



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE ESTUDIOS SUPERIORES  
ARAGÓN

“ANÁLISIS, DISEÑO Y DESARROLLO DE UNA PROPUESTA  
DE BASE DE DATOS UNIFICADA PARA LOS CENTROS DE  
VERIFICACIÓN VEHICULAR DEL VALLE DE MÉXICO”

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:  
**INGENIEROS EN COMPUTACIÓN**

**P R E S E N T A N:**

**DÍAZ JÁCOME ARTURO**  
**MORALES DÍAZ RUBÉN EFRAÍN**

ASESOR:

M. EN C. MARCELO PÉREZ MEDEL





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## DEDICATORIAS

*A mi madre, que gracias a cada sacrificio que realizo hasta el día de hoy, me dio la fuerza y determinación necesaria para lograr lo que me propuse, espero que la vida me otorgue el tiempo suficiente para recompensarla.*

*Arturo Díaz Jácome*

*A todas esas cosas que susurran y se mueven en la oscuridad, pero sobre todo a esos futuros regalos divinos de la vida, aun no llegas y ya tengo planes para ti TARA ALI.*

*Rubén Efraín Morales Díaz*

## INDICE

INTRODUCCIÓN.....	1
JUSTIFICACIÓN DEL TEMA.....	2
PLANTEAMIENTO DEL PROBLEMA.....	4
OBJETIVOS.....	5
HIPÓTESIS.....	6
ALCANCE.....	7
METODOLOGIA.....	8
CRONOGRAMA.....	9
<b>CAPITULO 1. “ANTECEDENTES DE LA VERIFICACIÓN VEHICULAR Y LAS BASES DE DATOS”.....</b>	<b>10</b>
1.1. Historia de la Verificación Vehicular en México.....	10
1.2. Inicios e historia de los Centros de Verificación.....	18
1.3. Tecnología y proceso de verificación.....	18
1.4. Normas establecidas del programa de verificación vehicular y aspectos más importantes.....	20
1.5. Historia del uso de las Bases de Datos, tipos y usos más frecuentes.....	25
1.5.1. Tipos de Bases de Datos.....	27
1.5.1.1. Bases de Datos según su función.....	28
1.5.1.2. Bases de Datos según Modelos Administrativos.....	29
1.6. Ventajas e Inconvenientes de las Bases de Datos.....	31
1.6.1. Ventajas por Integración de Datos.....	31
1.6.2. Ventajas por existencia del SGBD.....	32
1.6.3. Inconvenientes de los sistemas de Bases de Datos.....	34
1.7. Estado Actual.....	35
1.8. Tendencias para el Futuro.....	37

1.9. Normas de estandarización de una Base de Datos.....	37
--	----

**CAPÍTULO 2. “ENTENDIMIENTO DE LAS NECESIDADES DEL MANEJO DE INFORMACIÓN EN LA VERIFICACIÓN VEHICULAR”.....39**

2.1. Configuración General.....	42
2.1.1. Diseño modular.....	43
2.1.2. Alta Capacidad de Prueba.....	44
2.1.3. Resistencia a la Alteración.....	44
2.1.4. Diseño del Equipo.....	45
2.1.5. Documentación del software.....	45
2.2. Especificaciones de la base de datos.....	45
2.2.1. Datos del Centro y de la Línea de Verificación.....	46
2.3. Directorios y Archivos.....	46
2.3.1. Localización de los Archivos.....	46
2.3.2. Acceso a las Tablas.....	46
2.3.3. Limpieza de las Tablas.....	47
2.3.4. Tablas Maestra y Trabajo.....	48
2.3.5. Estructuras y contenidos de las tablas.....	48
2.4. Homologación de equipos.....	51

<b>CAPÍTULO 3. “DISEÑO DE LA BASE DE DATOS UNIFICADA”</b> .....	52
3.1    Abstracción de las Reglas de Negocio.....	52
3.2    Establecimiento de Estándares.....	54
3.3    Niveles de Estudio de la Base de Datos.....	57
3.4    Visualización de Entidades.....	58
3.5    Modelo Entidad-Relación.....	60
3.5.1    Restricciones en el Modelo Entidad-Relación.....	61
3.5.2    Integridades.....	62
3.5.3    Cardinalidad.....	63
3.6    Normalización.....	64
3.7    Selección de la Plataforma Tecnológica y Metodología.....	66
3.8    Planteamiento de Seguridad.....	70
3.9    Requerimientos mínimos de Hardware.....	70
3.10    Requerimientos de Software.....	71
3.11    Características del motor de base de datos compatibles con las ediciones de SQL Server 2000.....	73
 <b>CAPÍTULO 4. “DESARROLLO”</b> .....	 75
4.1    Creación de Tablas.....	75
4.2    Creación de Integridades.....	79
4.2.1    Integridad por Entidad.....	79
4.2.2    Integridad por Domino.....	79
4.2.3    Integridad por Referencia.....	81
4.3    Creación de Vistas.....	82
4.4    Creación de Respaldos y Recuperación de la Base de Datos con Procedimientos Almacenados.....	83
4.4.1    Procedimiento Almacenado que genera un respaldo completo de la Base de Datos.....	83
4.5    Creación de Disparadores.....	84
4.6    Conversión de Datos.....	85
4.6.1    Importación.....	85
4.6.2    Exportación.....	85
4.7    Código fuente.....	86

<b>CAPÍTULO 5. “ANÁLISIS DE LAS VENTAJAS E INCONVENIENTES DE LA PROPUESTA”</b> .....	104
5.1    Ventajas de la Base de Datos.....	104
5.2    Inconvenientes de la Base de Datos.....	105
<b>CONCLUSIONES</b> .....	107
<b>ANEXOS</b> .....	108
<b>BIBLIOGRAFÍA</b> .....	111

## INTRODUCCIÓN

La Secretaría de Ecología y en específico la Dirección de Control Ambiental dicta que en el Valle de México cada automóvil está obligado a acreditar una verificación vehicular semestral que garantice el buen estado y funcionamiento de su maquinaria previniendo el exceso de emisiones contaminantes que dañan al medio ambiente y a la atmósfera.

Para satisfacer la necesidad de verificar cada vehículo se ha creado un Programa de Verificación Vehicular el cual consiste en realizar una prueba física en los Centros de Verificación Vehicular los cuales están equipados con maquinaria especializada en software y hardware que realizan las mediciones necesarias para garantizar que cada automóvil cumple con los límites de emisiones de gases especificados.

Actualmente el Valle de México cuenta con un parque vehicular de gran magnitud el cual produce el 25% de la contaminación del ambiente, así como hay muchos automóviles de reciente modelo que no genera demasiadas emisiones contaminantes también existen una gran cantidad de vehículos con más de 7 años de antigüedad que generan grandes cantidades de emisiones contaminantes.

La mezcla que compone esta llamada “emisión contaminante” se compone de cinco gases, los cuales son Hidrocarburos (HC), Monóxido de Carbono (CO), Dióxido de Carbono (CO<sub>2</sub>), Oxígeno (O<sub>2</sub>) y Óxido de Nitrógeno (NO<sub>x</sub>), este último es el gas más nocivo para el ambiente, en especial daña la atmósfera.



## JUSTIFICACIÓN DEL TEMA

### **Situación Actual**

El Programa de Verificación Vehicular es un esfuerzo de varias Instancias del Gobierno para tener un control de la contaminación por emisiones de gases emitidas por cada vehículo realizando un esfuerzo por mantener la Ecología Ambiental, lamentablemente este esfuerzo no ha sido suficiente ya que el proceso de verificación vehicular no es confiable hasta la fecha, debido a la alteración en los procesos de verificación y en la información que se maneja, ya que hasta ahora no se cuenta con una Base de Datos confiable e integral que proporcione la certeza necesaria para tomar otro tipo de medidas que reflejen los resultados deseados por lo cuales fue creado este Programa Ambiental.

La falta de Organización de la información para el diseño de la Base de Datos que controle la información sobre las emisiones contaminantes de los vehículos ha dado como resultado que las empresas proveedoras de los sistemas informáticos de verificación generen prototipos por separado causando una falta de homologación hablando en términos de estructuras en las Bases de Datos y de las técnicas para su uso, facilitando a los operadores la manipulación de los datos a conveniencia.

Otra consecuencia aparece en la poca confiabilidad de los resultados que arroja esta información, ya que los sistemas informáticos también no están de alguna forma del todo normalizados y con esto se da la facilidad de alteración de dicha información. Además no se cuenta con sistemas confiables de respaldos, para ser utilizados en casos de pérdida de información por causas accidentales, ajenas al sistema o deliberadas.

### **Situación Deseada**

Tener una Base de Datos confiable, robusta y extensible que dé la suficiente confiabilidad de la información, cuidando la integridad en su estructura para que no se produzcan o dificultar la manipulación de su contenido. Además de contar con un modelo el cual esté homologado para todos y cada uno de los proveedores de los sistemas informáticos, evitando la desorganización, esfuerzos y técnicas separadas para el almacenamiento y manejo de la información. También contando con un plan de respaldos programados que permitan el reporte puntual de la información confiable a las autoridades correspondientes, además de tener la posibilidad de restablecer la Base de Datos en caso de contingencias no previstas.

## PLANTEAMIENTO DEL PROBLEMA

Existen varios proveedores para los sistemas de verificación vehicular en la zona del Valle de México, lo cual causa que no exista un sistema único y centralizado para todos los centros de verificación vehicular, esto permite que personas autorizadas y no autorizadas modifiquen el sistema y la base de datos sin que se lleve control alguno, esto implica que en la práctica corran diferentes aplicaciones en diversos centros de verificación vehicular que inclusive tienen el mismo proveedor de software. Por consecuencia no se tiene una seguridad en la integridad de la información que estos centros vehiculares emiten.

En resume el Problema es *“No se cuenta con un sistema confiable y único que controle todas las operaciones de verificación vehicular del Valle de México y que sea robusto y confiable presentando facilidades para el monitoreo y rastreo de operaciones normales e inusuales sobre la base de datos.”*

Con todo esto se ha venido abajo la credibilidad del programa de Verificación y en ocasiones se a dudado completamente que se haya creado en verdad para la reducción de la contaminación ambiental.

## **OBJETIVOS**

### **GENERAL**

Analizar, Diseñar y Desarrollar una Propuesta de Base de Datos Unificada para los Centro de Verificación Vehicular del Valle de México, que cuente con una mejor funcionalidad, mayor control, flexibilidad y adaptabilidad a las nuevas necesidades de los programas vehiculares, logrando con esto incorporar las últimas tecnologías y un máximo desempeño.

### **PARTICULARES**

1. Lograr la exactitud de una Base de Datos Unificada que realice las tareas precisas tal como las define su especificación.
2. Contar con la robustez necesaria para reaccionar apropiadamente a las condiciones anormales.
3. Permitir la facilidad de la adaptación de la Base de Datos a los cambios de especificación.
4. Facilitar la compatibilidad de la Base de Datos con los elementos existentes y futuros con los que cuenta y contará el Programa Vehicular de Verificación.
5. Tener una Base de Datos eficiente que demande la cantidad mínima de recursos al sistema operativo.
6. Demandar los recursos mínimos de hardware tales como tiempo de procesador, espacio ocupado en memorias internas o externas, ancho de banda usado en dispositivos de comunicación.

7. Crear una Base de Datos verificable e integral con procedimientos de seguridad, para detectar acciones ajenas al sistema tales como accesos o modificaciones no autorizadas.
8. Realizar procedimientos para corregir las contingencias producidas por las condiciones ajenas al sistema y a los usuarios.

## **HIPÓTESIS**

Con el desarrollo de la propuesta de la base de datos se establece la base para lograr un software único, confiable y auditable que cumpla con las necesidades que los diferentes programas de verificación vehicular actuales no cumplen. Para lograr este objetivo totalmente se espera que al terminar la base de datos venga el esfuerzo de otro grupo para implantar la aplicación que debe de conectarse con esta base de datos y cerrar el ciclo completo para el proceso de verificación vehicular.

Es deseable que la única instancia que guarde y almacene la información sobre este proyecto fuera la Secretaria de Ecología, la cual también se encargaría de las diferentes modificaciones y de la distribución del programa y la instalación de la Base de Datos para lo Centros de Verificación, evitando que terceros tenga la posibilidad de realizar modificaciones a su beneficio.

## **ALCANCE**

Obtener la información requerida sobre los Programas de Verificación Vehicular del Valle de México, su historia, estado actual y posibles tendencias futuras, así como recopilar los fundamentos, metodologías y tecnologías sobre las Bases de Datos que se puedan aplicar a este proyecto, logrando así tomar una elección sobre el tipo de tecnología que sea más conveniente.

Analizar para poder integrar las reglas de la realidad que se quiere modelar en un concepto denominado reglas de negocio, estas reglas serán precisamente el código del comportamiento del discurso que debemos de modelar y que deben de estar implícitas en este análisis y en el diseño de la Base de Datos.

Partiendo del análisis se revisarán las diferentes alternativas de solución para diseñar un modelo de una Base de Datos Unificada que será propuesta para la posible implementación en los Programas de Verificación Vehicular.

Desarrollar el modelo resultante en la tecnología elegida partiendo del universo modelado obtenido de las reglas de negocio.

Documentar cada etapa de la elaboración de la Base de Datos para guardar registro de las decisiones tomadas y para tener en control el proceso antes mencionado.

## METODOLOGÍA

La base en la cual está sustentado este trabajo es la investigación documental y de campo, mediante la recopilación y la extracción de teorías sobre las que se sostienen las bases de datos, la visita y la revisión de los procesos de verificación en los Centros de Verificación Automotriz en su trabajo diario.

Parte importante del extracto de esta tesis fue obtenido por medio de la observación de los procesos de los Centros de Verificación, al tener que recavar todo lo posible de la experiencia de algunas personas para poder documentarla y presentarla en forma desmenuzada.

Otra forma de hacer la investigación de campo en este trabajo será lograr la conceptualización y formulación de las reglas de negocio que deberán definir los procesos en la Verificación Vehicular metodológicamente.

En general nuestro papel en este caso será el de recopilar, analizar, diseñar, desarrollar y criticar nuestra propia propuesta de Base de Datos.

Las fuentes primarias donde se obtendrá la información serán algunos libros y manuales, lo que cabe resaltar es que referente al tema de la Verificación Vehicular mucha información se encuentra restringida por las autoridades con respecto a los procesos informáticos, por lo cual hemos decidido realizar visitas a los Centros de Verificación Vehicular para observar y documentar cada etapa y los procesos, que toman parte en el ciclo de la verificación de un automóvil.

## CRONOGRAMA

<b>TABLA DE ACTIVIDADES DEL PROYECTO</b>					
Id	Nombre	Duración	Comienzo	Fin	Predecesoras
1	Elección del Tema	6.d	20 de Febrero de 2005	6 de Marzo de 2005	
2	Recopilación de Información sobre el tema elegido	5.333d	6 de Marzo de 2005	20 de Marzo de 2005	1
3	Revisión del Material Recopilado	2.667d	20 de Marzo de 2005	27 de Marzo de 2005	2
4	Planteamiento del Problema	1.333d	27 de Marzo de 2005	2 de Abril de 2005	3
5	Elaboración de la Propuesta	3.d	2 de Abril de 2005	9 de Abril de 2005	4
6	Creación de la hipótesis	2.667d	9 de Abril de 2005	16 de Abril de 2005	5
7	Obtención de Objetivos	1.333d	9 de Abril de 2005	10 de Abril de 2005	5
8	Definición del Alcance	1.333d	10 de Abril de 2005	16 de Abril de 2005	7
9	Justificación del Tema	1.333d	16 de Abril de 2005	17 de Abril de 2005	8
10	Revisión de Propuesta con el ASESOR	1.333d	10 de Abril de 2005	16 de Abril de 2005	7
11	Búsqueda de los antecedentes de la Verificación Vehicular en México	10.d	16 de Abril de 2005	14 de Mayo de 2005	10
12	Recopilación de las teorías sobre Bases de Datos	10.d	14 de Mayo de 2005	5 de Junio de 2005	11
13	Edición del del Primer Capítulo	5.d	5 de Junio de 2005	19 de Junio de 2005	12
14	Revisión de avances Primer Capítulo con el ASESOR	2.667d	19 de Junio de 2005	26 de Junio de 2005	13
15	Análisis y abstracción de la información que torneara la BD	6.d	5 de Junio de 2005	25 de Junio de 2005	12
16	Edición del Segundo Capítulo	5.d	26 de Junio de 2005	3 de Julio de 2005	15
17	Revisión de avances Segundo Capítulo con el ASESOR	2.667d	9 de Julio de 2005	10 de Julio de 2005	16
18	Ajustes y corrección a Capítulo 1 y 2	4.d	16 de Julio de 2005	23 de Julio de 2005	17
19	Elaboración del prototipo de la BD	16.d	24 de Julio de 2005	3 de Septiembre de 2005	18
20	Elección de la Plataforma Tecnológica	1.333d	4 de Septiembre de 2005	4 de Septiembre de 2005	19
21	Revisión del Diseño de futura BD	5.d	10 de Septiembre de 2005	18 de Septiembre de 2005	20
22	Revisión de avances Tercer Capítulo con el ASESOR	4.d	18 de Septiembre de 2005	1 de Octubre de 2005	21
23	Desarrollo del Código	13.d	18 de Septiembre de 2005	23 de Octubre de 2005	21
24	Pruebas	4.d	23 de Octubre de 2005	5 de Noviembre de 2005	23
25	Correcciones	6.d	5 de Noviembre de 2005	19 de Noviembre de 2005	24
26	Revisión y creación de insert's para prueba integrales	8.d	20 de Noviembre de 2005	10 de Diciembre de 2005	25
27	Pruebas Integrales	4.d	11 de Diciembre de 2005	18 de Diciembre de 2005	26
28	Presentación de la BD al Asesor	2.667d	24 de Diciembre de 2005	25 de Diciembre de 2005	27
29	Correcciones al diseño por omisiones	4.d	31 de Diciembre de 2005	7 de Enero de 2006	28
30	Correcciones al Código	4.d	8 de Enero de 2006	15 de Enero de 2006	29
31	Pruebas Integrales 2.	2.667d	21 de Enero de 2006	22 de Enero de 2006	30
32	Redacción del Cuarto Capítulo	4.d	28 de Enero de 2006	4 de Febrero de 2006	31
33	Quinto Capítulo (análisis de las Ventajas e Inconvenientes de nuestra propues	5.d	5 de Febrero de 2006	18 de Febrero de 2006	32
34	Revisión del cuarto y quinto Capítulo	4.d	18 de Febrero de 2006	26 de Febrero de 2006	33



## **CAPÍTULO 1.**

### **ANTECEDENTES DE LA VERIFICACIÓN VEHICULAR Y LAS BASES DE DATOS**

En este capítulo examinaremos los antecedentes de los programas vehiculares desde sus inicios, desde la etapa en el cual se encuentran ya aunados a la tecnología de computo, así también recopilaremos información general sobre las Bases de Datos.

#### **1.1 HISTORIA DE LA VERIFICACIÓN VEHICULAR EN MÉXICO**

Desde los años setenta en la Ciudad de México se han llevado a cabo esfuerzos para entender y confrontar los problemas de calidad del aire. Desde aquella década se instaló una rudimentaria red de monitoreo que permitió las primeras mediciones objetivas de concentraciones de algunos contaminantes<sup>1</sup> sin embargo, este empeño de verificación y registro de las cantidades de contaminantes no estuvo vinculado aún a procesos de política pública.

Este proyecto no logro operar a través de períodos suficientemente continuos y sobre bases adecuadas de credibilidad; su vida fue corta debido a problemas de mantenimiento y a la inexistencia de un compromiso permanente por parte de las instancias gubernamentales a cargo del tema durante aquellos años y la falta de un interés significativo por parte de la opinión pública. Este fracaso puede ser explicado también por limitaciones presupuestarias y por la ausencia de capacidades técnicas adecuadas y, desde luego por la baja jerarquía y escasa influencia política del organismo gubernamental que durante la década

---

<sup>1</sup> Fuente: [http://omega.ilce.edu.mx:3000/sites/ciencia/volumen3/ciencia3/159/htm/sec\\_8.htm](http://omega.ilce.edu.mx:3000/sites/ciencia/volumen3/ciencia3/159/htm/sec_8.htm)

de los setenta tuvo a su cargo una política ambiental todavía inexperta: la Subsecretaría del Mejoramiento del Ambiente dependiente de la Secretaría de Salubridad y Asistencia.

Sin embargo, esta experiencia inicial arrojó algunos saldos valiosos, en particular una alianza cada vez más cercana entre las entidades de gobierno y las universidades, lo que permitió un involucramiento creciente de investigadores y académicos, quienes más adelante facilitarían los progresos realizados.

Al inicio de los ochenta, la creación de la Secretaría de Desarrollo Urbano y Ecología (SEDUE) impulsó de manera definitiva las capacidades de manejo de la calidad del aire. Un número importante de profesores universitarios fueron reclutados por la nueva instancia gubernamental, quienes en coordinación con consultores y funcionarios internacionales fueron capaces de instalar y operar la primera Red Automática de Monitoreo Atmosférico en la Ciudad de México, de proponer los primeros estándares en materia de calidad del aire y un correspondiente sistema de reporte al público.

A mediados de la década de los ochenta la Ciudad de México fue testigo de problemas graves de contaminación del aire, al punto que al comienzo de 1986 se declararon las primeras contingencias. Las concentraciones de ozono alcanzaron niveles de entre 0.4 y 0.5 partes por millón (más de 400 IMECAs<sup>2</sup>), mientras que las partículas suspendidas totales (PST) llegaron a niveles extremos, probablemente cerca de 1,000 microgramos por metro cúbico; peor aun, esto se dio en combinación con elevadas concentraciones de óxidos de azufre.

La emergencia tuvo el efecto inmediato de movilizar a la opinión pública; por primera vez en la historia de nuestro país grupos de ecologistas organizados fueron capaces de convocar a importantes manifestaciones demandando una respuesta por parte de las autoridades.

---

<sup>2</sup> Índice metropolitano de calidad del aire. Fuente : [http://www.ni.gob.mx/?P=med\\_amb\\_mej\\_amb\\_sima\\_imeca](http://www.ni.gob.mx/?P=med_amb_mej_amb_sima_imeca)

A mediados de los 80's el gobierno reaccionó con una serie de acciones que incluían la introducción de gas natural en las plantas termoeléctricas y en algunas industrias, y ciertos compromisos para mejorar los combustibles automotrices en la Ciudad de México. Sin embargo, la magnitud del problema excedía claramente el alcance de estas iniciativas propuestas.

El plomo se convirtió en uno de los problemas ambientales más sobresalientes de la ciudad. Un amplio proceso de investigación médica y académica documentó los grandes riesgos a la salud pública relacionados con altas concentraciones de plomo en la sangre y el consecuente daño neurológico.

Petróleos Mexicanos inició esfuerzos muy serios para reformular las gasolinas, eliminando gradualmente el contenido de plomo (en forma de tetraetilo, utilizado como antidetonante), sustituyéndolo por compuestos oxigenados, aromáticos y otros hidrocarburos que de acuerdo a algunas opiniones, incitaron una mayor reactividad fotoquímica en la atmósfera, algo que pudo haberse traducido en concentraciones más elevadas de ozono.

Hacia el final de los ochenta creció la sensibilidad y conciencia del público frente al problema, lo que impulsó la aplicación de nuevas acciones. Un grupo importante de organismos no gubernamentales ecologistas lanzaron la iniciativa voluntaria "*Un Día sin Auto*", la cual fue apoyada por las autoridades de la ciudad, para evolucionar años más adelante hasta constituirse en una restricción obligatoria a la circulación vehicular: "*Hoy No Circula*". En paralelo se estableció por primera vez un programa piloto de inspección y mantenimiento (*verificación para los vehículos privados*).

También las autoridades ambientales federales construyeron el primer inventario oficial de emisiones para la Ciudad de México. Este mostró claramente, que el ozono es resultado básicamente de las emisiones vehiculares, a pesar de una sensación ampliamente difundida en la población que ha atribuido el problema a fuentes industriales.

La calidad del aire se consolidó como un tema importante en la opinión pública, al grado que durante la toma de posesión del nuevo Presidente de la República en diciembre del 1988, éste destacó la importancia del problema y responsabilizó al nuevo Regente de la Ciudad de México de encontrar y aplicar soluciones viables. En 1990 fue integrado y lanzado el primer plan sistemático para combatir la contaminación atmosférica de la Ciudad de México (Programa Integral Contra la Contaminación Atmosférica, PICCA), el cual involucró a todos los actores institucionales relevantes al problema, como el Gobierno del Distrito Federal, el Gobierno del Estado de México, Petróleos Mexicanos, Comisión Nacional de Electricidad, la *Secretaría de Desarrollo Urbano y Ecología* y la *Secretaría de Salud*<sup>3</sup>.

El plan tuvo el apoyo de agencias internacionales como la EPA (*Environmental Protection Agency, Estados Unidos*), de JICA (*Japan International Cooperation Agency*), TÜV (*Entidad de Control Técnico de Alemania*) y del Banco Mundial. Fue sustentado en las medidas identificadas previamente por las autoridades mexicanas durante las crisis ambientales de los ochenta, y desde luego, en las experiencias norteamericanas. Sus principales componentes incluían la introducción de gasolina sin plomo y diesel de bajo azufre, compuestos oxigenados en las gasolinas, controles en las emisiones de óxidos de azufre, control de emisiones evaporativas en el almacenamiento de combustibles, restricciones a la circulación para todos los vehículos, uso de gas LP en vehículos comerciales, instalación obligatoria de convertidores catalíticos en todos los autos nuevos, uso de gas natural en plantas termoeléctricas en sustitución del combustible, y, verificación obligatoria de todos los vehículos registrados en la ciudad.

