

DISEÑO E IMPLEMENTACIÓN DE UN SCANNER AUTOMOTOR PARA EL
DIAGNÓSTICO DE LOS SENSORES DE OXIGENO, TEMPERATURA Y
CHOQUE PARA EL VEHÍCULO CHEVROLET BLAZER 1995 PARA EL
TALLER AUTOMOTRIZ 1-A.

YUNELIS GUTIÉRREZ SERRANO
LESLY MILAGRO RODRIGUEZ MONTAÑO
HÉCTOR RAFAEL TORRES CABALLERO

Asesor:
JOHAN MANCERA

UNIVERSIDAD SIMÓN BOLÍVAR
FACULTAD DE INGENIERÍA DE SISTEMA
INVESTIGACIÓN FORMATIVA IV
SEMESTRE X DIURNO
BARRANQUILLA
2006.



DISEÑO E IMPLEMENTACIÓN DE UN SCANNER AUTOMOTOR PARA EL
DIAGNÓSTICO DE LOS SENSORES DE OXIGENO, TEMPERATURA Y
CHOQUE PARA EL VEHÍCULO CHEVROLET BLAZER 1995 PARA EL TALLER
AUTOMOTRIZ 1-A.

A Dios, quien nos da su Sabiduría
Yunelis Gutiérrez Serrano.

A Dios, el amigo que nunca nos falla.
Lesly M. Rodríguez Montaña.

A Dios, el único que nos puede proteger.

Héctor R. Torres Caballero

AGRADECIMIENTO

Primeramente a Dios le agradecemos que nos haya facilitado el camino para llegar a la meta.

A nuestros padres que con su esfuerzo hicieron posible realizar nuestros estudios superiores y poderlos culminar.

A nuestros hermanos que nos mantuvieron siempre con ánimo.

Un agradecimiento especial a nuestro tutor Johan Mancera quien nos ayudó a desarrollar poco a poco este proyecto con sus tutorías y mucha dedicación hasta el final de las distintas etapas de formativa.

A los profesores de Formativa quienes nos guiaron durante todo el proceso investigativo.

A la Universidad Simón Bolívar por ser quien ha permitido llevar a cabo esta investigación y formarnos como unos profesionales idóneos.

Por facilitarnos la impresora y computadores al CAMFI (Centro de Apoyo a las Matemáticas y Física) de la Universidad Simón Bolívar.

Por facilitarnos la portátil a los Ingenieros Luís Brito, Enrique Martelo, Yuleima Medina.

Por prestarnos la tarjeta de crédito para la obtención de los materiales al Prof. Víctor Márceles.

Por dejarnos trabajar en sus instalaciones al taller Automotriz 1-A.

Por su inestimable apoyo a los Ingenieros Roberto Osio y Leonardo Martínez, a los profesores . Albeiro Zabala, Diego Rolong,

A nuestros amigos que nos brindaron su apoyo y conocimientos para que este proyecto se llevara a cabo.

En fin a todos aquellos, que de una u otra forma nos ayudaron en la realización de este proyecto de investigación.

CONTENIDO

	Pág.
RESUMEN	1
INTRODUCCIÓN	2
1. PLANTEAMIENTO DEL PROBLEMA	3
1.1 DESCRIPCIÓN DEL PROBLEMA	3
1.2 FORMULACIÓN DEL PROBLEMA	5
1.3 SISTEMATIZACIÓN DEL PROBLEMA	5
2. OBJETIVOS DEL PROYECTO	6
2.1. OBJETIVO GENERAL	6
2.2 OBJETIVOS ESPECÍFICOS	6
3. JUSTIFICACIÓN	7
4. MARCO DE REFERENCIA	8
4.1 MARCO TEÓRICO	8
4.2 MARCO CONCEPTUAL	24
4.3 MARCO ESPACIAL	26
4.4 MARCO LEGAL	26
5. METODOLOGÍA	27
5.1 TIPO DE ESTUDIO	27
5.2 LÍNEA DE INVESTIGACIÓN	27
5.3 POBLACIÓN Y MUESTRA	27
5.3.1 POBLACIÓN	27
5.3.2 MUESTRA	27
5.4 RECOLECCIÓN DE INFORMACIÓN	28
5.5 CRONOGRAMA DE ACTIVIDADES	29
6. RECURSOS	31
6.1 RECURSOS HUMANOS	31
6.2 RECURSOS FINANCIEROS	31
6.3 RECURSOS DEL ENTORNO	31



7. INGENIERÍA DE REQUISITOS	32
7.1 ÁMBITO DEL SOFTWARE	32
7.2 IDENTIFICACIÓN DE REQUISITOS	32
7.3 ANÁLISIS DE REQUISITOS	33
7.4 ESPECIFICACIÓN DE REQUISITOS	33
8. ANÁLISIS DEL SISTEMA	34
8.1 MODELO ENTIDAD-RELACIÓN	34
8.2 MODELO RELACIONAL	35
8.3 CASOS DE USOS	36
9. DISEÑO DEL SISTEMA	38
9.1 MODELADOR DE PROCESO	38
9.2 DISEÑO ELETRÓNICO DE LA INTERFAZ	39
10. GESTIÓN DE RIESGO	47
11. GESTION DE CALIDAD	49
12 CONCLUSIONES	58
BIBLIOGRAFÍA	
ÍNDICE	

Resumen

A pesar de los avances tecnológicos en la industria automotriz, por muchos años se ha mantenido en secreto la forma en que se leían las computadoras de los vehículos. Cada fabricante tenía su propia forma de realizarlo. Obligando así a que cuando un carro presentaba avería tenía que ir donde el fabricante, o los talleres tenían que adquirir costosos scanner proveídos por los propios fabricantes y, en el peor de los casos los pequeños talleres no podían realizar un diagnóstico a los vehículos.

Por medio de la especificaciones dada por la Sociedad de Ingenieros de Automoción o SAE (Society of Automotive Engineers) y los protocolos de comunicación tales como las normas ISO9141, ISO9141-2, ISO14230, SAEJ1850, se trata de conseguir un sistema a un bajo costo y confiable que sea capaz de leer los códigos de fallas, los valores reales de los sensores que están presentes en la computadora de un automóvil con tecnología OBD (On-Board Diagnostic) II y protocolo ISO J1850 VPW (Variable Pulse Width). El sistema debe recibir información de la computadora del vehículo a través del cable OBD, el sistema decodificará los datos los buscará en la base de datos para luego ser mostrados al usuario con su descripción.

Al inicio del proceso de investigación se hará un estudio de la documentación actualizada sobre los temas relacionados con el Sistema OBD de los automóviles para luego comenzar a desarrollar el scanner automotor mediante el paradigma de la ingeniería del software llamado modelo espiral.

INTRODUCCIÓN

Hoy en día los vehículos están equipados con una computadora a bordo llamado sistema OBD (On Board Diagnostic), encargado de guardar los códigos de fallas y los valores reales de los sensores presentes en el vehículo.

Para poder acceder a esta computadora es necesario un software encargado de interactuar con el sistema OBD y así poder realizar el diagnóstico por el cual está fallando el vehículo.

Estos softwares, llamados Scanner, son demasiado costosos en el mercado para pequeños talleres automotores que deseen modernizarse y utilizar la tecnología para el mejoramiento del servicio.

Con este Proyecto de Investigación se quiere desarrollar un Scanner automotor para el taller Automotriz 1-A, que sea capaz de acceder a la computadora de un Chevrolet Blazer 1995 y realice un auto diagnóstico del mismo.

1. PLANTEAMIENTO DEL PROBLEMA.

1.1 DESCRIPCION DEL PROBLEMA.

El Taller Automotriz 1-A ubicado en la calle 74 # 41D-11 en la ciudad de Barranquilla, tiene como finalidad prestar servicio automotor a los propietarios de vehículos cuando estos tengan fallas en su funcionamiento.

En el momento de la revisión del automóvil los mecánicos realizan un diagnóstico de una forma manual utilizando un tester que lo conecta en el conector OBD que todos los vehículos actuales tienen equipados, el mecánico interpreta los códigos arrojados por medio del check engine y detecta los sensores que están fallando y de esta manera sabrá donde dirigirse y solucionar el problema. Esta tarea por ser manual gasta un tiempo alrededor de 10 a 15 minutos.

Los códigos obtenidos, deben ser interpretados, en forma específica, recurriendo al manual del vehículo, ya que, cada fabricante programa su computadora con sus propios códigos.

Dicho de otra forma, el vehículo mediante la activación de señales luminosas, le informa al mecánico que elementos están funcionando mal, con el fin de prevenirle y éste tome las medidas de reparación correspondientes.

Conociendo los avances que existen en la actualidad, el Taller Automotriz 1-A quiere optimizar el proceso de revisión de posibles fallas que se le presentan a los vehículos con ayuda de la tecnología.

1.2 FORMULACIÓN DEL PROBLEMA.

¿Cómo incrementar la eficiencia en el Taller Automotriz 1-A en la revisión de fallas detectadas en el auto Chevrolet Blazer 1995 para modernizar y rentabilizar el taller?

1.3 SISTEMATIZACION DEL PROBLEMA

¿Cómo prestar un mejor servicio a los clientes del taller automotriz 1-A?

¿Cómo tener un buen personal que garantice la confiabilidad de los clientes?

¿Cómo tener un mejor control sobre los datos del vehículo?

2. OBJETIVOS DEL PROYECTO

2.1 OBJETIVO GENERAL

Implementar un scanner automotor para diagnosticar los sensores de oxígeno, temperatura y choque para incrementar la eficiencia en la revisión de las fallas mecánicas del auto Chevrolet Blazer 1995.

2.2 OBJETIVOS ESPECIFICOS

- Reducir los costes del taller con el fin de tener mayor ingreso.
 - Incrementar la rentabilidad global del taller.
 - Solucionar las carencias de conocimientos técnicos.
 - Disminuir las tareas administrativas.
 - Mejorar el rendimiento de los operarios.
 - Implementar nuevas tecnologías en el taller
-

3. JUSTIFICACIÓN.

La mecánica automotriz ha evolucionado los últimos años, tanto es así, que los vehículos vienen equipados con computadoras. El OBD es un sistema que generaliza la forma de leer los códigos de ésta computadora a bordo, también, monitorea y detecta fallas permanentes o intermitentes de los sensores, estos son pequeños dispositivos que miden las condiciones de operación y las traducen en señales que la computadora puede entender.

Existen dispositivos llamados técnicamente "scanner" (lectores de códigos), utilizados en algunos talleres automotores, los cuales son conectados al puerto de salida OBD de la computadora de los vehículos, y se inicia una interacción entre el lector y la máquina, pero estos scanner en el mercado son demasiado costosos y el Taller Automotriz 1-A no dispone de los recursos económicos para adquirir uno e instalarlo en su taller. Es por eso que el siguiente proyecto logrará diseñar el scanner automotriz que diagnosticará los sensores de oxígeno, temperatura y de choque para el vehículo Chevrolet Blazer 1995, de una manera más económica.

De esta manera los mecánicos del Taller Automotriz 1-A se le facilitarán leer los códigos de la computadora del vehículo para saber que, cual, o cuantos sensores tienen malos y junto con su experiencia, conocimiento y capacidad podrán controlar los daños del vehículo.

Con el éxito de este proyecto se estará impulsando el desarrollo tecnológico local ya que este producto se fabrica únicamente en los países desarrollados como por ejemplo EUA.

4. MARCO DE REFERENCIA.

4.1 MARCO TEÓRICO.

Hoy en día los automóviles vienen equipados con un sistema de diagnóstico a bordo llamado OBD (On-Board_Diagnostic) encargado de monitorear de cerca el consumo de combustible, minimizar la emisión de gases contaminantes y ayuda a los talleres de reparación, en el diagnóstico de fallas.

Cuando un vehículo presenta fallas en el sistema de control de emisiones la computadora a bordo le avisa al conductor a través del check engine ubicado en el cuadro de instrumentos indicándole la necesidad de llevarlo a mantenimiento. El mecánico en el taller puede recuperar el código de avería almacenado en la memoria de la computadora a bordo del auto, para diagnosticar las causas del problema. Esta tarea se puede realizar de dos formas posibles, la primera es hacerlo manualmente y la segunda, motivo de este trabajo de investigación, por medio de un scanner automotor.

Para desarrollar el scanner automotor hay que tener en cuenta que se comporta como un software de tiempo real¹, según Roger Pressman un software de tiempo real es aquel que coordina, analiza y controla sucesos del mundo real conforme ocurren, entonces el scanner automotor estará en la capacidad de saber lo que sucede en el interior del vehículo en cualquier momento que sea conectado con la computadora a bordo.

1 ¹ PRESSMAN, Roger S. Ingeniería del Software: Un Enfoque Práctico. 5^{ta} Edición. Ciudad de México, México: MCGRAW HILL

El hardware del scanner automotor será la interfaz entre el sistema OBD de la Chevrolet Blazer 1995 y el software de tiempo real ubicado en la computadora del taller. A través del conector ISO 15031-3 de 16 pines ubicado en la zona del conductor, el scanner automotor puede obtener los códigos de fallas (DTC - Diagnostic Trouble Code). Existen tres tipos de protocolos de comunicación entre el Scanner automotor y el conector, definidos por la SAE (Society of Automotive Engineers) el J1850 VPW, J1850 PWM e ISO 9141-2. La Chevrolet Blazer 1995 utiliza el J1850 VPW (Variable Pulse Width) lo que traduce modulación por ancho de pulso variable, aquí la información se transmite en un bus sencillo, entre los pines 2 y 5, el pin 16 es para la batería y el "1" lógico es de +7v con una velocidad de transmisión de 10.4 kbits/sec.

El scanner automotor se encargará de diagnosticar los sensores de temperatura, oxígeno y de choque de la Chevrolet Blazer 1995. El sensor de de temperatura tiene la función de monitorear la temperatura dentro del motor, el de oxígeno se encarga de olfatear los gases residuales de la combustión y el de choque tiene la tarea de cortar el flujo de gasolina cuando el vehículo se choca.

El scanner automotor será una integración de hardware más software desarrollado por el grupo de investigación para proveer al Taller Automotriz 1A de una excelente herramienta para facilitar la diagnosis de averías en los sistemas electrónicos del automóvil.

ANTECEDENTES HISTÓRICOS:

A finales de los 70 se empieza a incorporar la electrónica a los automóviles, se

añaden los primeros sensores a los motores para verificar su correcto funcionamiento, también se añadieron unidades de control del motor que manejaban dichos sensores. El objetivo inicial de estos elementos electrónicos era el control de las emisiones de gases contaminantes y facilitar la diagnosis de averías.

A partir de la década de los 80 la mayor parte de las innovaciones provienen principalmente de la incorporación de la electrónica, y no de la incorporación de mejoras mecánicas. Se añadieron multitud de sensores y se fueron mejorando las unidades de control del motor. Hoy en día un automóvil puede incorporar más de 200 sensores y más una unidad de control. Hay unidades de control para el motor, climatizador, airbag, etc.

Las primeras unidades de control eran Módulos de Control de Motor o ECM (Engine Control Module), con el tiempo estas ECMs se hicieron más complejas y pasaron a convertirse en Unidades de Control Electrónico o ECU (Electronic Control Unit), estas ECUs son las conocidas como centralistas o UCEs (siglas en español, Unidad de Control Electrónico).

Actualmente los sensores se encargan de la medición de temperaturas, presiones, rotaciones, volúmenes, y multitud de parámetros de funcionamiento. La información que captan los sensores es enviada y almacenada en las centralistas. Toda esta información permite que el propio automóvil "conozca" su estado. En realidad, los sensores se limitan a detectar una serie de valores que envían a la centralista y una vez allí son comparados con los valores óptimos que están



almacenados en las memorias. Cuando se encuentra un valor incorrecto, la centralista notifica un fallo avisando al conductor de alguna forma (indicadores luminosos, sonidos, etc.), los fallos quedan almacenados para su posterior verificación por un mecánico. En algunas ocasiones, las centralistas simplemente fallan sin notificar ningún error.

Cuando un coche sufre una avería, el taller mecánico usará un scanner para conectarse con la centralista del vehículo. Un scanner es un dispositivo que se conecta a una centralista para acceder a los datos que esta tiene almacenados en la memoria, en teoría es capaz de consultar los fallos y otros valores almacenados.

APARICIÓN DEL OBD

Para combatir los problemas de polución en Los Ángeles, el Estado de California exigió sistemas de control de emisiones de gases en los modelos de automóvil posteriores a 1966. El Gobierno Federal de los Estados Unidos extendió estos controles a toda la nación en 1968. El Congreso aprobó el Clean Air Act (Acta Antipolución) en 1970 y creó la Agencia de Protección Medioambiental o EPA (Environmental Protection Agency). La EPA inició el desarrollo de una serie de estándares en la emisión de gases y unos requerimientos para el mantenimiento de los vehículos con el fin de ampliar su vida útil.

Para cumplir estos estándares los fabricantes implementaron sistemas de

encendido y de alimentación controlada de combustible, con sensores que median las prestaciones del motor y ajustaban los sistemas para conseguir una mínima polución. Estos sensores también permitían una cierta ayuda en la reparación.

En abril de 1985 un organismo estatal de California, el CARB (California Air Resources Board), aprobó una regulación para un sistema de diagnóstico a bordo u OBD Esta regulación especifica que el Modulo de Control de Motor o ECM (Engine Control Module) debe monitorizar ciertos componentes del vehículo relacionados con las emisiones de gases para asegurar un correcto funcionamiento, y que se ilumine una lámpara Indicadora de Fallo o MIL (Malfunction Indicator Lamp) en el cuadro de mandos cuando se detecta un problema. El sistema OBD también aporta un sistema de códigos de Error de Diagnóstico o DTC (Diagnostic Trouble Codes) y unas tablas de errores en los manuales de reparación para ayudar a los técnicos (mecánicos) a determinar las causas más probables de avería en el motor y problemas en las emisiones. Los objetivos básicos de esta regulación son fundamentalmente dos:

- Reforzar el cumplimiento de las normativas de la regulación de la emisión de gases alertando al conductor cuando se presenta un fallo.
- Ayudar a los técnicos de reparación de automóviles (mecánicos) en la identificación y reparación de fallos en el sistema de control de emisiones.

Los Sistemas OBD se encuentran en la mayoría de los automóviles y camiones ligeros actuales. Durante la década de los 70 y principio de los 80 se introdujeron componentes electrónicos para cumplir los estándares de emisión de gases de la EPA, posteriormente la implantación de sistemas OBD para controlar funciones del motor y diagnosticar problemas supuso una mayor complejidad en la

electrónica integrada en los vehículos. A través de los años los sistemas OBD se han hecho más sofisticados. OBD-II, un nuevo estándar introducido a mediados de los 90, aporta un control casi completo del motor también monitoriza partes del chasis y otros dispositivos del vehículo, asimismo es el centro de control de diagnóstico del coche. Con el tiempo los primitivos Módulos de Control de Motor o ECMs se han hecho más complejas y han pasado a convertirse en las actuales Unidades de Control Electrónico o ECU (Electronic Control Unit), verdaderas 'cajas negras' de un vehículo.

Inicialmente hubo varios estándares y cada fabricante tenía sus propios sistemas y códigos. En 1988 la Sociedad de Ingenieros de Automoción o SAE (Society of Automotive Engineers) definió un conector estándar y un conjunto de códigos de diagnóstico. La EPA adoptó la mayoría de los estándares y recomendaciones de la SAE sobre las aplicaciones OBD. Posteriormente con OBD-II, un conjunto más amplio de estándares y sistemas también definidos por la SAE y adoptado por la EPA y el CARB es aprobado para su implementación el 1 de enero de 1996.

Para aplicar los estándares OBD, OBD-II se utilizan protocolos de comunicación tales como las normas ISO9141, ISO9141-2, ISO14230 (KWP2000), SAEJ1850, SAEJ1979, CAN BUS y VAN BUS entre otros. Algunos protocolos han sido definidos por ISO o SAE.

ELEMENTOS DEL SCANNER.

El scanner esta compuesto por un cable OBD (hardware) y un software del que trata este proyecto de investigación que unidos entre sí extraen los datos de las UCEs de los vehículos.

FUNCIONAMIENTO GENERAL DE LA UCE

En los vehículos fabricados hasta el año 1996, generalmente había una única UCE en la que sus distintos circuitos electrónicos actuaban como UCEs independientes, de manera que al entrar en cada una de ellas parece que están separadas y sin embargo forman un mismo conjunto. Un mecánico intentando realizar una reparación aparentemente se encontraría con varias UCEs, aunque realmente solo habría una.

Para comunicarse con las UCEs lo primero que debe hacer el scanner es despertarlas o activarlas, intentando buscar una respuesta de la UCE con la que quiere comunicarse. Si hay respuesta, entonces Se sabe que existe dicha UCE y se puede comunicarse con ella para extraer códigos de averías, borrar averías, cambiar parámetros que utiliza como referencia o incluso leer valores que esta recibiendo la UCE del vehículo, consiguiendo así aumentar o disminuir las rpm, codificar el cuadro para que utilice un idioma u otro, etc.

Una vez establecida la comunicación con la UCE lo primero que se hace es identificarla, y en los tiempos muertos para mantener la comunicación se

intercambian alternativamente paquetes de datos indicando confirmación (ACK).

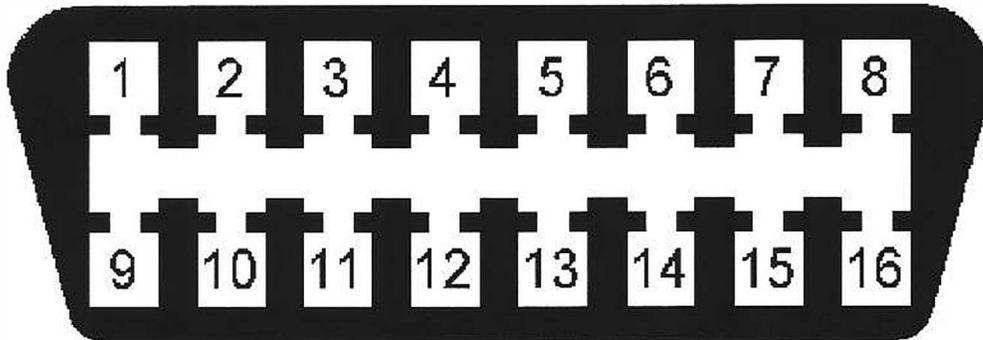
PROTOCOLOS

Los protocolos de comunicaciones empleados para la conexión de las centralistas o UCEs a los scanners presentan incompatibilidades entre si. Por desgracia cada fabricante ha optado por utilizar sus propios protocolos, es decir, un fabricante puede utilizar distintos protocolos en distintos modelos de vehículo e incluso un mismo fabricante puede utilizar distintos protocolos en un mismo modelo según varíe su año de fabricación. En ocasiones las diferencias pueden ser mínimas pero suficientes para que existan incompatibilidades.

PROTOCOLO J1850 VPW

El conector debe tener los pines 2, 4,5 y 16 pero no el 10. Puede o no tener más pines presente. La información se transmite en un bus sencillo, entre el contacto 2 y 5 el "1" lógico es de +7v. La velocidad de transmisión es 10.4 kbit/seg.

CONECTOR OBD II ISO 15031-3



2	-			J1850	+
4	-	Masa	del	vehículo	5 - Masa de la Señal
6	-			CAN	High (J-2284)
7	-			ISO	9141-2 Línea K
10	-				J1850
14	-			CAN	Low (J-2284)
15	-			ISO	9141-2 Línea L
16	-				Batería +

FORMATO DE LOS CÓDIGOS DE FALLAS²

La SAE (Sociedad Americana de Ingenieros) publicó la Norma J2012 para estandarizar el formato de los códigos de diagnóstico.

El formato asigna códigos alfanuméricos a las fallas y provee una guía de mensaje uniformes asociados con estos códigos, los DTC consiste en un código numérico de tres dígitos, precedidos por un designador alfanuméricos definido de la siguiente manera:

- PO: Código del PCM.
- El tercer dígito representa al sistema en el cual la falla ocurre.
- El cuarto y quinto representan al DTC específico para dicho sistema.

