio de venta	238
Existencias	1
Stock	1
Marca	FRITEC
Proveedor	ADELAR
Grupo de refaccion	FRENOS
Marca auto	CHEVROLET JAGUAR
Aplicacion	delant CHEVY JOY 94-03,CHEVY SWING 94-03,CHEVY POP 96-03,CHEVY MONZA 97-03,CHEVY VAGONETA 00-03(opcional),CHEVY PICK UP 99-03 (opc.), CHEVY C2 SEDAN 04-07,CHEVY HATCH BACK 04-07
Ruta imagen	C:/imagenes/cok.jpg
Fecha de registro	17/04/2012

Additional interface elements include a "Vender" panel with a "CANTIDAD A VENDER" field and a "Vender" button, a "Consultar Autos" panel with two dropdown menus, and "Eliminar" and "Guardar cambios" buttons at the bottom.

Figura 30. Edición de información de refacciones

En la ventana donde se Guardan y administran las facturas de compra, el administrador podrá realizar distintas consultas, a los documentos de pago. Con los informes que se generan se podrán tener en cuenta las facturas próximas a

vencer, así como para saber cuál es la situación que se tiene con cada proveedor (Figura 31).

The screenshot shows a software interface for managing purchase invoices. It includes search filters, sorting options, and a data table.

**Facturas de Compra**

Software: Sotwork

Exportar | Imprimir | Ayuda

**filtros de búsqueda**

N. factura:  Fecha compra:

Monto total:

Edo. factura:

Id proveedor:

visualizar todo

**Ordenar**

prioridad de pago

proveedor

Total

**Consulta entre 2 fechas de pago**

febrero 2010 | noviembre 2010

31 1 2 3 4 5 6 | 31 1 2 3 4 5 6

7 8 9 10 11 12 13 | 7 8 9 10 11 12 13

14 15 16 17 18 19 20 | 14 15 16 17 18 19 20

21 22 23 24 25 26 27 | 21 22 23 24 25 26 27

28 1 2 3 4 5 6 | 28 29 30 1 2 3 4

7 8 9 10 11 12 13 | 5 6 7 8 9 10 11

Hey: 12/09/2012 | Hey: 12/09/2012

N_FACTURA_COM	MONTO_TOTAL	ID_PROVEEDOR	DIAS_CREDITO	ESTADO_FACTURA	FECHA_COMPRA	FECHA_VENCIMIENTO	OBSERVACIONES
135395	1024.36	ADELAR	30	ADEUDO	2012-07-31	2012-08-30	(NINGUNA)
135341	2838.74	ADELAR	30	ADEUDO	2012-07-31	2012-08-30	(NINGUNA)
14449	6723.00	MARGAB	30	ADEUDO	2012-07-31	2012-08-30	(NINGUNA)
135070	1466.35	ADELAR	30	ADEUDO	2012-07-28	2012-08-27	(NINGUNA)
134800	4737.88	ADELAR	30	ADEUDO	2012-07-27	2012-08-26	(NINGUNA)
453734	4971.97	AUTOTODO	30	ADEUDO	2012-07-26	2012-08-25	(NINGUNA)
28832	11280.00	FORD E	30	ADEUDO	2012-07-25	2012-08-24	(NINGUNA)
56012	2502.41	QUAKER	30	ADEUDO	2012-07-24	2012-08-23	(NINGUNA)
133646	2026.00	ADELAR	30	ADEUDO	2012-07-24	2012-08-23	(NINGUNA)
83802	2593.11	HESA	30	ADEUDO	2012-07-24	2012-08-23	(NINGUNA)
13772	1450.00	MARGAB	30	ADEUDO	2012-07-24	2012-08-23	(NINGUNA)
133647	2538.06	ADELAR	30	ADEUDO	2012-07-24	2012-08-23	(NINGUNA)
133299	2675.20	ADELAR	30	ADEUDO	2012-07-21	2012-08-20	(NINGUNA)
133298	2732.78	ADELAR	30	ADEUDO	2012-07-21	2012-08-20	(NINGUNA)
133300	589.44	ADELAR	30	ADEUDO	2012-07-21	2012-08-20	(NINGUNA)
1390	1632.00	CUELLAR	30	ADEUDO	2012-07-20	2012-08-19	(NINGUNA)

Figura 31. Administración de facturas de compra

Desde la “VENTANA COMPRAS INGRESAR AL INVENTARIO” el administrador podrá ingresar de forma fácil y rápida a los productos que se van resurtiendo, así como, ir verificando y actualizando los precios de los productos para proteger la inversión (Figura 32).