También se incluyeron acciones de reforestación y transporte público, que tuvieron alcances muy limitados. En lo que respecta al transporte, fueron sustituidas la mayor parte de las combis por nuevos microbuses, los cuales

---

<sup>3</sup> Sitios de Interés: <http://saladeprensa.cfe.gob.mx/noticia/index.alia?PHPSESSID=d560e5ecd4530eb83674e7a60e29ea21>  
<http://www.seduvi.df.gob.mx/>  
<http://www.salud.gob.mx/>

aunque eran de tecnología improvisada a partir de un chasis de carga, estaban equipados con convertidor catalítico. Al no resolverse los problemas institucionales y políticos sobre las que se desarrolló el sistema de transporte urbano, éste llegó a constituirse en el principal reto a vencer hacia finales de los noventa.

Es de destacarse también en el PICCA un Programa de Contingencias para combatir episodios extremos de concentración de ozono, el cual incluía restricciones al tránsito vehicular y limitaciones a la actividad industrial. Por medio del PICCA fue posible lograr la convergencia de responsabilidades institucionales en una instancia coordinadora que adoptó la forma de Comisión Metropolitana, la cual fue creada por decreto presidencial.

Esta Comisión fue la iniciativa de un nuevo arreglo institucional que evolucionaría hasta incorporar a otras entidades gubernamentales e incluso a instituciones académicas y organismos no gubernamentales y representantes del sector privado, en un Consejo Consultivo. En la mitad de los años noventa esta Comisión modificó su estatuto legal transformándose en un acuerdo de coordinación entre el Distrito Federal y el Estado de México con la participación de la nueva Secretaría de Medio Ambiente, Recursos Naturales y Pesca (SEMARNAP).

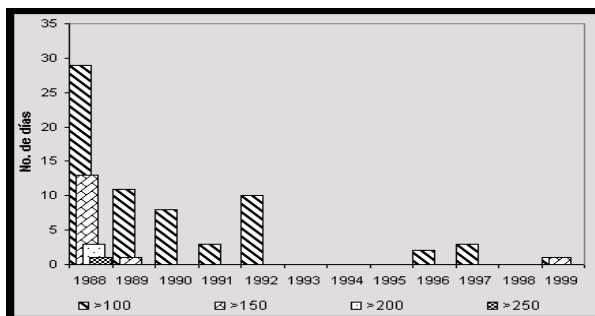
Ya para entonces otras ciudades de nuestro país habían experimentado problemas más o menos importantes en calidad del aire, como Guadalajara, Toluca y Monterrey, y a algunas ciudades fronterizas como Ciudad Juárez, que más adelante también serían objeto de atención gubernamental a través de nuevos instrumentos y programas. Todas las medidas adoptadas en la Ciudad de México en los primeros años de los noventa y por el PICCA, estuvieron rezagadas con respecto al crecimiento en el número de vehículos automotores y a la larga sobrevivencia de una flota vehicular vieja y tecnológicamente obsoleta. Ésta fue y ha sido responsable de una proporción muy grande de las emisiones totales que generan el ozono.

Entre 1991 y 1994 se presentaron los peores episodios de contaminación por ozono registrados en la Ciudad de México como podemos ver en la figura 1.1 y 1.2. Sin embargo, a partir de esos momentos, comenzaron a relajarse ciertas presiones al deterioro en la calidad del aire. El plomo dejó de representar una preocupación gracias a nuevas gasolinas reformuladas. Lo mismo sucedió con el bióxido de azufre, que fue limitado drásticamente al prohibirse en la ciudad el consumo de combustible con alto contenido de azufre, al mejorarse el diesel y al lograrse un mayor uso de gas natural en industrias y plantas termoeléctricas.

**Fig. 1.1**

**CIUDAD DE MÉXICO**

**Bióxido de Azufre: Número de días con lecturas IMECA superiores a 100, 150, 200 y 250 (1988-1999)**

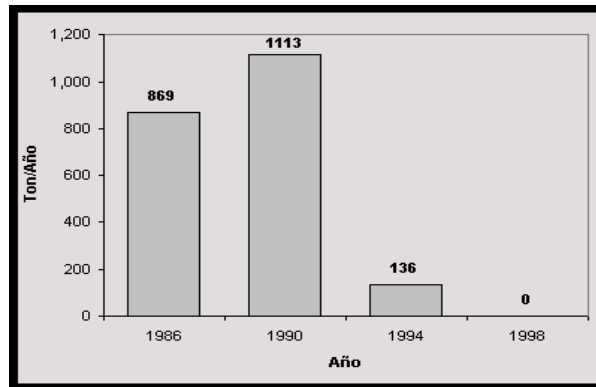


Fuente: Elaborado por CESPEDES con datos de la Red Automática de Monitoreo Atmosférico de la Ciudad de México.

**Fig. 1.2**

**CIUDAD DE MÉXICO**

**Plomo: Emisiones generadas por uso de gasolinas**



Fuente: Elaborado por el CESPEDES con datos de PEMEX.

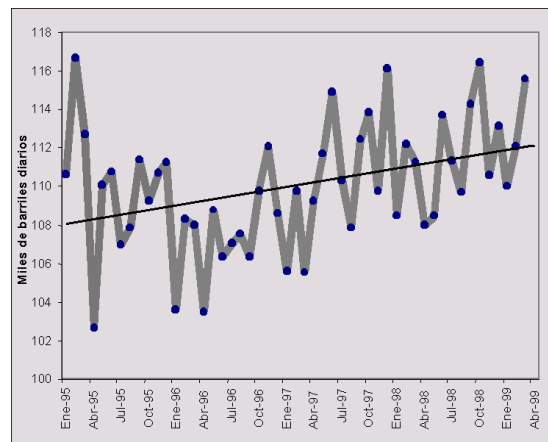
Las gasolinas reformuladas en combinación con los convertidores catalíticos y nuevas tecnologías vehiculares como podemos ver en la figura 1.3, fueron el producto de una normatividad cada vez más estricta, hicieron ceder al ozono, al menos en sus lapsos más extremos y en el promedio de sus concentraciones diarias.

Hasta esos momentos los problemas de calidad del aire fueron considerados casi exclusivamente como algo técnico, desvinculado de procesos urbanos como el transporte y los usos del suelo. Sin embargo, a partir de nuevas ideas surgidas en la Organización para la Cooperación y el Desarrollo Económico (OCDE) sobre la sustentabilidad urbana, se generaron nuevas perspectivas. En 1996 se diseñó y promulgó un nuevo plan: Programa para Mejorar la Calidad del Aire en el Valle de México en los años 1995-2000, el cual no sólo se reconoció como continuador de las acciones anteriores sino que intentó ofrecer un horizonte conceptual más amplio en el que se integraron políticas urbanas y de transporte con instrumentos de control de emisiones.

Por primera vez en este nuevo programa se plantearon objetivos en materia de calidad del aire en un tiempo definido, que se basaron en un conjunto de metas, estrategias e instrumentos orientados a la industria, a los vehículos, a la planeación del desarrollo urbano y al transporte público. Incluyó algunos fundamentos de análisis económico sobre la demanda de combustibles y la utilización de vehículos, así como la consideración de medidas en el precio de la gasolina y a incentivos fiscales, complementarios de los enfoques regulatorios tradicionales. Cabe destacar que se contempló de manera detallada en este plan al tema del gas LP y de su papel como inductor importante de los problemas de ozono. Este programa cayó en el olvido hacia finales de los noventa, al igual que el funcionamiento de la Comisión Ambiental Metropolitana y de su órgano consultivo.

**Fig. 1.3**

### **Consumo de gasolinas en la Ciudad de México**



Fuente: PEMEX



## **1.2 INICIOS E HISTORIA DE LOS CENTROS DE VERIFICACIÓN**

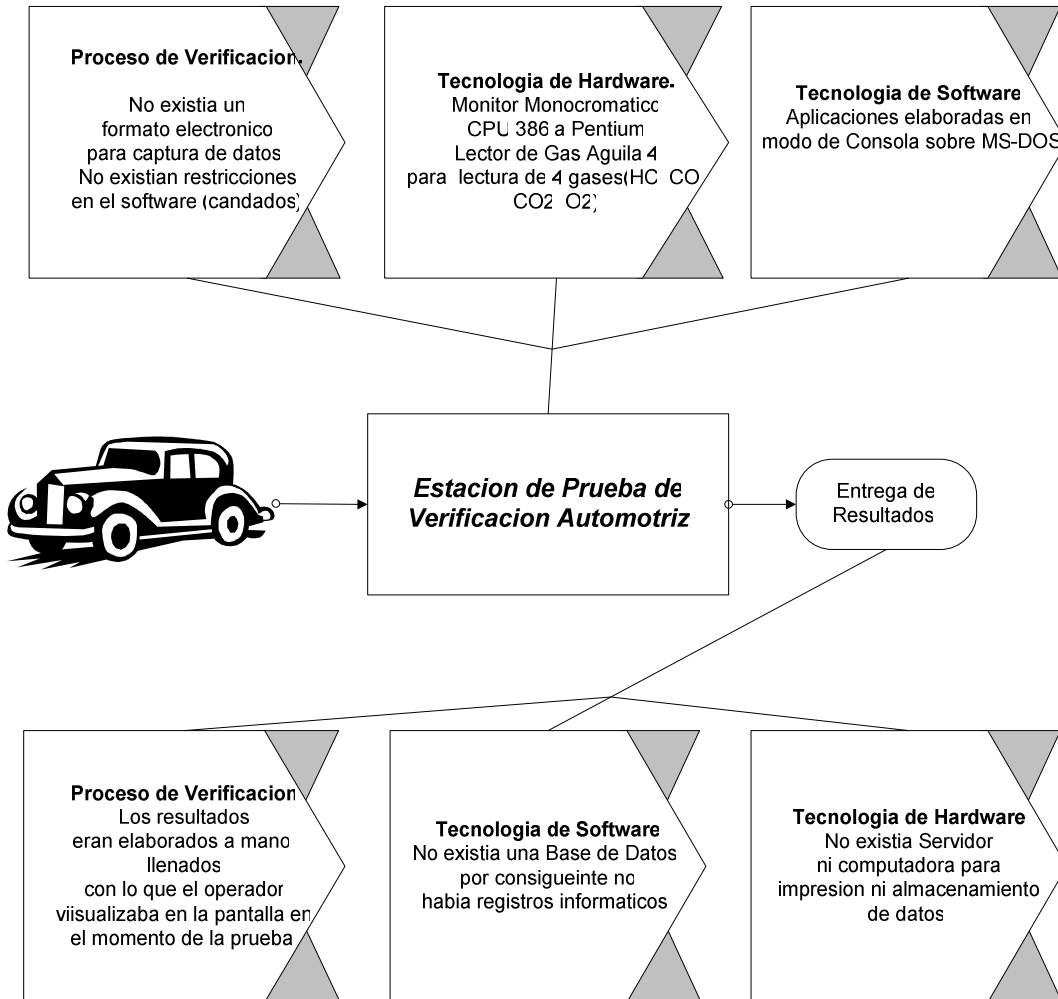
A finales de la década de los 80's se implementaron los primeros establecimientos de verificación vehicular en el Valle de México, los cuales únicamente contaban con un rudimentario mecanismo de chequeo de emisiones vehiculares analizando solamente cuatro de los cinco gases que mezclados son contaminantes para la atmósfera: Hidrocarburos, Bióxido y Monóxido de carbono y Oxígeno (HC, CO<sub>2</sub>, CO, O<sub>2</sub>), careciendo de lineamientos técnicos especializados, provocando la fácil manipulación a conveniencia de la información además de ser una medida no obligatoria.

Durante los años 90's se fueron implementando y mejorando los sistemas técnicos, legales y de logística de los programas de verificación vehicular, estableciendo la obligatoriedad a los usuarios de los vehículos en el Valle de México, logrando por fin a mediados de la década pasada la creación y el establecimiento de los actualmente conocidos como Centros de Verificación Vehicular, los cuales en la actualidad ya son totalmente especializados y cuentan con las últimas tecnologías de software y hardware para el monitoreo de las emisiones contaminantes del parque vehicular.

## **1.3 TECNOLOGÍA Y PROCESO DE VERIFICACIÓN**

En la figura 1.4 se muestra el diagrama de procesos de la verificación en sus inicios, además de la tecnología que se contaba hasta ese momento.

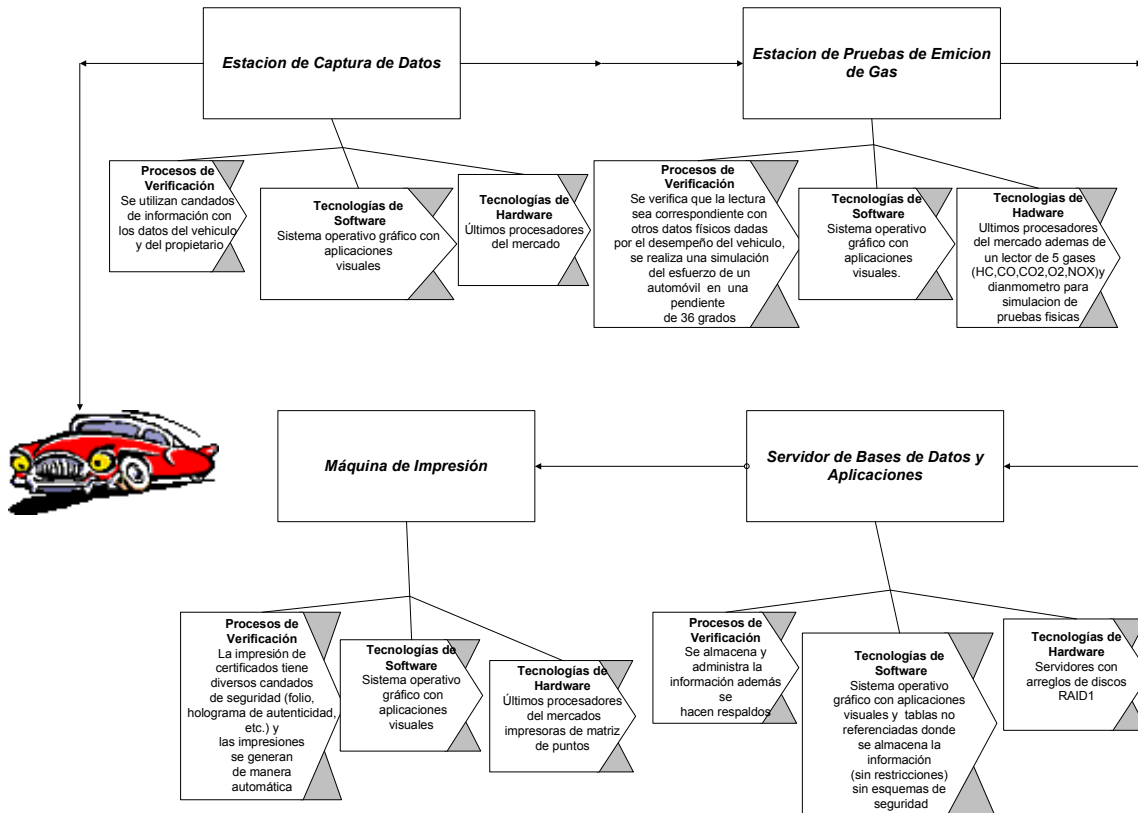
**Fig.1.4**  
**Proceso Original de los Verificentros**



En la figura 1.5 se muestra el diagrama de procesos de la verificación hoy en día, además de la tecnología con la que cuenta en el años 2005.

Fig. 1.5

**Proceso de Verificentros Actualmente**



**1.4. NORMAS ESTABLECIDAS DEL PROGRAMA DE VERIFICACIÓN VEHICULAR Y ASPECTOS MÁS IMPORTANTES.**

Los Programas de Verificación Vehicular están basados en determinadas normas establecidas por la SECRETARIA DE MEDIO AMBIENTE, RECURSOS NATURALES Y PESCA (SEMARNAP), a continuación se nombran algunas de las más importantes:

### ***NORMA OFICIAL MEXICANA NOM-041-ECOL-1999***

Establece los límites máximos permisibles de emisión de hidrocarburos, monóxido de carbono, oxígeno; nivel mínimo y máximo de dilución, medición de óxidos de nitrógeno, y es de naturaleza obligatoria para los responsables de los vehículos automotores que circulan en el país, que usan gasolina como combustible, así como para los responsables de los centros de verificación autorizados, a excepción de vehículos con peso bruto vehicular menor de 400 kilogramos, motocicletas, tractores agrícolas, maquinaria dedicada a las industrias de la construcción y minera.

### ***NORMA OFICIAL MEXICANA NOM-042-ECOL-1999***

Establece los límites máximos permisibles de emisión de hidrocarburos no quemados, monóxido de carbono, óxidos de nitrógeno y partículas suspendidas provenientes del escape de los vehículos automotores nuevos en planta, así como de hidrocarburos evaporativos provenientes del sistema de combustible que usan gasolina, gas licuado de petróleo, gas natural y diesel de los mismos, cuyo peso bruto vehicular no exceda los 3,856 kilogramos y es de de naturaleza obligatoria para los fabricantes e importadores de dichos vehículos.

### ***NORMA OFICIAL MEXICANA NOM-044-ECOL-1993,***

Establece los niveles máximos permisibles de emisión de hidrocarburos, monóxido de carbono, óxidos de nitrógeno, partículas suspendidas totales y opacidad de humo provenientes del escape de motores nuevos que usan diesel como combustible y que se utilizarán para la propulsión de vehículos automotores con peso bruto vehicular mayor de 3,857 kilogramos.

### ***NORMA OFICIAL MEXICANA NOM-047-ECOL-1999***

Establece las características del equipo y el procedimiento de medición para la verificación de los límites de emisión de contaminantes, provenientes de

los vehículos automotores en circulación que usan gasolina, gas licuado de petróleo, gas natural u otros combustibles alternos.

***NORMA OFICIAL MEXICANA NOM-048-ECOL-1993***

Establece los niveles máximos permisibles de emisión de hidrocarburos, monóxidos de carbono y humo, provenientes del escape de las motocicletas en circulación que utilizan gasolina o mezcla de gasolina-aceite como combustible.

***NORMA OFICIAL MEXICANA NOM-049-ECOL-1993***

Establece las características del equipo y el procedimiento de medición, para la verificación de los niveles de emisión de gases contaminantes, provenientes de las motocicletas en circulación que usan gasolina o mezcla de gasolina-aceite como combustible.

***NORMA OFICIAL MEXICANA NOM-050-ECOL-1993***

Establece los niveles máximos permisibles de emisión de gases contaminantes provenientes del escape de los vehículos automotores en circulación que usan gas licuado de petróleo, gas natural u otros combustibles alternos como combustible.

***NORMA OFICIAL MEXICANA NOM-076-ECOL-1995***

Establece los niveles máximos permisibles de emisión de hidrocarburos no quemados, monóxido de carbono y óxidos de nitrógeno provenientes del escape, así como de hidrocarburos evaporativos provenientes del sistema de combustible, que usan gasolina, gas licuado de petróleo, gas natural y otros combustibles alternos y que se utilizarán para la propulsión de vehículos automotores, con peso bruto vehicular mayor de 3,857 kilogramos nuevos en planta.

## ***NORMA OFICIAL MEXICANA NOM-077-ECOL-1995***

Establece el procedimiento de medición para la verificación de los niveles de emisión de la opacidad del humo proveniente del escape de los vehículos automotores en circulación que usan diesel como combustible.

En la ciudad de México se están revirtiendo las tendencias favorables de calidad del aire y de reducción del ozono observadas entre 1994 y 1997. Ello debe llevar a un replanteamiento de las políticas urbanas y desde luego a un compromiso por parte de las principales casas automotrices para una producción vehicular más limpia, inducida por un nuevo marco normativo que debe establecerse a la brevedad.

Dada la actual dinámica de los contaminantes atmosféricos deben verse como prioritarias aquellas medidas que disminuyan su generación y que induzcan la renovación del parque vehicular, considerando que la aportación de contaminantes de modelos viejos y en mal estado supera en decenas de veces a la de nuevos modelos.

En este sentido, es preciso modificar los máximos permisibles de la NOM 041 para vehículos en circulación y hacerlos más estrictos. Deberá además reestructurarse las categorías de edades para automóviles particulares y camiones ligeros, medianos y pesados, y fijarse sólo dos categorías, definida para modelos 1993 y anteriores y 1994 y posteriores, en lugar de mantener las actuales. Se incorporarían así avances tecnológicos importantes y se establecerían nuevos límites que forzarían a que los modelos viejos se mantuvieran en excelentes condiciones o a cambiarse por modelos más recientes.

La NOM 042 para vehículos nuevos debe buscar definir límites para emisiones de vehículos en planta homologados con los establecidos en Estados Unidos. Si las empresas automotrices existentes en México fabrican autos y camionetas para exportación a Estados Unidos con todos los requisitos que ese país fija, entonces existe la capacidad y la factibilidad para que en México se

produzcan automóviles de bajas emisiones que deterioren en menor medida la calidad del aire. Los óxidos de nitrógeno y el monóxido de carbono deben quedar homologados con Tier1<sup>4</sup> entre 1999 y el año 2000, y desde luego homologarse con Tier2<sup>5</sup> cuando esta norma entre en vigor en Estados Unidos, entre el 2004 y el 2006. También deben hacerse obligatorios periodos mínimos de cumplimiento similares a los estipulados por la EPA (Agencia de Protección Espacial).

1.4 <http://www.chevron.com/products/prodserv/fuels/techrongas/es/>.

El programa TOP TIER establece los requisitos para niveles de detergencia en las gasolinas, niveles que son superiores a aquellos actualmente impuestos por la Agencia para la Protección del Medio Ambiente (EPA). El programa establece requerimientos para el desarrollo de gasolinas que mantendrán más limpias las partes de los motores, reduciendo los depósitos causados por gasolina de baja calidad, ayudando a dar un mejor desempeño.

En lo que respecta a los vehículos de bajas emisiones y ultra bajas emisiones, es necesario estimular programas voluntarios similares a los estadounidenses Low Emission Vehicles (LEV) y Ultra Low Emission Vehicles (ULEV) y homologar los estándares entre los mismos.

Finalmente, las normas de emisiones vehiculares deben modificarse para favorecer el uso y la reconversión de vehículos a combustibles alternos limpios, como el gas natural.

---

<sup>4</sup> Compañías que poseen diseños avanzados y sistemas de integración producen y entregan sus planes de ingeniería al ensamblador final de la planta. Fuente:

[http://www.alianzaautomotriz.com.mx/articulos.php?id\\_sec=7&id\\_art=394&id\\_ejemplar=21](http://www.alianzaautomotriz.com.mx/articulos.php?id_sec=7&id_art=394&id_ejemplar=21).

<sup>5</sup> Compañías típicamente que entregan submódulos de sus archivos a tier1 los cuales se especializan en ciertos productos y reducirá sus costos competitivamente. Fuente:

[http://www.alianzaautomotriz.com.mx/articulos.php?id\\_sec=7&id\\_art=394&id\\_ejemplar=21](http://www.alianzaautomotriz.com.mx/articulos.php?id_sec=7&id_art=394&id_ejemplar=21)

## **1.5 HISTORIA DEL USO DE LAS BASES DE DATOS, TIPOS Y USOS MÁS FRECUENTES**

Los predecesores de los sistemas de bases de datos fueron los sistemas de ficheros. No está definido un momento concreto en que los sistemas de ficheros hayan cesado y dado comienzo a los sistemas de bases de datos. De hecho, todavía existen sistemas de ficheros en uso.

Se dice que los sistemas de bases de datos tienen sus raíces en el proyecto estadounidense Apolo (mandar al hombre a la luna, en los años sesenta). En aquella época, no había ningún sistema que permitiera gestionar la gran cantidad de información que requería el proyecto. La primera empresa encargada del proyecto, NAA (North American Aviation), desarrolló un software llamado GUAM (General Update Access Method) que estaba basado en el concepto de que varias piezas pequeñas se unen para formar una pieza más grande, y así hasta que el producto final está ensamblado. Esta estructura, que tiene la forma de un árbol, es lo que se denomina una estructura jerárquica. A mediados de los sesenta, IBM se unió a NAA para desarrollar GUAM en lo que ahora se conoce como IMS (Information Management System)..

A mitad de los sesenta, se desarrolló IDS (Integrated Data Store), de General Electric fue dirigido por uno de los pioneros en los sistemas de bases de datos, Charles Bachmann. IDS era un nuevo tipo de sistema de bases de datos conocido como sistema de red, que produjo un gran efecto sobre los sistemas de información de aquella generación. El sistema de red se desarrolló, en parte, para satisfacer la necesidad de representar relaciones entre datos más complejos que las que se podían modelar con los sistemas jerárquicos, y, en parte, para imponer un estándar de bases de datos. Para ayudar a establecer dicho estándar, CODASYL (Conference On Data Systems Languages), formado por representantes del gobierno de EEUU y representantes del mundo empresarial, formaron un grupo denominado DBTG (Data Base Task Group), cuyo objetivo era definir especificaciones estandarizadas que permitieran la creación de bases de datos y el manejo de los datos. El DBTG presentó su informe final en el año de



1971 y aunque éste no fue formalmente aceptado por ANSI (American National Standards Institute), muchos sistemas se desarrollaron siguiendo la propuesta del DBTG. Estos sistemas son los que se conocen como sistemas de red, o sistemas CODASYL o DBTG. Los sistemas jerárquicos y de redes constituyen la primera generación de los Sistemas Gestores de Bases de Datos. Pero estos sistemas presentan algunos inconvenientes:

- *Es necesario escribir complejos programas de aplicación para responder a cualquier tipo de consulta de datos, por simple que ésta sea.*
- *La independencia de datos es mínima.*
- *No tienen un fundamento teórico.*

En 1970 Edgar Frank Codd, de los laboratorios de investigación de IBM, escribió un artículo presentando el modelo relacional. En este artículo, presentaba también los inconvenientes de los sistemas previos como el jerárquico y el de red. Entonces, se comenzaron a desarrollar muchos sistemas relacionales, apareciendo los primeros a finales de los setenta y principios de los ochenta. Uno de los primeros es, System R, de IBM, que se desarrolló para probar la funcionalidad del modelo relacional, proporcionando una implementación de sus estructuras de datos y sus operaciones. Esto condujo a dos grandes desarrollos:

El desarrollo de un lenguaje de consultas estructurado denominado Structure Query Language (SQL), que se ha convertido en el lenguaje estándar de los sistemas relacionales.