CODIGOS P0100-P0199

- P0100: Mal funcionamiento del circuito del flujo masa o volumen de aire
- P0101: Circuito de variación* de resistencia; del flujo de masa o volumen de aire, con problemas de funcionamiento
- P0102: entrada débil del circuito del flujo de masa o volumen de aire
- P0103: Entrada Intensa del Circuito del Flujo de Masa o Volumen de Aire
- P0104: Circuito del Flujo de Masa o Volumen de Aire Intermitente
- P0105: Mal Funcionamiento del circuito; Presión Barométrica /presión Absoluta del Múltiple de Admisión

² PETER, Davis. OBD II Fault Code Referents. Florida, Estados Unidos: KOTZING PUBLISHING, 1999. PP 200. ISBN 0971541140



- P0106: Circuito de variación* de señal de la Presión Barométrica /Presión Absoluta del Múltiple de Admisión; Problemas de Funcionamiento.
 - P0107: Presión Absoluta del Manifold/Sensor Barométrico, Señal Entrada Baja
 - P0108: Presión Absoluta del Manifold/Sensor Barométrico, señal de Entrada Alta
 - P0109: Presión Absoluta del Manifold/Sensor Barométrico, señal intermitente
 - P0110: Mal funcionamiento del, Circuito de temperatura de entrada de aire.
 - P0111: Circuito de variación* de señal, de la Temperatura del aire en el múltiple de entrada con problemas de funcionamiento.
 - P0112: Señal de entrada débil del Circuito de la Temperatura del aire en el múltiple de entrada.
 - P0113: Señal de entrada intensa del Circuito de la Temperatura del aire en el múltiple de entrada.
 - P0114: Señal Intermitente del Circuito de la Temperatura del aire en el múltiple de entrada.
 - P0115: Mal Funcionamiento del Circuito de Temperatura del agua en el motor
 - P0116: Circuito de variación de la señal de la temperatura de agua en el motor con problemas de funcionamiento.
 - P0117: Señal de entrada débil, del circuito de temperatura de refrigerante en el motor
 - P0118: Señal de entrada intensa, del circuito de temperatura de refrigerante en el motor
 - P0119: Señal Intermitente del circuito de temperatura de refrigerante en el motor
-

- P0120: (Hyundai) Sensor de posición del acelerador, Mal Funcionamiento del circuito del conmutador [circuito de variación de voltaje]
 - P0121: Circuito de variación de resistencia del, Sensor-interruptor de posición de la garganta de aceleración, con problemas de funcionamiento
 - P0122: Señal entrada débil, del circuito del sensor interruptor de posición de la garganta de aceleración
 - P0123: Señal Intensa de entrada del circuito del sensor interruptor de posición, de la garganta de aceleración
 - P0124: Señal Intermitente del circuito del sensor interruptor de posición, de la garganta de aceleración
 - P0125: Temperatura insuficiente refrigerante en el motor, para cerrar el circuito de control de combustible.
 - P0126: Temperatura Insuficiente refrigerante, para una operación estable
 - P0128: Range/Performance Problem In Thermostat (Acura)
 - P0128: ECT Below Thermostat Regulating Temperature (GM)
 - P0130: Sensor de Oxígeno, Mal funcionamiento del circuito [banco 1 Sensor 1]
 - P0131: Sensor de Oxígeno, Voltaje Bajo en el circuito [banco 1 Sensor 1]
 - P0132: Sensor de Oxígeno, Voltaje alto en el circuito [banco 1 Sensor 1]
 - P0133: Sensor de Oxígeno. Respuesta lenta [Banco 1 sensor 1]
 - P0134: Sensor de Oxígeno, no se detecta actividad en el circuito [banco 1 Sensor 1]
 - P0135: Sensor de Oxígeno, mal funcionamiento del circuito, de la resistencia que lo calienta [Banco 1 Sensor1]
-

- P0136: Sensor de Oxígeno Mal Funcionamiento del circuito [Banco 1 sensor 2]
 - P0137: Sensor de Oxígeno, Voltaje Bajo [banco 1 Sensor2]
 - P0138: Sensor de Oxígeno, Voltaje Alto [Banco 1 sensor 2]
 - P0139: Sensor de Oxígeno, respuesta Lenta del circuito [banco 1 Sensor 2]
 - P0140: Sensor de Oxígeno, no se detecta actividad en el circuito [banco 1 sensor 2]
 - P0141: Sensor de Oxígeno, mal funcionamiento del circuito de la resistencia que lo calienta, [Banco 1 sensor 2]
 - P0142: Sensor de Oxígeno, Mal funcionamiento del circuito [banco 1 Sensor 3]
 - P0143: Sensor de Oxígeno, Voltaje Bajo en el circuito [Banco1 Sensor3]
 - P0144: Sensor de Oxígeno Voltaje Alto en el circuito [banco 1 sensor3]
 - P0145: Sensor de Oxígeno, respuesta lenta del circuito [banco 1 sensor 3]
 - P0146: Sensor de Oxígeno, no se detecta actividad en el circuito [banco1 sensor 3]
 - P0147: Sensor de Oxígeno, mal funcionamiento del circuito de la resistencia que lo calienta, [Banco 1 sensor 3]
 - P0150: Sensor de Oxígeno, Mal funcionamiento del circuito [banco 2 Sensor 1]
 - P0151: Sensor de Oxígeno, Mal funcionamiento del circuito [banco 2 Sensor 1]
 - P0152: Sensor de Oxígeno Voltaje Alto en el circuito [banco 2 sensor 1]
 - P0153:) Sensor de Oxígeno, respuesta lenta del circuito [banco 2sensor 1]
-

- P0154: Sensor de Oxígeno, no se detecta actividad en el circuito [banco2 sensor 1]
 - P0155: Sensor de Oxígeno, mal funcionamiento del circuito de la resistencia que lo calienta, [Banco 2 sensor 1]
 - P0156: Sensor de Oxígeno, Mal funcionamiento del circuito [banco 2 Sensor 2]
 - P0157: Sensor de Oxígeno, Voltaje Bajo en el circuito [Banco 2 Sensor 2]
 - P0158: Sensor de Oxígeno Voltaje Alto en el circuito [banco 2 sensor 2]
 - P0159: Sensor de Oxígeno, respuesta lenta del circuito [banco 2 sensor 2]
 - P0160: Sensor de Oxígeno, no se detecta actividad en el circuito [banco2sensor 2]
 - P0161: Sensor de Oxígeno, mal funcionamiento del circuito de la resistencia que lo calienta, [Banco 2 sensor 2]
 - P0162: Sensor de Oxígeno, Mal funcionamiento del circuito [banco 2 Sensor 3]
 - P0163: Sensor de Oxígeno, Voltaje Bajo en el circuito [Banco1 Sensor3]
 - P0164: Sensor de Oxígeno Voltaje Alto en el circuito [banco 2 sensor 3]
 - P0165: Sensor de Oxígeno, respuesta lenta del circuito [banco 2 sensor 3]
 - P0166: Sensor de Oxígeno, no se detecta actividad en el circuito [banco 2 sensor 3]
 - P0167: Sensor de Oxígeno, mal funcionamiento del circuito de la resistencia que lo calienta, [Banco 2 sensor 3]
 - P0170: Mal funcionamiento del compensador de gasolina [Banco 1]
 - P0171: Sistema Mezcla demasiado pobre [banco 1]
 - P0172: Sistema Mezcla demasiado rica [banco 1]
-

- P0173: Mal funcionamiento del compensador de gasolina [Banco 2]
 - P0174: Sistema Mezcla demasiado pobre [banco 2]
 - P0175: sistema Mezcla demasiado pobre [banco2]
 - P0176: Circuito Averiado, del Sensor de composición de combustible
 - P0177: Circuito de variación* de resistencia, del Sensor de composición de combustible, con problemas de funcionamiento.
 - P0178: Señal de entrada débil del Circuito, del Sensor de composición de combustible
 - P0179: Señal Alta del Circuito, del Sensor de composición de combustible
 - P0180: Mal funcionamiento del circuito, sensor "A" de temperatura del combustible
 - P0181: Circuito de variación* de resistencia, del sensor "A" de temperatura de combustible, con problemas de funcionamiento.
 - P0182: Señal débil de entrada, sensor A de temperatura de combustible
 - P0183: Señal intensa de entrada, sensor A de temperatura de combustible
 - P0184: Señal Intermitente, sensor A de temperatura de combustible
 - P0185: Mal funcionamiento del circuito, sensor "B" de temperatura del combustible
 - P0186: Circuito de variación* de resistencia, del sensor "B" de temperatura de combustible, con problemas de funcionamiento.
 - P0187: Señal débil de entrada, sensor "B" de temperatura de combustible
 - P0188: Señal intensa de entrada, sensor "B" de temperatura de combustible
 - P0189: Señal Intermitente, sensor "B" de temperatura de combustible
-

- P0190: Mal funcionamiento en el circuito del sensor de presión de combustible en el riel de inyectores.
 - P0191: Circuito de variación* de resistencia, del sensor de presión de combustible en el riel de inyectores, con problemas de funcionamiento.
 - P0192: Señal de entrada débil, en el circuito del sensor de presión de combustible, en el riel de inyectores
 - P0193: Señal intensa de entrada, en el circuito del sensor de presión de combustible, en el riel de inyectores
 - P0194: Señal Intermitente, en el circuito del sensor de presión de combustible, en el riel de inyectores
 - P0195: mal funcionamiento del sensor de temperatura de aceite del motor
 - P0196: Circuito de variación* de resistencia, del sensor de temperatura de aceite en el motor, con problemas de funcionamiento.
 - P0197: Señal débil del sensor de temperatura de aceite del motor
 - P0198: Señal intensa del sensor de temperatura de aceite del motor
 - P0199: Señal Intermitente del sensor de temperatura de aceite del motor
-

4.2 MARCO CONCEPTUAL.

- 1 **AUTODIAGNÓSTICO:** La computadora a bordo hace un diagnóstico de todos los sensores del automóvil.
- 2 **CHECK ENGINE:** Lámpara de advertencia ubicada en el cuadro de instrumento del vehículo que destella cuando el carro tiene algún problema en el funcionamiento interno.
- 3 **CÓDIGO DE FALLA:** Es un número que se almacena en la memoria de la computadora asociado a un problema específico del automóvil.
- 4 **CONECTOR TIPO ISO 15031-3:** Conector de prueba normalizado de 16 pines, localizado en la zona del conductor, debajo del panel de instrumentos.
- 5 **DTC:** Abreviatura de Diagnostic Trouble Code. Diagnóstico de códigos de fallas.
- 6 **MANUAL DTC:** Contiene todos los códigos de fallas con la descripción de problema.
- 7 **OBD:** Abreviatura de On Board Diagnostic, Este es un sistema que generaliza la forma de leer los códigos de la computadora a bordo. Reúne los requisitos para monitorear y detectar las fallas permanentes que podrían hacer que el auto funcione mal.
- 8 **SENSOR DE CHOQUE:** Su función es cortar el flujo de gasolina al motor cuando el vehículo se choca.



- 9 **SENSOR DE OXÍGENO:** Su función es olfatear los gases residuales de la combustión. Solo trabaja o funciona estando caliente. Tiene la particularidad de generar corriente, variando el voltaje de 1v en cuanto siente residuos altos o bajos de oxígeno interpretado como una mezcla rica o pobre, dando lugar a que la computadora ajuste la mezcla tratando de equilibrar una mezcla correcta.
- 10 **SENSOR DE TEMPERATURA:** Su función es monitorear la temperatura dentro del motor; de esta manera la computadora al recibir la señal de que el motor alcanzó la temperatura de trabajo procede a ajustar la mezcla y el tiempo de encendido.
- 11 **SENSORES:** Son todos los componentes colocados en diferentes partes del motor que están conectados con la computadora a bordo del vehículo.
- 12 **SISTEMA DE EMISIONES:** Se le llama así a todos los componentes encargados de ventilar y quemar, los residuos contaminantes.
-

4.3 MARCO ESPACIAL.

Este trabajo de investigación se hará en la ciudad de Barranquilla para el Taller Automotriz1-A. Ubicado en la calle 74 # 41D-11.

4.4 MARCO LEGAL

El reglamento CEE 1400/2002 sobre venta y posventa de vehículos expone con claridad que a fin de garantizar la competencia efectiva del mercado de servicios de reparación y mantenimiento. Los fabricantes de vehículos de motor deberán permitir a los talleres independientes el acceso a la información técnica, equipos de diagnóstico y otros, incluidos los programas informáticos de acceso a la información técnica necesaria para reprogramar los dispositivos electrónicos de un vehículo de motor.

5. METODOLOGÍA.

5.1 TIPO DE ESTUDIO.

El tipo de estudio de este trabajo de investigación es explorativo porque es la primera vez que se trabaja el sistema OBD de los vehículos en esta Universidad.

5.2 LÍNEA DE INVESTIGACIÓN.

La línea de investigación a seguir en este proyecto de investigación es la de estándares y paradigma de Ingeniería de software.

5.3 POBLACION Y MUESTRA.

5.3.1 POBLACION

La población está conformada por los trabajadores y los vehículos del taller automotriz 1-A.

5.3.2 MUESTRA

Cómo la población es pequeña se toma como muestra

5.4 RECOLECCION DE INFORMACION

Al inicio del proceso de investigación se hará un estudio de la documentación actualizada sobre los temas relacionados con el Sistema OBD de los automóviles para luego comenzar a desarrollar el scanner automotor mediante el paradigma de la ingeniería del software llamado modelo espiral. Este modelo comprende 6 marcos de trabajos llevados a cabo de una forma iterativa durante los cuatro semestres del proyecto de investigación las cuales son:

- 1 Comunicación con el taller Automotriz 1-A.
- 2 Planificación.
- 3 Análisis de riesgo.
- 4 Ingeniería.
- 5 Construcción y acción.
- 6 Prueba, entrega y evaluación del cliente.

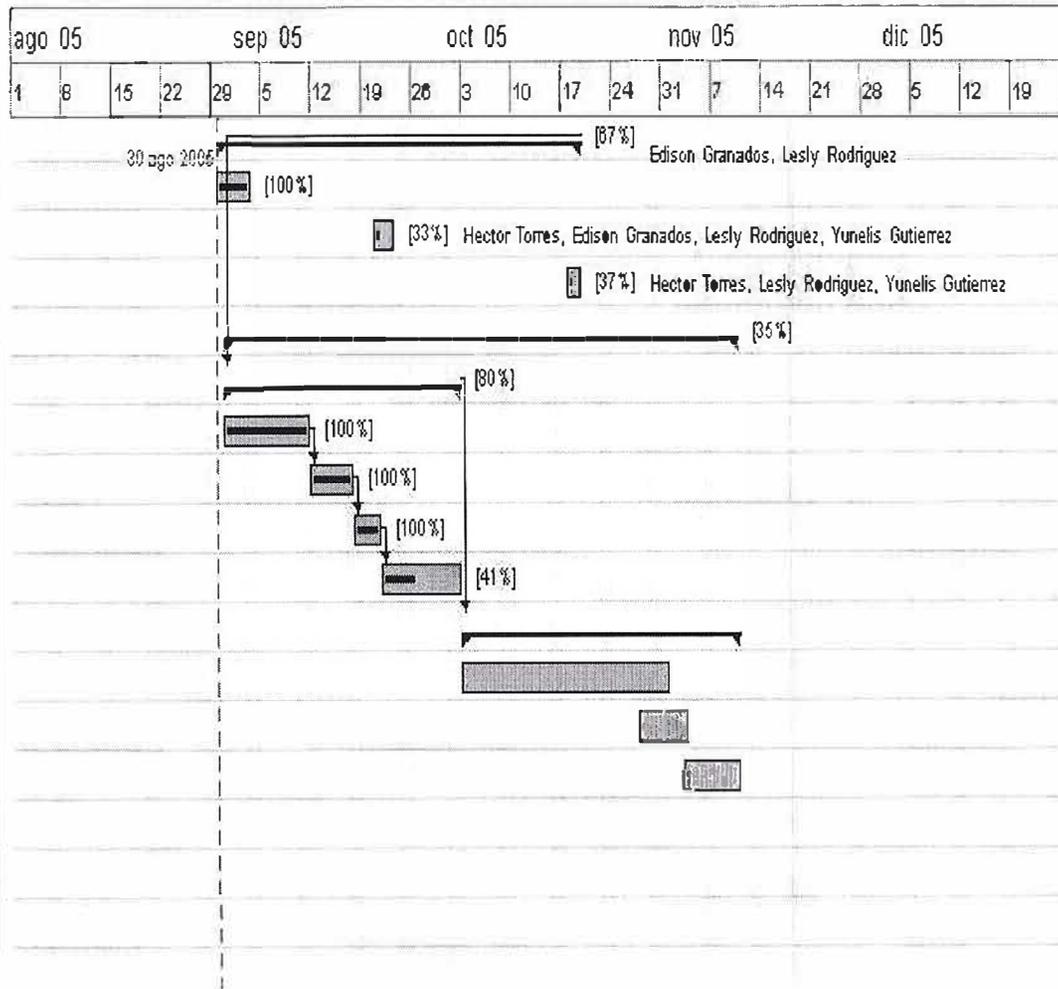
Este proceso vuelve e inicia en la etapa 1 para conocer la evaluación del cliente.

La recolección de información se llevo a cabo a través de una entrevista con el propietario del taller automotriz 1-A la cual se filmó.

5.5 CRONOGRAMA DE ACTIVIDADES

Entrevista con el Taller Automoto 1-a
Primera Entrevista
Segunda Entrevista
Tercera Entrevista
Planeación y Elaboración
Análisis
Identificación de Requisitos
Análisis de Requisitos
Especificación de Requisitos
Elaboración de Casos de Usos
Diseño
Diseño de Interfaces
Diseño de Datos
Diseño Prototipo
Codificación
Puebas
Puesta en Marcha

GanttProject(1.10.3)



Cronograma # 3

12 Nombre	13 Inicio	14 Fin	15 Notas	16 Asignar a
Encriptación	25/09/06	26/09/06	Esta tarea se realizó con éxito	Lesly Rodríguez
Documentación	26/09/06	5/11/06		
Manual de Usuario	26/09/06	1/11/06	En desarrollo	Yunelis Gutiérrez
Manual de Sistema	16/10/06	5/11/06	En desarrollo	Héctor Torres
Ayudas En líneas	20/10/06	27/10/06	En desarrollo	Lesly Rodríguez, Yunelis Gutiérrez
Importar/ Exportar DTC	2/10/06	8/10/06		Lesly Rodríguez
Desarrollo	30/09/06	15/10/06		
Velocímetro	30/09/06	8/10/06	Esta tarea se realizó con éxito	Héctor Torres
RPM	9/10/06	15/10/06	No se ha realizado	Héctor Torres
Gráfico	9/10/06	14/10/06		
Temperatura	9/10/06	14/10/06	No se ha realizado	Héctor Torres
Velocidad-Tiempo	9/10/06	12/10/06	No se ha realizado	Héctor Torres
Pruebas	1/11/06	30/11/06		Grupo De Investigación, Taller Automotriz 1-A, Tutor
Puesta en Marcha	27/10/06	11/11/06	No se ha realizado	Taller Automotriz 1- A
Implementación	1/11/06	16/11/06	No se ha realizado	
Asesoráis	30/09/06	26/11/06		Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres
Asesora 1	30/09/06	1/10/06	Esta tarea se realizó con éxito ver actas de asesorías	Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres
Asesora 2	7/10/06	8/10/06	Esta tarea no se realizó actas de asesorías	Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres
Asesora 3	14/10/06	15/10/06	Esta tarea se realizó con éxito actas de asesorías	Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres
Asesora 4	21/10/06	22/10/06	Esta tarea se realizó con éxito actas de asesorías	Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres
Asesora 5	28/10/06	29/10/06	Esta tarea se realizó con éxito actas de asesorías	Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres
Asesora 6	4/11/06	5/11/06	Esta tarea se realizó con éxito actas de asesorías	Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres
Asesora 7	11/11/06	12/11/06	Esta tarea se realizó con éxito actas de asesorías	Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres
Asesora 8	18/11/06	19/11/06	Esta tarea se realizó con éxito actas de asesorías	Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres
Asesora 9	25/11/06	26/11/06	Esta tarea se realizó con éxito actas de asesorías	Joan Mancera, Lesly Rodríguez, Yunelis Gutiérrez, Héctor Torres

6. RECURSOS.

6.1 RECURSOS HUMANOS.

EL desarrollo del proceso investigativo se realizará con el respaldo intelectual y metodológico de los integrantes del grupo de investigación, además, contará con el apoyo del ingeniero de sistema Johan Mancera.

6.2 RECURSOS FINANCIEROS.

CD.	\$20.000
Disquete.	10.000
Video 8.0.	9.000
Transporte al taller.	90.000
Horas de Internet.	30.000
1 Circuito ELM322	270.000
1 Cable convertidor de USB a Serial	50.000
Gasolina	30.000

6.3 RECURSOS DEL ENTORNO.

- 1 Un computador de escritorio y portátil.
 - 2 Impresora.
 - 3 CD, disquetes
 - 4 Memoria USB.
 - 5 Vehículo Chevrolet Blazer 1995
-

7 INGENIERÍA DE REQUISITOS

7.1 ÁMBITO DE SOFTWARE: SCANNER AUTOMOTOR

Se trata de conseguir un sistema que sea capaz de leer los códigos de fallas, los valores reales de los sensores de oxígeno, temperatura y choque que están almacenados en la computadora de un auto Chevrolet Blazer Modelo 95 con tecnología OBD II y protocolo ISO J1850 VPW. El sistema debe recibir información de la computadora del vehículo a través del cable OBD, el sistema decodificará los datos los buscará en la base de datos para luego ser mostrados al usuario con su descripción, Además el sistema permitirá guardar información referente a los vehículos y propietarios que lleguen al taller los cuales podrán ser consultado más adelante.

7.2 IDENTIFICACIÓN DE REQUISITOS

Los requisitos que solicitó el Taller Automotriz 1-A son:

- Llevar un historial de los vehículos que llegan al taller con sus respectivas fallas.
 - Detectar los códigos de fallas del vehículo.
 - Interpretar los códigos detectados.
 - Desplegar una posible solución.
 - Leer los sensores de los vehículos.
-

7.3 ANÁLISIS DE REQUISITOS

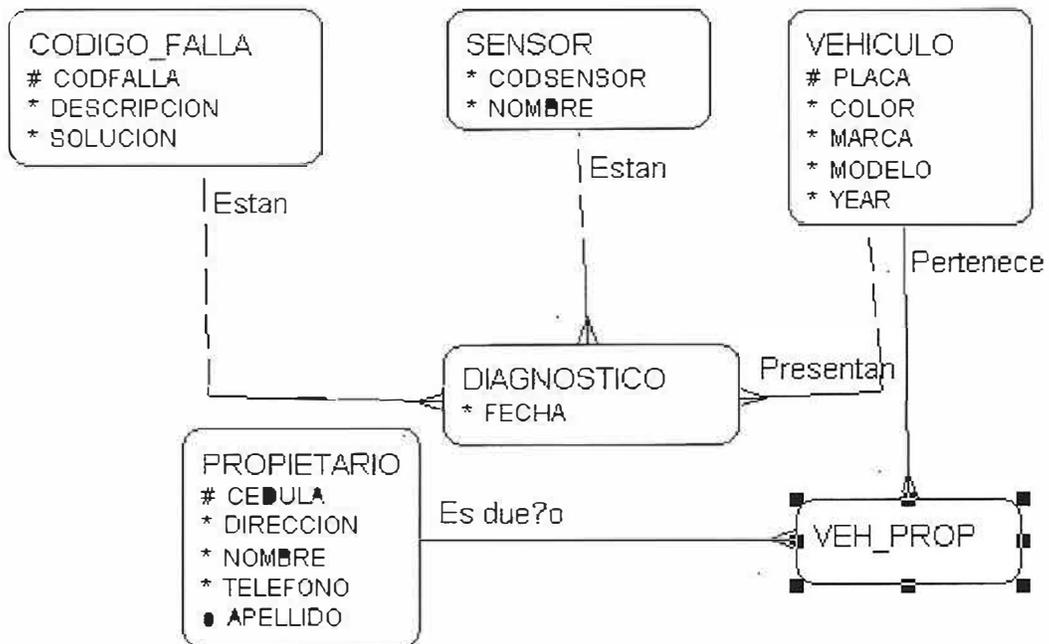
- **Llevar un historial de los vehículos:** El sistema será capaz de mantener una base de datos de todos los vehículos que lleguen al taller para su reparación y/o mantenimiento, donde se podrá consultar las veces y el motivo por el cual cada vehículo ha llegado al taller.
- **Detectar los códigos de fallas:** El sistema estará en capacidad de hacer las lecturas de las averías del automóvil cuando se conecten al scanner
- **Interpretar los códigos detectados:** El sistema decodificará los códigos de fallas leídos y los comparará con los códigos de fallas almacenados en la base de datos.
- **Desplegar una posible solución:** El sistema dispondrá de una guía completa con soluciones que orientarán al mecánico.
- **Leer los sensores del vehículo:** El sistema será capaz de leer los valores de los sensores en tiempo real.

7.4 ESPECIFICACIÓN DE REQUISITOS

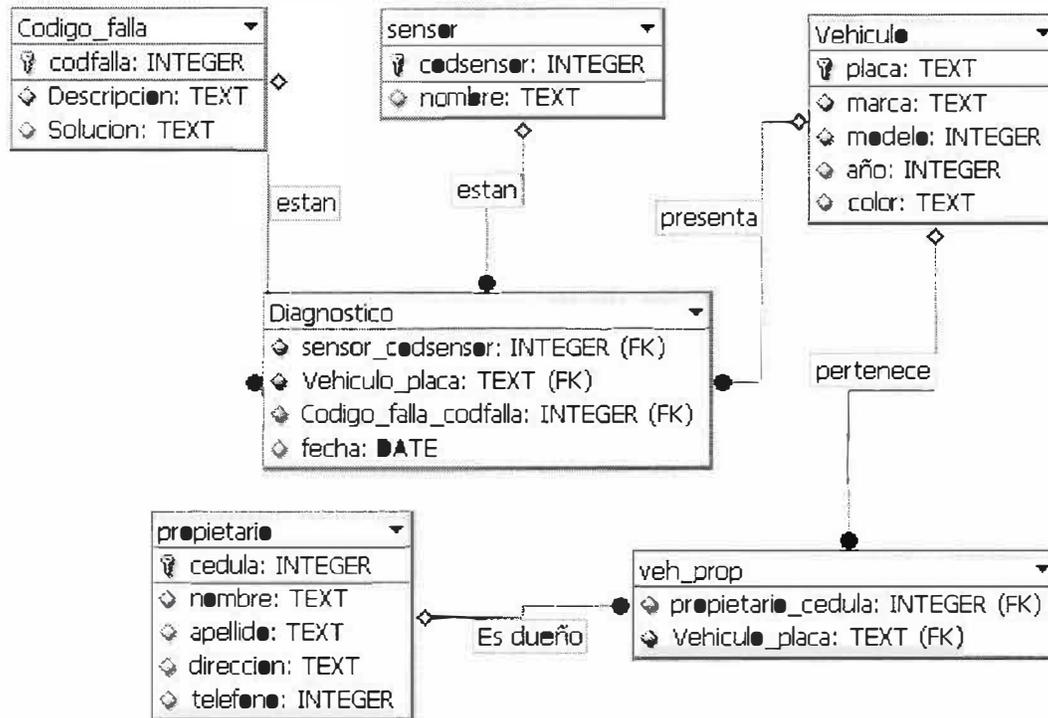
- Llevar un historial de los vehículos que llegan al taller con sus respectivas fallas.
 - Detectar los códigos de fallas del vehículo.
 - Interpretar los códigos detectados.
 - Desplegar una posible solución.
 - Leer los sensores de los vehículos.
-

8. ANÁLISIS DEL SISTEMA

8.1 MODELO ENTIDAD – RELACIÓN



8.2 MODELO RELACIONAL



8.3 CASOS DE USOS DEL SISTEMA

Caso de uso: Gestionar vehículos

Actores: Cliente, mecánico, Sistema.