COMPRAS INGRESAR AL INVENTARIO

Softwork Imprimir Ayuda

Datos para ingresar compras

Clave refaccion

Cantidad de compra

Numero de factura

Utilidad  %

Precio Compra

Precio venta

Filtros de búsqueda

Clave refaccion

N. Factura

Control de registros

Añadir mercancía al inventario

borrar registro

Visualizar todo

CLAVE_REFACCION	CANTIDAD_COMPRA	N_FACTURA_COMPRA	PRECIO_COMPRA	IMPORTE
CR928	12.0	1	12.00	144.000
CR928	1.0	1	12.00	12.000

Figura 32. Ventana compras ingresar al inventario

A través de la ventana “PROVEEDORES”, el administrador podrá ingresar los nuevos proveedores ligados a las refacciones que se compran para vender, así como, información importante para la localización de cada proveedor. (Figura 33)

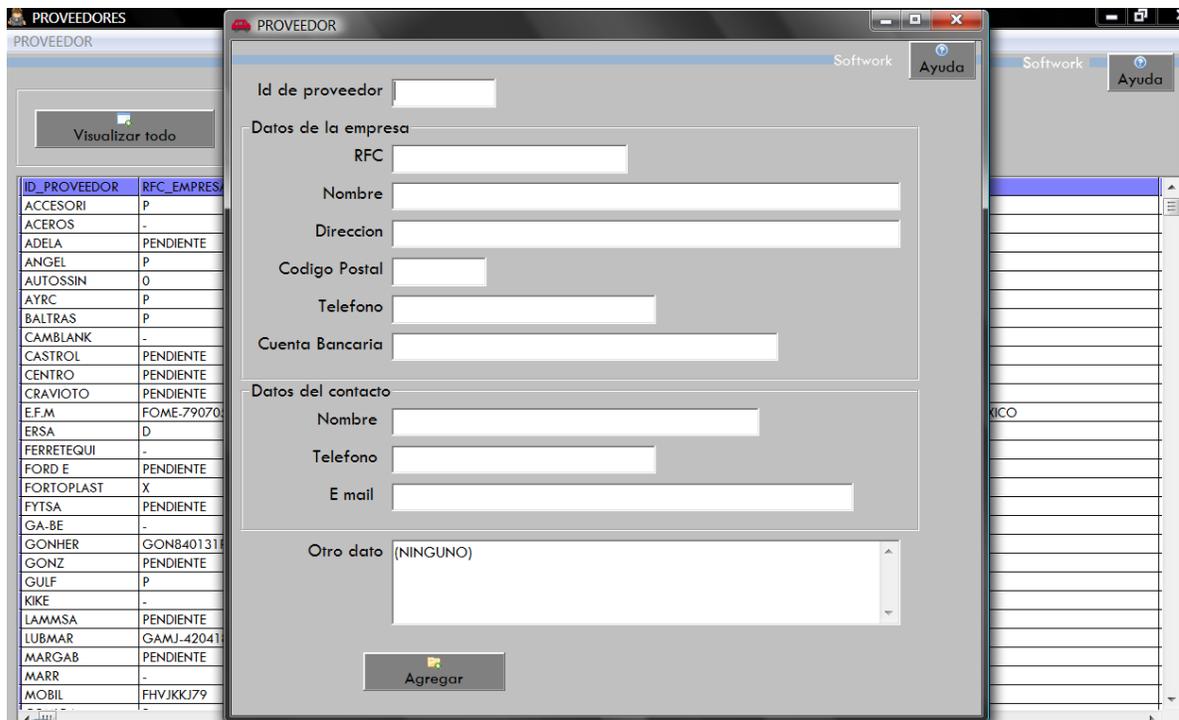


Figura 33. Ventana Proveedores

En la “VENTANA CONSULTAR VENTAS”, el administrador podrá consultar en cualquier momento cada una de las ventas realizadas por los vendedores y de esta forma obtener en cualquier momento los datos acerca de la venta total, de un periodo de tiempo determinado, o bien facturar las ventas generales del mes, para que el contador realice el reporte fiscal correspondiente (Figura 34).