La producción de varios SGBD relacionales durante los años ochenta, como DB2 y SLQ/DS de IBM, SQL Server 2000 de Microsoft y ORACLE de ORACLE Corporation.

Hoy en día, existen cientos de SGBD relacionales, tanto para microordenadores como para sistemas multiusuario, aunque muchos no son completamente fieles al modelo relacional.

Otros sistemas relacionales multiusuario son INGRES de Computer Associates, Informix de Informix Software Inc. y Sybase de Sybase Inc. Ejemplos de sistemas relacionales de microordenadores son Paradox y dBase IV de Borland, Access de Microsoft, FoxPro y R (base de Microrim).

Los SGBD relacionales constituyen la segunda generación de los SGBD. Sin embargo, el modelo relacional también tiene sus fallos, siendo uno de ellos su limitada capacidad al modelar los datos. Se ha hecho mucha investigación desde entonces tratando de resolver este problema. En 1976, Peter Chen presentó el modelo entidad-relación, que es la técnica más utilizada en el diseño de bases de datos. En 1979, Edgar Frank Codd intentó subsanar algunas de las deficiencias de su modelo relacional con una versión extendida denominada RM/T (1979) y más recientemente RM/V2 (1990).

Como respuesta a la creciente complejidad de las aplicaciones que requieren bases de datos, han surgido dos nuevos modelos: el modelo de datos orientado a objetos y el modelo relacional extendido. Sin embargo, a diferencia de los modelos que los preceden, la composición de estos modelos no está clara. Esta evolución representa la tercera generación de los SGBD.

### **1.5.1 TIPOS DE BASES DE DATOS**

Una base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En la actualidad, y en gran parte gracias a la tecnología y recursos disponibles provenientes de campos como la informática y la electrónica, las bases de datos pueden adquirir diversas formas, ofreciendo un amplio rango de soluciones al problema de almacenar datos.

En informática existen los Sistemas Gestores de Bases de Datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

### **1.5.1.1 BASES DE DATOS SEGÚN SU FUNCIÓN**

Hay diferentes formas de dividir a las bases de datos, pero en esta ocasión la dividiremos en dos grupos considerando su función:

#### *Analíticas*

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

#### *Dinámicas*

Éstas son bases de datos más dinámicas, orientadas a almacenar información que es modificada con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta.

### 1.5.1.2 BASES DE DATOS SEGÚN MODELOS ADMINISTRATIVOS

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como *contenedor de datos* (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas; son abstracciones que permiten la implementación de un sistema eficiente de *base de datos*, por lo general se refieren a algoritmos y conceptos matemáticos.

#### *Bases de datos jerárquicas*

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol visto al revés, en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres se le conoce como *raíz*, y a los nodos que no tienen hijos se les conoce como *hojas*.

Las limitaciones de este modelo, es su incapacidad de representar eficientemente la redundancia de datos.

#### *Bases de datos de red.*

Éste es un modelo distinto del jerárquico, en donde su diferencia fundamental es la modificación del concepto de un *nodo*, permitiendo que un mismo nodo tenga varios padres (algo no permitido en el modelo jerárquico).

Se mejoró con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos, pero aun así, la dificultad que significa administrar la información en una base de datos de red, ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

### *Bases de datos relacionales.*

Éste es el modelo que más se utiliza en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postuladas sus bases en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "tablas", compuestas de *registros* (las filas de una tabla) y *campos* (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario casual de la base de datos. La información puede ser recuperada o almacenada por medio de "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más común para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje de Consultas Estructurado*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

### *Bases de datos orientadas a objetos.*

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento). Esto quiere decir que en este modelo de Bases de Datos todo se ve como objetos, ya no como relaciones o entidades.

### *Base de datos distribuidas.*

Una base de datos distribuida (BDD) es la unión de las bases de datos con redes.

La base de datos está almacenada en varias computadoras conectadas en red, (ya sea en el mismo lugar físicamente o distribuidas a lo largo de la red) lo que permite al acceso de datos desde diferentes máquinas. Está manejada por el Sistema de Administración de Datos Distribuida (SABDD) o Sistema de Gestión de Base de Datos Distribuida. Son la evolución del modelo cliente-servidor.

La razón principal detrás de las BDD son los organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a la información, sin tener todo centralizado en un sólo punto. Ejemplo: bancos, cadenas de hoteles, campus de distintas universidades, sucursales de tiendas departamentales, etc.

## **1.6 VENTAJAS E INCONVENIENTES DE LAS BASES DE DATOS**

Los sistemas de bases de datos presentan numerosas ventajas que se pueden dividir en dos grupos: las que se deben a la integración de datos y las que se deben a la interfase común que proporciona el SGBD.

### **1.6.1 VENTAJAS POR INTEGRACIÓN DE DATOS**

Control sobre la redundancia de datos. Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos. En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

Consistencia de datos. Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes. Desgraciadamente, no todos los SGBD de hoy en día se encargan de mantener automáticamente la consistencia.

Más información sobre la misma cantidad de datos. Al estar todos los datos integrados, se puede extraer información adicional sobre los mismos.

Compartir datos. En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados. Además, las nuevas aplicaciones que se vayan creando pueden utilizar los datos de la base de datos existente.

Mantenimiento de estándares. Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

### **1.6.2 VENTAJAS POR LA EXISTENCIA DEL SGBD**

Mejora en la integridad de datos. La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargarse de mantenerlas.

Mejora en la seguridad. La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros. Sin embargo, los SGBD permiten mantener la seguridad mediante el establecimiento de claves para identificar al personal autorizado a utilizar la base de datos. Las autorizaciones se pueden realizar a nivel de operaciones, de modo que un usuario puede estar autorizado a consultar ciertos datos pero no a actualizarlos.

Mejora en la accesibilidad a los datos. Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

Mejora en la productividad. El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación. El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel. Muchos SGBD también proporcionan un entorno de cuarta generación consistente en un conjunto de herramientas que simplifican, en gran medida, el desarrollo de las aplicaciones que acceden a la base de datos. Gracias a estas herramientas, el programador puede ofrecer una mayor productividad en un tiempo menor.

Mejora en el mantenimiento gracias a la independencia de datos. En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan. Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados. Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como



independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

Aumento de la concurrencia. En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o, incluso, que se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

Mejora en los servicios de copias de seguridad y de recuperación ante fallos. Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos. En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

### **1.6.3 INCONVENIENTES DE LOS SISTEMAS DE BASES DE DATOS.**

Complejidad. Los SGBD son conjuntos de programas muy complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder sacar provecho de ellos.

Tamaño. Los SGBD son programas complejos y muy extensos que requieren una gran cantidad de espacio en disco y de memoria para trabajar de forma eficiente.

Costo económico del SGBD. El costo de un SGBD varía dependiendo del entorno y de la funcionalidad que ofrece.

Costo del equipamiento adicional. Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las ventajas deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

Costo de la conversión. En algunas ocasiones, el costo del SGBD y el costo del equipo informático que sea necesario adquirir para su buen funcionamiento, es insignificante comparado al costo de convertir la aplicación actual en un sistema de bases de datos. Este precio incluye la capacitación para utilizar estos sistemas y, probablemente, el precio del personal especializado para ayudar a realizar la conversión y poner en marcha el sistema. Este precio es una de las razones principales por las que algunas empresas y organizaciones se resisten a cambiar su sistema actual de ficheros por un sistema de bases de datos.

Vulnerable a los fallos. El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse.

## **1.7 ESTADO ACTUAL**

Aunque las bases de datos orientadas a objetos se están volviendo más populares, en el entorno de desarrollo actual, la base de datos relacional sigue siendo el método predominante para almacenar datos. Los diagramas de clases de UML se pueden usar para modelar la base de datos relacional en la que el sistema está basado, sin embargo, los diagramas tradicionales de modelado de datos capturan más información sobre la base de datos relacional y son más adecuados para modelarla. Los Diagramas de Relaciones de Entidad (ER) son

una extensión importante de UML<sup>6</sup> para el modelado de bases de datos relacionales.

UML es una notación, no un método. No impera un proceso para modelar un sistema. No obstante, como UML incluye los diagramas de casos de uso, se le considera estar dotado de una aproximación al diseño centrada en el problema con los casos de uso. El Diagrama de Caso de Uso nos da el punto de entrada para analizar los requisitos del sistema, y el problema que necesitamos solucionar.

Un caso de uso se modela para todos los procesos que el sistema debe llevar a cabo. Los procesos se describen dentro del caso de uso por una descripción textual o una secuencia de pasos ejecutados. Los Diagramas de Actividad se pueden usar también para modelar escenarios gráficamente. Una vez que el comportamiento del sistema está captado de esta manera, los casos de uso se examinan y amplían para mostrar qué objetos se interrelacionan para que ocurra este comportamiento. Los Diagramas de Colaboración y de secuencia se usan para mostrar las relaciones entre los objetos. Además el UML nos ofrece una serie de herramientas de diagramación de Bases de Datos.

Es por esto, que es común en la actualidad apoyarse de herramientas para el diseño y modelado de la Base de Datos, la más utilizada es el lenguaje UML, debido a las ventajas antes descritas al modelar las reglas de negocio en sus diferentes entidades y relaciones. La tendencia más fuerte es utilizar Bases de Datos Relacionales ya que tienen ventajas sobre el almacenamiento de datos. Podremos decir que estamos sobre la tercera generación de los Sistemas Manejadores de Bases de Datos.

---

<sup>6</sup> Unified Model Lenguaje. Sitios de Interés:  
<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/>  
<http://www.intersoftware.com.mx/>

## **1.8 TENDENCIAS PARA EL FUTURO**

El futuro siempre es incierto y por esta razón es difícil dar un diagnóstico exacto de cual será el modelo de Base de Datos predominante a mediano plazo, lo que si podemos acertar es que nos encontraremos en otra generación de DBMS, sólo cabe destacar que las tendencias indican que la corriente de Orientado a Objetos será la que determinará la pauta para los Sistemas de Información a corto plazo todo esto después de haber desplazado a las Bases de Datos Relacionales.

## **1.9 NORMAS DE ESTANDARIZACIÓN DE UNA BASE DE DATOS**

Para trabajar con los datos de una base de datos, debe utilizar un conjunto de comandos e instrucciones (lenguaje) definidos por el software del Data Base Management System (Sistema Manejador de Base de Datos). En las bases de datos relacionales se pueden utilizar varios lenguajes, de los que SQL es el más común y además utilizaremos para la creación de la propuesta de la Base de Datos Unificada.

El American National Standards Institute (ANSI) que durante casi noventa años ha servido como el coordinador de las normas de las organizaciones en vías de desarrollo, asociaciones de comercio, profesional y sociedades técnicas, gobierno, etc., y que influye en las actividades de regularización globales y el desarrollo de normas internacionales, además, de ISO (Organización Internacional para Regularización) es el diseñador más grande del mundo de normas. Aunque la actividad principal de ISO es el desarrollo de normas técnicas, las normas de ISO también tienen repercusiones económicas y sociales importantes. Estos dos institutos o asociaciones definen estándares de software, incluidos los estándares para el lenguaje SQL. Los diferentes Manejadores de Base de Datos se apegan a los diversos estándares de los antes mencionados para la creación y manipulación

de las Bases de Datos, cada DBMS además de basarse en estos estándares, contienen su propia y diferente interfaz, haciendo propia de su autoría su manera de realizar sus scripts y demás operaciones, como manipulación, administración, etc.

Por ejemplo, SQL Server 2000 admite el nivel de entrada de SQL-92, el estándar de SQL publicado por ANSI e ISO en 1992. El dialecto de SQL compatible con Microsoft SQL Server se llama Transact-SQL (TSQL). T-SQL es el lenguaje principal utilizado por las aplicaciones de Microsoft SQL Server.

En nuestro siguiente capítulo analizaremos a fondo la información recopilada aquí y las necesidades del medio para el cual generaremos la Base de Datos, otorgando más forma a nuestra información.

## CAPÍTULO 2

### ENTENDIMIENTO DE LAS NECESIDADES DEL MANEJO DE INFORMACIÓN EN LA VERIFICACIÓN VEHICULAR

Ya que hemos recopilado la información necesaria en el capítulo anterior, nos dedicaremos ahora a entender las necesidades que requerirá nuestro proyecto para poder tener una buena aceptación en el ambiente de la verificación vehicular.

En 1992 el Departamento del Distrito Federal especificó que los equipos deben cumplir con las especificaciones del “Bureau of Automotive Repair” del Gobierno del Estado de California EE.UU. (BAR) y contar con su certificación BAR-90 a lo cual este tipo de normas son Internacionales.

El BAR ha certificado analizadores durante más de 15 años. Cada generación de analizadores ha sido más confiable, precisa y compleja. Las primeras especificaciones de analizadores se publicaron en 1974 y se hicieron publicaciones después en 1980, en 1984 y en 1990. A los analizadores que cumplieron con las especificaciones el BAR les otorgó un certificado.

Los analizadores BAR-74 y BAR-80 medían únicamente hidrocarburos (HC) y monóxido de carbono (CO), los analizadores BAR-74 y BAR-80 no fueron contemplados en el capítulo 1 en la historia de la Verificación Vehicular ya que estos carecían de un sistema de computadora (PC). Los BAR-80 eran considerablemente más precisos que los BAR-74 debido a mejoras en el diseño de la banca óptica infrarroja, a auto-diagnósticos rudimentarios y a un cilindro instalado de gas de calibración. Sin embargo, no fue sino hasta que se desarrollaron las especificaciones BAR-84 que los analizadores se volvieron computarizados y requirieron que se realizara una calibración de gas (chequeo de

los parámetros de medición de gases) cada siete días o no permitiría ninguna prueba adicional.

El uso de una computadora permitió al analizador tomar automáticamente la decisión de aprobación o de rechazo del vehículo. Permitted también especificar otras características que hacía posible detectar la alteración del analizador y para dar instrucciones especiales al técnico.

El BAR comenzó a hacer planes para desarrollar una nueva especificación para analizadores en noviembre de 1987 como resultado de un estudio que mostró que los beneficios del programa podrían incrementarse si se hacían modificaciones a los analizadores BAR-84. El estudio mostró que tendrían que ser efectuados algunos cambios en el software y también en el equipo para reducir el número de fallas que generaban aprobaciones por equivocación.

Las características del BAR-90 eran:

- Mediciones de emisiones vehiculares de HC, CO y CO<sub>2</sub> (O<sub>2</sub> se ofrecerá como opción).
- Mediciones de R.P.M. (pulsos de las bobinas del motor);
- Determinación de la dilución (exceso de oxígeno en la muestra) de los gases de escape.
- La capacidad para agregar un lector de código de barras para una entrada de datos más precisa y conveniente.
- Una impresora dedicada para los reportes de inspección del vehículo y otras impresoras para las constancias técnicas de verificación y de propósito general;
- La capacidad para grabar los datos en diskettes flexibles de 1.44 Mb de 3,5" y en discos duros de 40 megabytes.
- La capacidad de desplegar en pantalla información para el operador del equipo de verificación.

- Una operación controlada por microprocesador totalmente manejada por medio de un menú interactivo de fácil interpretación.

La especificación BAR-90 fue diseñada para proveer un servicio confiable y preciso en un ambiente de un centro de verificación o en un taller mecánico.

Esta especificación de BAR en California fue tomada como base para el desarrollo de las especificaciones para los equipos en la Ciudad de México. Desde la licitación inicial, su desarrollo siguió dos vertientes:

- Para los equipos destinados a Centros Autorizados en talleres mecánicos.
- Para los equipos destinados a los Macro Centros y Centros Especializados de Verificación Vehicular.

Aunado a lo anterior, a pesar de que existe coherencia entre las especificaciones de los equipos de diversos fabricantes, sus programas computarizados de control (el software) varían considerablemente ya que el programador de cada proveedor lo desarrolló a su manera dentro de algunos lineamientos generales, la mayoría no escritos.

Parte de la causa de estas diferencias ha sido la carencia de especificaciones escritas. El no contar con una documentación completa ha facilitado que cada proveedor interprete de una forma diferente los requisitos del programa.

Con base en lo anterior los Estatutos del BAR-90 a BAR-97 internacional, la Secretaria de Ecología de Medio Ambiente, la Dirección General de Prevención y Control de la Contaminación Ambiental de la Secretaría de Ecología del Departamento del Distrito Federal y Estado de México han tratado de homologar criterios que han dado como resultado ciertas normas de control de emisiones de contaminantes del parque vehicular del Valle de México.



En este capítulo sólo nos enfocaremos a las reglas de negocio que tengan que ver con la manipulación y el almacenamiento de la información y sus aspectos más cercanos en el proceso de la verificación vehicular.

## **2.1 CONFIGURACIÓN GENERAL**

El equipo de verificación vehicular debe consistir de los siguientes elementos:

- Un servidor central de archivos y programas. Debe tener suficiente espacio en disco duro para almacenar los resultados de verificaciones realizadas así como los demás archivos durante 14 meses de operación. Debe almacenar todos los programas involucrados en el proceso de verificación vehicular, incluso los que se utilizan exclusivamente en las líneas de verificación. El servidor central debe contar con dispositivos de almacenamiento secundario y mantener en disco duro los índices asociados con su consulta.
- Un servidor de comunicación corriendo en plataforma Windows NT que conecta la red local con una red amplia uniendo todos los centros de verificación. Esta máquina debe tener suficiente espacio en disco duro para mantener un espejo de todos los archivos de datos que residen en el servidor de la red. Debe unir también a los resultados de aforo de la computadora de aforo, con los de las cámaras de video y las verificaciones realizadas. Esta computadora es la única que podrá correr programas diferentes a los especificados por la DGCA (Dirección General de Control al Ambiente) para efectos de administración del centro de verificación. Dichos programas deben residir en directorios diferentes a los de verificación. El acceso del Administrador a Windows NT siempre será de la DGCA.

- El servidor y la Base de Datos debe permitir la operación simultánea de cuando menos 5 líneas de verificación a su capacidad máxima sin sufrir deterioro en su comportamiento o pérdida de datos. Cada línea de verificación debe ser de dos estaciones (uno de captura y inspección visual, uno de prueba física) con la posibilidad de agregar una tercera estación en el futuro para pruebas evaporativas.

### **2.1.1 DISEÑO MODULAR**

El diseño modular debe permitir realizar cambios en un módulo sin afectar la operación de los demás módulos. Debe existir una librería perfectamente definida de objetos y variables a través de los cuales existe toda clase de interacción entre módulos.

El software de Base de Datos utilizado en sistema de verificación vehicular debe constar de los siguientes módulos.

Módulos de Administración de Información para:

- La obtención de datos en la línea de verificación de diversas Bases de Datos y Tablas de Referencia.
- El envío de datos de la línea a diversas bases de datos centralizados.
- La administración de las Bases de Datos centralizadas.
- Lectura de datos de discos compactos (CD)
- Lectura de datos de códigos de barra.

Módulos de Transmisión Electrónica para:

- Integrar el LAN del Centro de Verificación a una WAN conectado a la DGCA.

### **2.1.2. ALTA CAPACIDAD DE PRUEBA**

El equipo de verificación vehicular debe ser diseñado de modo que sea capaz de verificar a por lo menos 15 vehículos por hora por línea de verificación sin experimentar cuellos de botella u otros efectos negativos. La cifra arriba mencionada considera un tiempo promedio de prueba dinámica de 90 segundos por vehículo. En el caso de que el tiempo promedio sea mayor, la diferencia podrá sumarse a los 4 minutos por prueba arriba estipulados.

Esta información será importante en el diseño de nuestra Base de Datos para obtener los cálculos acertados sobre el incremento diario de la Base de Datos.

La Base de Datos debe ser capaz de almacenar, administrar, procesar y realizar otras tareas no decrementando su desempeño, aun conteniendo grandes bloques de información.

### **2.1.3 RESISTENCIA A LA ALTERACIÓN**

La Base de Datos debe ser diseñada y construida para asegurar que solo el personal autorizado podrá tener acceso a las áreas restringidas de ésta.

El control de acceso debe prohibir a los técnicos y operadores de los equipos, crear o cambiar cualquier resultado de las pruebas, programas de verificación o datos de los archivos contenidos en los equipos. Debe prohibir la introducción de cualquier dato al sistema que no sea proveniente de un elemento del sistema de gasolina o en su defecto debe contar con los privilegios para hacerlo, no violando las reglas de integridad; no podrá haber ninguna posibilidad de conectar una computadora a la LAN para introducir o modificar datos relacionados con la verificación de vehículos o impresión de certificados.

Se debe utilizar particiones especiales en disco duro, así como también otras propiedades en el equipo y software para proteger los archivos y programas de Verificación Vehicular.

#### **2.1.4 DISEÑO DEL EQUIPO**

El equipo debe ser de una construcción robusta que permita una operación confiable sin problemas de mantenimiento en un ambiente típico de centro de verificación. Su diseño debe facilitar, en todos momentos, el servicio a sus componentes.

Todos los programas deberán ser perfectamente integrados con acceso vía un menú principal, o submenú en cada una de las computadoras. A estos programas se deberá acceder desde el Menú Principal o un submenú, y cuando salgan, volverán directamente al menú o submenú desde el cual se accedieron, sin requerir reiniciar la computadora o sistema operativo.

#### **2.1.5 DOCUMENTACIÓN DEL SOFTWARE**

La Base de Datos del Sistema deberá estar totalmente documentada. La documentación del software incluirá por lo menos lo siguiente:

- a. Las especificaciones funcionales.
- c. Diagramas entidad/relación de la Base de Datos.
- d. Ejemplos de entradas y salidas de todos los procesos.
- f. Todos los formatos de archivo con nombres de archivo.

### **2.2 ESPECIFICACIONES DE LA BASE DE DATOS**

El programa de cómputo para las líneas de verificación así como cualquier tabla de referencia debe quedar almacenada exclusivamente en el disco duro del

servidor. No es aceptable que estos programas o archivos residen en los discos duros locales de las computadoras de línea, este requerimiento es para asegurar que todas las líneas de verificación funcionan con el mismo software. Con la comprobación de los componentes de la computadora, el software de aplicaciones debe cargarse desde el servidor.

### **2.2.1 DATOS DEL CENTRO Y DE LA LÍNEA DE VERIFICACIÓN**

La Base de Datos permitirá que el número del centro, el número de cada línea de verificación, la marca y número de serie del equipo instalado en cada línea sean cambiados únicamente por los representantes del gobierno.

## **2.3 DIRECTORIOS Y ARCHIVOS**

### **2.3.1 LOCALIZACIÓN DE LOS ARCHIVOS**

El directorio raíz del Servidor de la red debe contener todos los programas y archivos requeridos para el funcionamiento del sistema.

El directorio raíz del Servidor de la red también debe contener las tablas de límites, la Tabla Maestra de Vehículos y todas las tablas auxiliares generadas por la DGCA (Dirección General de Control Ambiental) en un subdirectorío en raíz.

### **2.3.2 ACCESO A LAS TABLAS**

Se debe tomar las precauciones necesarias para asegurar que ninguna computadora ajena podrá conectarse por red y acceder las tablas de la BD

### 2.3.3 LIMPIEZA DE LAS TABLAS

Para limitar el espacio requerido para almacenar estas tablas, se debe implementar un procedimiento de limpieza automático de registros de la siguiente forma:

#### *Limpieza de 14 meses*

Los siguientes archivos deben tener una limpieza diaria de los registros con más de 14 meses de almacenamiento:

- ✓ Acceso
- ✓ Calaudi
- ✓ Calbanco
- ✓ Calcomp
- ✓ Caldino
- ✓ Calfugas
- ✓ Fallas
- ✓ Vehículo

#### *Limpieza diaria de 14 días*

Los siguientes archivos deben tener una limpieza diaria de los registros con más de 14 días de almacenamiento:

- ✓ Segundo por Segundo

Tablas que no se deben de limpiar

- ✓ Personal

### **2.3.4 TABLAS MAESTRA Y TRABAJO**

Esta tabla proveerá información sobre los límites de emisiones, los protocolos de prueba, marcas y otros datos para ayudar al técnico a desarrollar la prueba con mayor precisión. Además existen otras tablas las cuales contienen datos para el desarrollo de las verificaciones, las cuales son: Carga, Carrocer, Estado, Marcas, Municip, Robados, Submarca, etc.

### **2.3.5 ESTRUCTURAS Y CONTENIDOS DE LAS TABLAS**

Todos los campos numéricos, cuando se encuentren vacíos deberán llenarse con ceros a menos que se especifique de otra manera y los espacios de los campos alfanuméricos deberán llenarse con espacio. Cualquier referencia a un campo en "blanco" significa que debe llenarse con un espacio ASCII.

Todos los campos se escribirán al archivo como datos de caracteres ASCII.

Todas las descripciones de datos en esta tabla bajo "Descripciones de Campo" se refieren a los caracteres permitidos dentro del campo.

Para los campos de fecha, el registro debe contener la fecha en el formato (DDMMAAAA): M= mes, D= día, A= año.

Los siguientes formatos de tablas servirán como referencia para realizar el nuevo diseño de una BD unificada, esto significa que existirán nuevas tablas y campos, pero sin omitir la información necesaria, así también no será obligatorio en la propuesta contemplar exactamente toda la información como está estructurada por ahora.