Propósito: Capturar la información del vehículo

Tipo: Primario

Descripción: Este caso de uso empieza cuando un cliente llega con el vehículo para su mantenimiento y/o reparación. El mecánico guarda la información respectivamente en el sistema.

Caso de uso: Leer Códigos de fallas

Actores: Mecánicos

Propósito: Leer los códigos de fallas de la computadora del vehículo

Tipo: Primario

Descripción: Este caso de uso comienza cuando el mecánico conecta el scanner con el puerto OBDII del vehículo para leer códigos de fallas

Caso de uso: Decodificar los códigos arrojados por el scanneo

Actores: Sistema

Propósito: Saber cuál o cuales son los códigos de fallas del vehículo

Tipo: Primario

Descripción: Este caso de uso comienza cuando el sistema ya tiene los códigos leídos del scanneo, luego son consultados en la base de datos.

Caso de uso: Leer sensores

Actores: Sistema

Propósito: Saber cual es el valor de los sensores

Tipo: Primario

Descripción: Este caso de uso comienza cuando el sistema lee los valores de los sensores del vehículo.



9.2 DISEÑO ELECTRONICO DE LA INTERFAZ

La información de la computadora a bordo del vehículo es transferida serialmente por el puerto OBD de una manera específica por unos estándares de la SAE (Sociedad Americana de Ingenieros). Este puerto no es compatible con el estándar de los computadores personales. Por tal razón se implementará un circuito integrado de 8 pines conocido como ELM322³, el cual es capaz de cambiar la información y reformatear las señales OBD en caracteres ASCII. Este integrado permite a cualquier computador personal comunicarse con el vehiculo equipado con OBD usando solo un puerto serial y programa terminal.

Diagrama del IC (Circuito Integrado)

Descripciones de los pines

Vdd (pin 1): Voltaje.

XT1 (pin 2) y XT2 (pin 3): Para conectar un 3.579545 MHz.

OBDIn (pin 4): Entrada de información desde el OBD.

RX (pin 5): Entrada de información al RS232 del computador.

TX (pin 6): Salida de información del RS232 del computador.

OBDOut (pin 7): Salida de Información del OBD.

Vss (pin 8): Tierra.

³ ELM Electronics- Circuit for the Hobbyists. En Documentación [en línea].
<http://www.elmelectronics.com/DSheets/ELM322Ds.pdf> [citado el 4 de mayo de 2004].

El ELM322 depende de un tipo de conexión serial estándar RS232 para comunicarse con el usuario. El intercambio de información es a 9600 baudios, con 8 bits, bit de no paridad, 1 bit de parada y no handshaking, para abreviar (9600 8N1). Todas las respuestas del IC terminan con un retorno de carro.

Cuando el ELM322 está bien conectado se visualizará

```
ELM322 v2.0
```

```
>
```

El carácter '>' es el prompt del IC e indica que el aparato está listo para recibir caracteres del puerto RS232. Los caracteres enviados del computador pueden ser entendidos para uso interno del ELM322 o para reformatearlo y pasarlos al OBD del Vehículo.

Los comandos para el uso interno del ELM322 son distinguidos porque siempre empiezan por los caracteres AT (comandos AT), mientras los comandos para el OBD contienen solo caracteres ASCII por dígitos hexadecimales (0 a 9 y A hasta F).

El ELM322 no es sensible a mayúsculas o minúscula, así, el comando 'ATZ' es equivalente a 'atz', 'AtZ', 'ATz', etc. También ignora los espacios en blancos, tabulaciones. Para ejecutar el comando enviado debe terminar en un retorno de carro '\r'.

Comandos AT.

Los comandos AT son usados para uso interno del ELM322.

Solo usaremos:

D: Restaurar a los valores por defecto.

I: Muestra el ID ej. ELM322.

Z: Resetear EL IC ELM322.

Comandos OBD

Si los bytes recibidos del RS232 no empiezan con las letras AT, el ELM322 asume que son comandos OBD, los bytes serán probados para asegurarse de que ellos son parejas válidas de dígitos hexadecimales y si los son, serán combinados en bytes para transmitirlos al vehículos.

Después de enviar un comando, el ELM322 escucha al OBD para alguna respuesta. Cada byte recibido es convertido a su equivalente pareja hexadecimal de caracteres ASCII y transmitido al puerto serial de la computadora, si no hay respuesta del vehículo o el comando no es válido un mensaje de "NO DATA" será enviado.

Los estándares de la SAE especifican que el primer byte, conocido como el modo, siempre describe el tipo de información que está siendo solicitada, mientras los

otros especifican la información actual solicitada, dada por un 'Parameter Identification' o numero PID.

En cada modo el PID 00 es normalmente reservado para saber cuales PID son soportados por el modo.

Los nueve modos más usados son:

01: Muestra la información actual

02: Muestra la información Estática.

03: Muestra los códigos de fallas.

04: Limpia los códigos de fallas.

05: Prueba los resultados, sensor de oxígeno.

06: Prueba los resultados, sin monitoreo continuo.

07: Prueba los resultados, monitoreo continuo.

08: Control especial.

09: Solicitar información del vehículo.

La mayoría de los vehículos no responden si no tienen la llave en posición ON.

Para saber la información de los sensores se utiliza el modo 01, por ejemplo para solicitar la temperatura del refrigerante (ECT Engine coolant temperature) se usa el PID 05, el comando para enviar es

>0105

Y la respuesta de l vehículo será de la forma

4105 7B

Lo cual se interpreta:

4105: Respuesta del modo 01 del PID 05.

7B: Valor solicitado en hexadecimal.

Para saber el valor real del sensor se trasforma a decimal y se le aplica la fórmula, si la tiene.

El valor del ECT seria en decimal 123, menos 40 de la fórmula: 83 °C.

Tabla 1.0. Nombre de los Sensores con sus comandos

Nombre del Sensor	Comando
Posición abs. del estrangulador	0111
Velocidad del motor RPM	010C
Velocidad del vehículo	010D
Valor de carga calculado	0104
Avance de tiempo	010E
Presión múltiple de admisión	010B
Flujo de aire	0110
Short term fuel trim Bank 1	0106
Long term fuel trim Bank 1	0107
Short term fuel trim Bank 2	0108
Long term fuel trim Bank 2	0109
Temperatura aire de entrada	010F
Temperatura de refrigerante	0105
Presión combustible	010A
OBD conforme a:	011C
Tiempo desde el encendido	011F
Presión de gas rel. al vacío	0122
Presión de gasolina	0123
Comando EGR	0112
Comando Purga del evaporador	012E

Entrada de nivel de gasolina	012F
Encendido desde reset al ECU	0130
Distancia desde reset al ECU	0131
Presión de vapor sist.de evap.	0132
Distancia de activación de MIL	0121
Presión barométrica	0133
Temperatura CAT,B1S1	013C
Temperatura CAT,B2S1	013E
Temperatura CAT,B1S2	013D
Temperatura CAT,B2S2	013F

Interpretar Los Códigos de Fallas o DTC (Diagnostic Trouble Codes)

Primero que todo hay que saber cuántos DTC están almacenados en la computadora del vehiculo, para esto usamos el modo 01 y PID 01

>0101

Una típica respuesta sería

41 01 81 07 65 04

El 4101 significa una respuesta del modo 01 y PID 01, el primer byte de información (81), 129 en decimal, es el resultado que estamos buscando.

Para determinar el número de DTC almacenados se resta 128, si el número es mayor que 128, sino se lee directamente.

En este ejemplo $129-128=1$ DTC.

Sabiendo el número de DTC, el próximo paso es solicitar el actual DTC con el modo 03.

>03

Una respuesta podría ser

43 01 33 00 00 00 00

El 43 significa una respuesta del modo 03. Los otros seis bytes en la respuesta tienen que ser leídos en parejas para mostrar el DTC (Arriba sería interpretado como 0133, 000 y 0000), aquí solo hay un código de falla que es 0133 y los otros cero lo requiere el estándar y no representan nada.

La siguiente tabla se usa para interpretar el primer dígito de un DTC, la primera columna representa el primer dígito hexadecimal recibido y la segunda

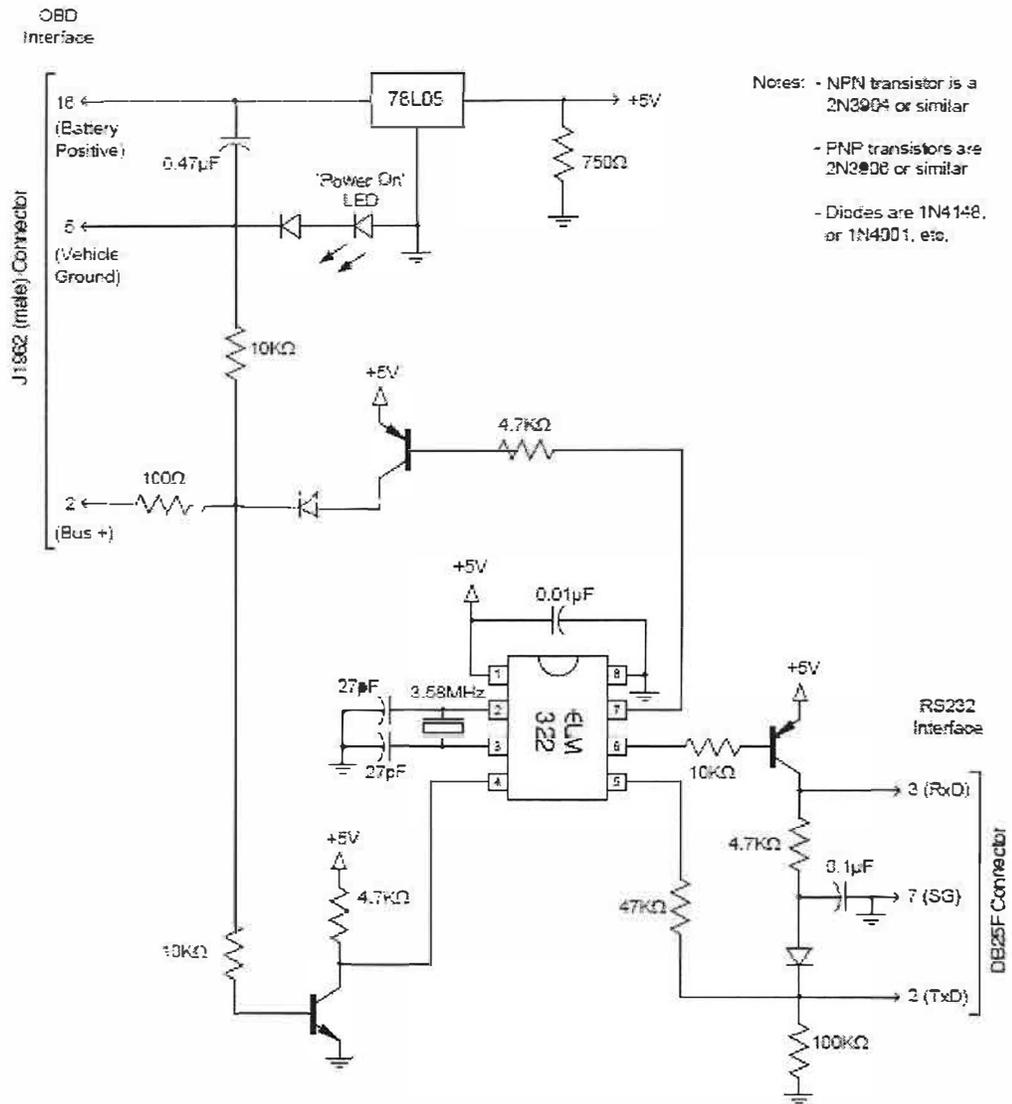
Columna con lo cual se sustituirá

0	P0
1	P1
2	P2
3	P3

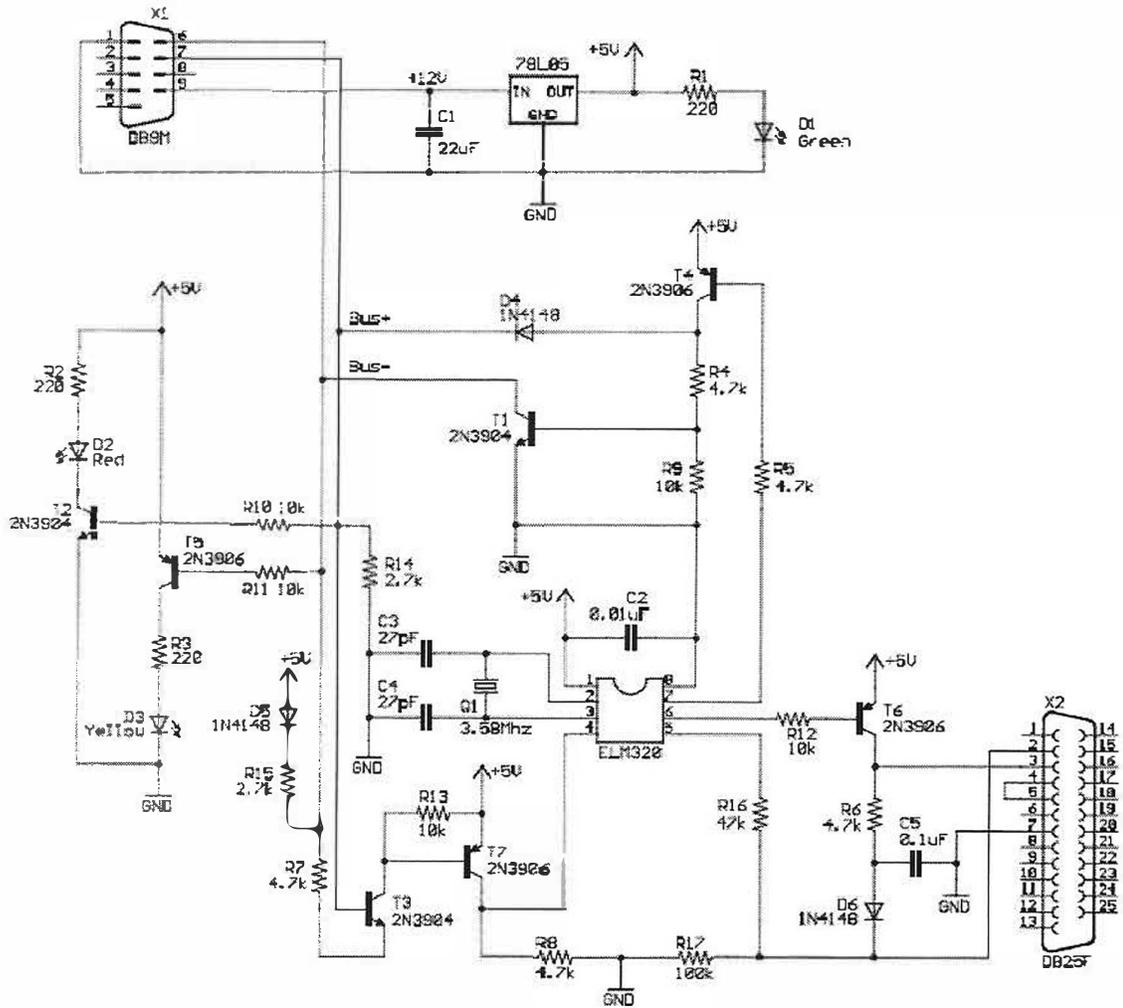
Tomando el ejemplo 0133, el primer dígito es 0, será sustituido por P0 y el DTC reportado es P0133 cuya descripción es Respuesta lenta en el circuito del sensor de O2 (Banco 1 Sensor 1).

Posibles configuraciones del circuito:

Configuración 1



Configuración 2



10 GESTIÓN DE RIESGOS

Id Riesgo	Descripción	Prob.	Impacto	Costo \$	Asignad o a	RSGR
Scann-01	Mal estimación del tamaño	20%	3	200000	YG	Utilizar métricas del tamaño de software, por medio de puntos de función
Scann-02	Mayor número de usuario de lo provisto	10%	3	1500000	LR	Realizar constantemente pruebas
Scann-03	La entrega de fecha límite estará ajustada	80%	2	300000	HT	Establecer un cronograma de actividades, especificando tareas, recursos y responsables.
Scann-04	Falta de capital	70%	3	300000	GD	Proyectar los costos, medios y/o actividades para conseguir los recursos económicos
Scann-05	El cliente cambiará los requisitos	20%	3	200000	LR	Especificar bien los requisitos a través entrevistas.
Scann-06	Falta de Material	50%	3	200000	YG	Planificar las actividades con

						sus recursos.
Scann-07	Personal Inexperto	60%	4	100000	HT	Los integrantes deben comprometerse con el proyecto y llevar actas de seguimiento con el tutor
Scann-08	Elevada movilidad del personal	5%	4	50000	LR	Dejar todo los procesos, actividades, etc. documentados
Scann-09	Número de personas adecuadas	10%	4	120000	YG	Buscar expertos en el tema por si se necesitase.
Scann-10	Ámbito del proyecto estable	10%	4	100000	HT	Definir bien los requisitos del sistemas
Scann-11	Experiencia con la tecnología	60%	2	100000	HT	Buscar asesorias por Internet

1 Catastrófico

2 Crítico

3 Marginal

4 Despreciable

HT Héctor Torres

YG Yunelis Gutiérrez

LR Lesly Rodríguez

11. GESTION DE CALIDAD

ISO 90001 # 7 EN EL PROYECTO

11.1 PLANIFICACIÓN DEL DISEÑO (7.3.1)

Etapas del D/D

- Ver Recolección de Información
- Ver actividades de Entrevista en el cronograma # 1

Revisión, verificación y validación para cada etapa D/D

- Ver Identificación de Requisitos
- Ver Análisis de Requisitos
- Ver Especificación de Requisitos
- Ver Modelo Relacional
- Ver Modelador de procesos
- Ver actividades de planeación y elaboración en el cronograma # 1

Las responsabilidades y autoridades para el D/D

- Ver Gantt. Cronograma # 1 y # 2.

11.2 ELEMENTOS DE ENTRADA PARA EL D/D (7.3.2)

Requisitos funcionales y de desempeño

- Identificación de Requisitos
 - Análisis de Requisitos
 - Especificación de Requisitos
-

- Ver ámbito del software

Requisitos legales y reglamentarios aplicables

- Ver marco legal

Información proveniente de diseños previos similares

- Toda la información es proveniente de Internet, porque en la universidad no se encuentra aplicaciones similares.

11.3 RESULTADOS DEL D/D (7.3.3)

Cumplir con los requisitos de entrada

- El ScannAmt06 cumple en un 80% con los requisitos mínimos funcionales

Proporcionar información apropiada para la compra, producción y prestación del servicio

- Se están desarrollando los manuales. Ver en el cronograma # 2 las actividades de documentación.

Especificar las características del producto

- Se están desarrollando los manuales. Ver en el cronograma # 2 las actividades de documentación.

11.4 REVISION DEL D/D (7.3.4)

Evaluar la capacidad de los resultados del D/D

- Se han hecho, pocas pruebas, de las cuales no quedaron registradas. Y se tienen planeadas realizar pruebas durante todo el mes de noviembre del 2006.

Identificar cualquier riesgo y proponer las acciones necesarias

- No se ha realizado el análisis de riesgo, aun cuando se proyectó en el cronograma de la primera fase de formativa 1
-

11.5 VERIFICACION DEL D/D (7.3.5)

Se deben realizar la verificación de acuerdo a lo planificado

- Ver Gantt. Cronograma # 1 , # 2 y # 3

Mantenerse los resultados de la verificación

- No se han realizado ningún registro
-

11.6 VALIDACION DEL D/D (7.3.6)

Se deben realizar la verificación de acuerdo a lo planificado

- Ver Gantt. Cronograma # 1 , # 2 y # 3

Mantenerse los resultados de la verificación

- No se han realizado ningún registro
-

11.7 CONTROL DE CAMBIOS (7.3.7)

Deben identificarse y mantenerse los registros

- Ver control de cambios

Control de cambios

- 1) El primer prototipo del Scannmt06 era con una interfaz sencilla y sin ninguna funcionalidad



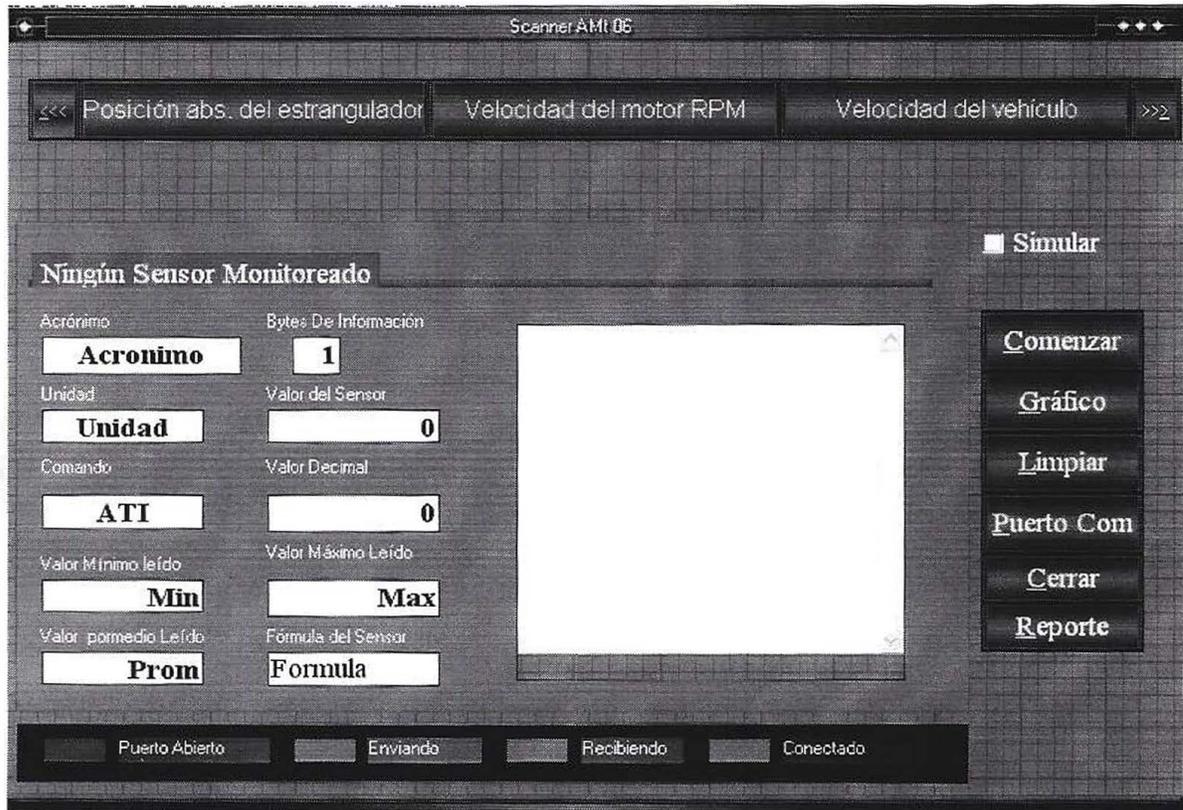
- 2) El segundo prototipo del ScannAmt06 implementó los Skins y cumplía con los requisitos mínimos del sistema.



La lectura de los sensores era manual.



3) La tercera y última versión del ScannAmt06 tiene una apariencia mejor, se pueden cambiar de Skin. La lectura de los sensores es automática.



Gantt

Cronogramas

A través de D/D se han planificado 4 cronogramas, de los cuales solo se conservan 3.
Uno por cada semestre.

Cronograma # 1

Actividades/Semana	1	2	4	5	8	9	10	11	12	13	14	15	Nota
Elaboración del anteproyecto													Esta se realizo con éxito
Asesoramiento con el Ingeniero Eléctrico Fernando Méndez													Esta se realizo con éxito
Asesoramiento con el Ingeniero Eléctrico Johan Mancera													Esta se realizo con éxito
Análisis de requisitos													Esta se realizo con éxito. Ver análisis de requisitos.
Entrevista con el dueño del taller automotriz 1-A													Ver CD anexo
Análisis de riesgo													No se realizo
Análisis de diseño													No se realizo
Elaboración prototipo													No se realizo
Pruebas de prototipo													No se realizo
Inducción del personal mecánico para la utilización del Scanner													No se ha realizado
Puesta en marcha													No se ha realizado



Cronograma # 2

17 Nombre	18 Inicio	19 Fin	20 Progreso	21 Notas	22 Asignar a
Entrevista con el Taller Automotor 1-a	30-ago-2005	20-oct-2005	67		Lesly Rodríguez, Edisón Granados
Primera Entrevista	30-ago-2005	04-sep-2005	100	Esta actividad se desarrolló con éxito	
Segunda Entrevista	21-sep-2005	24-sep-2005	33	Esta actividad se desarrolló con éxito	Yunelis Gutiérrez, Lesly Rodríguez, Edison Granados, Héctor Torres
Tercera Entrevista	18-oct-2005	20-oct-2005	37	Esta actividad se desarrolló con éxito	Yunelis Gutiérrez, Lesly Rodríguez, Héctor Torres
Planeación y Elaboración	31-ago-2005	11-nov-2005	35		
Análisis	31-ago-2005	03-oct-2005	80	Esta actividad se desarrolló con éxito	Yunelis Gutiérrez, Lesly Rodríguez,
Identificación de Requisitos	31-ago-2005	03-oct-2005	100	Esta actividad se desarrolló con éxito	Lesly Rodríguez,
Análisis de Requisitos	31-ago-2005	03-oct-2005	100	Esta actividad se desarrolló con éxito	Lesly Rodríguez, Héctor Torres
Especificación de Requisitos	31-ago-2005	03-oct-2005	100	Esta actividad se desarrolló con éxito	Yunelis Gutiérrez,
Elaboración de Casos de Usos	31-ago-2005	03-oct-2005	41	Esta actividad se desarrolló con éxito	Héctor Torres

CONCLUSIONES

Para desarrollar un scanner automotor se debe tener en cuenta.