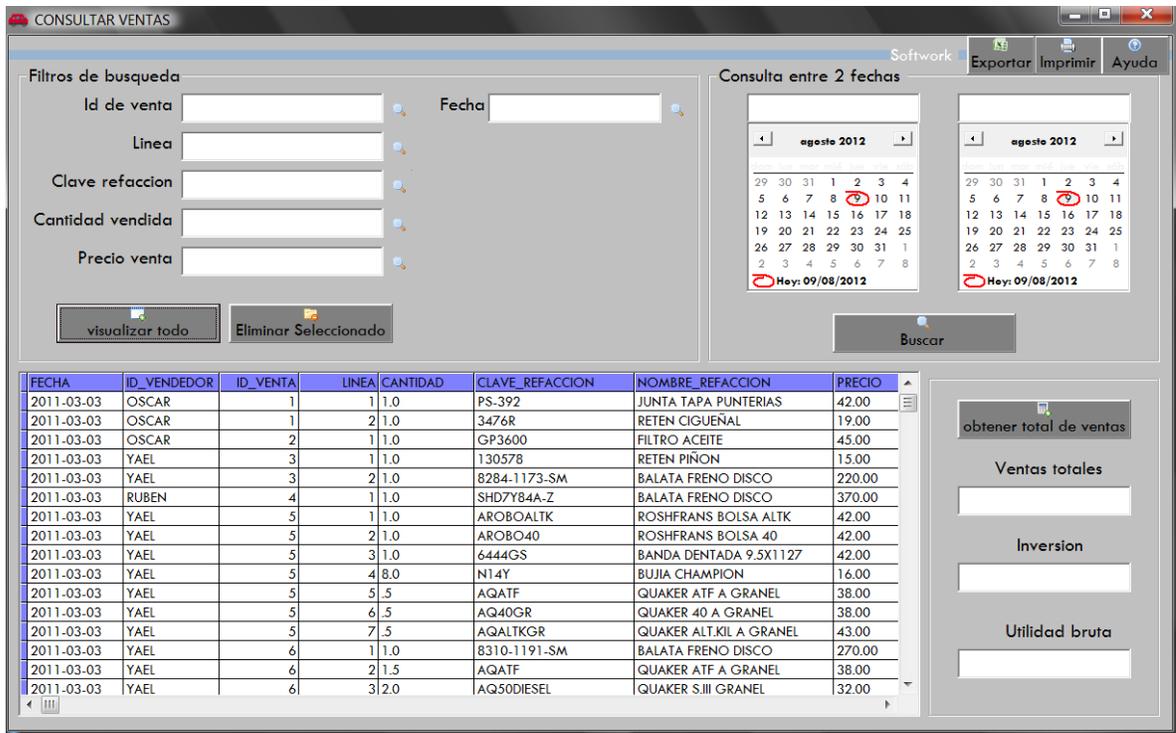


Figura 34. Ventana consultar ventas.

En la “VENTANA VENDEDORES”, el administrador obtiene los datos de cada uno de los vendedores con el fin de ligar las ventas que cada vendedor ha realizado (Figura 35).

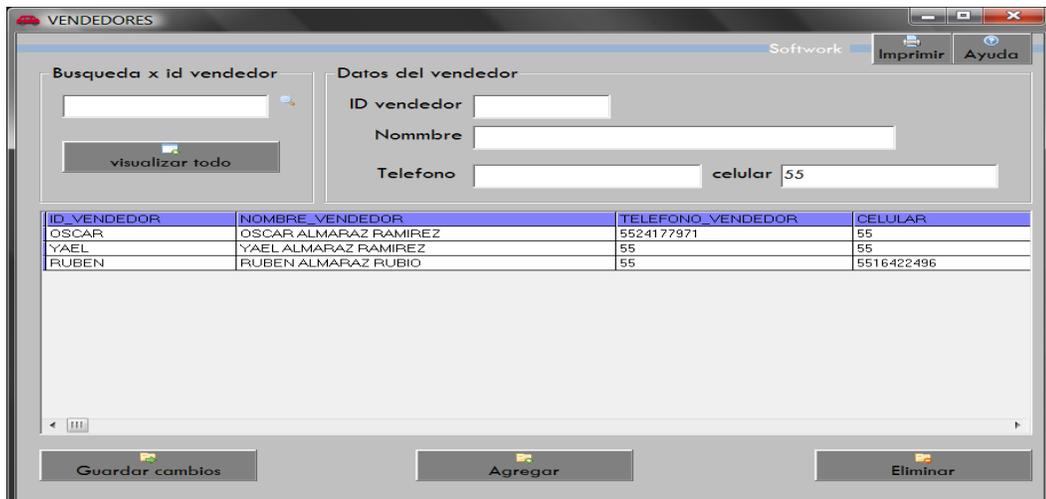


Figura 35. Ventana vendedores

Por medio de la “VENTANA CLIENTES” el administrador tiene disponible la información de cada uno de los clientes, esta información será guardada desde el momento que un cliente hace una compra, sin la necesidad de mostrar su cédula fiscal en cada compra que realice para la facturación, además estos datos son llevados para la impresión de la factura, esto se logra, con tan sólo presionar unas teclas sin necesidad de escribir manualmente estos datos (Figura 36).