## ***Entidades Actuales***

*Vehículo.* Esta tabla debe contener los registros de cada prueba o intento de prueba de verificación vehicular. Aún los datos de cualquier prueba abortada deben escribirse a esta tabla.

*Fallas.* Esta tabla debe contener el estatus de bloqueo / desbloqueo de los equipos del centro, así como fallas en sus sistemas mecánicos, siempre y cuando el programador halla generado una rutina de sistema para contemplarlo como error.

*Personal.* Esta tabla debe contener los registros de cada persona autorizada para trabajar en el centro con sus claves de acceso.

*Calaudi.* Esta tabla debe de contener todas las auditorias de gas patrón desglosadas segundo a segundo, elaboradas por el laboratorio correspondiente.

*Calbanco.* Esta tabla debe contener todas las pruebas de calibración de gas que cada 24 horas debe de realizar los equipos de verificación.

*Calcomp.* Esta tabla debe contener todas las auto-pruebas de calibración elaboradas en forma aleatoria en los equipos para comprobar su buen estado para verificar con los valores de regla más óptimos.

*Caldino.* Esta tabla debe contener todas las pruebas de calibración del dinamómetro que cada 24 horas debe de realizar los equipos de verificación.

*Calfugas.* Esta tabla debe contener todas las pruebas de calibración de fugas que cada 24 horas debe de realizar los equipos de verificación.



*Acceso.* Esta tabla debe contener información de cada intento de acceder el sistema aún si no fuera exitosa. Contiene la fecha y hora del acceso, la clave utilizada, a quien corresponde y una breve descripción de las acciones posteriores al acceso.

*Segundo a segundo.* Esta tabla debe contener la información de segundo por segundo de cada prueba realizada bajo el protocolo de Prueba de Aceleración Simulada (PAS) de verificación vehicular. Cada archivo contiene un día de registros o una semana entera.

Se debe crear en cada centro de verificación un archivo "SPS" en formato ASCII por día con los datos segundo por segundo de las pruebas Acceleration Simulation Mode (ASM o PAS) realizadas.

El nombre del archivo tendrá la forma de 0XXXXAASS en donde:

- XXXX = 4 caracteres numéricos que son el número del Verificentro. En caso necesario, se rellena a la izquierda con ceros.
- AA= 2 caracteres numéricos que representan el año:
- SS=2 caracteres numéricos que representan la semana de año en curso.

<b><u>Nombre de la Tabla</u></b>	<b><u>Uso General de la Tabla</u></b>
Vehiculo	Datos del Vehiculo Verificado.
Fallas	Fallas del Sistema (Errores capturados por el programador)
Personal	Datos de los Usuarios del Sistema
Calaudi	Datos de la Auditoria de Laboratorio.
Calbanco	Datos de la prueba de gas de 24 horas.
Calcomp	Datos de la Auto prueba de gas aleatoria.
Caldino	Datos de la prueba de dinamómetro de 24 horas.
Calfugas	Datos de la prueba de fugas de 24 horas.
Acceso	Accesos al sistema y acciones realizadas.
Segundo a segundo	Emisiones de los vehículos detalladas segundo por segundo.

## 2.4 HOMOLOGACIÓN DE EQUIPOS

Estas especificaciones son una parte integral para el diseño de la Base de Datos, y se han desarrollado para asegurar que los equipos propuestos para el uso en el programa de verificación vehicular de la Ciudad de México, cumple con ciertos requerimientos mínimos. En este proyecto trataremos de homologar todos estos conceptos y las nuevas metodologías y técnicas más actuales para mejorar este producto.

Ya que analizamos las necesidades del ambiente de verificación en éste capítulo, seguiremos con la parte de diseño que se vera en el capítulo 3, abstraeremos toda la información de las reglas de negocio para poder darles una representación técnica y así poder generar un prototipo de la Base de Datos.

## **CAPITULO 3**

### **DISEÑO DE LA BASE DE DATOS UNIFICADA**

Después de haber hecho un levantamiento de los requerimientos de la Base de Datos y de analizar esta información en los capítulos anteriores nos avocaremos a obtener un nuevo diseño que cumpla con las necesidades actuales y satisfaga las futuras expectativas con respecto a la extensibilidad y modularidad.

#### **3.1 ABSTRACCIÓN DE LAS REGLAS DE NEGOCIO**

En este punto describiremos las reglas del negocio que conciernen y podrían afectar en su estructura al proyecto de la propuesta de base de datos para los verificadores pero de cierta manera también discriminaremos ciertas reglas que pertenecen a la aplicación únicamente.

En promedio se desea que en el proceso de verificación no se encuentren con cuellos de botella, como mínimo se debe de verificar a 15 vehículos por hora por cada línea de verificación con la cuenta el centro. Aprovecharemos las herramientas y las facilidades que nos brinde la plataforma seleccionada para conseguir rápidos accesos de lectura en las tablas de consulta y de escritura en las tablas donde se actualice la información.

Uno de los aspectos que preocupan mas a las Autoridades es la seguridad de la información, ya que actualmente no se cuenta con la confianza de que en los centros de verificación del Valle de México existan en sus bases de datos información confiable e integra que sirva para que el proceso de verificación sea transparente. Debido a esto y aprovechando las ventajas que nos brinda la plataforma en la cual desarrollaremos la base de datos decidimos integrar diversos mecanismos de control de seguridad tales como el control de usuarios y los permisos relacionados a estos, bitácoras de los movimientos de información dentro de la base de datos y también utilizando métodos como las diferentes integridades en el diseño de nuestra propuesta.

Actualmente la mayoría de las bases de datos de las compañías proveedoras del software de verificación automotriz no cuentan con una base de datos que tenga la mínima posibilidad de ser distribuidas físicamente, es decir que este almacenada en mas de un disco duro, es por esto que diseñaremos una propuesta de Base de Datos que en caso de ser necesario tenga la posibilidad de ser distribuida en su almacenamiento.

Los centros de verificación normalmente y por política de la Secretaria de Ecología realizan una purga o limpieza a las tablas contenidas en sus bases de datos cada

14 meses en algunas tablas y cada 14 días en otras, esto se debe a que en el trasfondo las bases de datos actuales del mercado de verificación están presentando inconsistencias en su desempeño por la baja capacidad de manejo de información, ya que la mayoría de estas no cuentan con un manejador de bases de datos formal.

No existen actualmente respaldos totales ni incrementales automáticos que garanticen la perpetuidad de la información siendo susceptibles a fallas físicas del hardware o intentos deliberados de eliminar o corromper la información.

La distribución y organización de los programas de los proveedores de software de verificación vehicular no cuenta con un esquema unificado entre sus tablas de trabajo y sus tablas de consulta, haciendo con esto que la aplicación que busca información en la base de datos requiere más tiempo en dirigirse a diferentes directorios en el disco duro, realizar la búsqueda de las tablas de consultas y regresar a las tablas de trabajo para insertar dicha información, nuestro diseño promueve la unificación de las tablas de consulta con las de trabajo reduciendo los tiempos en búsquedas de tablas y de información, con esto la aplicación solo le tendrá que hacer una petición al administrador de la base de datos.

Los manejadores, herramientas o programas que hoy en día se utilizan en los centros de verificación para almacenar sus datos carecen ampliamente de lo que llamamos integridades (por entidad, por dominio y por referencia), dando como consecuencia que únicamente se respete la integridad por entidad ya que es la mínima necesaria para efectuar una búsqueda en una tabla, además analizando tablas obtenidas de los centros de verificación encontramos que se generan ciertas inconsistencias en los valores de inicio de registro por ejemplo campos numéricos en donde sus contenidos son blancos en lugar de que formalmente sea un cero, generando errores en los algoritmos de cálculos. Aunado a esto se encontró que no se cumple con la integridad por referencia, ya que existen datos que para existir en una tabla debieron de haber estado dados de alta previamente en otra y que en algunas ocasiones no se cumple esta condición. Uno de los aspectos en los que pondremos mayor énfasis será en este tipo de integridades debido a que nos proporcionarían una confiabilidad tanto en seguridad para la base de datos como coherencia en la información.

Un dato importante que debemos de contemplar es la homologación de los tipos de datos, ya que en las tablas estudiadas anteriormente en el levantamiento de la información pudimos darnos cuenta de que existían tipos de datos diferentes que relacionaban el mismo tipo de información.

Por todo lo anterior es evidente que la Base de Datos resultante podrá satisfacer las necesidades actuales y futuras bajo una arquitectura segura cumpliendo con todos los lineamientos del modelo Entidad-Relación con una óptima normalización.

### 3.2 ESTABLECIMIENTO DE ESTÁNDARES

Para lograr la homologación de los parámetros es necesario establecer un estándar para el nombramiento de cada elemento de la Base de Datos, así como del mismo manejador, es por esto que sería ideal que el manejador de Base de Datos contara con la posibilidad de ser sensible al caso (que diferencie entre letras mayúsculas y minúsculas).

Base de Datos. El nombre de la Base de Datos estará compuesto por tres datos:

1. Leyenda "BD\_Verifcentro"
2. Prefijo de Municipio o Delegación del Centro de Verificación "EC"
3. Numero de Centro de Verificación. En el Distrito Federal son 4 caracteres y en el Estado de México son tres lo cual el cuarto carácter será rellenado con un cero.

Ejemplo: (Nombre Base de Datos Estado de México) "BD\_Verifcentro\_EC\_0920"  
(Nombre Base de Datos Distrito Federal) "BD\_Verifcentro\_IZ\_9420"

Municipio o Delegación	Abreviación
Ecatepec	EC
Iztapalapa	IZ
Nezahualcoyotl	NE
Venustiano Carranza	VC

La Base de Datos al ser generada crea archivos, los cuales deberán ser almacenados en una única ruta a la cual solo tenga acceso el personal que sea asignado por la Secretaria de Ecología, por ejemplo los archivos de respaldo generaos por el gestor de base de datos, los archivos .mdf y .ldf los cuales contienen toda la información de la base de datos, con lo cuales en determinado momento podríamos recuperar la información.

Entidades y Relaciones. Los nombres de las entidades y relaciones (tablas de la BD) se nombraran con letras mayúsculas, en caso de que el nombre este compuesto por mas de una palabra se utilizara como separador el carácter de guión bajo ("\_"), evitando utilizar los espacios en cualquier nombre de Tablas, también se convino que los nombres de las tablas quedaran en forma plural, por ejemplo:

(Nombre simple) "AUTOS "  
(Nombre compuesto) "PRUEBAS\_5024"

Campos. Los nombres de los campos de las tablas se nombraran con letras minúsculas, en caso de que el nombre este compuesto por más de una palabra se

utilizara como separador el carácter de guión bajo (“\_”), evitando utilizar los espacios en cualquier nombre de algún campo, por ejemplo:

(Nombre simple) “fecha “  
 (Nombre compuesto) “clave\_vehiculo“

Funciones. Los nombres de las funciones como Store Procedures (Procedimientos Almacenado), Triggers (Disparadores) y demás funciones que podamos utilizar se nombraran con minúsculas, el nombre estará compuesto o antecedido por dos caracteres que indiquen el tipo de función, seguido de un guión bajo y un nombre que describa su propósito, por ejemplo:

(Nombre Store Procedure) “sp\_respaldo\_diario“  
 (Nombre Trigger) “tg\_altera\_contraseña“

Factor de Relleno (Fill Factor). Se han identificado tablas con una alta demanda de lectura y otras de escritura y es por esto que se define que para toda aquella tabla con alta demanda de escritura se le asignara un FILL FACTOR BAJO en contraste con un FILL FACTOR ALTO que se le asignara a las tablas que tengan alta demanda en lectura de datos. En especial este lineamiento tomado puede ser modificado en cada caso según las necesidades de lectura-escritura de cada tabla. Por ejemplo en nuestro pre-prototipo de base de datos tenemos las algunas tablas identificadas por su alta demanda de lectura o escritura, que se muestran a continuación:

<b><u>Nombre de la Tabla</u></b>	<b><u>Lectura</u></b>	<b><u>Escritura</u></b>	<b><u>Fill Factor</u></b>
LABORATORIOS	X		ALTO
CALIBRACIONES LABORATORIO		X	BAJO
TECNICOS_LAB	X		ALTO
PROVEEDORES	X		ALTO
LINEAS	X		ALTO
CALIBRACIONES_FUGAS		X	BAJO
DINAMÓMETROS	X		ALTO
VERIFICACIONES		X	BAJO
CORRECCIONES	X	X	MEDIO
ESTACIONES	X		ALTO
CALIBRACIONES_DINAMÓMETRO		X	BAJO
CALIBRACIONES _ BANCO		X	BAJO
FALLAS _ CATALOGO	X		ALTO
PRUEBAS_GAS_ALEATORIAS		X	BAJO
ESTADOS	X		ALTO

MUNICIPIOS	X		ALTO
CENTROS	X		ALTO
SEGUNDOS_X_SEGUNDOS		X	BAJO
CATALOGO_MOVIMIENTOS	X		ALTO
MOVIMIENTOS		X	BAJO
PERSONAL	X		ALTO
PUESTOS	X		ALTO
PERMISOS	X		ALTO
VERSIONES_SOFTWARE	X		ALTO
ACCIONES		X	BAJO
CONTRASEÑAS		X	BAJO
PERSONAL	X		ALTO
USUARIOS	X		ALTO
DATOS_PROPIETARIO		X	BAJO
IMPRESIONES		X	BAJO
PRUEBA_2540		X	BAJO
PRUEBA_5024		X	BAJO
ACCESOS		X	BAJO
CLASES_VEHICULOS	X		ALTO
RECHAZOS		X	BAJO
LIMITES_GASES	X		ALTO
CERTIFICADOS_ROBADOS	X		ALTO
DATOS_CIRCULACION_PAGOS		X	BAJO
SERVICIOS	X		ALTO
CARGA	X		ALTO
ALIMENTACIÓN_COMBUSTIBLES	X		ALTO
CARROCERÍA	X		ALTO
DATOS_VEHICULO		X	BAJO
T_MAESTRA	X		ALTA
SUBMARCAS	X		ALTA
MARCAS	X		ALTA

Valores Default de los campos. Se especifica que si algún campo no es llenado con valores de su tipo deberá ser llenado con valores predefinidos:

Campos Numéricos: Valor "0" o "0.0" según sea entero o flotante

Campos Alfanuméricos: Valor " " (espacios)

Campos de Fechas: NOT NULL

Campos Lógicos: FALSE

Campos Binarios: Valor "0"

Nota.- Cuando sea necesario utilizar campos de tipo alfanumérico se utilizarán formatos “varchar” para evitar el desperdicio de la memoria.

Usuarios. El nombre de los usuarios estará compuesto por el prefijo “usr” más el número del centro de verificación, el nombre de la persona (letra inicial del nombre y apellido) y terminado con el número de empleado del centro de verificación (en caso de que sea alguna persona externa el Centro de verificación asignara un numero), todo esto será con letras minúsculas y separado por el carácter guión bajo. Por ejemplo:

(Nombre de Usuario) “usr\_935\_rmorales\_0001”

Respaldos: Los diferentes respaldos que se generen tendrán un nombre estandarizado que indicara que se trata de un respaldo incremental o total, el día que se realizo así como la hora. Por ejemplo:

Formato: ddmmaaaa\_hhmmss  
(Nombre respaldo total) “RT\_01012005\_120001”  
(Nombre respaldo incremental) “RI\_01012005\_120001”

### 3.3 NIVELES DE ESTUDIO DE LA BASE DE DATOS

Para concebir cualquier base de datos es necesario estudiar las reglas negocio en diferentes niveles tales como:

**Nivel Físico.** En este nivel se estudia como estarán guardados los datos para lo cual solo podrá ser realizado por personas calificadas.

Se tienen visualizadas ciertas etapas en el proceso de verificación, en cada una de estas hay información que será desglosada y almacenada en diferentes tablas correspondientes a su etapa.

Raúl	Pérez	Col. Anahuac	Num. 42	CP. 78900	D.F.	HC 50	CO 1.0	C02 10.5	C02 10.5	O2 0.6	NOX 1200
------	-------	--------------	------------	--------------	------	----------	-----------	-------------	-------------	-----------	----------

Como esta estructurado en las bases de datos de los verificentros actualmente la información anterior que es de diferente tipo (conceptualmente hablando) se encuentran almacenados en la misma tabla, por concerniente no se tiene una base de datos normalizada, esto conlleva a que el lugar físico de almacenamiento de esta información no sea el adecuado. Esto evita que bajo este esquema se este cuidando la integridad referencial, para evitar esto dentro de diseño que se



esta generando se podrá atención en la ubicación tanto lógica como física del almacenamiento de la información como se muestra a continuación:

Propietario	Raúl	Pérez	Col. Anahuac	Num. 42	CP. 78900	D.F.
-------------	------	-------	--------------	---------	-----------	------

Prueba de 24 Kms	Hc 50	Co 1.0	CO2 10.5	O2 0.6	NOX 1200
------------------	-------	--------	----------	--------	----------

**Nivel Conceptual.** La Base de Datos se admira desde una forma de mando sin importan como están almacenados los datos.

En este tipo de nivel sabemos que hay diferentes procesos en la verificación de un automóvil, es aquí donde ya identificamos diferentes entidades pero sin importar como estarán almacenados al nivel de datos o bits inclusive discriminando la tecnología en la que nos vamos a basar, solo se identifican los diferentes objetos participantes, por ejemplo:

Captura datos Verificación  $\longrightarrow$  Vehículo, propietario, marcas, etc.

**Nivel Visión.** Es la división parcial para tomar las decisiones de que información puede ser mostrada a cada persona.

Usuario Técnico Verificador

Nivel Visión :

Placas	Modelo	Marca	Submarca	Tipo Certificado Obtenido
--------	--------	-------	----------	---------------------------

Usuario Administrador Verificentro

Nivel Visión:

Placas	Modelo	Marca	Submarca	Tipo Certificado Obtenido
HC	CO2	NOX	CO	O2
Tiempo Prueba	Cve Tecnico	Tipo Multa	Causa Rechazo	Técnico que laboro la prueba

### 3.4 VISUALIZACIÓN DE ENTIDADES

Partiendo del levantamiento de la información y de las tablas anteriores ocupadas hoy en día, hemos visualizado diferentes entidades que cumplen con las necesidades del Programa de Verificación y con la seguridad necesaria.

<b><u>Nombre de la Tabla</u></b>	<b><u>Uso General de la Tabla</u></b>
LABORATORIOS	Datos descriptivos de los Laboratorios de Calibración
CALIBRACIONES LABORATORIO	Datos de las Pruebas de Calibración que realiza el Laboratorio
TECNICOS_LAB	Información acerca de los Empleados de los Laboratorios de Calibración
PROVEEDORES	Información descriptiva sobre el proveedor del software de Verificación Automotriz
LINEAS	Información sobre cada línea de verificación, números de serie, etc.
CALIBRACIONES_FUGAS	Información sobre la prueba de 24 horas sobre fugas del sistema.
DINAMÓMETROS	Información descriptiva sobre el Dinamómetro de cada línea de Verificación
VERIFICACIONES	Datos de los vehículos verificados.
CORRECCIONES	Bitácora de las correcciones al sistema en caso de alguna falla.
ESTACIONES	Catálogo de los tipos de maquinaria que operan en el sistema.
CALIBRACIONES_DINAMÓMETRO	Información acerca de la prueba de 24 horas del Dinamómetro
CALIBRACIONES _ BANCO	Información acerca de la prueba de 24 horas del Banco de Emisiones
FALLAS _ CATALOGO	Catálogo de las fallas que se van suscitando a lo largo de la vida del equipo de verificación.
PRUEBAS_GAS_ALEATORIAS	Información sobre las pruebas de Calcom aleatorias
ESTADOS	Catálogo de los Estados de la República
MUNICIPIOS	Catálogo de los Municipios por cada Estado de la República
CENTROS	Información descriptiva del Centro de Verificación
SEGUNDOS_X_SEGUNDOS	Información de las pruebas Dinámicas y sus lecturas de los diferentes gases, transcurriendo segundo a segundo
CATALOGO_MOVIMIENTOS	Catálogo de los diferentes movimientos que puede haber dentro de la BD.
MOVIMIENTOS	Bitácora de los movimientos
PERSONAL	Información del personal del Centro de Verificación
PUESTOS	Información de las jerarquías del sistema
PERMISOS	Catálogo del tipo de permisos asociados a los puestos
VERSIONES_SOFTWARE	Información sobre las versiones de software del programa de Verificación.
ACCIONES	Información sobre los eventos realizados por el personal del Verificentro
CONTRASEÑAS	Password de los usuarios del Sistema

PERSONAL	Datos de los Usuarios del Sistema
USUARIOS	Catálogo de los usuarios del Sistema
DATOS_PROPIETARIO	Datos acerca del propietario del vehiculo a verificar.
IMPRESIONES	Información sobre las impresiones realizadas a causa de las verificaciones.
PRUEBA_2540	Datos de la prueba dinámica a 40 kilómetros de velocidad
PRUEBA_5024	Datos de la prueba dinámica a 24 kilómetros de velocidad
ACCESOS	Accesos del personal al sistema
CLASES_VEHICULOS	Tipo *****
RECHAZOS	Catalogo de los tipos de rechazos que pueden existir
LIMITES_GASES	Limites de los maximos niveles de emisiones de gases para los vehiculos a verificar.
CERTIFICADOS_ROBADOS	Folios de los certificados de verificación que han sido robados y que no se deben de reutilizar ilegalmente
DATOS_CIRCULACION_PAGOS	Información sobre la documentación del vehiculo a verificar.
SERVICIOS	Catálogo de los tipos de modos de utilización del vehiculo.
CARGA	Catálogo del tipo de carga de frenado que se debe utilizar para el tipo de vehiculo a verificar.
ALIMENTACIÓN_COMBUSTIBLES	Catálogo del tipo de tecnología para alimentación de gasolina.
CARROCERÍA	Catálogo del tipo de carrocería al que pertenece el vehiculo.
DATOS_VEHICULO	Datos generales del vehiculo a verificar
T_MAESTRA	Tabla que contiene todas las combinaciones posibles de los parámetros de prueba para cada vehiculo
SUBMARCAS	Catálogo de las submarcas de las marcas registradas.
MARCAS	Catalogo de las marcas de vehículos registradas.

### 3.5 MODELO ENTIDAD-RELACIÓN.

El Modelo E/ R es propuesto entre 1976 y 1977 por Peter Chen, donde se busca resolver la problemática de que ningún modelo tiene la suficiente capacidad de abstraer la semántica del mundo real, donde visualizamos un mundo real con entidades y relaciones entre ellas.

El modelo E/ R permite al diseñador concebir la base de datos ubicándose en un nivel superior de abstracción, aislando temas y características relativas a la tecnología, colocándolo en un plano "Infológico" debido a el enfoque de

abstracción “Entidad” y sus “Relaciones”, donde la entidad es algo claramente identificable, y la interrelación es una vinculación paramétrica entre las entidades.

Según el modelo, tenemos los siguientes elementos:

- A) Entidad** (Entity Set o Entity Type)
- B) Interrelación**
- C) Dominio** (Value Set)
- D) Atributo**

**Entidad:** A la entidad la podemos definir como cualquier objeto (este puede ser real o abstracto) que en la realidad exista y el cual queramos representar por nuestro esquema mediante el modelo E/R.

Tenemos dos tipos de entidades:

**Regulares:** Sus ejemplares tienen existencia por si mismos

**Débiles:** La existencia de sus ejemplares depende de la existencia de otros ejemplares

La definición no es tan sencilla como lo parece, ya que en ocasiones un objeto real se puede considerar como una entidad, o como una propiedad o atributo de una entidad, o una interrelación entre entidades.

**Interrelación:** Una interrelación es un vínculo, asociación o correspondencia entre entidades.

**Dominio y Valor:** Las propiedades de un tipo de entidad toman valores para cada ejemplar. Este conjunto de posibles valores se denomina **dominio**. El valor debe cumplir el predicado asociado al tipo de entidad.

**Atributo:** Cada característica de un tipo de entidad se conoce como atributo. Estos pueden tomar valores de uno o de varios dominios.

Los Atributos no tienen vida propia como los dominios. La existencia de los atributos esta vinculada o condicionada a su correspondiente tipo de entidad.

### 3.5.1 RESTRICCIONES EN EL MODELO ENTIDAD-RELACIÓN

Una relación inherente al modelo es que solo permite colocar interrelaciones entre entidades, no estando permitidas el resto de las combinaciones. A continuación describimos las restricciones

#### **Inherentes al modelo**

Solo permite relaciones entre entidades  
Cada entidad debe tener un identificador

## **Integridad**

*Dominio:* Permite limitar los valores del dominio y los valores de los atributos por consecuencia, comprendiendo estos datos en un rango determinado

*Atributos e Interrelaciones:* Los atributos de una entidad identifican unívocamente a cada ejemplar de ese tipo de entidad, por lo que se le llama *Identificador Candidato (IC)*.

**Restricciones de Exclusividad:** Se dice que dos o más interrelaciones tienen una restricción de exclusividad con respecto a un tipo de entidad que participa en ambas interrelaciones cuando cada ejemplar de cada tipo de entidad sólo puede pertenecer a uno de los tipos de interrelación, pero en el momento que pertenezca a uno, ya no podrá pertenecer a otro.

**Restricción de Exclusión:** Se dice que dos o más interrelaciones tienen una restricción de exclusión con respecto a un tipo de entidad que participa en ambas interrelaciones cuando cada ejemplar de cada tipo de entidad puede pertenecer a más de uno de los tipos de interrelación, pertenecerá a uno y para pertenecer a otro tendrá que dejar de pertenecer al primero.