- El lenguaje de programación que permita la comunicación con el puerto serial.
- La interfaz eléctrica que permita cambiar el formato del sistema OBD al de los computadores personales.
- El protocolo usado de los vehículos, para poder configurar la comunicación.
- Conocer los comandos OBD, para poder interactuar con la computadora a bordo del vehículo.
- Conocer las fórmulas de los sensores para poder analizar los datos leídos.

Además se tiene que seguir un modelo de Ingeniería de Software donde se lleven las etapas de

- Ingeniería de requisitos
 - Análisis del sistema
 - Diseño del Sistema
 - Planificación
 - Gestión de Riesgo
-

BIBLIOGRAFÍA

ELM Electronics- Circuit for the Hobbyists. En Documentación [en línea]. <http://www.elmelectronics.com/DSheets/ELM322Ds.pdf> [citado el 4 de mayo de 2004].

PETER, Davis. OBD II Fault Code Referents. Florida, Estados Unidos: KOTZING PUBLISHING ,1999. PP 200. ISBN 0971541140

PRESSMAN, Roger S. Ingeniería del Software: Un Enfoque Práctico. 5^{ta} Edición. Ciudad de México, México: MCGRAW HILL

ÍNDICE

	PÁG
ÁMBITO DEL SOFTWARE	32
ANÁLISIS DE REQUISITOS	33
ANÁLISIS DEL SISTEMA	34
CASOS DE USOS	36
CRONOGRAMA DE ACTIVIDADES	29
DESCRIPCIÓN DEL PROBLEMA	3
DISEÑO DEL SISTEMA	38
DISEÑO ELETRÓNICO DE LA INTERFAZ	39
ESPECIFICACIÓN DE REQUISITOS	33
FORMULACIÓN DEL PROBLEMA	5
GESTION DE CALIDAD	
GESTIÓN DE RIESGO	47
IDENTIFICACIÓN DE REQUISITOS	32
INGENIERÍA DE REQUISITOS	32
INTRODUCCIÓN	2
JUSTIFICACIÓN	7
LÍNEA DE INVESTIGACIÓN	27
MARCO CONCEPTUAL	24
MARCO DE REFERENCIA	8
MARCO ESPACIAL	26
MARCO LEGAL	26
MARCO TEÓRICO	8
METODOLOGÍA	27
MODELADOR DE PROCESO	38
MODELO ENTIDAD-RELACIÓN	34
MODELO RELACIONAL	35
MUESTRA	27
OBJETIVO GENERAL	6

OBJETIVOS DEL PROYECTO	6
OBJETIVOS ESPECÍFICOS	6
PLANTEAMIENTO DEL PROBLEMA	3
POBLACIÓN	27
POBLACIÓN Y MUESTRA	27
RECOLECCIÓN DE INFORMACIÓN	28
RECURSOS DEL ENTORNO	31
RECURSOS FINANCIEROS	31
RECURSOS HUMANOS	31
RECURSOS	31
RESUMEN	1
SISTEMATIZACIÓN DEL PROBLEMA	5
TIPO DE ESTUDIO	27

DISEÑO E IMPLEMENTACIÓN DE UN SCANNER AUTOMOTOR PARA EL
DIAGNÓSTICO DE LOS SENSORES DE OXIGENO, TEMPERATURA Y DE
CHOQUE DEL VEHÍCULO CHEVROLET BLAZER 1995 PARA EL TALLER
AUTOMOTRIIZ 1-A

MANUAL DEL SISTEMA

YUNELIS GUTIÉRREZ SERRANO
LESLY MILAGRO RODRIGUEZ MONTAÑO
HÉCTOR RAFAEL TORRES CABALLERO

ING. ROBERTO OSIO

UNIVERSIDAD SIMÓN BOLÍVAR
FACULTAD DE INGENIERÍA DE SISTEMA
INVESTIGACIÓN FORMATIVA IV
SEMESTRE X DIURNO
BARRANQUILLA

2006.



TABLA DE CONTENIDO

INTRODUCCIÓN

1. OBJETIVOS DEL MANUAL
 - 1.1 OBJETIVO GENERAL
 - 1.2 OBJETIVOS ESPECÍFICOS
 2. JUSTIFICACIÓN
 3. DESCRIPCIÓN DEL PROBLEMA
 4. INGENIERIA DE REQUISITOS
 - 4.1 DESCRIPCIÓN DEL SISTEMA ACTUAL (Casos de usos)
 - 4.2 MODELADOR DE PROCESOS (Sistema actual)
 - 4.3 SCRIPT DE LA BASE DE DATOS
 5. ANÁLISIS DEL SISTEMA PROPUESTO
 - 5.1 MODELO ENTIDAD – RELACIÓN
 - 5.2 MODELO RELACIONAL
 - 5.3 DICCIONARIO DE DATOS (MODELO RELACIONAL)
 - 5.4 CODIGO FUENTE DEL SISTEMA DE INFORMACION
 6. RECURSOS MÍNIMOS PARA IMPLANTAR EL SISTEMA
 7. TERMINOLOGÍA TÉCNICA USADA
-

INTRODUCCIÓN

A lo largo del contenido de este manual, se planteará y se estudiará de forma organizada el sistema de información para el taller Automotriz 1-A al cual se le desarrollo este sistema de información para facilitar la organización y control de sus procesos lectura de los sensores de cada vehiculo y posibles soluciones para estos, siendo este un método para mejorar la calidad de sus servicios a través de este sistema de información.

Así mismo se describirá la problemática actual que presenta el taller y se planteará el análisis y diseño de la posible solución para suplir las necesidades que presenta el taller, empleando la metodología y las herramientas necesarias para el desarrollo del sistema.

En el contenido también se encontrarán diagramas y modelos de datos que servirán de soporte y ayuda para cualquier personal que quiera estudiar la conformación y el comportamiento del sistema.

1. OBJETIVOS DEL MANUAL

1.1 OBJETIVO GENERAL

Brindarle al usuario un conocimiento acerca de todas las herramientas que componen el sistema de información y los manejos de los procesos administrativos del sistema de información para el Taller Automotriz 1-A.

1.2 OBJETIVOS ESPECÍFICOS

- Lograr que la problemática que la cooperativa presenta sea comprendida a cabalidad por las personas que estudien este manual.
 - Presentar como una forma de estudio la documentación de la fase de ingeniería de requisitos.
 - Describir en forma de diagramas y modelos la situación actual que presenta el sistema en estudio, con el fin de comprender, analizar y diseñar los procesos que se están llevando a cabo.
 - Conocer las aplicaciones y el buen funcionamiento del Sistema de Información.
-

- Proporcionar una descripción de los procesos del sistema.

 - Servir de soporte en caso de un futuro mantenimiento a la aplicación.

 - Modelar de manera precisa y entendible el sistema propuesto para la solución de la problemática de la cooperativa, con el fin de dejar bien documentado lo que el grupo de investigación ha desarrollado.
-

2. JUSTIFICACIÓN

Con la construcción de este manual pretendemos plasmar las fases de ingeniería que se llevaron a cabo para lograr el desarrollo y la implementación del sistema de información, lo cual le permitirá taller Automotriz 1-A, optimizar y agilizar sus procesos de lectura, diagnóstico y posibles soluciones. Este sistema fortalecerá toda la estructura organizacional y técnica taller Automotriz 1-A, permitiendo la sincronización del mismo, ya que al tener un buen manejo de información llegara a cumplir todas las metas trazadas para brindar una calidad de servicio a cada uno de los clientes que deseen afiliarse al taller Automotriz 1-A. dándole así al propietario la seguridad y facilidad de respuesta a cada una de las diferentes decisiones e interrogantes para llevar a cabo una buena toma de decisión. Lo anterior se verá reflejado en un eficiente sistema de lectura de códigos de fallas.

Así mismo, cabe anotar que la organización tiene grandes proyecciones, y si no incorpora un sistema que soporte el crecimiento de esta, se verá afectada en sus procesos administrativos lo cual redundará en la ineficiente prestación de servicios a sus afiliados.

En el manual se contemplan a través de modelos las formas en que el sistema podrá satisfacer la necesidad de controlar los procesos y manipular la información del taller, ya que este tendrá características de tipo de acceso remoto, y contará con la seguridad necesaria para ello, cumpliendo de esta manera las exigencias actuales en cuanto la incorporación de sistemas de información, cuya tendencia es Orientada al desarrollo y crecimiento de los avances tecnológicos.

3. DESCRIPCIÓN DEL PROBLEMA

El Taller Automotriz 1-A ubicado en la calle 74 # 41D-11 en la ciudad de Barranquilla, tiene como finalidad prestar servicio automotriz a los propietarios de vehículos cuando estos tengan fallas en su funcionamiento.

En el momento de la revisión del auto los mecánicos realizan un diagnóstico de una forma manual utilizando un tester que lo conecta en el conector OBD (On Board Diagnostics) que todos los vehículos actuales tienen equipados, el mecánico interpreta los códigos arrojados por medio del check engine y detecta los sensores que están fallando y de esta manera sabrá donde dirigirse y solucionar el problema. Esta tarea por ser manual gasta un tiempo alrededor de 10 a 15 minutos.

Los códigos obtenidos, deben ser interpretados, en forma específica, recurriendo al manual del vehículo, ya que, cada fabricante programa su computadora con sus propios códigos.

Dicho de otra forma, el vehículo mediante la activación de señales luminosas, le informa al mecánico que elementos están funcionando mal, con el fin de prevenirle y éste tome las medidas de reparación correspondientes.

Conociendo los avances que existen en la actualidad, el Taller Automotriz 1-A quiere optimizar el proceso de revisión de posibles fallas que se le presentan a los vehículos con ayuda de la tecnología.

4. INGENIERÍA DE REQUISITOS

4.1 DESCRIPCIÓN DEL SISTEMA ACTUAL (Casos de usos)

Caso de uso: Gestionar vehículos

Actores: Cliente, mecánico, Sistema.

Propósito: Capturar la información del vehículo

Tipo: Primario

Descripción: Este caso de uso empieza cuando un cliente llega con el vehículo para su mantenimiento y/o reparación. El mecánico guarda la información respectivamente en el sistema.

Caso de uso: Leer Códigos de fallas

Actores: Mecánicos

Propósito: Leer los códigos de fallas de la computadora del vehículo

Tipo: Primario

Descripción: Este caso de uso comienza cuando el mecánico conecta el scanner con el puerto OBDII del vehículo para leer códigos de fallas



Caso de uso: Decodificar los códigos arrojados por el scanneo

Actores: Sistema

Propósito: Saber cuál o cuales son los códigos de fallas del vehículo

Tipo: Primario

Descripción: Este caso de uso comienza cuando el sistema ya tiene los códigos leídos del scanneo, luego son consultados en la base de datos.

Caso de uso: Leer sensores

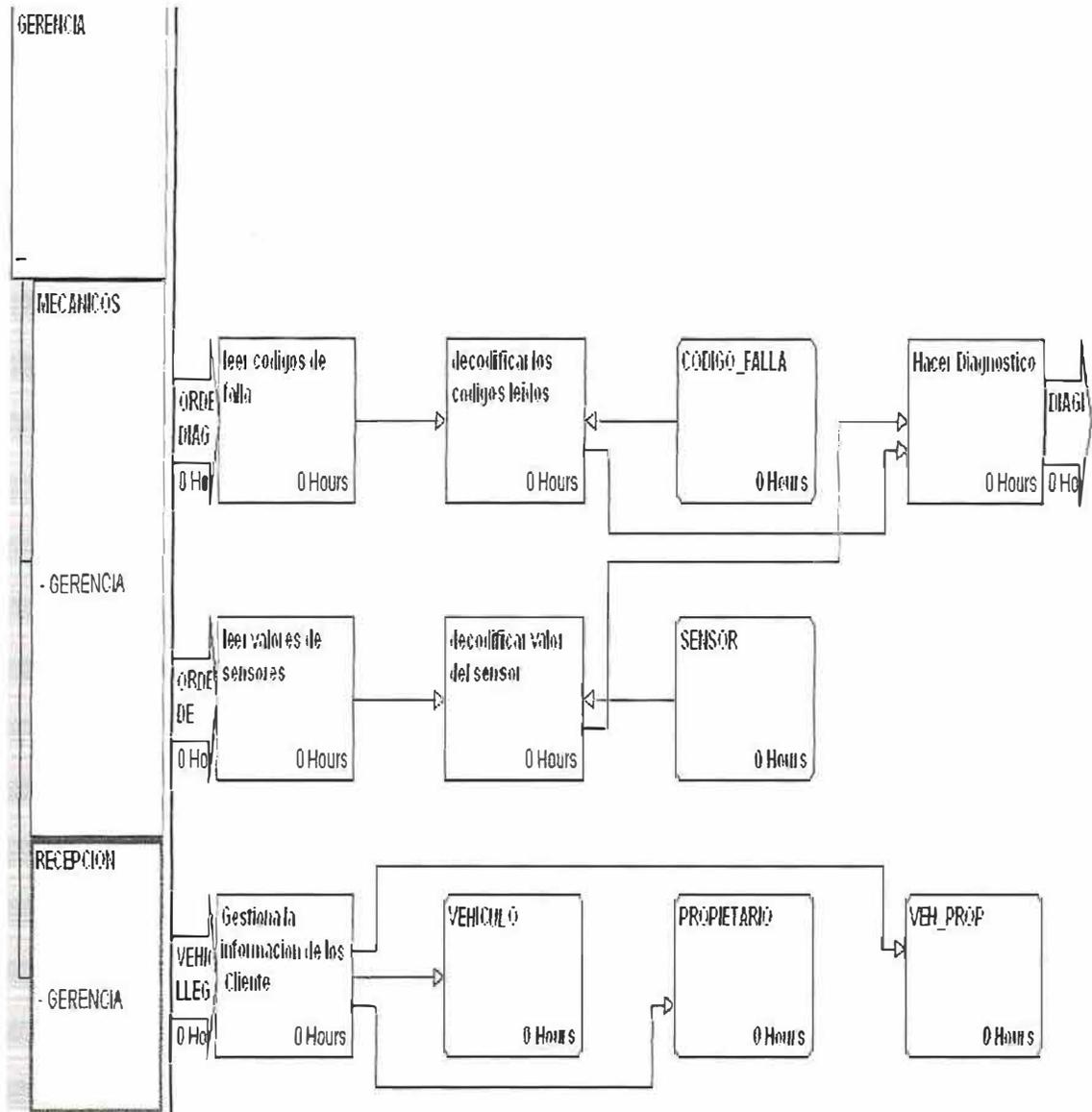
Actores: Sistema

Propósito: Saber cual es el valor de los sensores

Tipo: Primario

Descripción: Este caso de uso comienza cuando el sistema lee los valores de los sensores del vehículo.

4.2 MODELADOR DE PROCESOS (Sistema actual)



4.3. SCRIPT DE LA BASE DE DATOS.

Descripción de la tabla Diagnostico

```
CREATE TABLE IF NOT EXISTS diagnostico (  
    PlacaVeh varchar (6) NOT NULL DEFAULT "",  
    CodSensor int(11) unsigned NOT NULL DEFAULT "",  
    Valor varchar(25) NOT NULL DEFAULT '0',  
    fecha date NOT NULL DEFAULT ""  
);
```

Descripción de la tabla DTC.

```
CREATE TABLE `dtc` (  
    `codigo` varchar(5) default NULL,  
    `descripcion` varchar(255) default NULL,  
    `solucion` varchar(255) default NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Descripción de la Tabla Propietarios.

```
CREATE TABLE `propietarios` (  
  `cedula` varchar(50) default NULL,  
  `nombre` varchar(50) default NULL,  
  `direccion` varchar(50) default NULL,  
  `telefono` varchar(50) default NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Descripción de la tabla Sensor

```
CREATE TABLE `sensor` (  
  `codigo` int(11) default NULL,  
  `nombre` varchar(255) default NULL,  
  `comando` varchar(255) default NULL,  
  `bytes` varchar(50) default NULL,  
  `acronimo` varchar(50) default NULL,  
  `unidad` varchar(50) default NULL,  
  `formula` mediumtext,  
  `valor` varchar(50) default NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```



Descripción de la tabla Usuarios

```
CREATE TABLE `usuarios` (  
  `usuario` varchar(50) default NULL,  
  `clave` varchar(50) default NULL,  
  `tipo` varchar(50) default NULL,  
  `cedula` varchar(50) default NULL,  
  `nombre` varchar(50) default NULL,  
  `direccion` varchar(50) default NULL,  
  `telefono` varchar(50) default NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Descripción de la tabla Scsllaves

```
CREATE TABLE `scsllaves` (  
  `llav_nom1` varchar(20) NOT NULL default '0',  
  `llav_cod_tab1` varchar(30) NOT NULL default '0',  
  `llav_nom2` varchar(20) NOT NULL default '0',  
  `llav_cod_tab2` varchar(30) NOT NULL default '0'  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

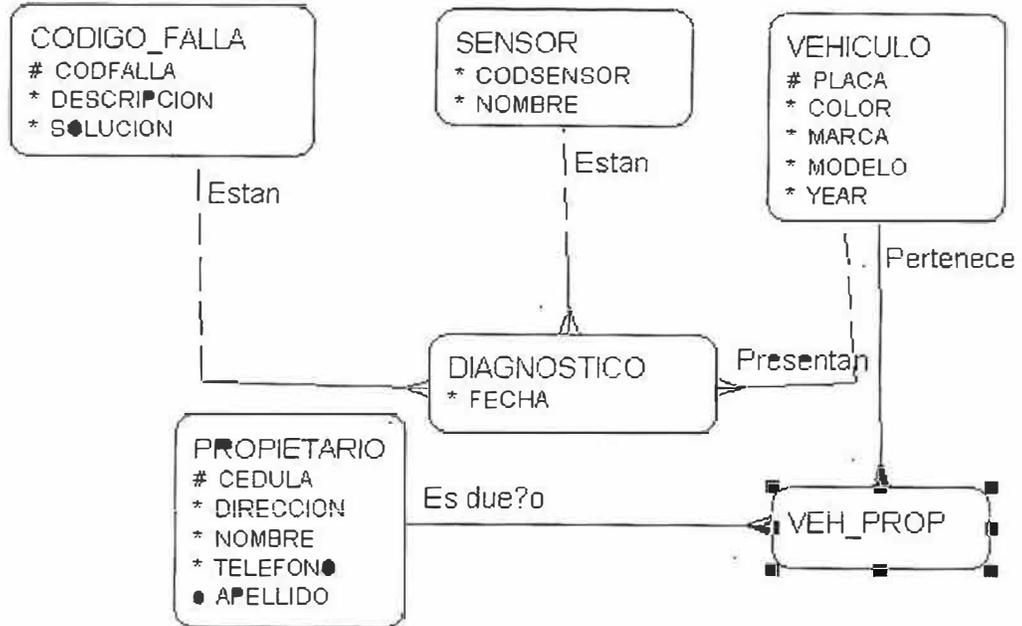
Descripción de la tabla Vehículos

```
CREATE TABLE IF NOT EXISTS vehiculos (  
  placa varchar(50) ,  
  marca varchar(50) ,  
  modelo varchar(50) ,  
  year varchar(50) ,  
  color varchar(50) ,  
  estado varchar(25) NOT NULL DEFAULT 'Desconectado'  
);
```

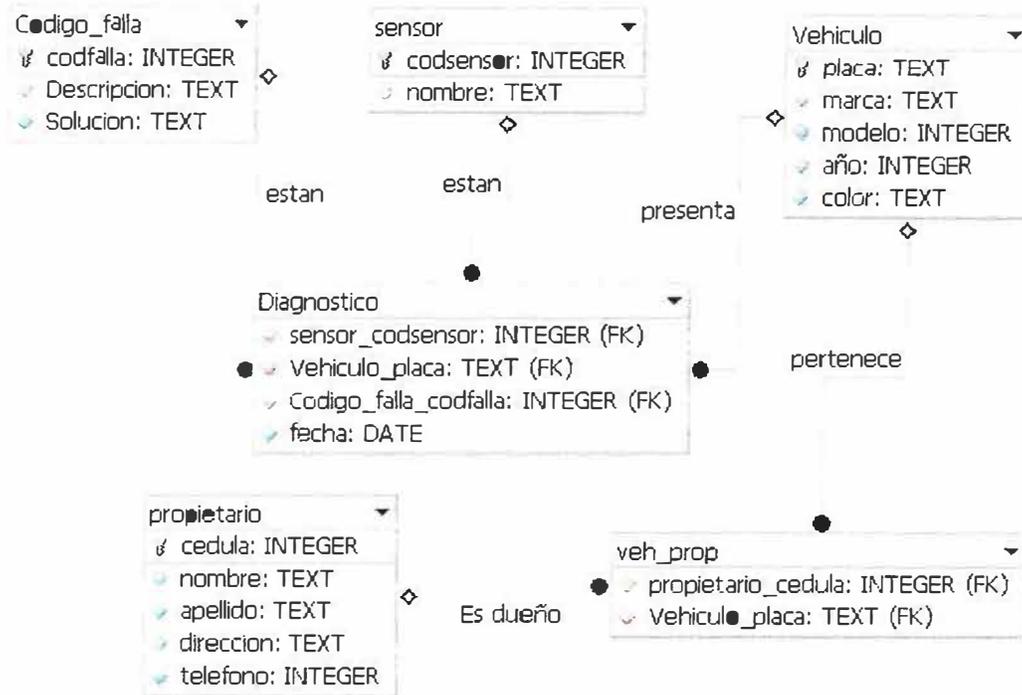
Descripción de la tabla Vehprop

```
CREATE TABLE `vehprop` (  
  `codveh` varchar(50) default NULL,  
  `Cedprop` varchar(50) default NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

5.1. MODELO ENTIDAD-RELACION



5.2. MODELO RELACIONAL



5.3. DICCIONARIO DE DATOS (MODELO RELACIONAL)

T.D = Tipo de Dato

L=Longitud

N=Nulo?