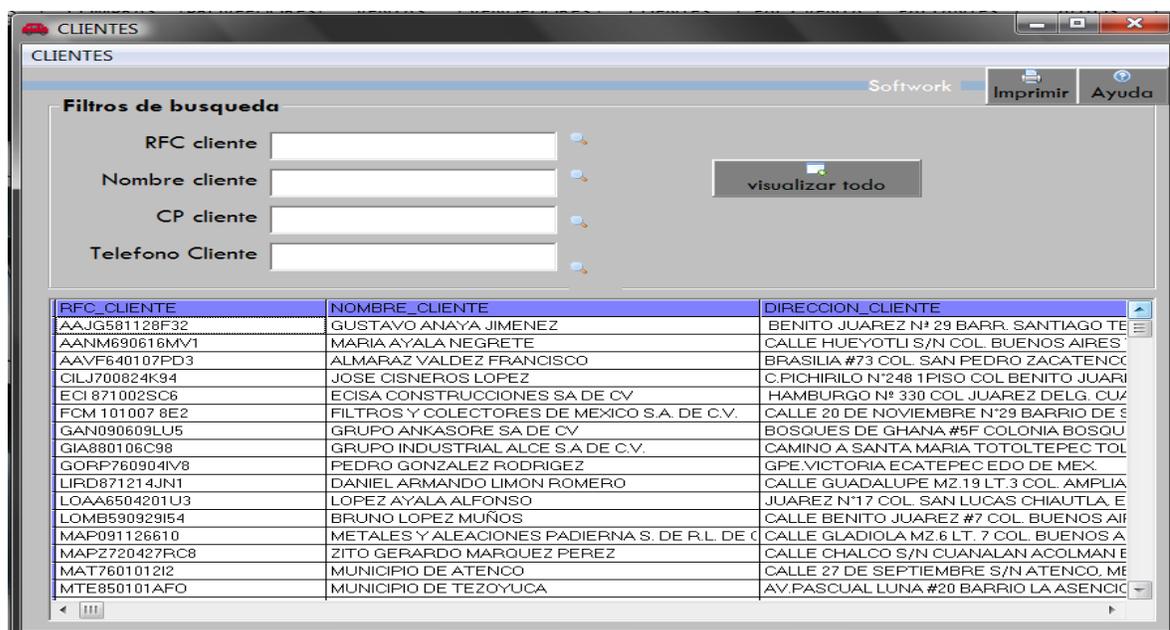


Figura 36. Ventana Clientes

A través de la “VENTANA FACTURAS DE VENTA”, el administrador puede revisar el registro de cada una de las facturas expedidas, de esta forma el administrador tiene una copia digital de cada factura. Además, es posible realizar desde esta ventana un informe o bosquejo del reporte fiscal del mes correspondiente a fin de proporcionar los datos al contador de una manera formal, completa y con cálculos sin errores, de esta forma el contador se ahorra tiempo en estar revisando y calculando cada documento (Figura 37).

FACTURAS DE VENTA

Softwork Exportar Imprimir Ayuda

**Filtros de búsqueda**

N factura

RFC cliente

Fecha  septiembre 2012

26 27 28 29 30 31 1  
2 3 4 5 6 7 8  
9 10 11 12 13 14 15  
16 17 18 19 20 21 22  
23 24 25 26 27 28 29  
30 1 2 3 4 5 6  
Hoy: 12/09/2012

**Consulta entre 2 fechas**

septiembre 2012 septiembre 2012

26 27 28 29 30 31 1  
2 3 4 5 6 7 8  
9 10 11 12 13 14 15  
16 17 18 19 20 21 22  
23 24 25 26 27 28 29  
30 1 2 3 4 5 6  
Hoy: 12/09/2012

Buscar

**Totales**

Obtener total

Subtotal

IVA

Venta Total

**Datos del informe**

Mes

Año

Eliminar Seleccionado

visualizar todo

N_FACTURA	RFC_CLIENTE	NOMBRE_CLIENTE	SUBTOTAL	IVA	TOTAL	FECHA
10	AAJG581128F32	GUSTAVO ANAYA JIMENEZ	81.90	13.10	95.00	2012-03-28
11	AAJG581128F32	GUSTAVO ANAYA JIMENEZ	81.90	13.10	95.00	2012-03-28
12	AAJG581128F32	GUSTAVO ANAYA JIMENEZ	616.38	98.62	715.00	2012-03-28

Figura 37. Ventana Facturas de venta

En la siguiente ventana muestra a detalle cada factura expedida por el sistema (Figura 38).

DETALLE DE FACTURA

N factura: 10      Softwork

**Datos del cliente**

Nombre: GUSTAVO ANAYA JIMENEZ

CP: CP 56000      RFC: AAJG581128F32

Direccion: BENITO JUAREZ N° 29 BARR. SANTIAGO TEZOYUCA

Telefono: PENDIENTE

LINEA	CANTIDAD	CLAVE_REF.	NOMBRE_REFACCION	PRECIO	IMPORTE
1	1.0	328688SKF	45449 BALERO	81.90	81.90

\*\*\* (NOVENTA Y CINCO PESOS 0/100 M.N) \*\*\*

Subtotal: 81.90

Iva: 13.10

total: 95.00

Figura 38. Ventana detalle de factura de venta

Este es un ejemplo de un bosquejo de facturas emitidas por el sistema, para una posterior declaración fiscal ante la Secretaria de Hacienda y Crédito Público (Figura 39).