**Restricciones de Inclusividad:** Se dice que dos o más interrelaciones tienen una restricción de inclusividad con respecto a un tipo de entidad que participa en ambas interrelaciones cuando cada ejemplar de cada tipo de entidad debió de pertenecer a la otra entidad en un tiempo pasado.

**Restricción de Inclusión:** Se dice que dos o más interrelaciones tienen una restricción de inclusión con respecto a un tipo de entidad que participa en ambas interrelaciones cuando cada ejemplar de cada tipo de entidad debió de pertenecer a la otra entidad en un tiempo pasado y en un orden lógico establecido.

### **3.5.2. INTEGRIDADES**

Es la habilidad de una Base de Datos para proteger sus distintos componentes (datos) contra los accesos o modificaciones no autorizados. Existen tres tipos de integridades que son necesarias en el diseño y desarrollo de la base de datos, ya que estas nos garantizaran la confiabilidad de los datos residentes.

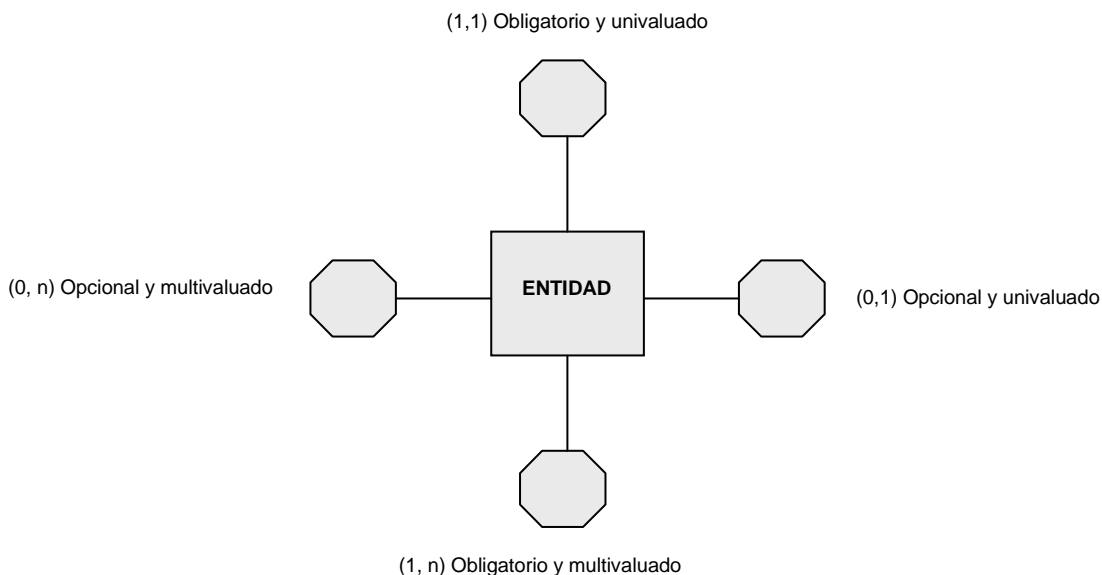
Integridad por Entidad. Los atributos secundarios de una entidad dependen directamente de los atributos primarios, es decir, que todas las tablas deben contar con al menos una llave, comúnmente llamada llave primaria, puede estar compuesta por uno o más campos.

Integridad por Dominio. Para cada atributo debe de haber un valor llamado base o default dependiendo del tipo de dato, dichos valores en la mayoría de los casos estarán altamente ligados con las reglas del negocio que le darán forma a nuestra futura base.

Integridad por Referencia. Para que pueda existir un dato en una tabla previamente deberá de existir en otra, siempre y cuando se cuente con una relación de información.

### 3.5.3 CARDINALIDAD

La cardinalidad mínima y máxima se refiere a los valores mínimos o máximos que puede tomar un atributo para cada ejemplar de un tipo de entidad a la cual pertenece. Las cardinalidades se representan asociando un par de números enteros (mínimo, máximo) al atributo. En la Figura 3.1 se muestra las diversas opciones de cardinalidad.



**Figura 3.1 Representación de la Cardinalidad.**

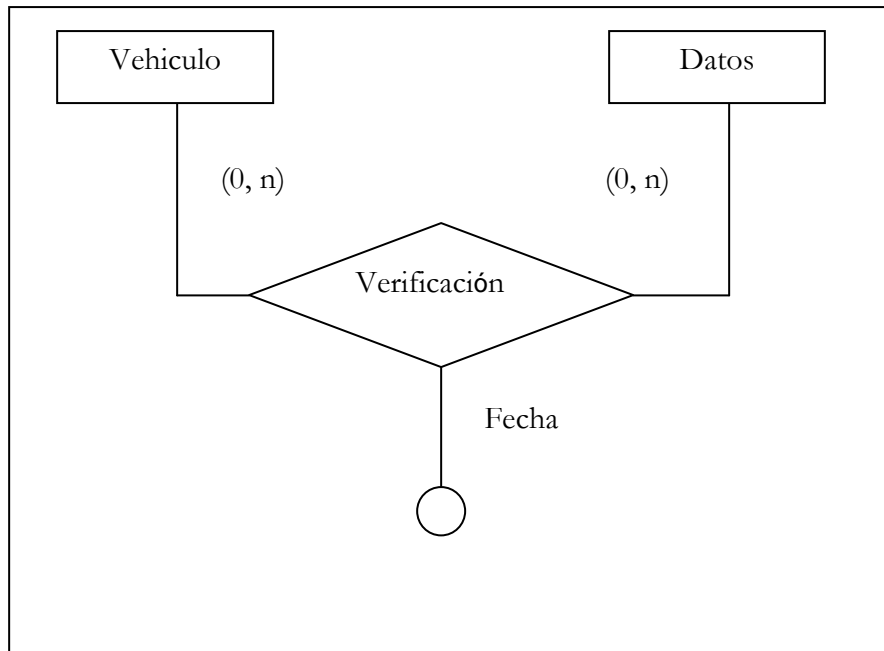


Figura 3.2 Ejemplo de Cardinalidad.

### 3.6 NORMALIZACIÓN

En esta teoría se consigue una formalización del diseño lógico de las bases de datos relacionales, lo que permite disponer de instrumentos algorítmicos de ayuda al diseñar, de esta forma podemos desarrollar programas que automaticen el diseño en el modelo relacional.

La información de nuestra base de datos puede representarse por medio de un conjunto de objetos (relaciones y dominios) y de un conjunto de reglas de integridad.

En el modelo relacional, como en todos los demás modelos de datos, el diseño de una base de datos se puede abordar de dos formas distintas:

Obteniendo el esquema relacional directamente a partir de la observación de nuestro universo de discurso, de forma que plasmemos nuestra percepción en un conjunto de esquemas de relación, los cuales contendrán los atributos y restricciones de integridad que representan los objetos y reglas que hemos podido captar en nuestro análisis del mundo real.

Realizando el proceso de diseño en dos fases. En la primera se lleva a cabo el diseño conceptual, por ejemplo, en el modelo E/ R, obteniendo el correspondiente

esquema conceptual; en la segunda, este se transforma en un esquema relacional, siguiendo unas determinadas reglas de transformación.

Estas relaciones que resultan de la observación del mundo real o de la transformación del esquema E/ R al modelo relacional, elaborado en la etapa de modelado conceptual, pueden presentar algunos problemas, derivados de fallos en la percepción del UD, en el diseño del esquema E/R o en el paso al modelo relacional; entre estos problemas tenemos:

- Incapacidad para almacenar ciertos hechos.
- Redundancias y, por tanto, posibilidad de inconsistencias.
- Ambigüedades.
- Pérdida de información (aparición de tuplas espurias).
- Pérdida de dependencias funcionales, es decir, de ciertas restricciones de integridad que dan lugar a interdependencias entre los datos
- Existencia de valores nulos (inaplicables)
- Aparición, en la base de datos, de estados que no son válidos en el mundo real (anomalías de inserción, borrado y modificación).

El esquema relacional debe ser siempre analizado para comprobar que no presenta los problemas mencionados anteriormente, evitando la pérdida de información y la aparición de estados que no son válidos en el mundo real.

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

La primera forma normal (1FN) es una restricción inherente al modelo relacional, por lo que su cumplimiento es obligatorio y afecta al número de valores que pueden tomar los atributos en una relación.

Para que una tabla pueda ser considerada una relación no debe admitir grupos repetidos, esto es, debe estar en la **primera forma normal**. Como una definición formal, podríamos decir que una relación está en 1FN cuando cada atributo sólo toma un valor del dominio simple subyacente.

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

La segunda forma normal (2FN) está basada en el concepto de dependencia plena y en las interrelaciones existentes entre los atributos principales (que se encuentran en alguna de las llaves) y no principales (que no forman parte de la llave) de una relación.

Se dice que una relación está en 2FN si:

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

La segunda forma normal no se cumple cuando algún atributo no principal depende funcionalmente de algún subconjunto de la clave. *Entonces la regla 2FN es, los atributos no principales deben depender funcionalmente de los atributos principales.*



Se puede afirmar que cualquier relación binaria se encuentra siempre en 2FN; así como también esta en 2FN cualquier relación en la que todas las llaves son simples, es decir, contienen un sólo atributo. Asimismo esta en 2FN cualquier relación en la que todos sus atributos son principales (forman parte de la llave). Siempre es posible transformar un esquema de relación que no este en 2FN en esquemas de relación 2FN, sin que se produzca pérdida de información, ni de dependencias.

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

La tercera forma normal (3FN) está basada en el concepto de dependencia transitiva.

Un esquema de relación R esta en tercera forma normal si y solo si:

- Esta en 2FN
- No existe ningún atributo no principal que dependa transitivamente de alguna llave de R.

La tercera forma normal no se cumple cuando existen atributos no principales que dependen funcionalmente de otros atributos no principales. *Entonces la regla 3FN es, los atributos no principales no deben depender funcionalmente de otros atributos no principales.*

Se puede afirmar que toda relación binaria se encuentra automáticamente en 3FN, así como toda relación cuyos atributos son todos principales, o bien cuando hay un único atributo no principal.

## **3.7 SELECCIÓN DE LA PLATAFORMA TECNOLÓGICA Y METODOLOGÍA.**

En el diseño de una Base de Datos es conveniente distinguir la fase de modelado conceptual que es la descripción del mundo real, según un modelo conceptual. Este modelo conceptual debe ser altamente semántico e independiente de alguna tecnología de Bases de Datos. A veces se prescinde de la etapa de modelado conceptual y del diseñador, sin una metodología precisa, siendo así, entonces los datos estarían carentes de sentido y significado.

### **¿Porque usar un Sistema Gestor de Base de Datos o “DataBase Management System”?**

Una base de datos es similar a un archivo de datos en que es también un lugar de almacenamiento de datos. Como en un archivo de datos (Hoja de Cálculo de Excel), una base de datos no presenta información directamente al usuario; el usuario ejecuta una aplicación que tiene acceso a la información de la base de datos y los presenta al usuario en un formato inteligible y sencillo.

Los sistemas de bases de datos (DataBase Management System) son más eficaces que los archivos de datos (Excel), ya que los datos están mejor

organizados. En una base de datos bien diseñada, no hay elementos de datos duplicados que el usuario o la aplicación deban actualizar al mismo tiempo. Los elementos de datos están agrupados en una única estructura o registro, y se pueden definir relaciones entre dichas estructuras y registros. Cuando se trabaja con archivos de datos, las aplicaciones se tienen que codificar para trabajar con la estructura específica de cada archivo de datos.

Generalmente, una base de datos tiene dos partes principales: los archivos que almacenan la base de datos física y el software del sistema de administración de la base de datos (DBMS, *Database Management System*), que las aplicaciones utilizan para tener acceso a los datos. El DBMS es el responsable de mantener la estructura de la base de datos, lo que incluye:

El mantenimiento de las relaciones entre los datos de la base de datos. La garantía de que los datos estén correctamente almacenados y de que no se infrinjan las reglas que definen las relaciones entre los datos. La recuperación de todos los datos hasta un punto coherente en caso de errores del sistema.

La base de datos es una colección de archivos interrelacionados almacenados en conjunto sin redundancia (repetición) y la DBMS es un conjunto de numerosas rutinas de software interrelacionadas, cada una de ellas es responsable de una determinada tarea.

La falta de un buen diseño es una razón importante por lo que el diseño evolutivo falla y los sistemas carecen de calidad, en consecuencia son deficientes.

## **Microsoft SQL Server 2000**

SQL Server 2000 incluye herramientas para extraer y analizar datos de resumen para el procesamiento analítico en línea. SQL Server incluye también herramientas para diseñar gráficamente las bases de datos y analizar los datos mediante preguntas en inglés.

Existen varios manejadores de Bases de Datos en el Mercado, algunos bastante eficientes capaces de administrar, almacenar, organizar y soportar grandes cantidades de información, sin embargo hay otros los cuales son bastante deficientes no cumpliendo con reglas ni estándares de una Base de Datos formal y es más, no se pueden llamar en ocasiones Sistemas Gestores de Base de Datos.

Hoy en día debemos elegir una tecnología para desarrollar nuestra Base de Datos dependiendo de varias situaciones que se adecuen al entorno para el cual serán diseñadas (Reglas de Negocio), no podemos simplemente decidir utilizar una tecnología por lo pequeño o grande que sea la cantidad de información, es

seguro que si elegimos una tecnología por este aspecto nuestro proyecto se enfrentara a grandes problemas en etapas posteriores.

Mencionaremos algunos aspectos importantes con los cuales cuenta un Sistema Gestor de Base de Datos en este caso SQL Server 2000.

1. SQL Server nos permite tener un nivel que se llama Isolation Level (Aislamiento), con esto quiere decir que no todos los usuarios podrán leer ni modificar los datos que quieran, sino que cada usuario tendrá los permisos necesarios para operar con la información que le corresponde sin alterar datos ajenos.
2. Cuenta con el Método de Paginación el cual permite que la base de Datos no crezca excesivamente en volumen (Megabytes) utilizando o terminando el espacio de los discos duros en menor tiempo que por ejemplo Programas como Access o Excel.
3. Cuenta con un método llamado PAD\_INDEX lo cual permite generar índices (herramientas de búsqueda de información) que permite encontrar y visualizar la información en tiempos rápidos.
4. Fill Factor, permite que cada vez que un registro nuevo se inserte, se ordene de forma que optimice la búsqueda de este para hacer más rápidas las consultas de información.
5. Permite crear procedimientos (herramientas) para el manejo de la información que serán automáticos y programados sin tener que hacerlos manualmente.
6. SQL Server permite almacenar cantidades enormes de información (registros) sin sufrir alguna deficiencia en su funcionamiento, por ejemplo el Manejador Access solamente soporta 20,000 a 30,000 registros ocasionando posteriormente deficiencias en los resultados de sus procesos de información.
7. La utilización de Integridad (Restricciones dentro de las Tablas) permite que en SQL Server se utilicen campos llave ocasionando que no se duplique la información (tuplas espurias) ocasionando falta de confiabilidad en la información

SQL Server 2000 está diseñado para admitir el tráfico de los sitios Web de mayor tamaño o sistemas de procesamiento de datos empresariales. Las instancias de SQL Server 2000 que se ejecutan en servidores multiprocesador de gran tamaño pueden admitir conexiones de miles de usuarios al mismo tiempo. Las características de escalabilidad y facilidad de uso de SQL Server 2000 permiten trabajar eficazmente en un único equipo sin consumir demasiados recursos y sin que sean necesarias tareas administrativas por parte del usuario independiente. Las mismas características permiten a SQL Server 2000 adquirir de forma dinámica los recursos necesarios para admitir miles de usuarios, al tiempo que minimizan la administración y la optimización de bases de datos.

## Ventajas de un Sistema de Base de Datos de Servidor

El tener los datos almacenados y administrados en una ubicación central ofrece varias ventajas:

Todos los elementos de datos están almacenados en una ubicación central donde todos los usuarios pueden trabajar con ellos. No se almacenan copias **separadas** del elemento en cada cliente, lo que elimina los problemas de hacer que todos los usuarios trabajen con la misma información. El sistema no necesita garantizar que todas las copias de los datos se actualizan con los valores actuales, porque hay una única copia en la ubicación central. Las reglas de empresa y de seguridad se pueden definir una sola vez en el servidor para todos los usuarios.

La exigencia de las reglas se puede llevar a cabo en una base de datos utilizando restricciones, procedimientos almacenados y desencadenadores. Las reglas se pueden exigir también en una aplicación de servidor, puesto que estas aplicaciones son, asimismo, recursos centrales a los que tienen acceso muchos clientes reducidos. Los servidores de base de datos relacionales optimizan el tráfico de la red al devolver sólo los datos que la aplicación necesita.

### Comparación de Gestores de Bases de Datos y Otros.

	EXCEL	ACCESS	SQL SERVER	ORACLE	DB2
Uso de Interidad			x	x	X
Factor de Relleno (Fill Factor)			x	x	X
Indexación Inteligente			x	x	X
Nivel de Aislamiento			x	x	X
Trabajo en Red			x	x	x
Seguridad			x	x	X
Duplicación de Datos	X	x			
Costo	Incluido en Office	Incluido en Office	Mediano Costo	Alto Costo	Alto Costo

Como podemos ver las aplicaciones como Excel y Acces en sus distintas versiones no aplican debido a que no son Gestores de Bases de Datos por lo cual no brindan las herramientas necesarias para lograr el control y la eficiencia en la base de datos a diseñar, en tanto los sistemas Gestores como Oracle de la compañía del mismo nombre, DB2 de IBM, SQL Server 2000 de Microsoft proporcionan todas las facilidades para tener en control la información contenida en una base de datos, sin embargo los dos primeros quedan descartados debido a su alto costo por lo cual sería difícil que todas los verificadores del Valle de México estuvieran dispuestos a pagarlo, ya que en ocasiones este tipo de licencias corren a cuenta totalmente de las ganancias de los centros de verificación.

### **3.8 PLANTEAMIENTO DE SEGURIDAD**

En lo que respeta al ámbito de seguridad y conociendo el ambiente donde se podría utilizar este nuevo prototipo de base de datos, es necesario enfocarnos más a la seguridad del control de accesos y al registro detallado de cada movimiento que ejecutan los diferentes usuarios con sus debidos privilegios.

Para esto se diseñó una estructura de base de datos donde se tenga el control de:

- Usuarios. Dependiendo del puesto y las funciones de cada personal del verificentro se han creado diferentes tipos de usuarios (diferentes tipos de permisos) que servirán para entrar a la aplicación que se conectara con esta base de datos, por medio de la contraseña privada de cada usuario se garantiza que la persona que se esta autenticando en el sistema sea quien dice ser.
- Acciones. Las posibles acciones a realizar en el sistema por los usuarios que dependerán de su tipo de jerarquía serán almacenadas detalladamente durante todos los lapsos de trabajo para tener la información de cuando, como, que y porque realizo esa acción en el sistema. NOTA: Cabe aclarar que estas acciones son las realizadas solamente en los menús de la aplicación que trabajara con la base de datos.
- Registro de movimientos. Aquí registraremos todos aquellos movimientos que tengan que ver con la manipulación de los datos.
- Respaldos. Como se analizó en capítulo dos es necesario tener implantado un mecanismo de respaldos automatizados en determinados horarios del día para asegurar la permanencia de la información y cumplir con algunas de las normas establecidas por las Autoridades de Gobierno.

### **3.9 REQUERIMIENTOS MÍNIMOS DE HARDWARE**

Debido a que no tenemos ingerencia sobre la adquisición y la compra de los diferentes equipos de Hardware para realizar las operaciones diarias de los diferentes procesos en el centro de verificación usaremos los equipos con los que actualmente cuentan los centros de verificación en el año 2005, cabe destacar que el diseño de esta base de datos no esta íntimamente ligado al hardware, estos equipos se describen a continuación:

Los equipos de los centros de computo varían en cada uno de estos, pero siempre cumpliendo con los requerimientos mínimos de las especificaciones que marca la ley y que son los suficientemente capaces de llevar a cabo el trabajo diario de la forma mas eficiente posible, entre algunas características generales que podremos nombrar son que tendremos un servidor central para la base de datos con disco duro suficiente para el almacenamiento de la base de datos con un sistema de discos raid 1 (espejo), con un procesador Xeon de aproximadamente

entre 2.5 y 3 Ghz de velocidad, la suficiente memoria ram, ancho de banda en tarjetas de red para soportar cargas robustas de información tomando como referencia la cantidad de trabajo que tiene los centros de verificación, así también con eficientes periféricos para la extracción e inserción por medio de estos.

### **3.10 REQUERIMIENTOS DE SOFTWARE**

Al igual que el punto anterior, tampoco tenemos ingerencia sobre el sistema operativo que los centros de verificación deben de elegir y esta es una razón más por la cual decidimos optar por la tecnología de SQL SERVER 2000 ya que es totalmente compatible con los sistemas operativos que hoy en día de utilizan en los centros de verificación como son Windows 2000 y 2003 Server en sus diferentes versiones.

Microsoft® SQL Server™ 2000 está disponible en las ediciones siguientes:

SQL Server 2000 Enterprise. Se utiliza como un servidor de base de datos de producción. Admite todas las características que están disponibles en SQL Server 2000 y es ampliable a los niveles de rendimiento que se requieren para ser compatible con los sitios Web más grandes, y el procesamiento de transacciones en línea (OLTP) y los sistemas de almacenamiento de datos corporativos.

SQL Server 2000 Standard. Se utiliza como un servidor de base de datos para un pequeño grupo de trabajo o departamento.

SQL Server 2000 Personal. La utilizan los usuarios móviles que están durante algún tiempo desconectados de la red, pero ejecutan aplicaciones que requieren un almacén de datos SQL Server. También se utiliza durante la ejecución de una aplicación independiente que requiere un almacén de datos.

SQL Server local en un equipo cliente.

SQL Server 2000 Developer. La utilizan los programadores que desarrollan aplicaciones que utilizan SQL Server 2000 como su almacén de datos. Aunque SQL Server 2000 Developer admite todas las características de SQL Server 2000 Enterprise, que permite a los programadores escribir y probar aplicaciones que pueden utilizar estas características, se autoriza el uso de SQL Server 2000 Developer sólo como un sistema de desarrollo y prueba, no como un servidor de producción.

SQL Server 2000 para Windows CE. Microsoft® SQL Server 2000™ para Windows® CE (SQL Server CE) se utiliza como almacén de datos en los dispositivos Windows CE. Capaz de duplicar datos con cualquier versión de SQL Server 2000 para mantener los datos de Windows CE sincronizados con la base de datos principal.

SQL Server 2000 Enterprise Evaluation. Versión con todas las características que está disponible para su descarga gratuita desde el Web. Está destinada sólo para utilizarla en la evaluación de características de SQL Server; esta versión dejará de funcionar cuando hayan transcurrido 120 días desde su descarga. Estas versiones de SQL Server 2000 también incluyen un componente que permite a los desarrolladores de aplicaciones distribuir una copia del motor de la base de datos relacional con sus aplicaciones. Este componente tiene dos nombres:

- SQL Server 2000 Desktop Engine
- Microsoft Data Engine (MSDE) 2000

En los Libros en pantalla de SQL Server se hace referencia al componente como Desktop Engine. La información acerca de Desktop Engine también es aplicable a MSDE 2000.

Aunque la funcionalidad del motor de base de datos de SQL Server 2000 Desktop Engine es parecida a la del motor de las otras versiones de SQL Server, el tamaño de sus bases de datos no puede sobrepasar los 2 GB.

Tanto SQL Server 2000 Personal como SQL Server 2000 Desktop Engine disponen de un regulador de carga de trabajo simultáneo que limita el rendimiento del motor de la base de datos si se ejecutan más de 5 procesos por lote al mismo tiempo. Para obtener más información acerca del regulador de carga de trabajo simultáneo, consulte Bases de datos de SQL Server 2000 en equipos de escritorio.

### Sistemas operativos compatibles con las ediciones de SQL Server 2000

En esta tabla se muestran los sistemas operativos compatibles con la ejecución del software de servidor desde cada edición de Microsoft® SQL Server™ 2000.

Sistema operativo	Enterprise	Standard	Personal	Developer	Desktop Engine	SQL Server CE	Enterprise Evaluation
Microsoft Windows® 2000 DataCenter	Compatible	Compatible	Compatible	Compatible	Compatible	N/D	Compatible
Windows 2000 Advanced Server	Compatible	Compatible	Compatible	Compatible	Compatible	N/D	Compatible
Windows 2000 Server	Compatible	Compatible	Compatible	Compatible	Compatible	N/D	Compatible
Windows 2000 Professional	N/D	N/D	Compatible	Compatible	Compatible	N/D	Compatible
Microsoft Windows NT® 4.0 Server Enterprise	Compatible	Compatible	Compatible	Compatible	Compatible	N/D	Compatible
Windows NT 4.0	Compatible	Compatible	Compatible	Compatible	Compatible	N/D	Compatible

Server							
Windows NT 4.0 Workstation	N/D	N/D	Compatible	Compatible	Compatible	N/D	Compatible
Microsoft Windows 98	N/D	N/D	Compatible	N/D	Compatible	N/D	N/D
Microsoft Windows CE	N/D	N/D	N/D	N/D	N/D	Compatible	N/D

### 3.11 CARACTERÍSTICAS DEL MOTOR DE BASE DE DATOS COMPATIBLES CON LAS EDICIONES DE SQL SERVER 2000

En esta tabla se muestran las características del motor de base de datos y las ediciones de SQL Server 2000 que las admiten.