D=Decimales

Nombre de la tabla: PROPIETARIOS	Descripción: Guarda información básica de los distintos propietarios.
Nombre largo: PROPIETARIOS	de los propietarios.
Módulo: GESTIÓN DE PROPIETARIOS	

N o	NOM ATRIBUTO	LLAVE S	T.D	L	D	N	DESCRIPCION
1	CEDULA	PK	VARCHA R	50	0	NO	Identifica de manera única cada propietario
2	NOMBRE		VARCHA R	50	0	NO	Nombre del propietario
3	DIRECCIÓN		VARCHA R	50	0	SI	Ubicación del propietario
4	TELEFONO		VARCHA R	40	0	NO	Teléfono del propietario

Nombre de la tabla: SENSOR	Descripción: Guarda información básica de los distintos sensores del automóvil.
Nombre largo: SENSORES	
Módulo: GESTIÓN DE LECTURA	

	NOM ATRIBUTO	LLAVES	T.D	L	D	N	DESCRIPCION
1	CODIGO	PK	INT	11	0	NO	Identifica de manera única cada sensor
2	NOMBRE		VARCHAR	255	0	NO	Nombres de los sensores
3	COMANDO		VARCHAR	255	0	NO	Abreviatura del sensor
4	BYTES		VARCHAR	50	0	NO	Bytes que arroja cada sensor
5	ACRONIMO		VARCHAR	50	0	SI	del sensor
6	UNIDAD		VARCHAR	50	0	NO	Unidad que arroja cada sensor
7	FORMULA		MEDIUMTEXT		0	NO	Como se calcula el valor del sensor
8	VALOR		VARCHAR	50	0	NO	Teléfono fijo

Nombre de la tabla: USUARIOS Descripción: Guarda información básica de los distintos afiliados o usuarios del taller automotriz 1-A
 Nombre largo: USUARIOS
 Módulo: GESTIÓN DE USUARIOS

N o	NOM ATRIBUTO	LLAVES	T.D	L	D	N	DESCRIPCION
1	USUARIO	PK	VARCHAR	50	0	SI	Identifica el afiliado o usuario en particular
2	CLAVE		VARCHAR	50	0	SI	Contraseña que posee un usuario
3	TIPO		VARCHAR	50	0	SI	El tipo de usuario que puede tener
4	CEDULA		VARCHAR	50	0	SI	Identificación única del usuario
5	NOMBRE		VARCHAR	50	0	SI	Nombre del usuario
6	DIRECCION		VARCHAR	50	0	SI	Ubicación del usuario
7	TELEFONO		VARCHAR	50	2	SI	Teléfono del usuario

Nombre de la tabla: DTC Descripción: Almacena todos códigos de fallas que puedan existir
Nombre largo: CODIGOS DE FALLAS
Módulo: GESTIÓN DE DTC

N o	NOM ATRIBUTO	LLAVES	T.D	L	D	N	DESCRIPCION
1	CODIGO	PK	VARCHA R	5	0	SI	Identifica de manera única Cada pago hecho a un proveedor
2	DESCRIPC IÓN		FECHA	25 5	0	SI	Fecha de pago
3	SOLUCION		TEXTO	25 5	0	SI	Tipo de pago realizado

Nombre de la tabla: DIAGNOSTICO Descripción: Almacena todos los registros que arrojan todos los sensores
Nombre largo: DIAGNOSTICO
Módulo: GESTIÓN DE DIAGNOSTICO

N o	NOM ATRIBUTO TO	LLAVES	T.D	L	D	N	DESCRIPCION
1	PLACAVEH	PK	VARCHA R	6	0	N O	Identifica de manera única Cada pago hecho a un proveedor
2	CODSENSO R		INT	11	0	N O	Código del sensor
3	VALOR		VARCHA R	25	0	N O	Valor que arroja el sensor
4	FECHA		DATE		0	N O	Fecha que fue realizada la lectura

Nombre de la tabla: VEHICULOS Descripción: Guarda información básica de los distintos automóviles.
Nombre largo: VEHICULOS
Módulo: GESTIÓN DE VEHÍCULOS

	NOM ATRIBUTO	LLAVES	T.D	L	D	N	DESCRIPCION
1	PLACA	PK	VARCHAR	50	0	SI	Identifica de manera única cada automóvil
2	MARCA		VARCHAR	50	0	SI	Marca del automóvil
3	MODELO		VARCHAR	50	0	SI	Modelo del automóvil
4	YEAR		VARCHAR	50	0	SI	Año del automóvil
5	COLOR		VARCHAR	50	0	SI	Color del automóvil
6	ESTADO		VARCHAR	25	0	NO	Estado del automóvil

Nombre de la tabla: VEHPROP	Descripción: Guarda información básica de
	los distintos
afiliados propietarios con la relación de su vehículo	
Nombre largo: VEHICULOS PROPIETARIOS	
Módulo: GESTIÓN DE VEHPROP	

N o	NOM ATRIBUTO	LLAVES	T.D	L	D	N	DESCRIPCION
1	CODVEH	PK	VARCH AR	50	0	SI	Identifica el afiliado o usuario en particular
2	CEDPROP	FK[PROPIET ARIOS(CEDU LA)]	VARCH AR	50	0	SI	Contraseña que posee un usuario



5.4 CÓDIGO FUENTE DEL SISTEMA DE INFORMACION SACANNAMT06

Formulario frmUsuario.frm

Public OK As Boolean

Private Sub cmdaceptar_Click()

Dim consulta As String

Set registro = New ADODB.Recordset

consulta = "select * from usuarios where usuario='" & txtusuario.Text & "' and clave='" & Encriptado(txtclave.Text) & "'"

Set registro = conexion.Execute(consulta)

If registro.EOF Then

sonError

frmMensaje.mensaje "Usuario Invalido. Acceso Denegado", "Login Incorrecto...", 9001&

txtusuario.SetFocus

txtusuario.SelStart = 0

txtusuario.SelLength = Len(txtusuario.Text)

txtclave.Text = Empty

Else

OK = True

Me.Hide

frmmain.barEstado.Panels(4).Text = "Usuario:" + registro!Usuario

If (registro!tipo = "Administrador") Then

frmmain.mnusuario.Enabled = True

frmmain.mnuExperto.Enabled = True

End If

frmmain.Show

End If

End Sub

Private Sub cmdcancelar_Click()

OK = False

cerrarBaseDato

Me.Hide

End

End Sub

```
Private Sub Form_Load()  
    skinActual.ApplySkin Me.hWnd  
End Sub
```

```
Private Sub txtclave_KeyDown(KeyCode As Integer, Shift As Integer)  
    If KeyCode = vbKeyF2 Then  
        AyudaScann.PopUp "Campo para digitar la contraseña de acceso", 1000,  
        1000  
    End If  
End Sub
```

```
Private Sub txtusuario_KeyDown(KeyCode As Integer, Shift As Integer)  
    If KeyCode = vbKeyF2 Then  
        AyudaScann.PopUp "Campo para digitar el nombre de usuario para poder  
    entrar", 1000, 1000  
    End If  
End Sub
```

Formulario frmVehiculo.frm

```
Public Sub Form_Load()  
  
    cmbcolor.AddItem ("Rojo")  
  
    cmbcolor.AddItem ("Blanco")  
  
    cmbcolor.AddItem ("Negro")  
  
    cmbcolor.AddItem ("Azul")  
  
    cmbcolor.AddItem ("Plateado")  
  
    cmbcolor.AddItem ("Dorado")  
  
    cmbcolor.AddItem ("Amarillo")  
  
    cmbcolor.AddItem ("Verde")  
  
    cmbmarca.AddItem ("Chevrolet")
```

```

        cargarPropietario
End Sub

Sub verFrm()
    skinActual.ApplySkin Me.hWnd
    Me.Show vbModal
End Sub

Private Sub cargarPropietario()
    Dim registro As New ADODB.Recordset
    registro.CursorType = adOpenStatic
    registro.CursorLocation = adUseClient
    registro.Open "select * from propietarios", cadena, adOpenKeyset,
adLockOptimistic
    J = registro.RecordCount
    For i = 1 To J
        cmbPropietario.AddItem (registro!cedula)
        registro.MoveNext
    Next i
    registro.Close
End Sub

Sub buscarPropietarios()
    Dim registro As New ADODB.Recordset

```



```
registro.CursorType = adOpenStatic
registro.CursorLocation = adUseClient
registro.Open "select * from propietarios where cedula=" & cmbPropietario & "",
cadena, adOpenKeyset, adLockOptimistic
If registro.EOF Then

Else

    txtNomPropietario.Text = registro!nombre
    txtdireccion.Text = registro!direccion
    txttelefono.Text = registro!telefono

End If

registro.Close

End Sub
```

```
Private Sub txtConectado_Change()

If (txtConectado.Text = "Conectado") Then

    btnConectar.Caption = "&Desconectar"

Elseif (txtConectado.Text = "Desconectado") Then

    btnConectar.Caption = "&Conectar"

End If

End Sub
```

```
Private Sub txtDireccion_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyF2 Then
        AyudaScann.PopUp "CaMpo para digitar la dirección del propietario", 1000,
        1000
    End If
End Sub
```

```
Private Sub txtmodelo_GotFocus()
    txtmodelo.BackColor = &HE0E0E0
End Sub
```

```
Private Sub txtmodelo_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyF2 Then
        AyudaScann.PopUp "Campo para digitar el modelo del vehículo", 1000, 1000
    End If
End Sub
```

```
Private Sub txtmodelo_KeyPress(KeyAscii As Integer)
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
End Sub
```

```
Private Sub txtmodelo_LostFocus()
    txtmodelo.BackColor = &HFFFFFF
End Sub
```

End Sub

```
Private Sub txtNomPropietario_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    If KeyCode = vbKeyF2 Then
```

```
        AyudaScann.PopUp "Campo para digitar el nombre del propietario", 1000,  
1000
```

```
    End If
```

```
End Sub
```

```
Private Sub txtplaca_GotFocus()
```

```
    txtplaca.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtplaca_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    If KeyCode = vbKeyF2 Then
```

```
        AyudaScann.PopUp "Campo para digitar la placa del vehiculo", 3200, 4000
```

```
    End If
```

```
End Sub
```

```
Private Sub txtplaca_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

End Sub

Private Sub txtplaca_LostFocus()

txtplaca.BackColor = &HFFFFFF

End Sub

Private Sub txtTelefono_KeyDown(KeyCode As Integer, Shift As Integer)

If KeyCode = vbKeyF2 Then

AyudaScann.PopUp "Campo para digitar el telefono del propietario", 1000,
1000

End If

End Sub

Private Sub txtyear_GotFocus()

txtyear.BackColor = &HE0E0E0

End Sub

Private Sub txtyear_KeyDown(KeyCode As Integer, Shift As Integer)

If KeyCode = vbKeyF2 Then

AyudaScann.PopUp "Campo para digitar el año del vehiculo", 1000, 1000

End If

End Sub



```
Private Sub txtyear_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txtyear_LostFocus()
```

```
    txtyear.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub cmbmarca_GotFocus()
```

```
    cmbmarca.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub cmbmarca_LostFocus()
```

```
    cmbmarca.BackColor = &HFFFFFF
```

```
End Sub
```

Formulario frmVel.frm

```
Public valor As Integer
```

```
Public i As Integer
```

```
Public enc As Boolean
```

```
Dim Y As Integer
```

```
Dim X As Integer
```

```
Dim delay As Integer
Dim toggle As Integer
Dim flap As Integer
Sub Velocidad(s As Double)
On Error Resume Next
If (s >= 0) Then
    Dim Xfin As Integer
    Dim Yfin As Integer
    On Error Resume Next
        picvel.Cls
        picvel.FontSize = 14
        picvel.DrawWidth = 2

        picvel.BackColor = vbBlue
        picvel.Scale (-100, 100)-(100, -100)
        picvel.ForeColor = RGB(36, 200, 255)
        pi = 4 * Atn(1)
        If (s > 180) Then
            s = 180
        End If

        If (s >= 80) Then
            Beep
```

```
picvel.BackColor = vbRed
picvel.ForeColor = RGB(20, 20, 20)
vmax = "DISMINUYA VELOCIDAD"
picvel.CurrentX = 0 - picvel.TextWidth(vmax) / 2
picvel.CurrentY = -40 - picvel.TextHeight(vmax) / 2
picvel.Print vmax
```

End If

```
teta = (s / 120) * pi - (5 * pi) / 4
```

```
Xfin = Cos(-teta) * 70
```

```
Yfin = Sin(-teta) * 70
```

```
X1 = Cos(-teta + pi / 30) * 60
```

```
Y1 = Sin(-teta + pi / 30) * 60
```

```
X2 = Cos(-teta - pi / 30) * 60
```

```
Y2 = Sin(-teta - pi / 30) * 60
```

```
picvel.Line (Xfin, Yfin)-(0, 0), vbGreen
```

```
picvel.Line (Xfin, Yfin)-(X1, Y1), vbGreen
```

```
picvel.Line (Xfin, Yfin)-(X2, Y2), vbGreen
```

```
cx = picvel.ScaleWidth / 2 + picvel.ScaleLeft
```

```
cy = picvel.ScaleHeight / 2 + picvel.ScaleTop
Msg = "KM HRS"
picvel.CurrentX = cx - picvel.TextWidth(Msg) / 2
picvel.CurrentY = 50 + cy - picvel.TextHeight(Msg) / 2
picvel.Print Msg
```

```
For g = 0 To 180 Step 5
    ang = (g / 120) * pi - (5 * pi) / 4
    x3 = Cos(-ang) * 4.5
    y3 = Sin(-ang) * 4.5
    If (g = 0 Or g = 30 Or g = 60 Or g = 90 Or g = 120 Or g = 150 Or g = 180)
```

Then

```
    picvel.DrawWidth = 4
    picvel.Line (x3 * 20, y3 * 20)-(x3 * 18, y3 * 18), vbYellow
    v = g
    picvel.FontSize = 12
    picvel.BorderStyle = 1
    xV = Cos(-ang) * 3.7
    yv = Sin(-ang) * 3.7
    picvel.CurrentX = 20 * xV + -3.12 - picvel.TextWidth(v) / 2
    picvel.CurrentY = 20 * yv + 8 - picvel.TextWidth(v) / 2
    picvel.Print v
```

Else

picvel.DrawWidth = 1

picvel.Line (x3 * 20, y3 * 20)-(x3 * 18, y3 * 18), vbGreen

End If

Next g

picvel.FillColor = vbBlue

picvel.FillStyle = vbSolid

picvel.DrawWidth = 1

picvel.Line (-24, -60)-(24, -82), vbRed, B

picvel.FillColor = vbBlue

picvel.FillStyle = vbCross

picvel.DrawWidth = 3

picvel.Line (-26, -59)-(26, -83), vbYellow, B

picvel.CurrentX = cx - picvel.TextWidth(s) - 2

picvel.CurrentY = cy - picvel.TextHeight(s) - 73

picvel.FontSize = 18

picvel.Print s

If s = 0 Then

t = 0

Else

t = 30

End If



```
Timer2.Interval = CInt(t + s / 20)
```

```
End If
```

```
End Sub
```

```
Private Sub btnDetener_Click()
```

```
    If (Timer1.Interval = 0) Then
```

```
        btnDetener.Caption = "Detener"
```

```
        Timer1.Interval = 1000
```

```
    Else
```

```
        btnDetener.Caption = "Comenzar"
```

```
        Timer1.Interval = 0
```

```
    End If
```

```
End Sub
```

```
Private Sub btnSalir_Click()
```

```
    Me.Hide
```

```
End Sub
```

```
Public Sub verFrm()  
    skinActual.ApplySkin Me.hWnd  
    Me.Show vbModal  
    Velocidad (frmSensorData.txtFormula.Text)  
End Sub
```

```
Sub moverVehiculos()  
    Y = Y + 1: If Y = 7 Then Y = 0  
    For img = 0 To 11  
        Picture1(img).Picture = Image1(Y)  
    Next img  
    Me.Icon = Image1(Y).Picture  
End Sub
```

```
Private Sub Form_Load()
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
    Velocidad (frmSensorData.txtFormula.Text)
```

```
End Sub
```

```
Private Sub Timer2_Timer()
```

```
    moverVehiculos
```

```
End Sub
```

```
Formulario frmTip.frm
```

```
Option Explicit
```

```
' La base de datos en memoria de sugerencias.
```

```
Dim Tips As New Collection
```

```
' Nombre del archivo de sugerencias
```

```
Const TIP_FILE = "tipScannAmt06.TXT"
```

```
' Índice en la colección de la sugerencia actualmente mostrada.
```

```
Dim CurrentTip As Long
```

```
Private Sub DoNextTip()
```

```
    ' Seleccionar una sugerencia aleatoriamente.
```

```
    CurrentTip = Int((Tips.Count * Rnd) + 1)
```

```
    frmTip.DisplayCurrentTip
```

```
End Sub
```

```
Function LoadTips(sFile As String) As Boolean
```

```
    Dim NextTip As String ' Leer cada sugerencia desde archivo.
```

```
    Dim InFile As Integer ' Descriptor para archivo.
```

```
    ' Obtener el siguiente descriptor de archivo libre.
```

```
    InFile = FreeFile
```

```
' Asegurarse de que se especifica un archivo.
If sFile = "" Then
    LoadTips = False
    Exit Function
End If

' Asegurarse de que el archivo existe antes de intentar abrirlo.
If dir(sFile) = "" Then
    LoadTips = False
    Exit Function
End If

' Leer la colección desde un archivo de texto.
Open sFile For Input As InFile
While Not EOF(InFile)
    Line Input #InFile, NextTip
    Tips.Add NextTip
Wend
Close InFile

' Mostrar una sugerencia aleatoriamente.
DoNextTip

LoadTips = True
End Function

Private Sub chkLoadTipsAtStartup_Click()
    ' guardar si este formulario debe mostrarse o no al iniciar
```

```
        SaveSetting App.EXENAME, "Opciones", "Mostrar sugerencias al iniciar",  
chkLoadTipsAtStartup.Value  
End Sub  
  
Private Sub cmdNextTip_Click()  
    DoNextTip  
End Sub  
  
Private Sub cmdOK_Click()  
    Unload Me  
End Sub  
  
  
Private Sub Form_Load()  
    ' Semilla aleatoria  
    Randomize  
    ' Leer el archivo de sugerencias y mostrar una sugerencia aleatoriamente.  
    LoadTips App.Path & "\" & TIP_FILE  
End Sub  
  
Public Sub DisplayCurrentTip()  
    If Tips.Count > 0 Then  
        lblTipText.Caption = Tips.Item(CurrentTip)  
    End If  
End Sub  
  
Sub verFrm()
```

```
skinActual.ApplySkin Me.hWnd
    Dim ShowAtStartup As Long
    ' Ver si debemos mostrar al iniciar
    ShowAtStartup = GetSetting(App.EXENAME, "Opciones", "Mostrar sugerencias
al iniciar", 1)
    If ShowAtStartup = 1 Then
        Me.Show
    End If
    ' Establecer la casilla de verificación, que obligará a que el valor se vuelva a
escribir en el Registro
    Me.chkLoadTipsAtStartup.Value = vbChecked
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    SaveSetting App.EXENAME, "Opciones", "Mostrar sugerencias al iniciar",
chkLoadTipsAtStartup.Value
End Sub
```

Formulario frmSiNo.frm

Option Explicit

Private parametro As Long



```
Public Function mensaje1(Msg As String, tit As String, Optional contex As Long =  
1000&)
```

```
    On Error Resume Next
```

```
    skinActual.ApplySkin Me.hWnd
```

```
        parametro = contex
```

```
        txtMensaje.Caption = Msg
```

```
        frmSiNo.Caption = tit
```

```
        frmSiNo.Show vbModal
```

```
        ' btnAceptar.SetFocus
```

```
End Function
```

```
Private Sub btnNo_Click()
```

```
    eliminarReg = False
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub btnSi_Click()
```

```
    On Local Error Resume Next
```

```
    eliminarReg = True
```

```
    Unload Me
```

```
End Sub
```

Formulario frmRPM.frm

Public valor As Integer

Public i As Integer

Public enc As Boolean

Dim Y As Integer

Dim X As Integer

Dim delay As Integer

Dim toggle As Integer

Dim flap As Integer

Sub Velocidad(s As Double)

On Error Resume Next

s = s / 1000

If (s >= 0) Then

Dim Xfin As Integer

Dim Yfin As Integer

On Error Resume Next

picvel.Cls

picvel.FontSize = 14

picvel.DrawWidth = 2

picvel.BackColor = &H808080

picvel.Scale (-100, 100)-(-100, -100)

```
picvel.ForeColor = RGB(36, 200, 255)
```

```
pi = 4 * Atn(1)
```

```
If (s > 10) Then
```

```
    s = 10
```

```
End If
```

```
teta = (7 * s / 50) * pi - (5 * pi) / 4 + pi / 16
```

```
Xfin = Cos(-teta) * 70
```

```
Yfin = Sin(-teta) * 70
```

```
X1 = Cos(-teta + pi / 30) * 60
```

```
Y1 = Sin(-teta + pi / 30) * 60
```

```
X2 = Cos(-teta - pi / 30) * 60
```

```
Y2 = Sin(-teta - pi / 30) * 60
```

```
picvel.Line (Xfin, Yfin)-(0, 0), vbGreen
```

```
picvel.Line (Xfin, Yfin)-(X1, Y1), vbGreen
```

```
picvel.Line (Xfin, Yfin)-(X2, Y2), vbGreen
```

```
cx = picvel.ScaleWidth / 2 + picvel.ScaleLeft
```

```
cy = picvel.ScaleHeight / 2 + picvel.ScaleTop
Msg = "RPM"
picvel.CurrentX = cx - picvel.TextWidth(Msg) / 2
picvel.CurrentY = 50 + cy - picvel.TextHeight(Msg) / 2
picvel.Print Msg
Msg = "1 X 1000"
picvel.CurrentX = cx - picvel.TextWidth(Msg) / 2
picvel.CurrentY = 20 + cy - picvel.TextHeight(Msg) / 2
picvel.Print Msg
```

```
For g = 0 To 10 Step 1
    ang = (7 * g / 50) * pi - (5 * pi) / 4 + pi / 16
    x3 = Cos(-ang) * 4.5
    y3 = Sin(-ang) * 4.5
    If (g = 0 Or g = 1 Or g = 2 Or g = 3 Or g = 4 Or g = 5 Or g = 6 Or g = 7 Or g =
8 Or g = 9 Or g = 10) Then
        picvel.DrawWidth = 4
        picvel.Line (x3 * 20, y3 * 20)-(x3 * 18, y3 * 18), vbYellow
        v = g
        picvel.FontSize = 12
        picvel.BorderStyle = 1
        xV = Cos(-ang) * 3.7
```

```
yv = Sin(-ang) * 3.7
```

```
picvel.CurrentX = 20 * xV + -3.12 - picvel.TextWidth(v) / 2
```

```
picvel.CurrentY = 20 * yv + 8 - picvel.TextWidth(v) / 2
```

```
picvel.Print v
```

```
Else
```

```
picvel.DrawWidth = 1
```

```
picvel.Line (x3 * 20, y3 * 20)-(x3 * 18, y3 * 18), vbGreen
```

```
End If
```

```
picvel.DrawWidth = 1
```

```
picvel.Line (x3 * 20, y3 * 20)-(x3 * 18, y3 * 18), vbGreen
```

```
Next g
```

```
picvel.FillColor = vbBlue
```

```
picvel.FillStyle = vbSolid
```

```
picvel.DrawWidth = 1
```

```
picvel.Line (-24, -60)-(24, -82), vbRed, B
```

```
picvel.FillColor = &H808080
```

```
picvel.FillStyle = vbCross
```

```
picvel.DrawWidth = 3
```

```
picvel.Line (-26, -59)-(26, -83), vbYellow, B
```

```
picvel.CurrentX = cx - picvel.TextWidth(s) - 2
picvel.CurrentY = cy - picvel.TextHeight(s) - 73
picvel.FontSize = 18
picvel.Print s * 1000
If s = 0 Then
t = 0
Else
t = 30
End If
Timer2.Interval = CInt(t + s / 20)
```

```
End If
```

```
End Sub
```

```
Private Sub btnDetener_Click()
If (Timer1.Interval = 0) Then
    btnDetener.Caption = "Detener"
    Timer1.Interval = 1000
Else
    btnDetener.Caption = "Comenzar"
```



```
Timer1.Interval = 0
```

```
End If
```

```
End Sub
```

```
Private Sub btnSalir_Click()
```

```
Me.Hide
```

```
End Sub
```

```
Public Sub verFrm()
```

```
skinActual.ApplySkin Me.hWnd
```

```
Me.Show vbModal
```

```
Velocidad (frmSensorData.txtFormula.Text)
```

```
End Sub
```

```
Sub moverVehiculos()
```

```
Y = Y + 1: If Y = 7 Then Y = 0
```

```
For img = 0 To 11
```

```
Picture1(img).Picture = Image1(Y)
```

```
Next img
```

```
Me.Icon = Image1(Y).Picture
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
    Velocidad (frmSensorData.txtFormula.Text)
```

```
End Sub
```

```
Private Sub Timer2_Timer()
```

```
    moverVehiculos
```

```
End Sub
```

Formulario frmReportesDinamicos.frm

Option Explicit

Dim CmpAgr As Boolean

```
Private Sub btnExport_Click()
```

```
On Error GoTo ErrHandler
```

```
Dlg.Filter = "Excel File(*.xls)|*.xls|Text File (*.txt)|*.txt"
```

```
Dlg.FilterIndex = 1
```

```
Dlg.ShowSave
```

```
' ExportarGrid MSFDatos, Dlg.FileName, Dlg.FilterIndex
```

```
ErrorHandler:
```

```
End Sub
```

```
*****
```

```
*****
```

```
Private Sub CmbAtab_Click()
```

```
Dim tStrAlias, i, vPosA
```

```
tStrAlias = CmbAlias & " AS " & UCase(TxtAliaTab)
```

```
vPosA = FindTab(CmbAlias)
```

```
For i = 1 To ListAlias.ListCount
```

```
    If (ListAlias.List(i - 1) = tStrAlias Or ListAlias.List(i) = tStrAlias) Then
```

```
        sonError
```

```
        frmMensaje.mensaje "No puede agregar el nuevo Alias, esta ya se  
encuentra en la lista.", "ScanAmt06"
```

```
        Exit Sub
```

```
    End If
```

```
Next
```

```
TxtAliaTab = ""
```

```
ListAlias.AddItem tStrAlias
```

```
If vPosA = CmbAlias.ListCount Then
    CmbConAlia.Enabled = True
End If
End Sub
```

```
*****
*****
```

```
Private Sub CmbCampC_Click()
    Dim vPosT, vPosC, Func, Oper, i
    vPosT = FindTab(CmbTabC)
    vPosC = FindCmp(CmbTabC, CmbCampC)
    With Tablas(vPosT).tCampo(vPosC)
        Select Case UCase(.cTipo)
            Case "INT"
                InfoCondicion = "Informacion: " + vbCrLf + "Campo: Valor Entero - Longitud
Maxima: " + .cLongitud
                Oper = "Between,Like,=,>,<,>=,<=,<>"
            Case "VARCHAR", "CHAR"
                InfoCondicion = "Informacion: " + vbCrLf + "Campo: Valor Alfanumerico -
Longitud Maxima: " + .cLongitud
                Oper = "Like,=,<>"
            Case "ENUM"
```

```

        InfoCondicion = "Informacion: " + vbCrLf + "Campo: Lista - Valores : " +
.cValores

        Oper = "Like,=,<>"

        Case "DATE"

            InfoCondicion = "Informacion: " + vbCrLf + "Campo: Fecha - Formato:
YYYY/MM/DD"

            Oper = "Between,=,>,<,>=,<=,<>"

            Case "TIME"

                InfoCondicion = "Informacion: " + vbCrLf + "Campo: Tiempo - Formato:
HH:MM:SS"

                Oper = "Between,=,>,<,>=,<=,<>"

            Case Else

                InfoCondicion = ""

        End Select

    End With

    CmbOperC.Clear

    Oper = Split(Oper, ",")

    For i = 0 To UBound(Oper)

        CmbOperC.AddItem Oper(i)