Reporte de facturas de venta

Zoom 100%

Ruben Almaraz Rubio Y/O  
Ferreteria y Refaccionaria Adonay

POLIZA DIARIO N. 1

CTA	CAJA			
		INGRESOS POR SERVICIOS		\$780.18
				\$780.18
Fact 10		GUSTAVO ANAYA JIMENEZ	\$81.90	
Fact 11		GUSTAVO ANAYA JIMENEZ	\$81.90	
Fact 12		GUSTAVO ANAYA JIMENEZ	\$616.38	
		IVA TRANSLADADO	\$124.82	\$124.82
SUMAS IGUALES				\$780.18

INGRESOS OBTENIDOS DURANTE EL MES DE MARZO DEL EJERCICIO 2012

HECHO	REVISADO	AUTORIZO	DIARIO

Página: 1 De: 1

Páginas: 1

Figura 39. Bosquejo de informe fiscal de ventas

Desde la “VENTANA DE FALTANTES”, el administrador podrá visualizar cada una de las refacciones que deben resurtirse y que están por debajo del stock de existencias que deben de tenerse como mínimo, de esta forma el administrador puede ordenar y consultar los faltantes para poder realizar el pedido y no quedar sin esa refacción (Figura 40).

FALTANTES

Softwork

Exportar Imprimir Ayuda

Filtros de búsqueda

Clave  Id proveedor

Nombre  Marca auto

Marca

Visualizar todo

CLAVE_REFACCION	NOMBRE_REFACCION	MARCA	ID_PROVEEDOR	MARCA_AUTO	COSTO	EXISTENCIAS	STOCK	FALTANTES
105044T	TERMINAL BATERIA 4TORNILLOS	NINGUNA	ACCESORI	UNIV	17.00	3.0	10.0	7.0
1141B-9	FOCO LED BLANCO	AREMI	ACCESORI	UNIV	30.00	.0	1.0	1.0
1141R-9	FOCO LED ROJO	AREMI	ACCESORI	UNIV	30.00	.0	1.0	1.0
1157B-9	FOCO LED AZUL	AREMI	ACCESORI	UNIV	30.00	.0	1.0	1.0
1157R-9	FOCO LED ROJO	AREMI	ACCESORI	UNIV	30.00	.5	1.0	.5
5W81	INTERRUPTOR. 2 PASOS ECON.	AMERICAN	ACCESORI	UNIVERSAL	11.00	1.0	10.0	9.0
9004ION	FOCO 9004 BLISTER	ION	ACCESORI	UNIVERSAL	70.00	2.0	3.0	1.0
ALMVIK125ML.	ALMOROL VIKINGO	VIKINGO	ACCESORI	UNIV	19.23	.0	1.0	1.0
ANTELEC.	ANTENA ELECTRONICA	P	ACCESORI	UNIV	27.05	.0	1.0	1.0
ANTVW	ANTENA SEDAN	AREMI	ACCESORI	VW	30.00	.0	1.0	1.0
ARLED9	ARILLOS LED. 9CM.	TUNIX	ACCESORI	UNIV	85.00	.0	1.0	1.0
BARRA 40CM	DE LED.	SIN	ACCESORI	UNIV	52.00	.0	1.0	1.0
CAMBIO	ACEITE	NO	ACCESORI	NO	.00	967.0	1000.0	33.0
CAPL4N	CABLE P/BATERIA	SIN	ACCESORI	UNIV	40.00	.5	2.0	1.5
CAPL4R	CABLE P/BATERIA	SIN	ACCESORI	UNIV	40.00	.3	2.0	1.7
CIREF1"	CINTA REFLEJANTE	3M	ACCESORI	UNIV	4.50	1.0	2.0	1.0
DS-175	SWITCH JALON 75AMP.	TOP AMERICAN	ACCESORI	UNIV	21.00	.0	1.0	1.0
ESTROBO AVA	ESTROBO	AREMI	ACCESORI	UNIV.	100.00	.0	1.0	1.0
FIAMM	BOCINAS CARACOL	FIAMM	ACCESORI	UNIV	115.00	.0	1.0	1.0
FLG	FARO LUPA GRANDE	CHINO	ACCESORI	UNIVERSAL	116.00	.0	2.0	2.0
FOLED7	FOCOS LED 7	SIN	ACCESORI	UNIV	6.00	6.0	10.0	4.0
FOLED9	FOCOS LED 9	SIN	ACCESORI	UNIV	6.00	6.0	10.0	4.0

Figura 40. Ventana de faltantes de refacciones.

El administrador o vendedor a través de la “VENTANA AUTOS” podrá actualizar el catálogo de autos y marcas que va aumentando año con año (Figura 41).