Característica del motor de base de datos	Enterprise	Standard	Personal	Developer	Desktop Engine	SQL Server CE	Enterprise Evaluation
Compatibilidad con múltiples instancias	Compatible	Compatible	Compatible	Compatible	Compatible	N/D	Compatible
Clúster de conmutación por error (hasta cuatro nodos)	Compatible	N/D	N/D	Compatible	N/D	N/D	Compatible
Compatibilidad con conmutación por fallo en el Administrador de SQL Server Enterprise	Compatible	N/D	N/D	Compatible	N/D	N/D	Compatible
Trasvase de registros	Compatible	N/D	N/D	Compatible	N/D	N/D	Compatible
DBCC en paralelo	Compatible	N/D	N/D	Compatible	N/D	N/D	Compatible
CREATE INDEX en paralelo	Compatible	N/D	N/D	Compatible	N/D	N/D	Compatible
Lectura anticipada y recorrido mejorados	Compatible	N/D	N/D	Compatible	N/D	N/D	Compatible
Vistas indizadas	Compatible	N/D	N/D	Compatible	N/D	N/D	Compatible
Servidor de base de datos federado	Compatible	N/D	N/D	Compatible	N/D	N/D	Compatible



Compatibilidad con la Red de área del sistema (SAN)	Compatible	N/D	N/D	Compatible	N/D	N/D	Compatible
Herramientas de desarrollo y DBA gráficas, asistentes	Compatible	Compatible	Compatible	Compatible	N/D	N/D	Compatible
Compatibilidad con utilidades gráficas para la configuración de idioma	Compatible	N/D	N/D	N/D	N/D	N/D	N/D
Búsqueda de texto	Compatible	Compatible	Compatible, excepto con Windows 98	Compatible	N/D	N/D	Compatible
SQL Mail	Compatible	Compatible	Compatible	Compatible	N/D	N/D	Compatible

Ya que obtuvimos un diseño partiendo de las reglas de negocio en éste capítulo, seguiremos con la parte de la construcción de la base de datos en el capítulo 4, basándonos en el modelo entidad relación construiremos los script's ya utilizando la tecnología seleccionada anteriormente (Microsoft SQL Server 2000), donde utilizaremos los niveles de abstracción de las bases de datos.

## CAPITULO 4.

### DESARROLLO

Con el diseño elaborado teórica y gráficamente respetando las reglas de negocio nos enfocaremos en la creación de los script's (código fuente) que generaran la base de datos, sus debidas restricciones, vistas y sus procedimientos almacenados para lograr un prototipo normalizado y robusto capaz de conseguir confiabilidad y seguridad en el manejo de la información durante el proceso de la verificación.

#### **4.1 CREACIÓN DE TABLAS**

La creación de tablas se puede llevar a cabo de dos formas, la primera y la más sencilla es utilizar la interfaz gráfica del gestor de la base de datos Microsoft SQL Server 2000 con la desventaja de que no se registran los pasos ni el procedimiento para regenerarlas de nuevo en un futuro, la segunda es utilizar el lenguaje Transact SQL<sup>1</sup> el cual asegura que la base de datos puede ser regenerada tantas veces como sea necesario, ya que con esta opción se debe generar todo el código correspondiente y si esto es acompañado de las buenas técnicas de codificación (estandarización y documentación ) se logra el objetivo propuesto por esta tesis.

A continuación se muestra un ejemplo de una tabla generada por las dos formas:

Transact SQL:

Se muestra el código de una tabla creada por medio del lenguaje propio de SQL Server, podremos notar que se asigna al servidor Federado (Primario), opción que no esta disponible de manera directa de la interfaz gráfica.

```
CREATE TABLE [dbo].[SEGUNDOS_X_SEGUNDOS]
(
    [clave_estado] [integer] NOT NULL,
    [clave_municipio] [integer] NOT NULL,
    [clave_proveedor] [integer] NOT NULL,
    [clave_centro] [integer] NOT NULL,
    [clave_vehiculo] [bigint] NOT NULL,
    [clave_linea] [integer] NOT NULL,
    [clave_banca] [integer] NOT NULL,
    [clave_version] [integer] NOT NULL,
    [fecha] [datetime] NOT NULL,
    [hora] [datetime] NOT NULL,
```

---

<sup>1</sup> Es el lenguaje de programación de MICROSOFT SQL Sever. Fuente:  
[http://www.netveloper.com/contenido2.aspx?IDC=47\\_0](http://www.netveloper.com/contenido2.aspx?IDC=47_0)

```

[protocolo]      [integer]      NOT NULL,
[etapa]          [integer]      NOT NULL,
[seg_etapa]     [integer]      NOT NULL,
[seg_prueba]   [integer]      NOT NULL,
[thp]           [float]       NOT NULL,
[kph]           [float]       NOT NULL,
[volts_dino]   [float]       NOT NULL,
[hc]           [integer]     NOT NULL,
[co]           [float]       NOT NULL,
[co2]          [float]       NOT NULL,
[o2]           [float]       NOT NULL,
[nox]          [integer]     NOT NULL,
[lambda]       [float]       NOT NULL,
[checksum]     [varchar] (8) NOT NULL,
[status]       [bit]         NOT NULL
) ON [PRIMARY]
GO

```

### Interfaz Grafica:

En la figura 4.1 se muestra donde se encuentra ubicada la opción para generar una tabla perteneciente a la base de datos seleccionada.

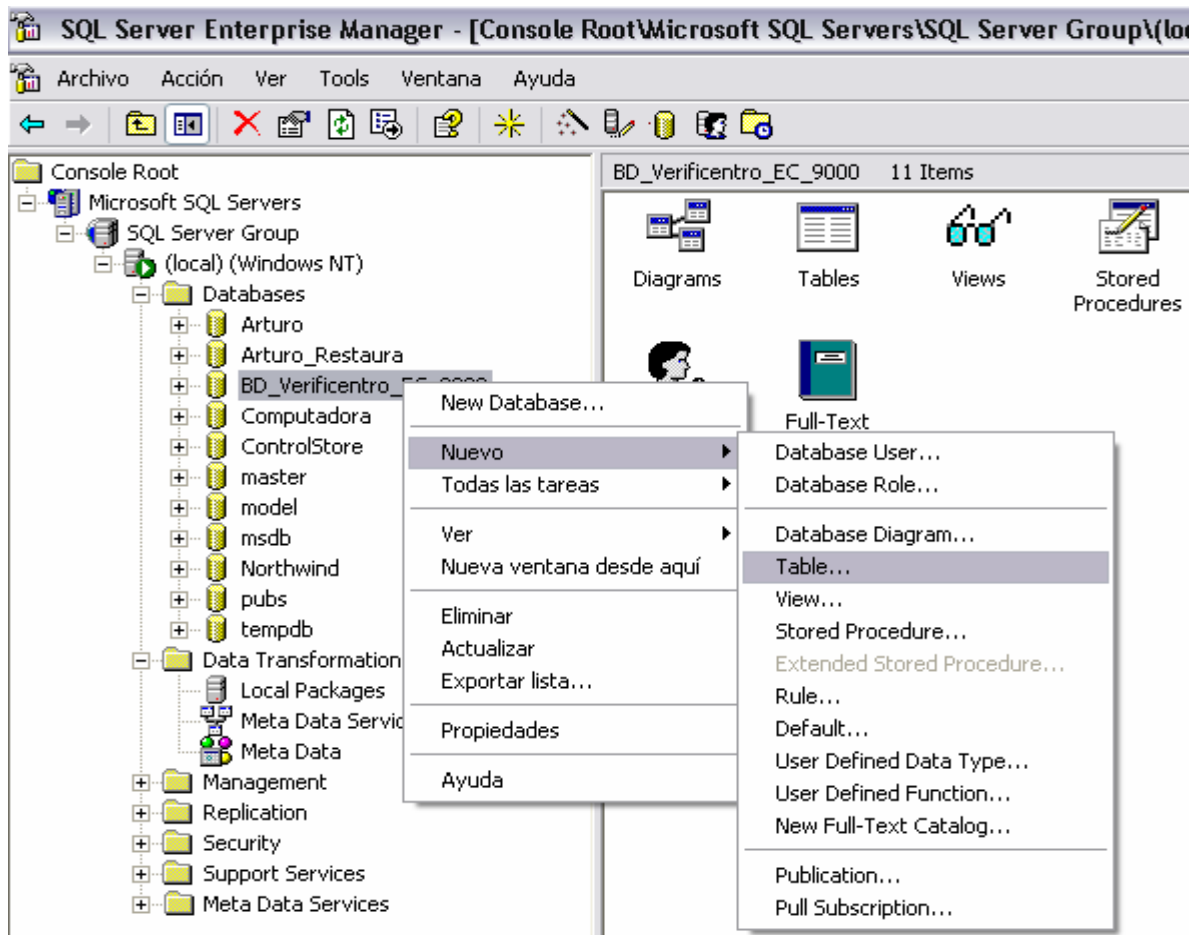
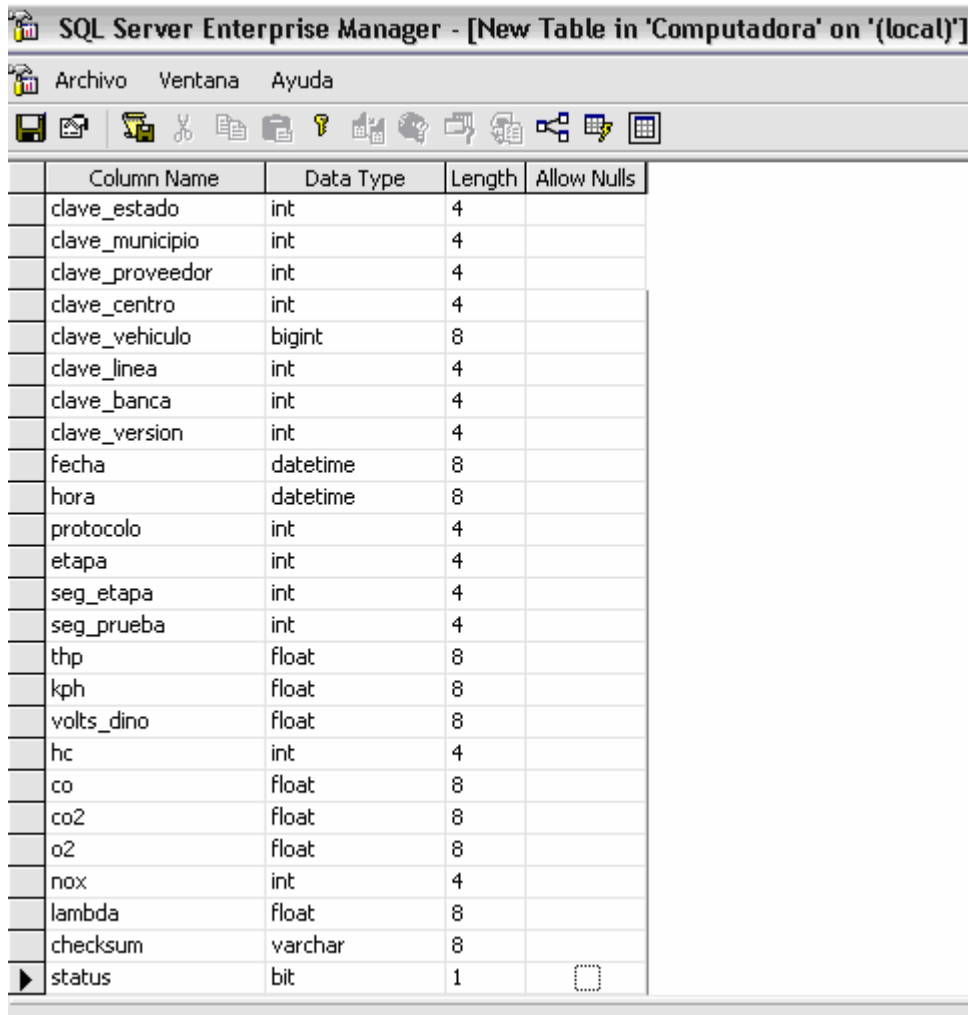


Figura 4.1

Aquí se muestra (figura 4.2) plenamente la estructura y los campos de la tabla, es importante notar que para todos los campos esta deshabilitada la opción que les permitiría guardar valores NULL, esta es una restricción que también se cuidó en la elaboración de la tabla por medio de Transact SQL 2000 Server de Microsoft.



SQL Server Enterprise Manager - [New Table in 'Computadora' on '(local)']

Archivo Ventana Ayuda

Column Name	Data Type	Length	Allow Nulls
clave_estado	int	4	<input type="checkbox"/>
clave_municipio	int	4	<input type="checkbox"/>
clave_proveedor	int	4	<input type="checkbox"/>
clave_centro	int	4	<input type="checkbox"/>
clave_vehiculo	bigint	8	<input type="checkbox"/>
clave_linea	int	4	<input type="checkbox"/>
clave_banca	int	4	<input type="checkbox"/>
clave_version	int	4	<input type="checkbox"/>
fecha	datetime	8	<input type="checkbox"/>
hora	datetime	8	<input type="checkbox"/>
protocolo	int	4	<input type="checkbox"/>
etapa	int	4	<input type="checkbox"/>
seg_etapa	int	4	<input type="checkbox"/>
seg_prueba	int	4	<input type="checkbox"/>
thp	float	8	<input type="checkbox"/>
kph	float	8	<input type="checkbox"/>
volts_dino	float	8	<input type="checkbox"/>
hc	int	4	<input type="checkbox"/>
co	float	8	<input type="checkbox"/>
co2	float	8	<input type="checkbox"/>
o2	float	8	<input type="checkbox"/>
nox	int	4	<input type="checkbox"/>
lambda	float	8	<input type="checkbox"/>
checksum	varchar	8	<input type="checkbox"/>
status	bit	1	<input type="checkbox"/>

Figura 4.2

Por último se asigna el nombre de la tabla como se muestra en la figura 4.3

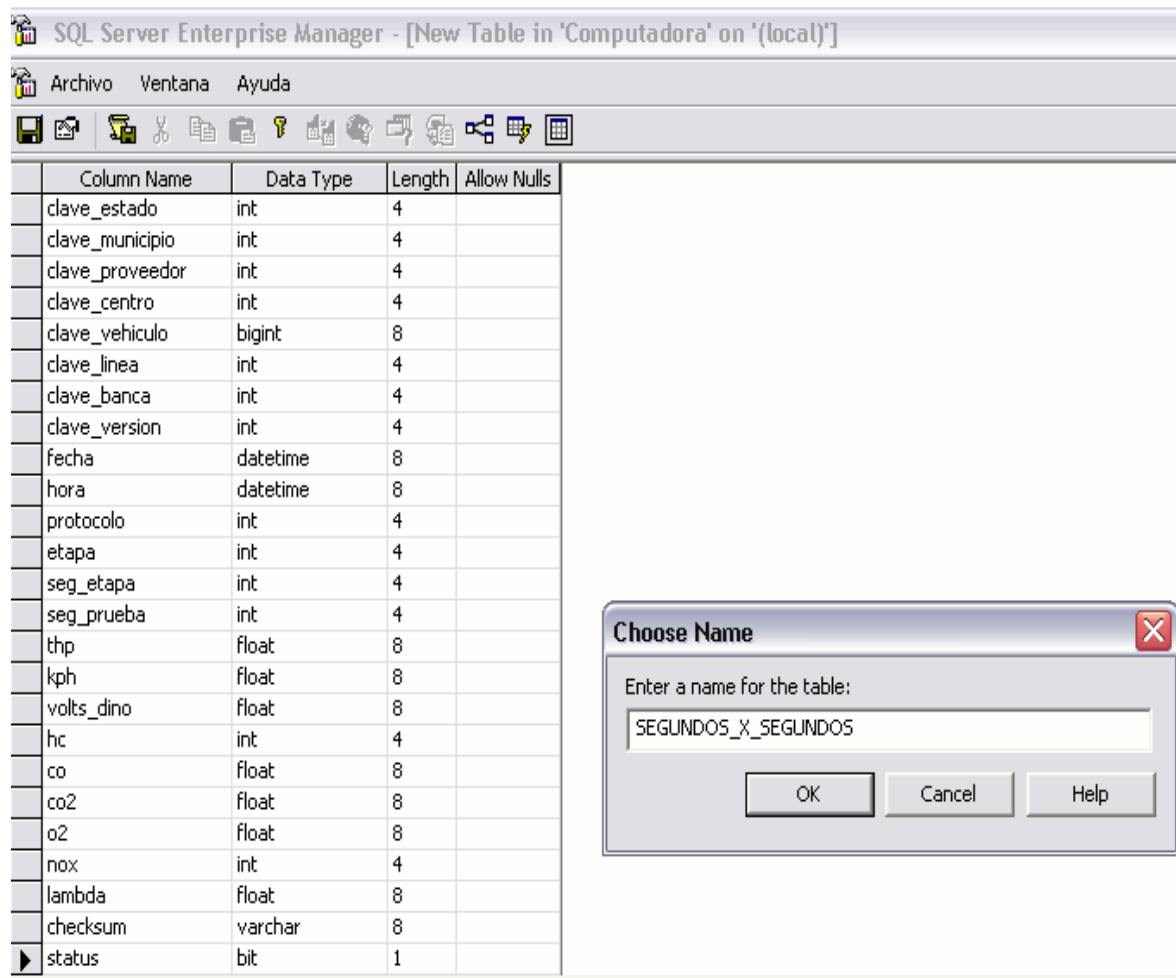


Figura 4.3

## 4.2 CREACIÓN DE INTEGRIDADES

Para la creación de las integridades tomamos como referencia el diagrama entidad relación y las reglas del negocio para determinar cuales son las restricciones que son propias del ámbito de una base de datos para el proceso de la verificación vehicular.

### 4.2.1 INTEGRIDAD POR ENTIDAD

La integridad por Entidad determina cual es la llave primaria de la tabla, con la cual se accede y se referencia a ella.

```
ALTER TABLE [dbo].[SEGUNDOS_X_SEGUNDOS] WITH NOCHECK ADD
    CONSTRAINT [PK_SEGUNDOS_X_SEGUNDOS] PRIMARY KEY CLUSTERED
    (
        [clave_estado],
        [clave_municipio],
        [clave_proveedor],
        [clave_centro] ,
        [clave_vehiculo] ,
        [clave_linea] ,
        [clave_banca] ,
        [clave_version] ,
        [fecha] ,
        [hora]
    )WITH FILLFACTOR = 10 ON [PRIMARY]
GO
```

### 4.2.2 INTEGRIDAD POR DOMINO

La integridad por Dominio muestra lo valores que se tomaran en el caso de no tener un valor dado por la aplicación o el usuario.

```
ALTER TABLE [dbo].[SEGUNDOS_X_SEGUNDOS] WITH NOCHECK ADD
    CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CLAVE_ESTADO]
        DEFAULT (0) FOR [clave_estado],
    CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CLAVE_MUNICIPIO]
        DEFAULT (0) FOR [clave_municipio],
    CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CLAVE_PROVEEDOR]
        DEFAULT (0) FOR [clave_proveedor],
```

```

CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CLAVE_CENTRO]
  DEFAULT (0)          FOR [clave_centro],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CLAVE_VEHICULO]
  DEFAULT (0)          FOR [clave_vehiculo],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CLAVE_LINEA]
  DEFAULT (0)          FOR [clave_linea],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CLAVE_BANCA]
  DEFAULT (0)          FOR [clave_banca],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CLAVE_VERSION]
  DEFAULT (0)          FOR [clave_version],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_FECHA]
  DEFAULT (01/01/1900) FOR [fecha],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_HORA]
  DEFAULT (000000)     FOR [hora],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_PROTOCOLO]
  DEFAULT (0)          FOR [protocolo],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_ETAPA]
  DEFAULT (0)          FOR [etapa],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_SEG_ETAPA]
  DEFAULT (0)          FOR [seg_etapa],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_SEG_PRUEBA]
  DEFAULT (0)          FOR [seg_prueba],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_THP]
  DEFAULT (0)          FOR [thp],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_KPH]
  DEFAULT (0)          FOR [kph],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_VOLTS_DINO]
  DEFAULT (0)          FOR [volts_dino],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_HC]
  DEFAULT (0)          FOR [hc],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CO]
  DEFAULT (0)          FOR [co],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CO2]
  DEFAULT (0)          FOR [co2],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_O2]
  DEFAULT (0)          FOR [o2],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_NOX]
  DEFAULT (0)          FOR [nox],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_LAMBDA]
  DEFAULT (0)          FOR [lambda],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_CHECKSUM]
  DEFAULT ( '')        FOR [checksum],
CONSTRAINT [DF_SEGUNDOS_X_SEGUNDOS_STATUS]
  DEFAULT (0)          FOR [status]

```

GO

### 4.2.3 INTEGRIDAD POR REFERENCIA

La integridad por Referencia relaciona diferentes tablas según las reglas de negocio y de integridad impuestas en el diseño detallado en el capítulo anterior.

```
ALTER TABLE [dbo].[SEGUNDOS_X_SEGUNDOS] ADD
  CONSTRAINT [FK_SEGUNDOS_X_SEGUNDOS_CENTROS] FOREIGN KEY
  (
    [clave_estado],
    [clave_municipio],
    [clave_proveedor],
    [clave_centro]
  )
  REFERENCES [dbo].[CENTROS]
  (
    [clave_estado],
    [clave_municipio],
    [clave_proveedor],
    [clave_centro]
  )
GO
```

```
ALTER TABLE [dbo].[SEGUNDOS_X_SEGUNDOS] ADD
  CONSTRAINT [FK_SEGUNDOS_X_SEGUNDOS_LINEAS] FOREIGN KEY
  (
    [clave_estado],
    [clave_municipio],
    [clave_proveedor],
    [clave_centro],
    [clave_linea]
  )
  REFERENCES [dbo].[LINEAS]
  (
    [clave_estado],
    [clave_municipio],
    [clave_proveedor],
    [clave_centro],
    [clave_linea]
  )
GO
```

```
ALTER TABLE [dbo].[SEGUNDOS_X_SEGUNDOS] ADD
  CONSTRAINT [FK_SEGUNDOS_X_SEGUNDOS_VERSIONES_SOFTWARE] FOREIGN
  KEY
  (
    [clave_version]
  )
  REFERENCES [dbo].[VERSIONES_SOFTWARE]
  (

```



```

        [clave_version]
    )
GO

ALTER TABLE [dbo].[SEGUNDOS_X_SEGUNDOS] ADD
    CONSTRAINT [FK_SEGUNDOS_X_SEGUNDOS_DATOS_VEHICULOS] FOREIGN KEY
    (
        [clave_vehiculo]
    )
    REFERENCES [dbo].[DATOS_VEHICULOS]
    (
        [clave_vehiculo]
    )
GO

```

### 4.3 CREACIÓN DE VISTAS

Es necesario contar con las siguientes vistas para restringir la información que puede ser vista por los diferentes tipos de usuario, facilitando el proceso las tareas diarias.

Estación 1 (maquina de captura)	Vista para llenar formulario de los datos de vehiculo a realizar la prueba
Servidor (Base de Datos)	Vista para el alta y baja de personal, asi como actualización de los datos del Centro de Verificación.
Impresión (Impresión de certificados)	Vista para la revisión de los certificados otorgados anteriormente.

## 4.4 CREACIÓN DE RESPALDOS Y RECUPERACIÓN DE LA BASE DE DATOS CON PROCEDIMIENTOS ALMACENADOS

Crearemos procedimientos almacenados en Transact SQL Server 2000 de Microsoft que sirve para la creación de respaldos totales y diferenciales (incremental) periódicamente. También se generan los script's para la recuperación de la base de datos en caso de pérdida total o de alguno de los dos archivos, el de transacciones o el de datos según sea el caso, además esta ultimo script se puede utilizar en caso de que el sistema operativo sufra algún daño evitando el acceso al gestor de la BD.

### 4.4.1 PROCEDIMIENTO ALMACENADO QUE GENERA UN RESPALDO COMPLETO DE LA BASE DE DATOS.

```
use [master]
GO
CREATE PROCEDURE sp_RespaldoCompleto AS
BACKUP DATABASE BD_Verificentro_EC_9000 TO DISK =
'c:\Respaldos\Completo\BD_Verificentro_EC_9000_respaldo.bak'
GO
```

### 4.4.2 Realiza un Respaldo Diferencial

```
use [master]
GO
CREATE PROCEDURE sp_RespaldoDiferencial
AS
    BACKUP DATABASE BD_Verificentro_EC_9000
        TO DISK =
'c:\Respaldos\Diferencial\BD_Verificentro_EC_9000_respaldo.bak'
        WITH DIFFERENTIAL
GO
```

## 4.5 CREACIÓN DE DISPARADORES

El motivo de la creación de los disparadores es básicamente para el registro de los diversos movimientos en las tablas en una bitácora que servirá como registro de todas y cada una de las acciones que acontezcan en la Base de Datos.

```
DROP TRIGGER [TRIGGER_INSERT_SEGUNDOS_X_SEGUNDOS]
GO

CREATE TRIGGER [TRIGGER_INSERT_SEGUNDOS_X_SEGUNDOS] ON
[SEGUNDOS_X_SEGUNDOS]
FOR INSERT
AS
SET NOCOUNT ON
INSERT INTO [MOVIMIENTOS]
(
    [clave_estado],
    [clave_municipio],
    [clave_proveedor],
    [clave_centro],
    [clave_personal],
    [clave_usuario],
    [clave_movimiento],
    [clave_permiso],
    [fecha],
    [hora],
    [entidad_relacion]
)
SELECT
    [clave_estado],
    [clave_municipio],
    [clave_proveedor],
    [clave_centro],
    0,
    0,
    1,
    0,
    GETDATE(),
    GETDATE(),
    'SEGUNDOS_X_SEGUNDOS'
FROM inserted
GO
```

## **4.6. CONVERSIÓN DE DATOS**

La conversión de datos se lleva acabo utilizando la interfaz gráfica integrando código Transact SQL , estas dos partes conforman una herramienta llamada DTS (Data Transformation Services) con la cual se importan y exportan datos a la BD.