    Next

    If CmbOperC.ListCount > 0 Then

        CmbOperC.ListIndex = 0

    End If

```

End Sub

Private Sub CmbCampF_Click()

Dim vPosT, vPosC, Func, i

vPosT = FindTab(CmbTabF)

vPosC = FindCmp(CmbTabF, CmbCampF)

With Tablas(vPosT).tCampo(vPosC)

Select Case UCase(.cTipo)

Case "INT"

InfoCampo = "Informacion: " + vbCrLf + "Campo: Valor Entero - Longitud

Maxima: " + .cLongitud

Func = "NINGUNO,COUNT,MAX,MIN,AVG,SUM"

Case "VARCHAR", "CHAR"

InfoCampo = "Informacion: " + vbCrLf + "Campo: Valor Alfanumerico -

Longitud Maxima: " + .cLongitud

Func = "NINGUNO,COUNT"

Case "ENUM"

InfoCampo = "Informacion: " + vbCrLf + "Campo: Lista - Valores : " +

.cValores

Func = "NINGUNO,COUNT"

Case "DATE"

InfoCampo = "Informacion: " + vbCrLf + "Campo: Fecha - Formato:

YYYY/MM/DD"

```
        Func = "NINGUNO,COUNT,MAX,MIN"

    Case "TIME"

        InfoCampo = "Informacion: " + vbCrLf + "Campo: Tiempo - Formato:
HH:MM:SS"

        Func = "NINGUNO,COUNT,MAX,MIN"

    Case Else

        InfoCampo = ""

    End Select

End With

CmbFuncF.Clear

Func = Split(Func, ",")

For i = 0 To UBound(Func)

    CmbFuncF.AddItem Func(i)

Next

CmbFuncF.ListIndex = 0

End Sub

Private Sub CmbCmp1_Click()

    Dim tPos1, cPos1, tPos2, cPos2

    Dim tab1, tab2 As String

    Dim sw As Boolean
```

```
tPos1 = FindATab(CmbTab1)
```

```
tPos2 = FindATab(CmbTab2)
```

```
cPos1 = FindACmp(CmbTab1, CmbCmp1)
```

```
cPos2 = FindACmp(CmbTab2, CmbCmp2)
```

```
Call con
```

```
ds.Open conex
```

```
Set rs = ds.Execute("select * from scsllaves where (llav_nom1 = "" +  
Tablas(tPos1).tNombre + "" and llav_nom2= "" + Tablas(tPos2).tNombre + "" )")
```

```
If Not rs.EOF Then
```

```
    tab1 = rs!llav_cod_tab1
```

```
    tab2 = rs!llav_cod_tab2
```

```
    sw = True
```

```
Else
```

```
    rs.Close
```

```
    Set rs = ds.Execute("select * from scsllaves where (llav_nom2 = "" +  
Tablas(tPos1).tNombre + "" and llav_nom1 = "" + Tablas(tPos2).tNombre + "" )")
```

```
If Not rs.EOF Then
```

```
    tab1 = rs!llav_cod_tab1
```

```
    tab2 = rs!llav_cod_tab2
```

```
    sw = True
```

```
Else
```

```
    sw = False
```



```
End If
End If
rs.Close
ds.Close

InfoJoin = "Informacion: " + vbCrLf

If (Tablas(tPos1).tCampo(cPos1).cTipo <> Tablas(tPos2).tCampo(cPos2).cTipo)
Then
    InfoJoin = InfoJoin + "No se puede realizar la union por estos campos, por no
ser del mismo tipo" + _
        vbCrLf + CmbTab1 + "." + CmbCmp1 + ": " +
Tablas(tPos1).tCampo(cPos1).cTipo + _
        vbCrLf + CmbTab2 + "." + CmbCmp2 + ": " +
Tablas(tPos2).tCampo(cPos2).cTipo
    CmdAddJoin.Enabled = False
Elseif ((CmbCmp1 = UCase(tab1) And CmbCmp2 <> UCase(tab2))) Then
    InfoJoin = InfoJoin + " No se puede realizar la union por estos campos, por no
tener Relacion la Relacion existente es:" + _
        vbCrLf + _
        vbCrLf + "De la tabla " + CmbTab1 + ": El Campo " + UCase(tab1) + _
        vbCrLf + _
        vbCrLf + "De la tabla " + CmbTab2 + ": El Campo " + UCase(tab2)
```

```
CmdAddJoin.Enabled = False

Elseif ((CmbCmp2 = UCase(tab1) And CmbCmp1 <> UCase(tab2))) Then

    InfoJoin = InfoJoin + " No se puede realizar la union por estos campos, por no
tener Relacion la Relacion existente es:" + _

        vbCrLf + _

        vbCrLf + "De la tabla " + CmbTab1 + ": El Campo " + UCase(tab1) + _

        vbCrLf + _

        vbCrLf + "De la tabla " + CmbTab2 + ": El Campo " + UCase(tab2)

CmdAddJoin.Enabled = False

Elseif ((CmbCmp1 <> UCase(tab1) And CmbCmp2 <> UCase(tab2)) And
(CmbCmp1 <> UCase(tab2) And CmbCmp2 <> UCase(tab1))) Then

    InfoJoin = InfoJoin + " No se puede realizar la union por estos campos, por no
tener Relacion la Relacion existente es:" + _

        vbCrLf + _

        vbCrLf + "De la tabla " + CmbTab1 + ": El Campo " + UCase(tab1) + _

        vbCrLf + _

        vbCrLf + "De la tabla " + CmbTab2 + ": El Campo " + UCase(tab2)

CmdAddJoin.Enabled = False

If Not (sw) Then

    frmMensaje.mensaje "Seleccione Tablas con Relacion", " ScanAmt06"

    CmbTab1.ListIndex = 0

    CmbTab2.ListIndex = 1

End If
```

```
Elseif Not (sw) Then
    frmMensaje.mensaje "Seleccione Tablas con Relacion", " ScanAmt06"
    CmbTab1.ListIndex = 0
    CmbTab2.ListIndex = 1
Else
    InfoJoin = InfoJoin + vbCrLf + "Puede continuar con la union."
    CmdAddJoin.Enabled = True
End If
End Sub

Private Sub CmbCmp2_Click()
    CmbCmp1_Click
End Sub

Private Sub CmbConAlia_Click()
    Dim i As Long
    Dim txt
    InitTab
    If (ListAlias.ListCount > 0) Then
        For i = 1 To ListAlias.ListCount
            txt = Split(ListAlias.List(i - 1), " ")
            TxtAliaTab = txt(2)
            addAlias (TxtAliaTab)
```

```
        Next
    End If
    If (pos > 1) Then
        SSTab.TabEnabled(1) = True
        SSTab.Tab = 1
    Else
        SSTab.TabEnabled(2) = True
        SSTab.Tab = 2
    End If
End Sub
```

```
Private Sub CmbLimAlia_Click()
    ListAlias.Clear
    TxtAliaTab = ""
End Sub
```

```
Private Sub CmbLimDato_Click()
    MSFDatos.Clear
    MSFDatos.Rows = 2
    MSFDatos.Cols = 1
End Sub
```

```
Private Sub CmbOperC_Click()
```

```
Select Case UCase(CmbOperC)
    Case "BETWEEN": ValC2.Enabled = True
    Case Else: ValC2.Enabled = False
End Select
End Sub

Private Sub CmbRegAlia_Click()
    SSTab.Tab = 0
End Sub

Private Sub CmbRegRes_Click()
    SSTab.Tab = 3
End Sub

Private Sub CmbTab1_Click()
    Dim vPos, i
    If (CmbTab1 <> CmbTab2) Then
        vPos = FindATab(CmbTab1)
        CmbCmp1.Clear
        For i = 1 To Tablas(vPos).tNumCamp
            CmbCmp1.AddItem Tablas(vPos).tCampo(i).cNombre
        Next
        CmbCmp1.ListIndex = 0
    End If
End Sub
```

```
Else
sonError
    frmMensaje.mensaje "La union debe ser entre dos tablas diferentes,
seleccione otra tabla para poder continuar.", "ScanAmt06"
    If (CmbTab1.ListIndex > 0) Then
        CmbTab1.ListIndex = 0
    Else
        CmbTab1.ListIndex = 1
    End If
    CmbTab1_Click
End If
End Sub
```

```
Private Sub CmbTab2_Click()
    Dim vPos, i
    If (CmbTab1 <> CmbTab2) Then
        vPos = FindATab(CmbTab2)
        CmbCmp2.Clear
        For i = 1 To Tablas(vPos).tNumCamp
            CmbCmp2.AddItem Tablas(vPos).tCampo(i).cNombre
        Next
        CmbCmp2.ListIndex = 0
    Else
```

sonError

frmMensaje.mensaje "La union debe ser entre dos tablas diferentes,
seleccione otra tabla para poder continuar.", "ScanAmt06"

If (CmbTab2.ListIndex > 0) Then

CmbTab2.ListIndex = 0

Else

CmbTab2.ListIndex = 1

End If

CmbTab2_Click

End If

End Sub

Private Sub CmbTabC_Click()

Dim vPos, i

vPos = FindATab(CmbTabC)

CmbCampC.Clear

For i = 1 To Tablas(vPos).tNumCamp

CmbCampC.AddItem Tablas(vPos).tCampo(i).cNombre

Next

CmbCampC.ListIndex = 0

End Sub

Private Sub CmbTabF_Click()

```

Dim vPos, i
vPos = FindATab(CmbTabF)
CmbCampF.Clear
For i = 1 To Tablas(vPos).tNumCamp
    CmbCampF.AddItem Tablas(vPos).tCampo(i).cNombre
Next
CmbCampF.ListIndex = 0
End Sub

```

```

Private Sub CmdAddCamp_Click()

```

```

    Dim KeyWord, tStrCmp, ListaAgr, xCmp, i, sw As Boolean
    KeyWord = " ' SELECT AS FROM WHERE GROUP BY ORDER HAVING SUM
MAX MIN AVG COUNT INSERT INTO VALUES UPDATE SET UNION
INTERSETION EXEPT EXECUTE DELETE DROP ALTER TABLE CREATE
MODIFY COLUMN ADD CONSTRAINT PRIMARY KEY RENAME VARCHAR
CHAR NUMBER INT ENUM TIME DATE SEQUENCE JOIN LEFT RIGTH "

```

```

    tStrCmp = CmbTabF + "." + CmbCampF

```

```

    If UCase(CmbFuncF) <> "NINGUNO" Then

```

```

        tStrCmp = CmbFuncF + "(" + tStrCmp + ")"

```

```

    End If

```

```

    If Trim(txtAlias) <> vbNullString Then

```

```

        If (InStr(1, KeyWord, UCase(Trim(txtAlias)))) = 0) Then

```



```
tStrCmp = tStrCmp + " AS " + txtAlias
Else
sonError
frmMensaje.mensaje "Los alias no pueden contener las siguientes
palabras" + vbCrLf + KeyWord, "ScanAmt06"
Exit Sub
End If
End If

For i = 1 To ListCondiciones.ListCount
If (ListCampos.List(i - 1) = tStrCmp Or ListCampos.List(i) = tStrCmp) Then
sonError
frmMensaje.mensaje "No puede agregar la nueva union, esta ya se
encuentra en la lista", "ScanAmt06"
Exit Sub
End If
Next

If CmbFuncF <> "NINGUNO" Then
If CmpAgr Or ListCampos.ListCount = 0 Then
ListCampos.AddItem tStrCmp
Else
ListaAgr = ""
```

```
For i = 1 To ListCampos.ListCount
```

```
  If (InStr(1, ListCampos.List(i - 1), "(")) Then
```

```
    xCmp = Split(ListCampos.List(i - 1), "(")
```

```
    xCmp = Split(xCmp(0), "(")
```

```
    xCmp = xCmp(1)
```

```
  Else
```

```
    xCmp = Split(ListCampos.List(i - 1), " AS ")
```

```
    xCmp = xCmp(0)
```

```
  End If
```

```
  If ListaAgr = vbNullString Then
```

```
    ListaAgr = xCmp
```

```
  Else
```

```
    ListaAgr = ListaAgr + "," + xCmp
```

```
  End If
```

```
Next
```

```
sonError
```

```
  If (frmMensaje.mensaje("Atencion al insertar esta funcion de agregacion  
" + vbCrLf + "usted debe agrupar la consulta por los siguientes campos: " + vbCrLf  
+ ListaAgr + vbCrLf + "Desea hacerlo?", vbYesNo + vbInformation, "Atencion!") =  
vbYes) Then
```

```
    ListCampos.AddItem tStrCmp
```

```
    CmpAgr = True
```

```
        ListaAgr = Split(ListaAgr, ",")
        For i = 0 To UBound(ListaAgr)
            ListGroup.AddItem Trim(ListaAgr(i))
        Next
    Else
        Exit Sub
    End If
End If

Else
    If (Not CmpAgr) Then
        sw = False
        xCmp = CmbTabF + "." + CmbCampF
        For i = 1 To ListCampos.ListCount
            If (InStr(1, ListCampos.List(i - 1), "COUNT") Or InStr(1, ListGroup.List(i
- 1), "MIN") Or InStr(1, ListGroup.List(i - 1), "MAX") Or InStr(1, ListGroup.List(i - 1),
"SUM") Or InStr(1, ListGroup.List(i - 1), "AVG")) Then
                sw = True
            Exit For
        End If
    Next
    If (Not sw) Then
        ListCampos.AddItem tStrCmp
    Else
```



sonError

frmMensaje.mensaje "Atencion para adicionar este campo, usted debe agrupar la consulta por el mismo desea hacerlo?", "ScanAmt06"

ListCampos.AddItem tStrCmp

ListGroup.AddItem xCmp

CmpAgr = True

End If

Else

sw = False

xCmp = CmbTabF + "." + CmbCampF

For i = 1 To ListGroup.ListCount

If xCmp = Trim(ListGroup.List(i - 1)) Then

sw = True

Exit For

End If

Next

If (sw) Then

ListCampos.AddItem tStrCmp

Else

sonError

If (frmMensaje.mensaje("Atencion para adicionar este campo, usted debe agrupar la consulta por el mismo desea hacerlo?", vbYesNo + "SacarAmt06") = vbYes) Then

```

        ListCampos.AddItem tStrCmp
        ListGroup.AddItem xCmp
    Else
        Exit Sub
    End If
End If
End If
End If
End Sub

Private Sub CmdAddCond_Click()
    Dim tStrCond, Msg, tPos, cPos, i
    tPos = FindTab(CmbTabC)
    cPos = FindCmp(CmbTabC, CmbCampC)
    If (UCase(CmbOperC) = "LIKE") Then
        tStrCond = CmbTabC + "." + CmbCampC + " " + CmbOperC + " " + ValC1
+ "%" + ""
    Else
        tStrCond = CmbTabC + "." + CmbCampC + " " + CmbOperC + " " + ValC1 +
""
    End If
    For i = 1 To ListCondiciones.ListCount

```

```
    If (ListCondiciones.List(i - 1) = tStrCond Or ListCondiciones.List(i) = tStrCond)
Then
    sonError
    frmMensaje.mensaje "No puede agregar la nueva union, esta ya se
encuentra en la lista.", "ScanAmt06"
    Exit Sub
End If
Next

With Tablas(tPos).tCampo(cPos)

    If Not ValidarDato(.cTipo, ValC1, .cValores) Then
        ValC1.SetFocus
        Exit Sub
    End If

    If (UCCase(CmbOperC) = "BETWEEN") Then
        If Not ValidarDato(.cTipo, ValC2, .cValores) Then
            ValC2.SetFocus
            Exit Sub
        End If

        tStrCond = tStrCond + " AND " + ValC2 + ""
    End If
```

```
If (CmbOpeLC.Enabled) Then
    tStrCond = CmbOpeLC + " " + tStrCond
End If
ListCondiciones.AddItem tStrCond
If (ListCondiciones.ListCount > 0) Then
    CmbOpeLC.Enabled = True
    CmbOpeLC.ListIndex = 0
End If
End With
End Sub

Private Sub CmdAddJoin_Click()
    Dim tStrJoin, tStrJoin2, i
    tStrJoin = CmbTab1 & "." & CmbCmp1 & " = " & CmbTab2 & "." & CmbCmp2
    tStrJoin2 = tStrJoin
    For i = 1 To ListJoin.ListCount
        If (ListJoin.List(i - 1) = tStrJoin Or ListJoin.List(i) = tStrJoin2) Then
            frmMensaje.mensaje "No puede agregar la nueva union, esta ya se
encuentra en la lista", vbInformation, "Atencion!"
            Exit Sub
        End If
    Next
    ListJoin.AddItem tStrJoin
```

End Sub

Private Sub CmdBackCampo_Click()

 SSTab.Tab = 2

End Sub

Private Sub CmdBackCond_Click()

 If (pos > 1) Then

 SSTab.Tab = 1

 Else

 SSTab.Tab = 0

 End If

End Sub

Private Sub CmdBackJoin_Click()

 SSTab.Tab = 0

End Sub

Private Sub CmdClearCampo_Click()

 ListCampos.Clear

 ListGroup.Clear

 CmpAgr = False

End Sub

Private Sub CmdClearCon_Click()

 CmbOpeLC.Enabled = False

 ListCondiciones.Clear

End Sub

Private Sub CmdClearJoin_Click()

 ListJoin.Clear

End Sub

Private Sub CmdDatos_Click()

 Dim vSql, i, J, a As Field

 On Error GoTo EtError

 Call con

 ds.Open conex

 cmd.CommandText = Replace(Trim(TxtSql), vbCrLf, " ")

 rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic

 i = 0: J = 0

 MSFDatos.Clear

 MSFDatos.Rows = 2

 MSFDatos.Cols = 1

 If Not rs.EOF Then



```

While Not rs.EOF

    MSFDatos.Rows = MSFDatos.Rows + 1

    For Each a In rs.Fields

        J = J + 1

        If i = 0 Then

            MSFDatos.Cols = MSFDatos.Cols + 1

            MSFDatos.TextMatrix(0, J - 1) = a.Name

        End If

        MSFDatos.TextMatrix(MSFDatos.Rows - 2, J - 1) = IIf(IsNull(a.Value), "
", a.Value)

    Next

    rs.MoveNext

    i = i + 1

    J = 0

Wend

Else

    sonError

    frmMensaje.mensaje "No se hayo ningun registro para la consulta
establecida", "ScanAmt06"

End If

rs.Close

ds.Close

MSFDatos.Cols = MSFDatos.Cols - 1

```

Exit Sub

EtError:

ds.Close

sonError

frmMensaje.mensaje "Error en su consulta SQL", "ScanAmt06"

End Sub

Private Sub CmdNext1_Click()

Dim i As Long

InitTab

If (ListTablas.SelCount > 0) Then

ds.Open conex

For i = 0 To ListTablas.ListCount - 1

If (ListTablas.Selected(i)) Then

addTab (ListTablas.List(i))

rs.Open SqlColumns & ListTablas.List(i), ds, adOpenDynamic,

adLockBatchOptimistic

While Not rs.EOF

addCmp ListTablas.List(i), rs.Fields(0), rs.Fields(1), rs.Fields(2),

rs.Fields(3)

rs.MoveNext

Wend

rs.Close

```
        End If
    Next
    ds.Close

    If (pos >= 1) Then
        SSTab.TabEnabled(5) = True
        SSTab.Tab = 5
    End If
Else
    sonError
        frmMensaje.mensaje "Para poder continuar, por lo menos debe seleccionar
una tabla", "ScanAmt06"
    End If

End Sub

Private Sub CmdNextCampo_Click()
    SSTab.Tab = 4
    SSTab.TabEnabled(4) = True
End Sub

Private Sub CmdNextCond_Click()
    SSTab.Tab = 3
```

```
SSTab.TabEnabled(3) = True
```

```
End Sub
```

```
Private Sub CmdNextJoin_Click()
```

```
    SSTab.Tab = 2
```

```
    SSTab.TabEnabled(2) = True
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
skinActual.ApplySkin Me.hWnd
```

```
    Dim i As Long
```

```
    Load_Sql_To_List ListTablas, SqlTab, 0
```

```
    For i = 1 To 5
```

```
        SSTab.TabEnabled(i) = False
```

```
    Next
```

```
    CmdClearJoin_Click
```

```
    CmdClearCon_Click
```

```
    CmdClearCampo_Click
```

```
    Toolbar1.Buttons(2).Enabled = False
```

```
    Toolbar1.Buttons(3).Enabled = False
```

```
    CmbConAlia.Enabled = False
```

```
End Sub
```



```

Sub verFrm()
    skinActual.ApplySkin Me.hWnd
    Me.Show vbModal
End Sub

Private Sub ListTablas_Click()
    Dim i, d, a, f As Long
    Dim J, e, c As Long
    Dim Ntab1, Ctab1, Ntab2, Ctab2 As String

    If (ListTablas.SelCount > 0) Then
        pos = 0
        ds.Open conex
        For i = 0 To ListTablas.ListCount - 1
            If (ListTablas.Selected(i)) Then
                addTab (ListTablas.List(i))
                rs.Open SqlColumns & ListTablas.List(i), ds, adOpenDynamic,
adLockBatchOptimistic
                While Not rs.EOF
                    addCmp ListTablas.List(i), rs.Fields(0), rs.Fields(1), rs.Fields(2),
rs.Fields(3)
                    rs.MoveNext
                Wend
            End If
        Next i
    End If
End Sub

```

```
        rs.Close
    End If
Next
ds.Close
ComTab
End If
End Sub
```

```
Private Sub SSTab_Click(PreviousTab As Integer)
```

```
    Dim i, J
```

```
    Select Case SSTab.Tab
```

```
        Case 0: 'TABLAS
```

```
            Form_Load
```

```
        Case 1: 'JOIN
```

```
            CmbTab1.Clear
```

```
            CmbTab2.Clear
```

```
            If pos > 1 Then
```

```
                For i = 1 To pos
```

```
                    CmbTab1.AddItem Alias(i).tNombre
```

```
                    CmbTab2.AddItem Alias(i).tNombre
```

```
                Next
```

```
                CmbTab1.ListIndex = 0
```

```
                CmbTab2.ListIndex = 1
```

End If

Case 2: 'WHERE

CmbTabC.Clear

If pos > 0 Then

For i = 1 To pos

CmbTabC.AddItem Alias(i).tNombre

Next

CmbTabC.ListIndex = 0

End If

Case 3: 'CAMPOS

CmpAgr = False

CmbTabF.Clear

If pos > 0 Then

For i = 1 To pos

CmbTabF.AddItem Alias(i).tNombre

Next

CmbTabF.ListIndex = 0

End If

Case 4: 'RESULTADOS

TxtSql = "SELECT" + vbCrLf

***** CAMPOS

If (ListCampos.ListCount > 0) Then

For i = 1 To ListCampos.ListCount

```
        TxtSql = TxtSql + ListCampos.List(i - 1) + ", "  
    Next  
    TxtSql = Mid(TxtSql, 1, Len(TxtSql) - 2) + vbCrLf  
Else  
    TxtSql = TxtSql + "*" + vbCrLf  
End If
```

```
'***** TABLAS
```

```
TxtSql = TxtSql + "FROM" + vbCrLf  
If (ListAlias.ListCount > 0) Then  
    For i = 1 To ListAlias.ListCount  
        TxtSql = TxtSql + ListAlias.List(i - 1) + ", "  
    Next  
Else  
    For i = 1 To pos  
        TxtSql = TxtSql + Tablas(i).tNombre + ", "  
    Next  
End If  
TxtSql = Mid(TxtSql, 1, Len(TxtSql) - 2) + vbCrLf
```

```
'***** JOIN
```

```
If (ListJoin.ListCount > 0) Then  
    TxtSql = TxtSql + "WHERE ( " + vbCrLf
```

```
For i = 1 To ListJoin.ListCount
    TxtSql = TxtSql + ListJoin.List(i - 1) + vbCrLf + "AND "
Next
TxtSql = Mid(TxtSql, 1, Len(TxtSql) - 4) + ")"
End If
```

```
'***** WHERE
```

```
If (ListCondiciones.ListCount > 0) Then
    If (ListJoin.ListCount = 0) Then
        TxtSql = TxtSql + "WHERE" + vbCrLf
    Else
        TxtSql = TxtSql + "AND "
    End If
End If
```

```
For i = 1 To ListCondiciones.ListCount
    TxtSql = TxtSql + ListCondiciones.List(i - 1) + " "
Next
'TxtSql = TxtSql + ")"
End If
```

```
'***** GROUP BY
```

```
If (ListGroup.ListCount > 0) Then
    TxtSql = TxtSql + vbCrLf + "GROUP BY"
    For i = 1 To ListGroup.ListCount
        TxtSql = TxtSql + vbCrLf + ListGroup.List(i - 1) + ", "
```

Next

 TxtSql = Mid(TxtSql, 1, Len(TxtSql) - 2) + vbCrLf

End If

 Toolbar1.Buttons(2).Enabled = True

 Toolbar1.Buttons(3).Enabled = True

 Toolbar1.Buttons(4).Enabled = True

Case 5: 'WHERE

 CmbAlias.Clear

 If pos > 0 Then

 For i = 1 To pos

 CmbAlias.AddItem Tablas(i).tNombre

 Next

 CmbAlias.ListIndex = 0

 End If

End Select

End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)

 Dim f, i, J, Linea, Abrir

 Select Case Button.Index

 Case 1: SSTab.Tab = 0



```
CmdClearJoin_Click
CmdClearCon_Click
CmdClearCampo_Click
CmbLimDato_Click
CmbLimAlia_Click
CmbConAlia.Enabled = False
```

Case 2:

```
'Guardar Datos
Dlg.Filter = "Archivo Texto|*.txt"
Dlg.ShowSave
If Trim(Dlg.FileName) <> vbNullString Then
    Abrir = Dlg.FileName
    f = FreeFile
    Open Dlg.FileName For Output As f
    Print #f, vbCrLf
    Print #f, Tab(3); "Sistema de Informacion "
    Print #f, Tab(3); "Generador de Reportes Dinamico"
    Print #f, Tab(3); "Reporte creado por: " + txtuserw
    Print #f, Tab(3); "Fecha          : " + FormatDateTime(Date, vbLongDate)
    Print #f, Tab(3); "Hora            : " + CStr(Time) + vbCrLf
    With Me.MSFDatos
        For i = 0 To .Rows - 1
            Linea = ""
```

```

    If i = 1 Then
        For J = 0 To .Cols - 1
            Linea = Linea + StrRepeat("_", MAXLONGCOL(Me.MSFDatos,
J) + 1) + " "
        Next
        Print #f, Tab(3); Linea
        Linea = ""
    End If
    For J = 0 To .Cols - 1
        Linea = Linea + .TextMatrix(i, J) + StrRepeat(" ",
MAXLONGCOL(Me.MSFDatos, J) + 2 - Len(.TextMatrix(i, J)))
    Next
    Print #f, Tab(3); Linea
Next
End With
Close f
If frmSiNo.mensaje1("Desea ver el contenido de este archivo .TXT?",
vbQuestion + vbYesNo, "Atencion") = vbYes Then
    Shell "notepad " + Abrir, vbNormalFocus
End If
End If
Case 3:
'Guardar SQL

```

```

Dlg.Filter = "Archivo Sql|*.sql"

Dlg.ShowSave

If Trim(Dlg.FileName) <> vbNullString Then

    Abrir = Dlg.FileName

    f = FreeFile

    Open Dlg.FileName For Output As f

    Print #f, TxtSql

    Close f

    If frmSiNo.mensaje1("Desea ver el contenido de este archivo .SQL?",
vbQuestion + vbYesNo, "Atencion") = vbYes Then

        Shell "notepad " + Abrir, vbNormalFocus

    End If

End If

Case 4: Unload Me

End Select

End Sub

```

```

Function MAXLONGCOL(msf As MSFlexGrid, ByVal COL As Long) As Long

    Dim i

    MAXLONGCOL = 0

    With msf

        For i = 0 To .Rows - 1

```

```
If Len(.TextMatrix(i, COL)) > MAXLONGCOL Then
    MAXLONGCOL = Len(.TextMatrix(i, COL))
End If
Next
End With
End Function
```

```
Function StrRepeat(ByVal Character As String, ByVal Cantidad As Long) As String
    Dim i
    StrRepeat = ""
    For i = 1 To Cantidad
        StrRepeat = StrRepeat + Character
    Next
End Function
```

Formulario fmRegistro.fm

```
Private Sub cmbtipo_GotFocus()
    cmbtipo.BackColor = &HE0E0E0
End Sub
```

```
Private Sub cmbtipo_LostFocus()
    cmbtipo.BackColor = &HFFFFFF
```



End Sub

Private Sub cmdbuscar_Click()

Dim consulta As String

If txtusuario.Text = Empty Then

sonError

frmMensaje.mensaje " Por Favor Digite el Usuario", "ScannAmt06"

Else

Set registro = New ADODB.Recordset

consulta = "select * from usuarios where usuario=" & _

txtusuario.Text & " "

'cmdlimpiar_Click

Set registro = conexion.Execute(consulta)

If registro.EOF Then

frmMensaje.mensaje "Este usuario no existe en la base de datos.",

"ScannAmt06"

txtusuario.SetFocus

txtusuario.SelStart = 0

txtusuario.SelLength = Len(txtusuario.Text)

Else

txtusuario.Text = registro!Usuario

txtclave.Text = registro!clave

```
cmbtipo = registro!tipo  
txtcedula.Text = registro!cedula  
txtnombre.Text = registro!nombre  
txtdireccion.Text = registro!direccion  
txttelefono.Text = registro!telefono
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub cmdeliminar_Click()
```

```
Dim consulta As String
```

```
Dim eliminar As String
```

```
If txtusuario.Text = Empty Then
```

```
sonError
```

```
frmMensaje.mensaje "Digite el Usuario", "ScannAmt06"
```

```
Else
```

```
Set registro = New ADODB.Recordset
```

```
consulta = "select * from usuarios where usuario=" & _
```

```
txtusuario.Text & " "
```

```
Set registro = conexion.Execute(consulta)
```

```
If registro.EOF Then
```

```
frmMensaje.mensaje "Este usuario no existe en la base de datos.",  
"ScannAmt06"
```

```
Else
```

```
txtusuario.Text = registro!Usuario
```

```
txtclave.Text = registro!clave
```

```
cmbtipo = registro!tipo
```

```
txtcedula.Text = registro!cedula
```

```
txtnombre.Text = registro!nombre
```

```
txtdireccion.Text = registro!direccion
```

```
txttelefono.Text = registro!telefono
```

```
frmSiNo.mensaje1 "Desea eliminar este Usuario?", "ScanAmt06"
```

```
If eliminarReg = True Then
```

```
eliminar = "delete from usuarios where usuario=" & txtusuario.Text & ""
```

```
conexion.Execute (eliminar)
```

```
frmMensaje.mensaje "Usuario Eliminado exitosamente.", "ScannAmt06"
```

```
End If
```

```
cmdlimpiar_Click
```

```
End If
```

```
End If
```

End Sub

Private Sub cmdguardar_Click()

Dim adicionar As String

Dim consulta As String

Set registro = New ADODB.Recordset

consulta = "select * from usuarios where usuario=" & _

txtusuario.Text & " "

Set registro = conexion.Execute(consulta)

If (validarRegistro(txtusuario, txtclave, cmbtipo, txtcedula, txtnombre, txtdireccion, txttelefono)) Then

If registro.EOF Then

adicionar = "insert into usuarios (usuario, clave, tipo, cedula, nombre,direccion,telefono) VALUES (" & txtusuario.Text & ", " & Encrypted(txtclave.Text) & ", " & cmbtipo & ", " & txtcedula.Text & ", " & txtnombre.Text & ", " & txtdireccion.Text & ", " & txttelefono.Text & ")"

conexion.Execute (adicionar)

frmMensaje.mensaje "usuario Ingresado Exitosamente.", "ScannAmt06"

cmdlimpiar_Click

Else

modificar = " update usuarios set usuario=" & txtusuario.Text & ",clave=" & Encrypted(txtclave.Text) & ",tipo=" & cmbtipo & ",cedula=" & txtcedula.Text & "

```
,nombre="" & txtnombre.Text & " ,direccion="" & txtdireccion.Text & ",telefono="" &
txttelefono.Text & " where usuario="" & txtusuario.Text & ""
```

```
conexion.Execute (modificar)
```

```
frmMensaje.mensaje "usuario Actualizado Exitosamente.", "ScannAmt06"
```

```
cmdlimpiar_Click
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub cmdlimpiar_Click()
```

```
txtusuario.Text = ""
```

```
txtusuario.SetFocus
```

```
txtclave.Text = ""
```

```
cmbtipo.Clear
```

```
cmbtipo.AddItem ("Administrador")
```

```
cmbtipo.AddItem ("usuario")
```

```
txtcedula.Text = ""
```

```
txtnombre.Text = ""
```

```
txtdireccion.Text = ""
```

```
txttelefono.Text = ""
```

```
End Sub
```



```
Private Sub cmdSalir_Click()
```

```
Me.Hide
```

```
End Sub
```

```
Public Sub Form_Load()
```

```
'abrirBaseDato
```

```
Dim nomskin As String
```

```
    cmbtipo.AddItem ("Administrador")
```

```
    cmbtipo.AddItem ("Usuario")
```

```
End Sub
```

```
Sub verFrm()
```

```
    skinActual.ApplySkin Me.hWnd
```

```
    Me.Show vbModal
```

```
End Sub
```

```
Private Sub txtcedula_GotFocus()
```

```
    txtcedula.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtcedula_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txtcedula_LostFocus()
```

```
    txtcedula.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub txtclave_GotFocus()
```

```
    txtclave.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtclave_LostFocus()
```

```
    txtclave.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub txtdireccion_GotFocus()
```

```
    txtdireccion.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtdireccion_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

End Sub

```
Private Sub txtdireccion_LostFocus()
```

```
txtdireccion.BackColor = &HFFFFFFF
```

End Sub

```
Private Sub txtnombre_GotFocus()
```

```
txtnombre.BackColor = &HE0E0E0
```

End Sub

```
Private Sub txtnombre_KeyPress(KeyAscii As Integer)
```

```
KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

End Sub

```
Private Sub txtnombre_LostFocus()
```

```
txtnombre.BackColor = &HFFFFFFF
```

End Sub

```
Private Sub txttelefono_GotFocus()
```

```
txttelefono.BackColor = &HE0E0E0
```

End Sub

```
Private Sub txttelefono_KeyPress(KeyAscii As Integer)
```



```
KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txttelefono_LostFocus()
```

```
txttelefono.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub txtusuario_GotFocus()
```

```
txtusuario.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtusuario_LostFocus()
```

```
txtusuario.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub txtusuario_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
If KeyCode = vbKeyF2 Then
```

```
AyudaScann.PopUp "ggjhkjk", 9000, 1800
```

```
End If
```

```
End Sub
```

Fomulario frmRecursos.frm

```
Private Sub mnuComenzar_Click()  
    frmSensorData.btnComenzar_Click  
End Sub
```

```
Private Sub mnuConfPuerto_Click()  
    frmSensorData.btnConfPuerto_Click  
End Sub
```

```
Private Sub mnuGrafico_Click()  
    frmSensorData.btnGrafico_Click  
End Sub
```

```
Private Sub mnuLimpiar_Click()  
    frmSensorData.btnLimpiar_Click  
End Sub
```

```
Private Sub mnuReporte_Click()  
    frmSensorData.btnReporte_Click  
  
End Sub
```

```
Private Sub mnuSalir_Click()  
frmSensorData.btnSalir_Click
```

```
End Sub
```

```
Formulario frmPropetarios.frm
```

```
Private Sub cmdbuscar_Click()
```

```
Dim consulta As String
```

```
    If txtcedula.Text = Empty Then
```

```
        sonError
```

```
        frmMensaje.mensaje "Por Favor Digite la Cedula .", "ScannAmt06..."
```

```
    Else
```

```
        Set registro = New ADODB.Recordset
```

```
        consulta = "select * from Propietarios where cedula=" & _
```

```
        txtcedula.Text & " " "
```

```
        'cmdlimpiar_Click
```

```
        Set registro = conexion.Execute(consulta)
```

```
        If registro.EOF Then
```

```
            frmMensaje.mensaje "Este Propietario no Existe en la Base de Datos .",
```

```
"ScannAmt06..."
```

```
            txtcedula.SetFocus
```

```
            txtcedula.SelStart = 0
```

```

        txtcedula.SelLength = Len(txtcedula.Text)

Else

    txtcedula.Text = registro!cedula

    txtnombre.Text = registro!nombre

    txtdireccion.Text = registro!direccion

    txttelefono = registro!telefono

End If

End If

End Sub

Private Sub cmdeliminar_Click()

Dim consulta As String

    Dim eliminar As String

    If txtcedula.Text = Empty Then

        sonError

        frmMensaje.mensaje "Digite la Cedula", "ScannAmt06..."

    Else

        Set registro = New ADODB.Recordset

        consulta = "select * from Propietarios where cedula=" & _
            txtcedula.Text & " "

        Set registro = conexion.Execute(consulta)

        If registro.EOF Then

```

```
frmMensaje.mensaje "Este Propietario no Existe en la Base de Datos .",  
"ScannAmt06..."
```

```
Else
```

```
txtcedula.Text = registro!cedula
```

```
txtnombre.Text = registro!nombre
```

```
txtdireccion.Text = registro!direccion
```

```
txttelefono = registro!telefono
```

```
frmSiNo.mensaje1 "Desea eliminar este Propietario? ", "ScannAmt06"
```

```
If eliminarReg = True Then
```

```
eliminar = "delete from Propietarios where cedula=" & txtcedula.Text & ""
```

```
conexion.Execute (eliminar)
```

```
frmMensaje.mensaje "Propietario Eliminado Exitosamente.",
```

```
"ScannAmt06..."
```

```
End If
```

```
cmdlimpiar_Click
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub cmdguardar_Click()
```

```
Dim adicionar As String
```

```
Dim consulta As String
```



```

Set registro = New ADODB.Recordset

    consulta = "select * from propietarios where cedula=" & _
    txtcedula.Text & " "

Set registro = conexion.Execute(consulta)

If (validarPropietario(txtcedula, txtnombre, txtdireccion, txttelefono)) Then

    If registro.EOF Then

        adicionar = "insert into propietarios (cedula, nombre, direccion, telefono)
VALUES (" & txtcedula.Text & ", " & txtnombre.Text & ", " & txtdireccion.Text & "
," & txttelefono & " )"

        conexion.Execute (adicionar)

        frmMensaje.mensaje      "Propietario      Ingresado      Exitosamente.",
"ScannAmt06..."

        cmdlimpiar_Click

    Else

        modificar = " update propietarios set nombre=" & txtnombre.Text & "
,direccion=" & txtdireccion.Text & " , telefono=" & txttelefono & " where cedula="
& txtcedula.Text & ""

        conexion.Execute (modificar)

        frmMensaje.mensaje      "Propietario      Actualizado      Exitosamente.",
"ScannAmt06..."

        cmdlimpiar_Click

    End If

```

End If

End Sub

Private Sub cmdlimpiar_Click()

txtcedula.Text = ""

txtcedula.SetFocus

txtnombre.Text = ""

txtdireccion.Text = ""

txttelefono.Text = ""

End Sub

Private Sub cmdSalir_Click()

Me.Hide

End Sub

Public Sub Form_Load()

'abrirBaseDato

End Sub

Sub verFrm()

skinActual.ApplySkin Me.hWnd

Me.Show vbModal

End Sub

```
Private Sub txtcedula_GotFocus()  
txtcedula.BackColor = &HE0E0E0  
End Sub  
Private Sub txtcedula_KeyPress(KeyAscii As Integer)  
KeyAscii = Asc(UCase(Chr(KeyAscii)))  
End Sub  
Private Sub txtcedula_LostFocus()  
txtcedula.BackColor = &HFFFFFF  
End Sub  
  
Private Sub txtdireccion_GotFocus()  
txtdireccion.BackColor = &HE0E0E0  
End Sub  
Private Sub txtdireccion_KeyPress(KeyAscii As Integer)  
KeyAscii = Asc(UCase(Chr(KeyAscii)))  
End Sub  
Private Sub txtdireccion_LostFocus()  
txtdireccion.BackColor = &HFFFFFF  
End Sub  
  
Private Sub txtnombre_GotFocus()  
txtnombre.BackColor = &HE0E0E0  
End Sub
```

```
Private Sub txtnombre_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txtnombre_LostFocus()
```

```
txtnombre.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub txttelefono_GotFocus()
```

```
txttelefono.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txttelefono_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txttelefono_LostFocus()
```

```
txttelefono.BackColor = &HFFFFFF
```

```
End Sub
```

Formulario frmMensaje.frm

Option Explicit

Private ayuda As AyudaEnLinea

Private parametro As Long

Public Function mensaje(Msg As String, tit As String, Optional contex As Long = 1000&)

On Error Resume Next

skinActual.ApplySkin Me.hWnd

parametro = contex

txtMensaje.Caption = Msg

frmMensaje.Caption = tit

frmMensaje.Show vbModal

' btnAceptar.SetFocus

End Function

Private Sub btnAceptar_Click()

On Local Error Resume Next

Unload Me

End Sub



```
Private Sub btnAyuda_Click()  
    AyudaScann.mostrarContex parametro
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
End Sub
```

```
Formulario frmmain.frm
```

```
Option Explicit
```

```
Dim a As String
```

```
Dim t As String
```

```
Dim J As String
```

```
Dim c As String
```

```
Dim b As Integer
```

```
Dim i As Integer
```

```
Public Skin As Boolean
```

```
Private Sub barH_ButtonClick(ByVal Button As ComctlLib.Button)
```

```
sonClick
```

```
Select Case Button.Index
```

```
    Case 1
```

frmvehiculo.verFrm

Case 2

frmPropietarios.verFrm

Case 3

frmBSensor.verFrm

Case 5

frmSensorData.verFrm

Case 6

frmDTC.verFrm

Case 7

frmExperto.verFrm

Case 9

AyudaScann.mostrarContenido

End Select

End Sub

Private Sub Command1_Click()

' ayuda.mostrarContex (9001)

End Sub

```
Private Sub Form_Load()
```

```
    Call ShockwaveFlash1.LoadMovie(0, App.Path + "/flash/scanner.swf")
```

```
    cargarSkin
```

```
    numPuerto = Val(GetSetting(App.Title, "configuracion", "numpuerto", ""))
```

```
    escucharSonido = GetSetting(App.Title, "conSonido", "escucharsonido", "")
```

```
    mnuSonido.Checked = escucharSonido
```

```
    mnuTitulo.Checked = GetSetting(App.Title, "conTitulo", "titulo", "")
```

```
    If (mnuTitulo.Checked = True) Then Timer1.Interval = 100
```

```
    If (numPuerto = 0) Then numPuerto = 4
```

```
    barEstado.Panels(5).Text = "Puerto Com: " & numPuerto & " Cerrado"
```

```
    titulo
```

```
    sonEntrar
```

```
    comenzarAyuda
```

```
End Sub
```

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y  
As Single)
```

```
If Button = 2 Then
    Me.Enabled = False
    PopupMenu mnuscannear
    Me.Enabled = True
    Me.SetFocus
End If
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    sonSalir
```

```
    cerrarBaseDato
```

```
    Set AyudaScann = Nothing
```

```
    SaveSetting App.Title, "configuracion", "numpuerto", numPuerto
```

```
    SaveSetting App.Title, "conSkin", "nomSkin", skinActual.Name
```

```
    SaveSetting App.Title, "conSonido", "escucharsonido", mnuSonido.Checked
```

```
    SaveSetting App.Title, "conTitulo", "titulo", mnuTitulo.Checked
```

```
End
```

```
End Sub
```

```
Sub cargarSkin()
```

```
    nomskin = GetSetting(App.Title, "conSkin", "nomSkin", "")
```

```
    If (skinBstudio.Name = nomskin) Then
```

```
        Set skinActual = skinBstudio
```

```
mnuBstudi0.Checked = True
Elseif (skinChizh.Name = nomskin) Then
    Set skinActual = skinChizh
    mnuChizh.Checked = True
Elseif (skinGalaxy.Name = nomskin) Then
    Set skinActual = skinGalaxy
    mnuGalaxy.Checked = True
Elseif (skinGreen.Name = nomskin) Then
    Set skinActual = skinGreen
    mnuGreen.Checked = True
Elseif (skinMedia.Name = nomskin) Then
    Set skinActual = skinMedia
    mnuMedia.Checked = True
Elseif (skinWeb.Name = nomskin) Then
    Set skinActual = skinWeb
    mnuWeb.Checked = True
Elseif (skinTopsecret.Name = nomskin) Then
    Set skinActual = skinTopsecret
    mnuTopSecret.Checked = True
Else
    Set skinActual = skinGalaxy
    mnuTopSecret.Checked = True
End If
```

```
skinActual.ApplySkin Me.hWnd
End Sub
Public Sub comenzarAyuda()
Set AyudaScann = New AyudaEnLinea
With AyudaScann
.archivoAyuda = App.HelpFile
.hWnd = Me.hWnd
End With
End Sub
```

```
Private Sub mnuacerca_Click()
frmAcerca.Show
End Sub
```

```
Private Sub mnuBar_Click()
mnuBar.Checked = Not mnuBar.Checked
barEstado.Visible = mnuBar.Checked
End Sub
```

```
Private Sub mnuDTC_Click()
```



sonClick

frmBDTC.verFrm

End Sub

Private Sub mnurdinamicos_Click()

sonClick

frmReportesDinamicos.verFrm

End Sub

Private Sub mnuSensor_Click()

sonClick

frmBSensor.verFrm

End Sub

Private Sub mnuSonido_Click()

mnuSonido.Checked = Not mnuSonido.Checked

escucharSonido = mnuSonido.Checked

End Sub

Private Sub mnuTitulo_Click()

mnuTitulo.Checked = Not mnuTitulo.Checked

Me.Caption = "<----- ScannAmt06 ----->"

If (mnuTitulo.Checked = True) Then

```
Timer1.Interval = 100
```

```
Else
```

```
Timer1.Interval = 0
```

```
End If
```

```
Me.Caption = "<----- ScannAmt06 ----->"
```

```
End Sub
```

```
Private Sub mnuBarHer_Click()
```

```
mnuBarHer.Checked = Not mnuBarHer.Checked
```

```
barH.Visible = mnuBarHer.Checked
```

```
End Sub
```

```
Private Sub mnubusqueda_Click()
```

```
AyudaScann.mostrarBusqueda
```

```
End Sub
```

```
Private Sub mnuCerrarAyuda_Click()
```

```
AyudaScann.cerrarAyuda
```

```
End Sub
```

```
Private Sub mnuConfPuerto_Click()
```

```
    frmConfigurarPuerto.verFrm
```

```
End Sub
```

```
Private Sub mnucontenido_Click()
```

```
    AyudaScann.mostrarContenido
```

```
End Sub
```

```
Private Sub mnuExperto_Click()
```

```
    frmExperto.verFrm
```

```
End Sub
```

```
Private Sub mnuexportar_Click()
```

```
    frmExportImport.verFrm
```

```
End Sub
```

```
Sub seleccionarSkin()
```

```
    mnuBstudi0.Checked = False
```

```
    mnuChizh.Checked = False
```

```
    mnuGalaxy.Checked = False
```

```
    mnuGreen.Checked = False
```

```
    mnuMedia.Checked = False
```

```
    mnuWeb.Checked = False
```

```
    mnuTopSecret.Checked = False
```

```
End Sub
```

```
Private Sub mnuBstudi0_Click()  
    skinBstudio.ApplySkin Me.hWnd  
    seleccionarSkin  
    mnuBstudi0.Checked = True  
    Set skinActual = skinBstudio  
End Sub
```

```
Private Sub mnuChizh_Click()  
    skinChizh.ApplySkin Me.hWnd  
    seleccionarSkin  
    mnuChizh.Checked = True  
    Set skinActual = skinChizh  
End Sub
```

```
Private Sub mnuGalaxy_Click()  
    skinGalaxy.ApplySkin Me.hWnd  
    seleccionarSkin  
    mnuGalaxy.Checked = True  
    Set skinActual = skinGalaxy  
End Sub
```

```
Private Sub mnuGreen_Click()  
    skinGreen.ApplySkin Me.hWnd  
    seleccionarSkin
```



```
        mnuGreen.Checked = True
    Set skinActual = skinGreen
End Sub

Private Sub mnuMedia_Click()
    skinMedia.ApplySkin Me.hWnd
    seleccionarSkin
    mnuMedia.Checked = True
    Set skinActual = skinMedia
End Sub

Private Sub mnuTopSecret_Click()
    skinTopsecret.ApplySkin Me.hWnd
    seleccionarSkin
    mnuTopSecret.Checked = True
    Set skinActual = skinTopsecret
End Sub

Private Sub mnuWeb_Click()
    skinWeb.ApplySkin Me.hWnd
    seleccionarSkin
    mnuWeb.Checked = True
    Set skinActual = skinWeb
End Sub
```

```
Private Sub mnuTip_Click()
```

```
    frmTip.verFrm
```

```
End Sub
```

```
Private Sub mnuindice_Click()
```

```
    AyudaScann.mostrarIndice
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
*****
```

```
t = Left(a, b)
```

```
Me.Caption = t
```

```
J = Left(c, b)
```

```
b = b + 1
```

```
If b > i Then b = 0
```

```
End Sub
```

```
Sub titulo()
```

```
a = "<----- ScannAmt06 ----->"
```

```
i = Len(a)
```

```
    b = 0
```

```
End Sub
```

```
Private Sub mnuleercodigos_Click()
```

```
    frmDTC.verFrm
```

```
End Sub
```

```
Private Sub mnuleersensor_Click()
```

```
    sonClick
```

```
    frmSensorData.verFrm
```

```
End Sub
```

```
Private Sub mnupropietario_Click()
```

```
    sonClick
```

```
    frmPropietarios.verFrm
```

```
End Sub
```

```
Private Sub mnusuario_Click()
```

```
    sonClick
```

```
    frmregistro.verFrm
```

```
End Sub
```

```
Private Sub mnuvehiculo_Click()
```

```
    sonClick
```

```
    frmvehiculo.verFrm
```

```
End Sub
```

Formulario frmExportImport.frm

Sub Limpiar()

Dim i

For i = 1 To ListTablas.ListCount

ListTablas.Selected(i - 1) = False

Next

PBar.Value = 0.0001

End Sub

Private Sub CmbTab_Click()

Load_Sql_To_List ListCmp, SqlColumns + " " + CmbTab, 0

End Sub

Private Sub CmdExport_Click()

Dim i, total, indice, vCmp As Field, Linea, f

Dim a As Field

Call con

ds.Open conex

total = 0

PBar.Value = 0.0001

InitTab

If ListTablas.SelCount > 0 Then

```

For i = 1 To ListTablas.ListCount
    If ListTablas.Selected(i - 1) Then
        addTab (ListTablas.List(i - 1))
        cmd.CommandText = "select count(*) from " & ListTablas.List(i - 1)
        rs.Open      cmd.CommandText,      ds,      adOpenDynamic,
adLockBatchOptimistic
        total = total + rs.Fields(0)
        rs.Close

    End If

Next

indice = 0

For i = 1 To pos
    Set rs = ds.Execute("select * from " & Tablas(i).tNombre & " ")

    If Not rs.EOF Then
        Dlg.Filter = "Archivo Imex|*.imex"
        Dlg.FileName = Tablas(i).tNombre
        Dlg.ShowSave
        If Trim(Dlg.FileName) <> vbNullString Then
            f = FreeFile
            Open Dlg.FileName For Output As f
            While Not rs.EOF
                Linea = ""