AUTOS

Softwork

Ayuda

Consultar

Marca Auto

Nombre Auto

Eliminar marca

Eliminar auto

Agregar registro

Marca Auto

Nombre Auto

Guardar

Figura 41. Ventana de catálogo de autos

## VIII. DISCUSIÓN

El sistema para la venta de refacciones automotrices realmente ha respondido a los requisitos que originaron su desarrollo. Para probarlo se ha medido la eficiencia de los procesos con el nuevo sistema y lo que se tardaba sin él.

El siguiente cuadro muestra tareas cotidianas que son realizadas por el personal de la refaccionaria, mostrando el tiempo que se tardaban en hacerlo de forma manual y posteriormente con el sistema que se ha implementado.

<b>Comparación de los sistemas utilizados por el negocio de refaccionaria</b>			
<b>Tarea</b>	<b>Descripción</b>	<b>Tiempo consumido con el sistema manual</b>	<b>Tiempo consumido con el sistema basado en computadora</b>
Búsqueda de una refacción en el catalogo	Implica tomar un catálogo de algún tipo de refacciones y buscar por nombre de auto, modelo, número de cilindros.	2 minutos	20 segundos
Búsqueda de dos refacciones en el catalogo	Implica tomar un catálogo de algún tipo de refacciones y buscar por nombre de auto, modelo, número de cilindros. Repetir el proceso para cada refacción.	4 minutos	20 segundos
Búsqueda de tres refacciones en el catalogo	Implica tomar un catálogo de algún tipo de refacciones y buscar por nombre de auto, modelo, número de cilindros. Repetir el proceso para cada refacción.	6 minutos	20 segundos
Búsqueda de cuatro refacciones en el catalogo	Implica tomar un catálogo de algún tipo de refacciones y buscar por nombre de auto, modelo, número de cilindros. Repetir el proceso para cada refacción.	8 minutos	20 segundos
Búsqueda de cinco refacciones en el catalogo	Implica tomar un catálogo de algún tipo de refacciones y buscar por nombre de auto, modelo, número de cilindros. Repetir el proceso para cada	10 minutos	20 segundos

	refacción.		
Búsqueda de seis refacciones en el catalogo	Implica tomar un catálogo de algún tipo de refacciones y buscar por nombre de auto, modelo, número de cilindros. Repetir el proceso para cada refacción.	12 minutos	20 segundos
Búsqueda de siete refacciones en el catalogo	Implica tomar un catálogo de algún tipo de refacciones y buscar por nombre de auto, modelo, número de cilindros. Repetir el proceso para cada refacción.	14 minutos	20 segundos
Elaboración de una nota de remisión	Tomar una nota de remisión en blanco y anotar refacción por refacción que se vende y posteriormente realizar los cálculos para obtener el total de la venta	De 1 a 15 refacciones  5 minutos promedio	30 segundos
Elaboración de una Factura	Elaborar línea por línea cada refacción vendida, desglosar el IVA a cada precio de, calcular el subtotal, calcular el IVA y el total finalmente	De 1 a 15 refacciones  8 minutos promedio	30 segundos
Suma de las ventas del día	Implica sumar línea por línea de las refacciones vendidas en el transcurso del día laboral.	5 minutos	10 segundos
Suma de las ventas de la semana	Implica sumar línea por línea de las refacciones vendidas en el transcurso de la semana laboral.	30 minutos	10 segundos
Suma de las ventas del mes	Implica sumar línea por línea de las refacciones vendidas en el transcurso del mes laboral.	2 horas 50 min  En meses de 30 días.	10 segundos
Reporte anual de ventas	Implica sumar línea por línea de las refacciones vendidas en el transcurso del año laboral.	30 horas	10 segundos
Listado de faltantes	Posterior a una venta el vendedor tiene que apuntar en la libreta de faltantes, las refacciones que se vendieron y que se encuentran ya en el stock mínimo de existencias y por lo		

	tanto debe resurtirse.	1 hora	10 segundos
Agendar documentos de compra a crédito	Anotar en una agenda. Información de una factura de compra de acuerdo a su fecha de pago	1 minuto	30 segundos
Ingreso de compras al inventario	Implica marcar cada artículo que se adquiere con la fecha en que ingresa al inventario y su precio correspondiente	1 hora	10 minutos

Cuadro 10. Comparación de los sistemas utilizados por el negocio de refaccionaria.

En el caso de consultas de refacciones con el sistema manual va aumentando el tiempo con cada refacción más que debe buscarse, ya que cada refacción debe buscarse en un catálogo distinto; mientras tanto con el sistema sólo basta escoger en los combos el auto para el que se están buscando refacciones y despliega todas las refacciones existentes, si se quiere ser un poco más específico se busca por grupo de refacciones, o si se quiere ser aún más específico se busca por nombre de parte, o bien por su clave y basta con unos movimientos para ubicarse en un contenedor para realizar posteriormente una factura o una remisión en unos cuantos segundos.