### **4.6.1 IMPORTACIÓN**

Con respecto a la importación la utilizaremos en los casos donde tengamos que extraer información de tablas externas de un diferente formato que no es propio de SQL Server, un ejemplo son las tablas como "Datos\_Maestros" las cuales no son generadas por los centros de Verificación sino que son actualizadas por las autoridades que rigen los programas de verificación vehicular en el valle de México y que las proporcionan a los Centro de Verificación para su importación a las Bases de Datos para la realización de las verificaciones vehiculares.

### **4.6.2 EXPORTACIÓN**

Los centros de Verificación se rigen por una política la cual consta en que semanalmente generan un respaldo de ciertas tablas y determinados campos, todo esto viene a que por medio de la herramienta DTS crearemos esos respaldos de archivos que actualmente su salida es en formato de hoja de cálculo.

## 4.7 CÓDIGO FUENTE.

Por razones de espacio ejemplificaremos únicamente el código de creación para la base de datos.

```
use master
--Creamos la BDs
CREATE DATABASE [BD_Verificentro_EC_9000] ON
(
  --Nombramos el archivo que contendrá os datos.
  NAME = N'BD_Verificentro_EC_9000_Dat',
  --Nombramos la ruta donde se alojara ese archivo
  FILENAME = N'c:\BD_Verificentro_EC_9000_Dat.mdf' ,
  --Valor de crecimiento del archivo de Datos
  SIZE = 10,
  --Valor en porcentaje para el incremento de la BDs cada vez que
  ocurra un llenado
  FILEGROWTH = 10%
)
--Creación de archivo de transacciones
LOG ON
(
  NAME = N'BD_Verificentro_EC_9000_log',
  FILENAME = N'c:\BD_Verificentro_EC_9000_log.ldf' ,
  SIZE = 2,
  FILEGROWTH = 10%
)
GO

exec sp_dboption N'BD_Verificentro_EC_9000', N'autoclose', N'true'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'bulkcopy', N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'trunc. log',
N'true'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'torn page
detection', N'true'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'read only',
N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'dbo use', N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'single', N'false'
GO
```

```

exec sp_dboption N'BD_Verificentro_EC_9000', N'autoshrink',
N'true'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'ANSI null default',
N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'recursive
triggers', N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'ANSI nulls',
N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'concat null yields
null', N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'cursor close on
commit', N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'default to local
cursor', N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'quoted identifier',
N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'ANSI warnings',
N'false'
GO
exec sp_dboption N'BD_Verificentro_EC_9000', N'auto update
statistics', N'true'
GO

```

```

--Se coloca en la Base de Datos Nueva
use [BD_Verificentro_EC_9000]
GO

```

```

--
--                                CREACION DE TABLAS
-----
--
--                                ENTIDADES FUERTES
-----

```

```

CREATE TABLE [dbo].[LABORATORIOS]
(
    [clave_municipio] [integer] NOT NULL,
    [clave_estado] [integer] NOT NULL,
    [clave_laboratorio] [integer] NOT NULL,
    [nombre_razonsoc] [varchar] (20) NOT NULL,
    [calle] [varchar] (20) NOT NULL,
    [colonia] [varchar] (20) NOT NULL,
    [num_ext] [varchar] (5) NOT NULL,
    [num_int] [varchar] (5) NOT NULL,
    [cp] [integer] NOT NULL,

```

```

        [checksum]          [varchar] (8) NOT NULL,
        [status]            [bit]          NOT NULL
    ) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[TECNICOS_LABORATORIOS]
(
    [clave_municipio]      [integer]       NOT NULL,
    [clave_estado]        [integer]       NOT NULL,
    [clave_laboratorio]   [integer]       NOT NULL,
    [clave_tecnico]       [integer]       NOT NULL,
    [nombre]              [varchar] (20) NOT NULL,
    [apel_paterno]        [varchar] (20) NOT NULL,
    [apel_materno]        [varchar] (20) NOT NULL,
    [checksum]            [varchar] (8) NOT NULL,
    [status]              [bit]          NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[PROVEEDORES]
(
    [clave_municipio]      [integer]       NOT NULL,
    [clave_estado]        [integer]       NOT NULL,
    [clave_proveedor]     [integer]       NOT NULL,
    [nombre_razonsoc]     [varchar] (20) NOT NULL,
    [calle]               [varchar] (20) NOT NULL,
    [colonia]             [varchar] (20) NOT NULL,
    [num_ext]             [varchar] (5) NOT NULL,
    [num_int]             [varchar] (5) NOT NULL,
    [cp]                 [integer]       NOT NULL,
    [checksum]            [varchar] (8) NOT NULL,
    [status]              [bit]          NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[LINEAS]
(
    [clave_municipio]      [integer]       NOT NULL,
    [clave_estado]        [integer]       NOT NULL,
    [clave_proveedor]     [integer]       NOT NULL,
    [clave_centro]        [integer]       NOT NULL,
    [clave_linea]         [integer]       NOT NULL,
    [clave_dinamometro]   [integer]       NOT NULL,
    [clave_banca]         [integer]       NOT NULL,
    [serie_chasis]        [integer]       NOT NULL,
    [factor_error_propa]  [float]          NOT NULL,
    [clave_estacion]      [integer]       NOT NULL,
    [clave_version]       [integer]       NOT NULL,
    [checksum]            [varchar] (8) NOT NULL,
    [status]              [bit]          NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[ESTACIONES]
(
    [clave_estacion]    [integer]        NOT NULL,
    [tipo_estacion]    [varchar] (10) NOT NULL,
    [descripcion]      [varchar] (20) NOT NULL,
    [checksum]         [varchar] (8)  NOT NULL,
    [status]           [bit]            NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[VERSIONES_SOFTWARE]
(
    [clave_version]    [integer]        NOT NULL,
    [fecha_inicio]    [datetime]       NOT NULL,
    [fecha_termino]   [datetime]       NOT NULL,
    [checksum]         [varchar] (8)  NOT NULL,
    [status]           [bit]            NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[FALLAS_CATALOGO]
(
    [clave_tipo_falla] [integer]        NOT NULL,
    [tipo_falla]      [varchar] (10) NOT NULL,
    [descripcion]     [varchar] (20) NOT NULL,
    [checksum]        [varchar] (8)  NOT NULL,
    [status]          [bit]            NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[EVALUACIONES]
(
    [clave_evaluacion] [integer]        NOT NULL,
    [valor_minimo]     [float]          NOT NULL,
    [valor_maximo]     [float]          NOT NULL,
    [valor_inferior_nox] [float]       NOT NULL,
    [valor_maximo_nox] [float]          NOT NULL,
    [pruebas_minimas] [integer]        NOT NULL,
    [pruebas_maximas] [integer]        NOT NULL,
    [acciones]         [varchar] (8)  NOT NULL,
    [checksum]         [varchar] (8)  NOT NULL,
    [status]           [bit]            NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[ESTADOS]
(
    [clave_estado] [integer]        NOT NULL,
    [nombre]       [varchar] (20) NOT NULL,
    [checksum]     [varchar] (8)  NOT NULL,
    [status]       [bit]            NOT NULL
)

```



```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[MUNICIPIOS]
```

```
(
  [clave_municipio] [integer] NOT NULL,
  [clave_estado] [integer] NOT NULL,
  [nombre] [varchar] (35) NOT NULL,
  [cp_inicial] [integer] NOT NULL,
  [cp_final] [integer] NOT NULL,
  [checksum] [varchar] (8) NOT NULL,
  [status] [bit] NOT NULL
```

```
) ON [PRIMARY]
```

```
GO
```

```
CREATE TABLE [dbo].[CENTROS]
```

```
(
  [clave_municipio] [integer] NOT NULL,
  [clave_estado] [integer] NOT NULL,
  [clave_proveedor] [integer] NOT NULL,
  [clave_centro] [integer] NOT NULL,
  [nombre_razonsoc] [varchar] (20) NOT NULL,
  [num_lineas] [integer] NOT NULL,
  [calle] [varchar] (20) NOT NULL,
  [colonia] [varchar] (20) NOT NULL,
  [num_ext] [varchar] (5) NOT NULL,
  [num_int] [varchar] (5) NOT NULL,
  [cp] [integer] NOT NULL,
  [rfc] [varchar] (20) NOT NULL,
  [checksum] [varchar] (8) NOT NULL,
  [status] [bit] NOT NULL
```

```
) ON [PRIMARY]
```

```
GO
```

```
CREATE TABLE [dbo].[PERSONAL]
```

```
(
  [clave_municipio] [integer] NOT NULL,
  [clave_estado] [integer] NOT NULL,
  [clave_proveedor] [integer] NOT NULL,
  [clave_centro] [integer] NOT NULL,
  [clave_personal] [integer] NOT NULL,
  [clave_puesto] [integer] NOT NULL,
  [nombre] [varchar] (20) NOT NULL,
  [apel_paterno] [varchar] (20) NOT NULL,
  [apel_materno] [varchar] (20) NOT NULL,
  [calle] [varchar] (20) NOT NULL,
  [colonia] [varchar] (20) NOT NULL,
  [num_ext] [varchar] (5) NOT NULL,
  [num_int] [varchar] (5) NOT NULL,
  [cp] [integer] NOT NULL,
  [checksum] [varchar] (8) NOT NULL,
  [status] [bit] NOT NULL
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[CATALOGO_MOVIMIENTOS]
(
    [clave_movimiento] [integer] NOT NULL,
    [tipo_movimiento] [varchar] (10) NOT NULL,
    [descripcion] [varchar] (20) NOT NULL,
    [checksum] [varchar] (8) NOT NULL,
    [status] [bit] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[PUESTOS]
(
    [clave_puesto] [integer] NOT NULL,
    [tipo_puesto] [varchar] (20) NOT NULL,
    [descripcion] [varchar] (20) NOT NULL,
    [checksum] [varchar] (8) NOT NULL,
    [status] [bit] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[USUARIOS]
(
    [clave_municipio] [integer] NOT NULL,
    [clave_estado] [integer] NOT NULL,
    [clave_proveedor] [integer] NOT NULL,
    [clave_centro] [integer] NOT NULL,
    [clave_personal] [integer] NOT NULL,
    [clave_usuario] [integer] NOT NULL,
    [clave_permiso] [integer] NOT NULL,
    [login_usuario] [varchar] (10) NOT NULL,
    [tipo_usuario] [varchar] (10) NOT NULL,
    [checksum] [varchar] (8) NOT NULL,
    [status] [bit] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[CONTRASENAS]
(
    [clave_municipio] [integer] NOT NULL,
    [clave_estado] [integer] NOT NULL,
    [clave_proveedor] [integer] NOT NULL,
    [clave_centro] [integer] NOT NULL,
    [clave_personal] [integer] NOT NULL,
    [clave_usuario] [integer] NOT NULL,
    [clave_contrasena] [integer] NOT NULL,
    [contrasena] [varchar] (10) NOT NULL,
    [fecha_inicio] [datetime] NOT NULL,
    [fecha_termino] [datetime] NOT NULL,
    [checksum] [varchar] (8) NOT NULL,

```

```

    [status]          [bit]          NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[PERMISOS]
(
    [clave_permiso] [integer]          NOT NULL,
    [tipo_permiso] [varchar] (20) NOT NULL,
    [checksum]     [varchar] (8)  NOT NULL,
    [status]      [bit]          NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[CLASES_VEHICULOS]
(
    [clave_clase_vehiculo] [integer]          NOT NULL,
    [tipo_clase_vehiculo] [varchar] (10) NOT NULL,
    [checksum]            [varchar] (8)  NOT NULL,
    [status]             [bit]          NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[LIMITES_GASES]
(
    [clave_combustible] [integer]          NOT NULL,
    [clave_peso_bruto_vehicular] [integer]          NOT NULL,
    [clave_taxi]       [integer]          NOT NULL,
    [año_desde]       [integer]          NOT NULL,
    [año_hasta]       [integer]          NOT NULL,
    [hc_max]          [integer]          NOT NULL,
    [co_max]          [float]            NOT NULL,
    [o2_max]          [float]            NOT NULL,
    [dil_min]         [float]            NOT NULL,
    [dil_max]         [float]            NOT NULL,
    [nox_max]         [integer]          NOT NULL,
    [max_nivel]       [float]            NOT NULL,
    [checksum]        [varchar] (8) NOT NULL,
    [status]          [bit]            NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[CERTIFICADOS_ROBADOS]
(
    [clave_estado] [integer]          NOT NULL,
    [fecha_robos] [datetime]          NOT NULL,
    [certificado_inicio] [integer]          NOT NULL,
    [certificado_final] [integer]          NOT NULL,
    [tipo_certificado] [varchar] (10) NOT NULL,
    [checksum]        [varchar] (8) NOT NULL,
    [status]          [bit]            NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[RECHAZOS]
(
    [clave_rechazo]          [integer]          NOT NULL,
    [tipo_rechazo]          [varchar] (10) NOT NULL,
    [descripcion]           [varchar] (20) NOT NULL,
    [checksum]              [varchar] (8)  NOT NULL,
    [status]                [bit]          NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[MOTIVOS_VERIFICACION]
(
    [clave_motivo]          [integer]          NOT NULL,
    [zona]                 [varchar] (1)  NOT NULL,
    [tipo_motivo]          [varchar] (10) NOT NULL,
    [descripcion]           [varchar] (20) NOT NULL,
    [checksum]              [varchar] (8)  NOT NULL,
    [status]                [bit]          NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[CARGAS]
(
    [clave_carroceria] [integer]          NOT NULL,
    [cilindros]        [integer]          NOT NULL,
    [ciclo]            [integer]          NOT NULL,
    [rodillos]        [integer]          NOT NULL,
    [potin]            [integer]          NOT NULL,
    [checksum]         [varchar] (8)  NOT NULL,
    [status]           [bit]            NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[SERVICIOS]
(
    [clave_servicio] [integer]          NOT NULL,
    [tipo_servicio] [varchar] (10) NOT NULL,
    [checksum]       [varchar] (8)  NOT NULL,
    [status]         [bit]            NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[CARROCERIAS]
(
    [clave_carroceria] [integer]          NOT NULL,
    [tipo_carroceria] [varchar] (10) NOT NULL,
    [checksum]         [varchar] (8)  NOT NULL,
    [status]           [bit]            NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[ALIMENTACION_COMBUSTIBLES]
(
    [clave_combustible]          [integer]          NOT NULL,
    [tipo_alimentacion_combustible] [varchar] (10) NOT NULL,
    [clave_peso_bruto_vehicular] [integer]          NOT NULL,
    [clave_taxi]                 [integer]          NOT NULL,
    [año_desde]                  [integer]          NOT NULL,
    [año_hasta]                  [integer]          NOT NULL,
    [checksum]                   [varchar] (8)     NOT NULL,
    [status]                     [bit]             NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[MARCAS]
(
    [clave_marca] [integer]          NOT NULL,
    [nombre]      [varchar] (20) NOT NULL,
    [checksum]    [varchar] (8)   NOT NULL,
    [status]     [bit]           NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[SUBMARCAS]
(
    [clave_marca]      [integer]          NOT NULL,
    [clave_submarca]  [integer]          NOT NULL,
    [nombre]           [varchar] (20) NOT NULL,
    [checksum]         [varchar] (8)   NOT NULL,
    [status]           [bit]           NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[DATOS_MAESTROS]
(
    [tabla_maestra]          [bigint]          NOT NULL,
    [clave_combustible]     [integer]          NOT NULL,
    [clave_marca]           [integer]          NOT NULL,
    [clave_submarca]       [integer]          NOT NULL,
    [forma]                 [integer]          NOT NULL,
    [cilindros]             [integer]          NOT NULL,
    [cilindrada]           [integer]          NOT NULL,
    [transmision]           [char] (1)       NOT NULL,
    [año_desde]             [char] (1)       NOT NULL,
    [año_hasta]            [integer]          NOT NULL,
    [clave_peso_bruto_vehicular] [integer]          NOT NULL,
    [peso_equivalente]     [float]           NOT NULL,
    [cl_inercia]           [integer]          NOT NULL,
    [potencia_total_5024]  [float]           NOT NULL,
    [potencia_total_2540] [float]           NOT NULL,
    [potencia_total_5024_8] [float]           NOT NULL,
    [potencia_total_2540_8] [float]           NOT NULL,
    [potencia_total_5024_20] [float]           NOT NULL,

```

```

[potencia_total_2540_20]      [float]      NOT NULL,
[fwdperm]                     [char] (1)   NOT NULL,
[c_tracc]                     [char] (1)   NOT NULL,
[c_tracc_off]                 [char] (1)   NOT NULL,
[abs]                         [char] (1)   NOT NULL,
[abs_off]                     [char] (1)   NOT NULL,
[dinoleje]                    [char] (1)   NOT NULL,
[obd]                         [char] (1)   NOT NULL,
[sistema_aire]                [char] (1)   NOT NULL,
[sistema_combustible]         [char] (2)   NOT NULL,
[covertidor_catalitico]       [char] (1)   NOT NULL,
[max_nivel]                   [char] (1)   NOT NULL,
[preacond]                    [char] (1)   NOT NULL,
[protocolo]                   [char] (2)   NOT NULL,
[rpm_sural]                   [integer]    NOT NULL,
[o2_sural]                    [float]      NOT NULL,
[o2_sural_5024]               [float]      NOT NULL,
[o2_sural_2540]               [float]      NOT NULL,
[dil_inral]                   [float]      NOT NULL,
[dil_inral_5024]              [float]      NOT NULL,
[dil_sural]                   [float]      NOT NULL,
[dil_su_5024]                 [float]      NOT NULL,
[dil_su_2540]                 [float]      NOT NULL,
[co_sural]                    [float]      NOT NULL,
[co_su_5024]                  [float]      NOT NULL,
[co_su_2540]                  [float]      NOT NULL,
[hc_sural]                    [integer]    NOT NULL,
[hc_su_5024]                  [integer]    NOT NULL,
[hc_su_2540]                  [integer]    NOT NULL,
[nox_sural]                   [integer]    NOT NULL,
[nox_su_5024]                 [integer]    NOT NULL,
[nox_su_2540]                 [integer]    NOT NULL,
[lam_inral]                   [float]      NOT NULL,
[lam_in_5024]                 [float]      NOT NULL,
[lam_in_2540]                 [float]      NOT NULL,
[lam_sural]                   [float]      NOT NULL,
[lam_su_5024]                 [float]      NOT NULL,
[lam_su_2540]                 [float]      NOT NULL,
[co_suacel]                   [integer]    NOT NULL,
[hc_suacel]                   [integer]    NOT NULL,
[nox_suacel]                  [integer]    NOT NULL,
[lam_inacel]                  [float]      NOT NULL,
[lam_suacel]                  [float]      NOT NULL,
[diesel]                       [integer]    NOT NULL,
[potencia_maxima_rpm]         [integer]    NOT NULL,
[diesel_suacel]               [float]      NOT NULL,
[spare]                        [char] (16)  NOT NULL,
[checksum]                    [varchar] (8) NOT NULL,
[status]                       [bit]        NOT NULL
) ON [PRIMARY]
GO

```

-- ENTIDADES DEBILES

---

```
CREATE TABLE [dbo].[ACCESOS]
(
    [clave_municipio] [integer] NOT NULL,
    [clave_estado] [integer] NOT NULL,
    [clave_proveedor] [integer] NOT NULL,
    [clave_centro] [integer] NOT NULL,
    [clave_personal] [integer] NOT NULL,
    [clave_usuario] [integer] NOT NULL,
    [clave_contrasena] [integer] NOT NULL,
    [clave_accion] [integer] NOT NULL,
    [fecha] [datetime] NOT NULL,
    [hora] [datetime] NOT NULL,
    [checksum] [varchar] (8) NOT NULL,
    [status] [bit] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[CALIBRACIONES_LABORATORIOS]
(
    [clave_municipio] [integer] NOT NULL,
    [clave_estado] [integer] NOT NULL,
    [clave_proveedor] [integer] NOT NULL,
    [clave_centro] [integer] NOT NULL,
    [clave_linea] [integer] NOT NULL,
    [clave_laboratorio] [integer] NOT NULL,
    [clave_banca] [integer] NOT NULL,
    [clave_version] [integer] NOT NULL,
    [fecha] [datetime] NOT NULL,
    [clave_tecnico] [integer] NOT NULL,
    [hora_inicial] [datetime] NOT NULL,
    [hora_final] [datetime] NOT NULL,
    [cilindro] [integer] NOT NULL,
    [hc] [integer] NOT NULL,
    [co] [float] NOT NULL,
    [co2] [float] NOT NULL,
    [o2] [float] NOT NULL,
    [nox] [integer] NOT NULL,
    [checksum] [varchar] (8) NOT NULL,
    [status] [bit] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[CALIBRACIONES_FUGAS]
(
    [clave_municipio] [integer] NOT NULL,
    [clave_estado] [integer] NOT NULL,
    [clave_proveedor] [integer] NOT NULL,
    [clave_centro] [integer] NOT NULL,
    [clave_linea] [integer] NOT NULL,
```

```

    [clave_banca]      [integer]      NOT NULL,
    [clave_version]   [integer]      NOT NULL,
    [fecha]           [datetime]    NOT NULL,
    [hora_inicial]    [datetime]    NOT NULL,
    [hora_final]      [datetime]    NOT NULL,
    [resultado]       [bit]         NOT NULL,
    [checksum]        [varchar] (8) NOT NULL,
    [status]          [bit]         NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[CORRECCIONES]
(
    [clave_municipio] [integer]      NOT NULL,
    [clave_estado]    [integer]      NOT NULL,
    [clave_proveedor] [integer]      NOT NULL,
    [clave_centro]    [integer]      NOT NULL,
    [clave_personal]  [integer]      NOT NULL,
    [clave_usuario]   [integer]      NOT NULL,
    [clave_linea]     [integer]      NOT NULL,
    [clave_estacion]  [integer]      NOT NULL,
    [clave_falla]     [integer]      NOT NULL,
    [clave_tipo_falla] [integer]      NOT NULL,
    [fecha]           [datetime]    NOT NULL,
    [hora]            [datetime]    NOT NULL,
    [desc_correccion] [varchar] (40) NOT NULL,
    [checksum]        [varchar] (8)  NOT NULL,
    [status]          [bit]         NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[VERIFICACIONES]
(
    [clave_municipio] [integer]      NOT NULL,
    [clave_estado]    [integer]      NOT NULL,
    [clave_proveedor] [integer]      NOT NULL,
    [clave_centro]    [integer]      NOT NULL,
    [clave_personal]  [integer]      NOT NULL,
    [clave_usuario]   [integer]      NOT NULL,
    [clave_linea]     [integer]      NOT NULL,
    [clave_estacion]  [integer]      NOT NULL,
    [clave_banca]     [integer]      NOT NULL,
    [clave_dinamometro] [integer]      NOT NULL,
    [clave_version]   [integer]      NOT NULL,
    [clave_vehiculo]  [bigint]      NOT NULL,
    [clave_servicio]  [integer]      NOT NULL,
    [clave_clase_vehiculo] [integer]      NOT NULL,
    [clave_combustible] [integer]      NOT NULL,
    [clave_carroceria] [integer]      NOT NULL,
    [tabla_maestra]   [bigint]      NOT NULL,
    [motivo_verificacion] [integer]      NOT NULL,
    [checksum]        [varchar] (8)  NOT NULL,

```



```

    [status]          [bit]          NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[CALIBRACIONES_DINAMOMETROS]

```

```

(
    [clave_municipio] [integer]      NOT NULL,
    [clave_estado]    [integer]      NOT NULL,
    [clave_proveedor] [integer]      NOT NULL,
    [clave_centro]    [integer]      NOT NULL,
    [clave_linea]     [integer]      NOT NULL,
    [clave_dinamometro] [integer]    NOT NULL,
    [clave_banca]     [integer]      NOT NULL,
    [clave_version]   [integer]      NOT NULL,
    [fecha]           [datetime]     NOT NULL,
    [hora_inicial]    [datetime]     NOT NULL,
    [hora_final]      [datetime]     NOT NULL,
    [dino_patron]     [float]         NOT NULL,
    [dicerolec]       [float]         NOT NULL,
    [lipatlectu]      [float]         NOT NULL,
    [vcero]           [float]         NOT NULL,
    [vspan]           [float]         NOT NULL,
    [resultado]       [bit]           NOT NULL,
    [checksum]        [varchar] (8)  NOT NULL,
    [status]          [bit]           NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[CALIBRACIONES_ANALIZADOR]

```

```

(
    [clave_municipio] [integer]      NOT NULL,
    [clave_estado]    [integer]      NOT NULL,
    [clave_proveedor] [integer]      NOT NULL,
    [clave_centro]    [integer]      NOT NULL,
    [clave_linea]     [integer]      NOT NULL,
    [clave_banca]     [integer]      NOT NULL,
    [clave_version]   [integer]      NOT NULL,
    [fecha]           [datetime]     NOT NULL,
    [hora_inicial]    [datetime]     NOT NULL,
    [hora_final]      [datetime]     NOT NULL,
    [clave_evaluacion] [integer]     NOT NULL,
    [fep_baja]        [float]         NOT NULL,
    [hc_gas_patro]    [float]         NOT NULL,
    [hc_cero_lect]    [float]         NOT NULL,
    [hc_patro_lect]   [float]         NOT NULL,
    [co_gas_patro]    [float]         NOT NULL,
    [co_cero_lect]    [float]         NOT NULL,
    [co_patro_lect]   [float]         NOT NULL,
    [co2_gas_patro]   [float]         NOT NULL,
    [co2_cero_lect]   [float]         NOT NULL,
    [co2_patro_lect]  [float]         NOT NULL,
    [no_gas_patro]    [float]         NOT NULL,