En el caso de la elaboración de documentos de compra, existe también una gran diferencia en cuanto al consumo de tiempo, ya que esta información ya se tiene ubicada en un contenedor y sólo basta con presionar algunas teclas para estar imprimiendo una factura o una remisión según sea el caso.

Si eso no es suficiente el obtener la suma total de ventas por día ya no es más una tarea rutinaria, ya que en el sistema sólo basta presionar la fecha del día en el calendario y hacer click en el botón totales y se tiene la venta del día, así como las utilidades brutas obtenidas (información que con el sistema manual era casi imposible de conseguir). No se diga realizar el reporte del mes, tan sólo basta con seleccionar dos fechas, hacer click y se tienen los datos de las ventas del mes, así

como, el desglose de utilidades. Se sigue el mismo procedimiento para obtener el informe de ventas del año y el tiempo consumido es exactamente el mismo, nótese que con el sistema manual en la medida que crece el intervalo de tiempo del informe es cada vez más complicada.

El listado de faltantes ya no es más un proceso repetido, ya que al momento de ejecutar el proceso de venta, la información sobre existencias se actualiza y en el momento de verificar los faltantes, se tendrá siempre actualizada la información.

Agendar las fechas de pago aparentemente no tiene gran diferencia en cuanto al tiempo gastado; sin embargo, los beneficios que trae el agendar dentro del sistema basado en computadora son mejores porque se puede consultar en segundos; cuanto se debe a cada proveedor, cuanto se debe en total, cuales son las facturas próximas a vencer, que facturas ya se pagaron entre otros más beneficios.

Ingresar los productos se vuelve una tarea más fácil debido a que no hay que etiquetar producto a producto, sino simplemente ingresar las actualizaciones en la base de datos, sólo los datos necesarios para este proceso. Y finalmente acomodar los productos en su lugar.

Algunos otros beneficios no mesurables, es que en cualquier momento el administrador puede corroborar el inventario con la información que se tiene en la base de datos, con el fin de evitar algún tipo de fuga en cuanto extracción de mercancía sin autorización.

Además con el sistema es posible obtener un reporte de facturas pagadas en el mes y facturas expedidas en el mes, con el fin de proporcionar un informe completo al contador para que realice su declaración correspondiente en tiempo y forma.

La presentación que tienen los documentos de venta, así como los reportes generados por el sistema es un beneficio más que añadir a los resultados obtenidos.

## **IX. CONCLUSIONES**

Después de analizar los datos de los tiempos obtenidos cuando se hace de forma manual y por el sistema, se determina que el sistema está cumpliendo con los objetivos planteados, que son primordialmente ahorrar tiempo en todos los procesos, y con ello se cumplió con el objetivo del trabajo planteado.

El personal que atiende la refaccionaria ha aceptado con agrado el sistema, debido a que facilita que trabajen con mayor rapidez y sin errores.

Este sistema puede ser implementado en cualquier refaccionaria sin temor a detectar muchos errores ya que el sistema actualmente está instalado en la refaccionaria "Adonay".

Un factor importante es que el contador tiene información de primera mano y libre de errores para hacer los movimientos ante la secretaría de hacienda y crédito público.

## X. Bibliografía

- [1] "The unified modeling language user guide", Grady Booch, James Rumbaugh, Ivar Jacobson. ed. Addison Wesley. 2a edición. USA 2005 pág. 16
- [2] El proceso unificado de desarrollo de software. Ivar Jacobson, Grady Booch, James Rumbaugh. ed. Addison Wesley. 1a edición. 2000 pág. 4
- [3] UML para programadores java. Rober C. Martin.ed. pearson prentice hall. 1a edicion 2004. pag 2.
- [4] Analisis de sistemas diseño y metodos. Jeffrey I. Whitten, lonnie d. bentley. mc grawhill.septima edicion 2008. pp. 186,187,189
- [5] UML, gota a got.martin fowler con Kendall Scott.addison wesley longman de mexico s.a de c.v. mexico, 1999.pp.10, 49,55
- [6] Fundamentos y modelos de bases de datos. Adoración de Miguel Castaño. Mario g. paittini velthuis. ed. alfaomega. 2ª edicion. 1999. pp.8, 10.
- [7] Analisis y diseño de sistemas de información. James A. Senm. ed.mc graw hill. 2a. edicion pág. 23 (qa76.9 s88/s44 1992)
- [8] Analisis y diseño de sistemas. Kennet e. Kendall. Jullie e. Kendall. pearson educacion.tercera edicion mexico, 1997.pp.2
- [9] Sistemas de informacion gerencial-Administracion de la empresa digital. Laudon Jane y kenneth. pearson educacion-prentice hall 2006.
- [10] Fundamentos de las Bases de Datos. Henry F. Korth, Abraham Siberschatz .2da Edición McGraw – Hill/Interamericana de España 1993
- [11] Fundamentos y Modelos de Bases de Datos. Piattini Miguel Mario. 2da Edicion Alfaomega Grupo Editor. México. septiembre 1999.
- [12] Programación de bases de datos con visual basic. net. Carsten Thomsen. ed. inforbooks. 1a. edición. 2002. pp. 29, 33.
- [13] Curso de programación de Visual Basic 6. Ceballos Sierra Fco. Javier. Alfaomega Grupo Editor. México DF, 2000.
- [14] MySQL para Windows y Linux. Pérez Cesar (2008). Segunda edición Alfaomega Grupo Editor. México. marzo 2008.
- [15] Administracion de bases de datos. Diseño y desarrollo de aplicaciones. Michael V. Mannino. ed. mc graw hill. 3ª edición. 2007. pp.4,7.

- [16] Fundamentos de diseño de bases de datos. Abraham Silberschatz. Henry f. korth. S.Dudarsham. mc grawhill. quinta edicion 2007
- [17] Sistemas de bases de datos diseño implementacion y administracion. Peter Rob. Carlos Coronel. thomson. quinta edicion 2004. p 124.
- [18] Introducción a los Sistemas de Bases de Datos. Date C. J. 7a edición. Pearson Prentice Hall, México 2001.
- [19] Sql server 7. Iniciacion y referencia. Jose Antonio Ramalho. mc graw-hill. 1ª. edición. 1999. pag. 5
- [20] SQL: a beginner's guide, 3ra edición ilustrada. Opper Andy, Sheldon Robert (2008). McGraw Hill Profesional.
- [21] Curso tutorial mysql <http://dev.mysql.com/doc/refman/5.0/es/tutorial.html>
- [22] SQL. León. Tata McGraw-Hill Education 1999.
- [23] PHP y MySQL Domine el desarrollo de un sitio Web dinámico e interactivo. Heurtel Olivier (2009). Ediciones ENI
- [24] SQL Server: programación y administración. González Alfons (1999). Alfaomega ra-ma.

## X. ANEXOS

### ANEXO I

Formato de la primera entrevista realizada para investigación de requerimientos		
Entrevistado: Rubén Almaraz, Administrador y dueño del negocio		
Fecha: 19 de marzo de 2011		
Hora: 1:30 p.m.		
Lugar: Refaccionaria Adonay, Administración		
Tema: Proceso de venta de refacciones		
Tiempo	Pregunta u Objetivo	Respuesta del entrevistado
2 min	Objetivo presentación	
5 min	1. ¿Cómo se lleva a cabo el proceso de venta de una o más refacciones?	
8 min	2. ¿Quién hace la venta, quien cobra, quien entrega? ¿Cómo se lleva a cabo cada proceso?	
2 min	3. ¿Qué tipo de comprobante se entrega al cliente?	
3 min	4. ¿Qué rango de tiempo es válido para poder canjear una remisión por una factura?	
8 min	5. ¿De qué personas requiere guardar información? ¿Qué información?	
3 min	6. ¿Cuál es el proceso que considera más tardado el realizar una venta?	
8 min	7. ¿Cómo guarda la información acerca de las ventas diarias? ¿Cómo se llevan a cabo los informes de ventas y cada cuando se hace?	
2 min	8. ¿Ha intentado utilizar un sistema en base de computadora dentro del negocio? ¿Qué sistema y Cuál ha sido su experiencia?	
1 min	9. ¿por qué considera que un sistema de información a la medida facilitaría sus procesos de negocio?	
42 minutos	Tiempo total de la entrevista	

## ANEXO II

<b>Formato de la Segunda entrevista realizada para investigación de requerimientos</b>		
Entrevistado: Rubén Almaraz, Administrador y dueño del negocio		
Fecha: 29 de marzo de 2011		
Hora: 1:30 p.m.		
Lugar: Refaccionaria Adonay, Administración		
Tema: Control de inventario		
Tiempo	Pregunta u Objetivo	Respuesta del entrevistado
2 min	Objetivo presentación	
5 min	1. ¿Cómo se lleva a cabo el control de inventario actualmente?	
5 min	2. una vez que la mercancía a añadir al inventario llega a su negocio ¿Cuál es el proceso a seguir para control de entradas al inventario?	
2 min	4. ¿En qué momento sabe que debe resurtir producto?	
2 min	5. ¿Existen documentos de compra de los cuales debe guardar información? ¿Qué información?	
2 min	6. ¿Maneja crédito de compra? ¿Cómo lo controla?	
1 min	7. ¿Maneja crédito de venta? Si es así ¿Cómo lo controla?	
2 min	8. ¿Al realizar una venta, que información debe almacenarse?	
2 min	9. ¿Cómo se llevan a cabo los cambios y devoluciones?	
3 min	10. ¿Qué información se requiere acerca de cada artículo contenido en el inventario?	
26 min	Tiempo total de la entrevista	