```

```

[no_cero_lect]          [float]          NOT NULL,
[no_patro_lect]        [float]          NOT NULL,
[pef_media]            [float]          NOT NULL,
[hc_media_gas_patro]   [float]          NOT NULL,
[hc_media_patro_lect] [float]          NOT NULL,
[co_media_gas_patro]   [float]          NOT NULL,
[co_media_patro_lect] [float]          NOT NULL,
[co2_media_gas_patro] [float]          NOT NULL,
[co2_media_patro_lect] [float]          NOT NULL,
[no_media_gas_patro]   [float]          NOT NULL,
[no_media_patro_lect] [float]          NOT NULL,
[hc_ant_cero_lect]    [float]          NOT NULL,
[hc_ant_patro_lect]   [float]          NOT NULL,
[co_ant_cero_lect]    [float]          NOT NULL,
[co_ant_patro_lect]   [float]          NOT NULL,
[co2_ant_cero_lect]   [float]          NOT NULL,
[co2_ant_patro_lect]  [float]          NOT NULL,
[o2_ant_cero_lect]    [float]          NOT NULL,
[o2_ant_patro_lect]   [float]          NOT NULL,
[no_ant_cero_lect]    [float]          NOT NULL,
[no_ant_patro_lect]   [float]          NOT NULL,
[hc_media_ant_patro]  [float]          NOT NULL,
[co_media_ant_patro]  [float]          NOT NULL,
[co2_media_ant_patro] [float]          NOT NULL,
[no_media_ant_patro]  [float]          NOT NULL,
[checksum]             [varchar] (8)   NOT NULL,
[status]               [bit]           NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[FALLAS]
(
  [clave_municipio] [integer]          NOT NULL,
  [clave_estado]    [integer]          NOT NULL,
  [clave_proveedor] [integer]          NOT NULL,
  [clave_centro]    [integer]          NOT NULL,
  [clave_personal]  [integer]          NOT NULL,
  [clave_usuario]   [integer]          NOT NULL,
  [clave_linea]     [integer]          NOT NULL,
  [clave_estacion]  [integer]          NOT NULL,
  [clave_falla]     [integer]          NOT NULL,
  [clave_tipo_falla] [integer]          NOT NULL,
  [fecha]           [datetime]         NOT NULL,
  [hora]            [datetime]         NOT NULL,
  [checksum]        [varchar] (8)     NOT NULL,
  [status]          [bit]              NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[PRUEBAS_GAS_ALEATORIAS]
(
  [clave_municipio] [integer]          NOT NULL,

```

```

[clave_estado] [integer] NOT NULL,
[clave_proveedor] [integer] NOT NULL,
[clave_centro] [integer] NOT NULL,
[clave_linea] [integer] NOT NULL,
[clave_banca] [integer] NOT NULL,
[clave_version] [integer] NOT NULL,
[fecha] [datetime] NOT NULL,
[hora_inicial] [datetime] NOT NULL,
[hora_final] [datetime] NOT NULL,
[hc] [integer] NOT NULL,
[co] [float] NOT NULL,
[co2] [float] NOT NULL,
[o2] [float] NOT NULL,
[nox] [integer] NOT NULL,
[intervalo] [integer] NOT NULL,
[resultado] [integer] NOT NULL,
[checksum] [varchar] (8) NOT NULL,
[status] [bit] NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[SEGUNDOS_X_SEGUNDOS]
(
[clave_municipio] [integer] NOT NULL,
[clave_estado] [integer] NOT NULL,
[clave_proveedor] [integer] NOT NULL,
[clave_centro] [integer] NOT NULL,
[clave_vehiculo] [bigint] NOT NULL,
[clave_linea] [integer] NOT NULL,
[clave_banca] [integer] NOT NULL,
[clave_version] [integer] NOT NULL,
[fecha] [datetime] NOT NULL,
[hora] [datetime] NOT NULL,
[protocolo] [integer] NOT NULL,
[etapa] [integer] NOT NULL,
[seg_etapa] [integer] NOT NULL,
[seg_prueba] [integer] NOT NULL,
[thp] [float] NOT NULL,
[kph] [float] NOT NULL,
[volts_dino] [float] NOT NULL,
[hc] [integer] NOT NULL,
[co] [float] NOT NULL,
[co2] [float] NOT NULL,
[o2] [float] NOT NULL,
[nox] [integer] NOT NULL,
[lambda] [float] NOT NULL,
[checksum] [varchar] (8) NOT NULL,
[status] [bit] NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[MOVIMIENTOS]
(
    [clave_municipio] [integer] NOT NULL,
    [clave_estado] [integer] NOT NULL,
    [clave_proveedor] [integer] NOT NULL,
    [clave_centro] [integer] NOT NULL,
    [clave_personal] [integer] NOT NULL,
    [clave_usuario] [integer] NOT NULL,
    [clave_movimiento] [integer] NOT NULL,
    [clave_permiso] [integer] NOT NULL,
    [fecha] [datetime] NOT NULL,
    [hora] [datetime] NOT NULL,
    [entidad_relacion] [varchar] (20) NOT NULL,
    [checksum] [varchar] (8) NOT NULL,
    [status] [bit] NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[ACCIONES]
(
    [clave_accion] [integer] NOT NULL,
    [tipo_accion] [varchar] (10) NOT NULL,
    [descripcion] [varchar] (20) NOT NULL,
    [checksum] [varchar] (8) NOT NULL,
    [status] [bit] NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[IMPRESIONES]
(
    [clave_vehiculo] [bigint] NOT NULL,
    [tabla_maestra] [bigint] NOT NULL,
    [clave_rechazo] [integer] NOT NULL,
    [clave_tipo_certificado] [integer] NOT NULL,
    [estado_anterior] [integer] NOT NULL,
    [certificado_anterior] [integer] NOT NULL,
    [tipo_cert_anterior] [integer] NOT NULL,
    [crc_anterior] [varchar] (8) NOT NULL,
    [estado_actual] [integer] NOT NULL,
    [certificado_actual] [integer] NOT NULL,
    [tipo_cert_actual] [integer] NOT NULL,
    [crc_actual] [varchar] (8) NOT NULL,
    [periodo] [integer] NOT NULL,
    [checksum] [varchar] (8) NOT NULL,
    [status] [bit] NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[PRUEBAS_2540_5024]
(
    [clave_vehiculo] [bigint] NOT NULL,
    [tipo_prueba] [integer] NOT NULL,

```

```

[humos]           [bit]           NOT NULL,
[thp_humo]        [float]         NOT NULL,
[hc_b]           [integer]       NOT NULL,
[co_b]           [float]         NOT NULL,
[co2_b]          [float]         NOT NULL,
[o2_b]           [float]         NOT NULL,
[nox_b]          [integer]       NOT NULL,
[lambda]         [float]         NOT NULL,
[temp]           [float]         NOT NULL,
[hr]             [float]         NOT NULL,
[psi]            [float]         NOT NULL,
[fcnox]          [float]         NOT NULL,
[fcdl]           [float]         NOT NULL,
[rpm]            [integer]       NOT NULL,
[kph]            [float]         NOT NULL,
[thp]            [float]         NOT NULL,
[volts]          [float]         NOT NULL,
[hc]             [integer]       NOT NULL,
[co]             [float]         NOT NULL,
[co2]            [float]         NOT NULL,
[coco2]          [float]         NOT NULL,
[o2]             [float]         NOT NULL,
[nox]            [integer]       NOT NULL,
[eficiencia]     [float]         NOT NULL,
[checksum]       [varchar] (8) NOT NULL,
[status]         [bit]           NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[CERTIFICADOS]
(
  [clave_tipo_certificado] [integer] NOT NULL,
  [certificado_inicial]   [integer] NOT NULL,
  [certificado_final]     [integer] NOT NULL,
  [checksum]               [varchar] (8) NOT NULL,
  [status]                 [bit]     NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[DATOS_CIRCULACION_PAGOS]
(
  [clave_vehiculo]       [bigint] NOT NULL,
  [tarjeta_circulacion] [varchar] (15) NOT NULL,
  [fecha_tarjeta_circulacion] [datetime] NOT NULL,
  [derechos_verificacion] [varchar] NOT NULL,
  [folios_multas]        [integer] NOT NULL,
  [folio_baja_placas]    [integer] NOT NULL,
  [folio_alta_placas]    [integer] NOT NULL,
  [nivel_holograma_solicitado] [integer] NOT NULL,
  [checksum]              [varchar] (8) NOT NULL,
  [status]                [bit]     NOT NULL
) ON [PRIMARY]

```

GO

```
CREATE TABLE [dbo].[DATOS_PROPIETARIO]
(
    [clave_municipio]          [integer]          NOT NULL,
    [clave_estado]            [integer]          NOT NULL,
    [clave_propietario]       [bigint]           NOT NULL,
    [nombre]                   [varchar] (20)    NOT NULL,
    [apel_paterno]             [varchar] (20)    NOT NULL,
    [apel_materno]             [varchar] (20)    NOT NULL,
    [calle]                     [varchar] (20)    NOT NULL,
    [colonia]                   [varchar] (20)    NOT NULL,
    [num_ext]                   [varchar] (5)     NOT NULL,
    [num_int]                   [varchar] (5)     NOT NULL,
    [cp]                        [integer]          NOT NULL,
    [checksum]                  [varchar] (8)     NOT NULL,
    [status]                    [bit]              NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[DATOS_VEHICULOS]
(
    [clave_vehiculo]           [bigint]           NOT NULL,
    [clave_propietario]       [bigint]           NOT NULL,
    [tabla_maestra]           [bigint]           NOT NULL,
    [clave_motivo]             [integer]          NOT NULL,
    [clave_servicio]           [integer]          NOT NULL,
    [clave_clase_vehiculo]     [integer]          NOT NULL,
    [clave_combustible]        [integer]          NOT NULL,
    [clave_carroceria]         [integer]          NOT NULL,
    [modelo]                    [integer]          NOT NULL,
    [placas]                   [varchar] (10)    NOT NULL,
    [serie_vehiculo]           [varchar] (20)    NOT NULL,
    [serie_motor]              [varchar] (20)    NOT NULL,
    [factura_serie]            [varchar] (20)    NOT NULL,
    [odometro]                  [integer]          NOT NULL,
    [peso_bruto_asm]           [float]            NOT NULL,
    [tipo_vehiculo]            [char] (1)        NOT NULL,
    [intentos_verificacion]    [integer]          NOT NULL,
    [sistema_escape]           [char] (1)        NOT NULL,
    [tapon_combustible]        [char] (1)        NOT NULL,
    [tapon_aceite]             [char] (1)        NOT NULL,
    [bayoneta]                  [char] (1)        NOT NULL,
    [ventilador_carta]         [char] (1)        NOT NULL,
    [canister]                  [char] (1)        NOT NULL,
    [filtro_aire]               [char] (1)        NOT NULL,
    [mangueras_motor]          [char] (1)        NOT NULL,
    [mangueras_tanque]         [char] (1)        NOT NULL,
    [protocolo_pruebas]        [char] (1)        NOT NULL,
    [checksum]                  [varchar] (8)     NOT NULL,
    [status]                    [bit]              NOT NULL
) ON [PRIMARY]
GO
```

## **CAPÍTULO 5**

### **ANÁLISIS DE LAS VENTAJAS E INCOVENIENTES DE LA PROPUESTA**

Ahora ya que tenemos completada la propuesta partiendo desde su fase de análisis hasta el desarrollo final es también importante analizar las ventajas en todos los aspectos posibles que tiene esta propuesta sobre las Bases de Datos que hoy en día están en operación en los Centros de Verificación, sin embargo también es importante estar concientes de las posibles “desventajas” que esta propuesta puede tener.

#### **5.1 VENTAJAS DE LA BASE DE DATOS**

Para satisfacer las necesidades del almacenamiento de datos en el proceso de verificación vehicular se creo una base de datos hecha a la medida, partiendo de un análisis que contemplo todas las posibles variantes.

La base de datos se diseño respetando el modelo entidad-relación, se identificaron las posibles entidades y todas las relaciones entre estas, lo cual asegura que la información almacenada tendrá coherencia y lógica.

Se puso mucho interés en lograr la robustez necesaria, la base de datos esta preparada para recuperarse de cualquier contingencia. Se cuenta con respaldos totales e incrementales y procedimientos de restauración.

La base de datos tiene varios niveles de seguridad:

- El primer nivel es el esquema de autenticación del usuario, para que una persona pueda acceder la base de datos por medio de la futura aplicación (fuera del alcance de esta tesis) deberá tener un usuario con su respectiva contraseña.
- El segundo nivel dicta que cada usuario esta asociado a un puesto y cada puesto esta asociado a una serie de privilegios, los cuales restringen el numero de acciones que esta persona pueda ejecutar en la base de datos, lo que significa que no por tener acceso a la base de datos tendrá control total. Con este punto hemos tenido que hacer un esfuerzo extra por identificar los posibles roles y su nivel de acción dentro de la base de datos. También se contemplo la elaboración de ciertas vistas para restringir la información que visualizan los diferentes usuarios.

- El tercer nivel son unos disparadores<sup>1</sup> que escriben a una tabla de movimientos (bitácora) todas las acciones sensitivas realizadas por los usuarios, es importante saber siempre que hizo cada usuario.

Todo el proceso de documentación y esquematización sobre el modelo de la base de datos asegura que futuras generaciones tengan un punto de arranque para diseñar la aplicación que se conectara con la base de datos, además de formar la principal herramienta para el entendimiento de la construcción de la Base de Datos.

Una de las columnas principales de este proyecto fue crear una rigurosa estandarización, al desarrollar la base de datos se obtuvo un código homogéneo y fácilmente legible para cualquier persona letrada en el lenguaje Transact SQL Server.

Actualmente existen diferentes proveedores que ponen en el mercado varios sistemas de software vehicular, todos estos están regulados por la Secretaria de Ecología. Pero no se tiene la seguridad de que estos sistemas no sean modificados ilegalmente con fines de lucro. Con la actual propuesta de base de datos y una futura aplicación la Secretaria de Ecología estaría en la posición de contar con un sistema único de Verificación Vehicular, de tal forma que se tendría la total confianza de que no se malversaría ningún interés. La Secretaria de Ecología tendría un control total en los futuros cambios y su mantenimiento.

## **5.2 INCONVENIENTES DE LA BASE DE DATOS.**

Pasando primero por el ámbito económico la información que se encuentra en muchos Centros de Verificación no cuenta con un Gestor de Bases de Datos formal como Microsoft SQL Server sino que la información en ocasiones cuenta con el programa de Access e incluso en la mayoría de los casos esta solamente se encuentra almacenada en tablas con formato “dbf” contenidas en alguna carpeta de los servidores y el manejo de estas tablas pasa a ser responsabilidad de la aplicación.

Debido a esto, al utilizar el diseño que nosotros proponemos significa adquirir licencias para la administración y funcionamiento de la Bases de Datos, las cuales representan un costo mas alto comparando con utilizar simplemente archivos con extensión dbf manejados por la aplicación o con programas como Access el cual viene incluido en un paquete de Oficina. Todo esto se podría ver como una desventaja monetaria la comparación de costos en licencias entre los programas que se utilizan en la actualidad en los Centros de Verificación y el Gestor que nosotros proponemos para la utilización de nuestra propuesta.

---

<sup>1</sup> Mejor conocidos como “triggers”.



Otro de los aspectos que es importante resaltar y aunque solamente se aplica en algunos casos es la utilización de un equipo adecuado para el manejo, administración y funcionamiento de la Base de Datos, ¿porqué comentamos que solamente en algunos casos? Porque el 80% de los Centros de Verificación ya cuentan con un equipo Servidor que aunque no es un equipo como el que podemos ver en operación en un banco que son totalmente especializados y grandes servidores este nos permite tener la suficiente eficiencia para las operaciones necesarias en el trabajo diario de un Centro de Verificación Vehicular, sin embargo todavía se encuentran algunos que todavía no les convence la idea de que al mejorar su equipo de cómputo por uno más eficiente podrán carecer de fallas que constantemente ocurren en sus procesos o simple y sencillamente la lentitud con la que trabajan hasta ahora en sus procesos debido al decadente equipo de cómputo que utilizan hasta ahora y que todo esto se traduce en la conocida frase de “el tiempo es dinero”.

Por este punto, podríamos ver como una segunda desventaja en nuestra propuesta *la obligación de utilizar un equipo servidor* lo suficientemente capaz de brindar un correcto funcionamiento al Gestor de la Base de Datos, ya que en los casos que no cuentan con servidor acostumbran utilizar una PC debido a que hasta ahora la utilización de Access o de archivos dbf les permite trabajar utilizando las PCS como servidores y en caso de poner a funcionar nuestra propuesta en estos equipos personales todo el trabajo por mejorar la eficiencia en los procesos de estos Centros se vendría abajo. Más claramente la desventaja sería que los Centros de Verificación que no han invertido en un mejor equipo de cómputo en cuanto a su equipo servidor ya con nuestra propuesta se verían obligados a hacerlo representando esto un costo para ellos, pero a su vez tendría que ver el aspecto de costo-beneficio que nuestra propuesta daría.

Ahora que sabemos las ventajas y las posibles desventajas de nuestra propuesta y que el proyecto de tesis que realizamos está terminado podremos dar nuestras propias conclusiones acerca del trabajo que realizamos durante este lapso de tiempo y la experiencia que adquirimos.

## **CONCLUSIÓN**

Se puede decir que en el tiempo en el cual realizamos este proyecto aprendimos mucho sobre todos y cada uno de los pasos con los cuales hicimos posible que este proyecto llegara a completarse.

Desde la fase de recopilación de información la cual por tratarse de un tema el cual se encuentra en operación hoy en día, se tuvo que hacer una investigación y estudio sobre información de las últimas tecnologías sobre el manejo de Bases de Datos pero a su vez reconocer que fueran tecnologías que pudieran ser compatibles con ciertas características de los Centros de Verificación y con sus reglas de negocio ya que no podíamos realizar una propuesta que generara un cambio total en la infraestructura de estos Centros de Verificación vehicular, sino que pensamos en una propuesta que junto con el material que ya se tiene mejorar en mucho los procesos, seguridad y confiabilidad de la información.

Una de las fases que resultó sin duda más complicada de realizar fue la de diseño de la Base de Datos, donde se obtuvo el diagrama del Modelo Entidad-Relación ya que teníamos que generar un diseño que fuera posible de ser extensible, robusto y exacto así que varias veces se obtuvo un diseño que constantemente y por varias semanas se modificó una y otra vez hasta hacerlo lo mejor posible.

El objetivo general de esta propuesta es romper con la costumbre de soportar un proceso tan importante como el de la verificación vehicular en software mal diseñado, no robusto y de dudosa confiabilidad por el mal uso al que en estos momentos esta siendo expuesto.

Esperamos que este primer esfuerzo sea seguido de distintas propuestas que critiquen y propongan una solución razonable a las diferentes problemáticas que caen en el ámbito gubernamental. De tal manera que será deseable que los futuros ingenieros que estén apunto de escoger tema de tesis contemplen problemáticas actuales y posibles áreas de oportunidad para mejorar cualquier ámbito social a mediano y largo plazo, ya que es cuestión del implicado demostrar que la Ingeniería en Computación es real, es decir, aplicable a cualquier ámbito de la vida.

## ANEXOS

<b>Glosario</b>	
10/100 baseT	Especificación de Ethernet
Analizador	Equipo de verificación de emisiones vehiculares
ASCII	Código que expresan los caracteres
ASM	Aceleración Simulation Mode. Modo de aceleración simulada
BAR	Bureau of Automotive Repair, California EE.UU..
Baudios	Velocidad de transferencia de datos en bytes por Segundo
BD	Base de Datos
BHP	Caballos de potencial al freno
BIOS	Sistema Básico de Entrada y Salida
Bps	Bits por segundo
C	Grados Celsius o centígrados
CAM	Comisión Ambiental Metropolitana
CENAM	Centro Nacional de Metrología
Checksum	Algoritmo de Verificación de los datos
CO	Monóxido de Carbono
CO2	Bióxido de carbono
CO + CO2	Dilución
CODASYL	Conference on Data System Languages
DBMS	Data Base Management System
DBGT	Data Base Task Group
DB_25	Conector de 25 contactos
DGCA	Dirección General de Control al Ambiente
DIS	Sistema de Ignición del vehículo
DS0	Especificación de líneas de comunicación
EPA	Environmental Protection Agency , Estados Unidos
FEP	Factor de Equivalencia del Propano
GLP	Gas Licuado de Petróleo
GNC	Gas Natural Comprimido
GUAN	General Update Access Method
HC	Hidrocarburos
Hg.	Mercurio
Hub	Nodo Central de Conexión
IBM	Internacional Business Machines Corporation
IDS	Integrate Data Store
INE	Instituto Nacional de Ecología
IMECAS	Unidad de medición de partículas contaminantes
ISO	Organización Internacional para la regulación
JICA	Japan Internacional Cooperation Agency
K	Coefficiente de Absorción de luz
Kg.	Kilogramo
Km./h	Kilómetros por hora
Km.	Kilómetro
KPa	KiloPascal
KW	KiloWatt
L	Litro
LAN	Red Local
LEV	Low Emisión Vehicles

Login	Procedimiento de conexión
Mb	Megabyte
Microswitch	Micro-interruptor
mm.	Milímetro
Ms	Milisegundos
MHz	MegaHertz
M	Metro
NA	No aplicable
NAA	North American Aviation
NDIR	Infrarrojo no dispersivo
NIST	National Institute of Science and Technology
NO	Oxido Nitrogeno
NOx	Óxidos de Nitrógeno
NOM	Norma Oficial Mexicana
O2	Oxigeno
OBD	On Board Diagnostics. Diagnostico a bordo
Odómetro	Medidor de kilómetros recorridos
OCDE	Organización para la Cooperación y el Desarrollo Económico
Opacimetro	Medidor de opacidad
Park	Estacionar
PAS	Prueba de Aceleración Simulada
PBV	Peso Bruto Vehicular
PCBench	Programa de Evaluación de computadoras
Ppm	Partes por millón
Ppmh	Partes por millón de hexano
Psig	Libras por pulgadas cuadrada relativo a la presión atmosférica
RF	Frecuencia de Radio
RFC	Registro Federal de Causantes
RJ45	Diseño de Conector
Rpm	Revoluciones por minuto
RS232	Interfase de Comunicación
SAE	Sociedad de Ingenieros Automotrices
SEDUE	Secretaría de Desarrollo Urbano y Ecología
SEMARNAT	Secretaria de Medio Ambiente y Recursos Naturales
SGBD	Sistema Gestor de Base de Datos
SQL	Structure Query Language
Tacómetro	Medidor de resoluciones del motor
Ton	Tonelada
T- SQL	Transac SQL
TÜV	Entidad de Control Técnico de Alemania
UL	Underwriter Laboratorios
ULEV	Ultra Low Emisión Vehicles
UML	Unified Model Language
UPS	Fuente no Interruptible de Poder
VEC	Etiqueta de Control de Emisiones del Vehiculo
VIN	Numero de Identificación del Vehiculo
WAN	Red Amplia de Comunicación
ZMVM	Zona Metropolitana del Valle de México

## Programa de Verificación Vehicular

Los propietarios de automóviles deberán realizar el trámite de Verificación Vehicular de acuerdo al siguiente calendario:

Engomado	Terminación	Período	Segundo Semestre
5y6	5 y 6	Primer Semestre Enero y Febrero	Julio y Agosto
7y8	7 y 8	Febrero y Marzo	Agosto y Septiembre
3y4	3 y 4	Marzo y Abril	Septiembre y Octubre
1y2	1 y 2	Abril y Mayo	Octubre y Noviembre
9y0	9 y 0	Mayo y Junio	Noviembre y Diciembre

Antes de presentar su automóvil a verificar, cerciórese de cumplir con lo siguientes requisitos:	
<b>Documentación</b> <ul style="list-style-type: none"> <li>▶ Certificado original de la verificación anterior.</li> </ul>	<b>Vehículo</b> <ul style="list-style-type: none"> <li>▶ Correctamente afinado</li> </ul>
<ul style="list-style-type: none"> <li>▶ Holograma anterior adherido al automóvil.</li> </ul>	<ul style="list-style-type: none"> <li>▶ Sistema de escape original en buen estado.</li> </ul>
<ul style="list-style-type: none"> <li>▶ Tarjeta de circulación (Original y copia por los dos lados.)</li> </ul>	<ul style="list-style-type: none"> <li>▶ Filtro de aire, tapón del depósito de combustible, tapón y bayoneta del depósito de aceite.</li> </ul>
<ul style="list-style-type: none"> <li>▶ Comprobante de pago de multa por verificación Extemporánea (Únicamente en caso de así requerirlo).</li> </ul>	<ul style="list-style-type: none"> <li>▶ Dispositivos anticontaminantes en buen estado y conectados.</li> </ul>

**Tabla de Programa de Verificación Vehicular**  
<http://www.portalautomotriz.com/tramites/verificacion/>

## **BIBLIOGRAFÍA**

- Pérez, César.  
Domine Microsoft. SQL Server 2000  
Administración y Análisis de Base de Datos.  
Ed. Alfaomega Ra-Ma Edición Original.  
ISBN. 84-7897-546-2.
  
- Coordinador responsable García Gutiérrez, Alfonso.  
Revisión e integración Franco Sandoval, Porfirio.  
Asesor Rogers Anderson, John Allen  
Manual técnico de verificación automotriz.  
Ed. Secretaría de Medio Ambiente y Recursos Naturales SEMARNAT  
e Instituto Nacional de Ecología INE.  
ISBN: 968-817-543-7
  
- [http://www.inegi.gob.mx/est/librerias/tabulados.asp?tabulado=tab\\_po02b&c=706&e=09](http://www.inegi.gob.mx/est/librerias/tabulados.asp?tabulado=tab_po02b&c=706&e=09)
  
- <http://www.portalautomotriz.com/tramites/verificacion/>
  
- <http://www.portalautomotriz.com/tramites/verificacion/verificentros/>
  
- <http://www.portalautomotriz.com/tramites/verificacion/verificentros/edomex.php>.