```

```

For Each a In rs.Fields
    Linea = Linea & "" & If(IsNull(a.Value), " ", a.Value) & ""
Next
Print #f, Mid(Linea, 1, Len(Linea) - 1)
indice = indice + 1
rs.MoveNext

Wend

Close f

PBar.Value = CInt((indice / total) * 100)

frmMensaje.mensaje "Proceso de la tabla " + Tablas(i).tNombre + " ha
terminado satisfactoriamente ", "ScanAmt06"

frmSiNo.mensaje1 "Desea ver el contenido de este archivo?",
"Atencion"

If (eliminarReg = True) Then
    Shell "notepad " + Dlg.FileName, vbNormalFocus
End If

End If

End If

rs.Close

Next

Limpiar

Else

sonError

```

```
        frmMensaje.mensaje "Debe selecciona por lo menos una tabla para continuar  
con la operacion", "ScanAmt06"
```

```
    End If
```

```
    ds.Close
```

```
End Sub
```

```
Private Sub CmdFind_Click()
```

```
    Dlg.Filter = "Archivos de Datos Import Export|*.imex"
```

```
    Dlg.FileName = ""
```

```
    Dlg.ShowOpen
```

```
    If Trim(Dlg.FileName) <> vbNullString Then
```

```
        If (Trim(Dlg.FileName) = Trim(TxtFileOut)) Then
```

```
            sonError
```

```
                frmMensaje.mensaje "No puede seleccionar este archivo ya lo esta usando  
para almacenar los errores", "ScanAmt06"
```

```
        Else
```

```
            TxtFileIn = Dlg.FileName
```

```
        End If
```

```
    End If
```

```
End Sub
```

```
Private Sub CmdImport_Click()
```

```
'*****  
' verificamos la existencia de los archivos  
If Trim(TxtFileOut) = vbNullString Then  
    sonError  
        frmMensaje.mensaje "No se ha definido el archivo para almacenar los errores  
en caso de haberlo", "ScanAmt06"  
        Exit Sub  
    End If  
If Trim(TxtFileIn) = vbNullString Then  
    sonError  
        frmMensaje.mensaje "No se ha definido el archivo para cargar los datos",  
"ScanAmt06"  
        Exit Sub  
    End If  
CmdImport.Enabled = False  
FileOut = TxtFileOut  
If ExistFile(FileOut) Then  
    Kill FileOut  
End If  
If Not ExistFile(TxtFileIn) Then  
sonError
```



frmMensaje.mensaje "No se encontro el archivo especificado para cargar los
datos", "ScanAmt06"

Exit Sub

End If

' *****

' ** cargamos los datos de la tabla seleccionada

InitTab

addTab CmbTab

Call con

ds.Open conex

cmd.CommandText = SqlColumns + " " + CmbTab

rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic

While Not rs.EOF

addCmp CmbTab, rs.Fields(0), rs.Fields(1), rs.Fields(2), rs.Fields(3)

rs.MoveNext

Wend

ds.Close

' verificar cantidad de campos de la tabla y sus valores

If VerificarCmp(TxtFileIn, FileOut) Then

LoadDB

```
        frmMensaje.mensaje "Los datos han sido cargados satisfactoriamente.",  
"ScanAmt06"  
        PIE.Value = 0.0001  
    End If  
    frmSiNo.mensaje1 "Se han encontrado errores en el archivo estos han sido  
almacenados en:" + vbCrLf + TxtFileOut + vbCrLf + "Desea ver el contenido de  
este archivo?", "ScanAmt06"  
    If (eliminarReg = True) Then  
        Shell "notepad " + TxtFileOut, vbNormalFocus  
    End If  
    CmdImport.Enabled = True  
    TxtFileOut = ""  
    TxtFileIn = ""  
    CmbTab.ListIndex = 0  
End Sub  
  
Private Sub CmdSave_Click()  
    Dlg.Filter = "Archivos de Errores Import Export[*].err"  
    Dlg.FileName = ""  
    Dlg.ShowSave  
    If Trim(Dlg.FileName) <> vbNullString Then  
        If (Trim(Dlg.FileName) = Trim(TxtFileIn)) Then
```

```
        MsgBox "No puede seleccionar este archivo ya lo esta usando extraer los  
datos"
```

```
    Else
```

```
        TxtFileOut = Dlg.FileName
```

```
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Load_Sql_To_List ListTablas, SqlTab, 0
```

```
    Load_Sql_To_Cmb CmbTab, SqlTab, 0
```

```
End Sub
```

```
Sub verFrm()
```

```
    skinActual.ApplySkin Me.hWnd
```

```
    Me.Show vbModal
```

```
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub LoadDB()
```

```
    Dim f, Linea, Lista, nLinea
```

```
f = FreeFile
nLinea = 0
PIE.Value = 0.0001
Open TxtFileIn For Input As #f
While Not EOF(f)
    nLinea = nLinea + 1
    Line Input #f, Linea
    Linea = Trim(Linea)
    If Linea <> vbNullString Then
        ds.Open conex
        ds.Execute ("insert into " + CmbTab + " values(" + Linea + ")")
        ds.Close
    End If
    PIE.Value = CInt((nLinea / NumeroLineas) * 100)
Wend
Close #f
End Sub
```

```
Function NewLinea(ByVal Linea As String) As String
```

```
    Dim cLinea, Lista, i
    cLinea = Mid(Linea, 2, Len(Linea) - 2)
    Lista = Split(cLinea, ",")
    For i = 0 To UBound(Lista)
```

```
If IsDate(Lista(i)) Then
    NewLinea = "" + FormatFecha2(Lista(i)) + ","
Else
    NewLinea = DateTime(Lista(i))
End If

Next

NewLinea = Mid(NewLinea, 1, Len(NewLinea) - 1)

End Function
```

Formulario frmDTC.frm

Option Explicit

Public OK

Dim bOpenPort As Boolean

Private Sub cargarDTC()

Dim registro As New ADODB.Recordset

registro.CursorType = adOpenStatic

registro.CursorLocation = adUseClient

registro.Open "select * from dtc where codigo=" & cmbdtc & "", cadena,
adOpenKeyset, adLockOptimistic

txtDtcActual.Text = cmbdtc

If registro.EOF Then

txtDescripcion.Text = "Código de falla desconocido"

```
        txtSolucion.Text = "Sin Solución"

    Else

        txtDescripcion.Text = registro!descripcion
        txtSolucion.Text = registro!solucion

    End If

    registro.Close

End Sub

Private Sub btnConfPuerto_Click()

    frmConfigurarPuerto.verFrm

End Sub

Private Sub cmbdtc_Change()

    cargarDTC

End Sub

Private Sub cmbdtc_Click()

    cargarDTC

End Sub

Private Sub cmdLeer_Click()

    On Error Resume Next

    Dim v As String
```



```
Dim dtc As String
Dim dato As String
Dim cuantos As Integer
Dim b As String
Dim pbyte As String
Dim c As String

    enviarComando ("0101")

    v = leerPuerto
    'v = "410187076504"
    cuantos = HexADec(Trim(Left(Mid(v, 7), 2 * 1 + 1)))
    If (cuantos > 128) Then
        lblTotal.Caption = cuantos - 128
    End If

    enviarComando ("03")
    dato = leerPuerto
    'dato = "43013300000000"
    b = Mid(dato, 3)
    pbyte = Left(b, 1)
    c = Mid(b, 2)
    dtc = Left(c, 3)
    If (pbyte = 0) Then
        dtc = "P0" + dtc
```

End If

cmbdtc.AddItem (dtc)

End Sub

Private Sub cmdSalir_Click()

Me.Hide

cerrarPuerto

End Sub

Sub verFrm()

skinActual.ApplySkin Me.hWnd

Me.Show vbModal

End Sub

Private Function abrirPuerto()

On Error GoTo error

With puerto

If .PortOpen = True Then

.PortOpen = False

bOpenPort = True

End If

.CommPort = numPuerto 'Paso 1: elijo el puerto

.Settings = "9600,N,8,1"

```
.PortOpen = True ' Paso 3: Intento abrir el
If bOpenPort = True Then
    .PortOpen = True
    bOpenPort = False
End If
End With
```

GoTo salir 'Todo Ok, continuar.

error:

frmMensaje.mensaje "El puerto com " & numPuerto & " esta siendo usado por otra aplicación. Para habilitar otro número de puerto en el Menu 'Opciones' --> Configurar Puerto ", "ScanAmt06, puerto no disponible"

frmmain.barEstado.Panels(5).Text = "Puerto Com:" & numPuerto & " no disponible"

Resume salir ' Resume me permite continuar con el programa.

salir:

```
If (puerto.PortOpen = True) Then
    frmmain.barEstado.Panels(5).Text = "Puerto Com " & numPuerto & ";
Abierto."
End If
```

```
End Function

Private Sub cerrarPuerto()
On Error GoTo error

If (puerto.PortOpen = True) Then
    puerto.PortOpen = False 'Puede haber error si
    'intento cerrar un puerto que está en uso por otro
    'programa, entre otras causas.

    frmmain.barEstado.Panels(5).Text = "Puerto Com " & numPuerto & ": Cerrado."
End If

GoTo salir

error:

    frmMensaje.mensaje "Error al intentar cerrar COM " & numPuerto & "puerto no
disponible." & Err.Description, "ScanAmt06"

Resume salir

salir:

End Sub

Private Sub enviarComando(cmd As String)

    Dim envio As Variant

    On Error GoTo error

        envio = cmd + Chr$(13)
```

```
If (puerto.PortOpen = True) Then
```

```
    puerto.OutBufferCount = 0
```

```
    puerto.Output = envio
```

```
End If
```

```
GoTo salir
```

```
error:
```

```
    frmMensaje.mensaje "Ocurrio un error al intentar enviar el texto. Error: " &  
Err.Description, "ScanAmt06..."
```

```
    Resume salir
```

```
salir:
```

```
End Sub
```

```
Function leerPuerto()
```

```
    Dim datoPuerto As Variant
```

```
    Dim datoString As String
```

```
    Dim prom As String
```

```
    If (puerto.InBufferCount > 0) Then
```

```
        datoPuerto = puerto.Input
```

```
    End If
```

```
    datoString = CStr(datoPuerto)
```

```
    prom = InStr(datoString, ">")
```

```
If (prom = 1) Then
    leerPuerto = PROMPT
Elseif (Len(datoString) = 0) Then
    leerPuerto = Vacio
Else
    datoString = Replace(datoString, "0101", "")
    datoString = Replace(datoString, "03", "")
    datoString = Replace(datoString, vbCr, "")
    datoString = Replace(datoString, vbCrLf, "")
    datoString = Replace(datoString, vbLf, "")
    leerPuerto = datoString
End If
```

End Function

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    cerrarPuerto
```

```
End Sub
```

Formulario frmConfigurarPuerto.frm

```
Private Sub btnAceptar_Click()  
    Dim num As Integer  
    Dim tam As Integer  
    tam = Len(cmbPuertos)  
    If (tam = 4) Then  
        i = Right(cmbPuertos, 1)  
    ElseIf (tam = 5) Then  
        i = Right(cmbPuertos, 2)  
    End If  
  
    If (Val(i) <> 0) Then  
        numPuerto = i  
        Me.Hide  
        frmMensaje.mensaje "Puerto Com " & numPuerto & " Seleccionado",  
"ScannAmt06"  
    Else  
        sonError  
        frmMensaje.mensaje "Seleccione un Puerto, por favor", "ScannAmt06"  
    End If  
  
End Sub  
  
Private Sub Command2_Click()
```



```
Me.Hide
```

```
End Sub
```

```
Private Sub btnCancelar_Click()
```

```
Me.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
llenarPuertosComDisponible
```

```
End Sub
```

```
Sub verFrm()
```

```
skinActual.ApplySkin Me.hWnd
```

```
llenarPuertosComDisponible
```

```
Me.Show vbModal
```

```
End Sub
```

```
Sub llenarPuertosComDisponible()
```

```
Dim l As Long, pos As Long, aux As String
```

```
Dim buffer(1024) As Byte, BytesNecesarios As Long, InfoDevueltos As Long
```

```
Dim InfoPort As PORT_INFO_2
```

```
Dim num As String
```

```
cmbPuertos.Clear

I = EnumPorts(vbNullString, 2, buffer(0), 1024, BytesNecesarios, InfoDevueltos)

pos = 0

For I = 1 To InfoDevueltos

    CopyMemory InfoPort, buffer(pos), LenB(InfoPort)

    num = PointerToString(InfoPort.pPortName)

    num = Replace(num, ":", "")

    If (Left(num, 3) = "COM") Then

        cmbPuertos.AddItem num

        aux = ""

    End If

    pos = pos + LenB(InfoPort)

Next I

End Sub
```

Formulario frmBSensor.frm

```
Sub verFrm()

skinActual.ApplySkin Me.hWnd

Me.Show vbModal

End Sub

Private Sub cmdbuscar_Click()

Dim consulta As String
```

```
If txtcodBsensor.Text = Empty Then

sonError

frmMensaje.mensaje "Por Favor Digite el Código del Sensor .", "ScannAmt06..."

Else

Set registro = New ADODB.Recordset

    consulta = "select * from sensor where codigo=" & _
    txtcodBsensor.Text & " "

    'cmdlimpiar_Click

Set registro = conexion.Execute(consulta)

If registro.EOF Then

frmMensaje.mensaje "Este Sensor no Existe en la Base de Datos .",
"ScannAmt06..."

    txtcodBsensor.SetFocus

    txtcodBsensor.SelStart = 0

    txtcodBsensor.SelLength = Len(txtcodBsensor.Text)

Else

    txtcodBsensor.Text = registro!codigo

    txtnomBsensor.Text = registro!nombre

    txtcomanBsensor.Text = registro!comando

    txtbytesBsensor.Text = registro!bytes

    txtacroBsensor.Text = registro!acronimo

    txtunidBsensor.Text = registro!unidad

    txtforBsensor.Text = registro!formula
```

```

    End If

    End If

End Sub

Private Sub cmdeliminar_Click()

    Dim consulta As String

    Dim eliminar As String

    If txtcodBsensor.Text = Empty Then

        sonError

        frmMensaje.mensaje "Digite el Código del Sensor", "ScannAmt06..."

    Else

        Set registro = New ADODB.Recordset

        consulta = "select * from sensor where codigo=" & _
            txtcodBsensor.Text & " "

        Set registro = conexion.Execute(consulta)

        If registro.EOF Then

            frmMensaje.mensaje "Este Sensor no Existe en la Base de Datos .",
                "ScannAmt06..."

        Else

            txtcodBsensor.Text = registro!codigo

            txtnomBsensor.Text = registro!nombre

            txtcomanBsensor.Text = registro!comando

            txtbytesBsensor.Text = registro!bytes

        End If

    End If

End Sub

```

txtacroBsensor.Text = registro!acronimo

txtunidBsensor.Text = registro!unidad

txtforBsensor.Text = registro!formula

frmSiNo.mensaje1 "Desea Eliminar este Sensor? ", "ScannAmt06"

If eliminarReg = True Then

eliminar = "delete from sensor where codigo=" & txtcodBsensor.Text & ""

conexion.Execute (eliminar)

frmMensaje.mensaje "Sensor Eliminado Exitosamente.", "ScannAmt06..."

End If

cmdlimpiar_Click

End If

End If

End Sub

Private Sub cmdguardar_Click()

Dim adicionar As String

Dim consulta As String

Set registro = New ADODB.Recordset

consulta = "select * from sensor where codigo=" & _

txtcodBsensor.Text & ""

Set registro = conexion.Execute(consulta)



```
If (validarBSensor(txtcodBsensor, txtnomBsensor, txtcomanBsensor,
txtbytesBsensor, txtacroBsensor, txtunidBsensor, txtforBsensor)) Then
```

```
    If registro.EOF Then
```

```
        adicionar = "insert into sensor (codigo, nombre, comando,
bytes,acronimo,unidad,formula) VALUES ('" & txtcodBsensor.Text & "', '" &
txtnomBsensor.Text & "', '" & txtcomanBsensor.Text & "', '" & txtbytesBsensor.Text
& "', '" & txtacroBsensor.Text & "', '" & txtunidBsensor.Text & "', '" &
txtforBsensor.Text & " ')"
```

```
        conexion.Execute (adicionar)
```

```
        frmMensaje.mensaje "SensorIngresado Exitosamente.", "ScannAmt06..."
```

```
        cmdlimpiar_Click
```

```
    Else
```

```
        modificar = " update sensor set nombre='" & txtnomBsensor.Text & "'
,comando='" & txtcomanBsensor.Text & "' , bytes='" & txtbytesBsensor.Text & "'
,acronimo='" & txtacroBsensor.Text & "',unidad='" & txtunidBsensor.Text & "'
,formula='" & txtforBsensor.Text & "' where codigo='" & txtcodBsensor.Text & "'"
```

```
        conexion.Execute (modificar)
```

```
        frmMensaje.mensaje "Sensor Actualizado Exitosamente.", "ScannAmt06..."
```

```
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub cmdlimpiar_Click()
```

```
    txtcodBsensor.Text = ""
```

```
    txtcodBsensor.SetFocus
```

```
    txtnomBsensor.Text = ""
```

```
    txtcomanBsensor.Text = ""
```

```
    txtbytesBsensor.Text = ""
```

```
    txtacroBsensor.Text = ""
```

```
    txtunidBsensor.Text = ""
```

```
    txtforBsensor.Text = ""
```

```
End Sub
```

```
Private Sub cmdSalir_Click()
```

```
    Me.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    abrirBaseDato
```

```
End Sub
```

```
Private Sub txtcodBsensor_GotFocus()
```

```
    txtcodBsensor.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtcodBsensor_LostFocus()  
txtcodBsensor.BackColor = &HFFFFFF  
End Sub
```

```
Private Sub txtnomBsensor_GotFocus()  
txtnomBsensor.BackColor = &HE0E0E0  
End Sub
```

```
Private Sub txtnomBsensor_KeyPress(KeyAscii As Integer)  
KeyAscii = Asc(UCase(Chr(KeyAscii)))  
End Sub
```

```
Private Sub txtnomBsensor_LostFocus()  
txtnomBsensor.BackColor = &HFFFFFF  
End Sub
```

```
Private Sub txtcomanBsensor_GotFocus()  
txtcomanBsensor.BackColor = &HE0E0E0  
End Sub
```

```
Private Sub txtcomanBsensor_KeyPress(KeyAscii As Integer)
```

```
KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txtcomanBsensor_LostFocus()
```

```
txtcomanBsensor.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub txtbytesBsensor_GotFocus()
```

```
txtbytesBsensor.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtbytesBsensor_LostFocus()
```

```
txtbytesBsensor.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub txtacroBsensor_GotFocus()
```

```
txtacroBsensor.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtacroBsensor_KeyPress(KeyAscii As Integer)
```

```
KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txtacroBsensor_LostFocus()
```

```
txtacroBsensor.BackColor = &HFFFFFFF
```

```
End Sub
```

```
Private Sub txtunidBsensor_GotFocus()
```

```
txtsolBDTC.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtunidBsensor_KeyPress(KeyAscii As Integer)
```

```
KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txtunidBsensor_LostFocus()
```

```
txtunidBsensor.BackColor = &HFFFFFFF
```

```
End Sub
```

```
Private Sub txtforBsensor_GotFocus()
```

```
txtforBsensor.BackColor = &HE0E0E0  
End Sub  
Private Sub txtforBsensor_KeyPress(KeyAscii As Integer)  
KeyAscii = Asc(UCase(Chr(KeyAscii)))  
End Sub  
Private Sub txtforBsensor_LostFocus()  
txtforBsensor.BackColor = &HFFFFFFF  
End Sub
```

Formulario frmBrowser.frm

```
Public StartingAddress As String  
Dim mbDontNavigateNow As Boolean  
Private Sub Form_Load()  
On Error Resume Next  
Me.Show  
tbToolBar.Refresh  
Form_Resize  
  
cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15  
  
StartingAddress = "file:/// " & App.Path & "/scannamt06.htm"
```



```
If Len(StartingAddress) > 0 Then
    cboAddress.Text = StartingAddress
    cboAddress.AddItem cboAddress.Text
    'intenta explorar la dirección inicial
    timTimer.Enabled = True
    brwWebBrowser.Navigate StartingAddress
End If
```

```
End Sub
```

```
Sub verFrm()
```

```
skinActual.ApplySkin Me.hWnd
```

```
Me.Show vbModal
```

```
End Sub
```

```
Private Sub brwWebBrowser_DownloadComplete()
```

```
On Error Resume Next
```

```
Me.Caption = brwWebBrowser.LocationName
```

```
End Sub
```

```
Private Sub brwWebBrowser_NavigateComplete2(ByVal pDisp As Object, URL As Variant)
```

```
    On Error Resume Next
```

```
    Dim i As Integer
```

```
    Dim bFound As Boolean
```

```
    Me.Caption = brwWebBrowser.LocationName
```

```
    For i = 0 To cboAddress.ListCount - 1
```

```
        If cboAddress.List(i) = brwWebBrowser.LocationURL Then
```

```
            bFound = True
```

```
            Exit For
```

```
        End If
```

```
    Next i
```

```
    mbDontNavigateNow = True
```

```
    If bFound Then
```

```
        cboAddress.RemoveItem i
```

```
    End If
```

```
    cboAddress.AddItem brwWebBrowser.LocationURL, 0
```

```
    cboAddress.ListIndex = 0
```

```
    mbDontNavigateNow = False
```

```
End Sub
```

```
Private Sub cboAddress_Click()
```

```
If mbDontNavigateNow Then Exit Sub
    timTimer.Enabled = True
    brwWebBrowser.Navigate cboAddress.Text
End Sub
```

```
Private Sub cboAddress_KeyPress(KeyAscii As Integer)
    On Error Resume Next
    If KeyAscii = vbKeyReturn Then
        cboAddress_Click
    End If
End Sub
```

```
Private Sub Form_Resize()
    On Error Resume Next
    cboAddress.Width = Me.ScaleWidth - 100
    brwWebBrowser.Width = Me.ScaleWidth - 100
    brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top +
picAddress.Height) - 100
End Sub
```

```
Private Sub timTimer_Timer()  
    If brwWebBrowser.Busy = False Then  
        timTimer.Enabled = False  
        Me.Caption = brwWebBrowser.LocationName  
    Else  
        Me.Caption = "Trabajando..."  
    End If  
End Sub  
  
Private Sub tbToolBar_ButtonClick(ByVal Button As Button)  
    On Error Resume Next  
  
    timTimer.Enabled = True  
  
    Select Case Button.Key  
        Case "Atrás"  
            brwWebBrowser.GoBack  
        Case "Adelante"  
            brwWebBrowser.GoForward  
        Case "Actualizar"
```

```
        brwWebBrowser.Refresh
    Case "Inicio"
        brwWebBrowser.GoHome
    Case "Búsqueda"
        brwWebBrowser.GoSearch
    Case "Detener"
        timTimer.Enabled = False
        brwWebBrowser.Stop
        Me.Caption = brwWebBrowser.LocationName
End Select
End Sub
```

Formulario frmBDTC.frm

```
Sub verFrm()
    skinActual.ApplySkin Me.hWnd
    Me.Show vbModal
End Sub
```

```
Private Sub cmdbuscar_Click()
```

```
    Dim consulta As String
```

```
    If txtcodBDTC.Text = Empty Then
```

```
        sonError
```

```
        frmMensaje.mensaje "Por Favor Digite el Código del DTC .", "ScannAmt06..."
```

```
    Else
```

```

Set registro = New ADODB.Recordset

    consulta = "select * from dtc where codigo=" & _
    txtcodBDTC.Text & " "
    'cmdlimpiar_Click

Set registro = conexion.Execute(consulta)

If registro.EOF Then

    frmMensaje.mensaje "Este Código del DTC no Existe en la Base de Datos .",
    "ScannAmt06..."

        txtcodBDTC.SetFocus

        txtcodBDTC.SelStart = 0

        txtcodBDTC.SelLength = Len(txtcodBDTC.Text)

    Else

        txtcodBDTC.Text = registro!codigo

        txtdesBDTC.Text = registro!descripcion

        txtsolBDTC.Text = registro!solucion

    End If

End If

End Sub

Private Sub cmdeliminar_Click()

    Dim consulta As String

```

```
Dim eliminar As String
If txtcodBDTC.Text = Empty Then
sonError
frmMensaje.mensaje "Digite el Código del DTC", "ScannAmt06..."
Else
Set registro = New ADODB.Recordset
    consulta = "select * from dtc where codigo=" & _
    txtcodBDTC.Text & " "
Set registro = conexion.Execute(consulta)
If registro.EOF Then
frmMensaje.mensaje " Código del DTC no Existe en la Base de Datos .",
"ScannAmt06..."
Else
    txtcodBDTC.Text = registro!codigo
    txtdesBDTC.Text = registro!descripcion
    txtsolBDTC.Text = registro!solucion

frmSiNo.mensaje1 "Desea eliminar este Código DTC? ", "ScannAmt06"

If eliminarReg = True Then
    eliminar = "delete from dtc where codigo=" & txtcodBDTC.Text & ""
    conexion.Execute (eliminar)
```

```

        frmMensaje.mensaje "Código del DTC Eliminado Exitosamente.",
"ScannAmt06..."
    End If

    cmdlimpiar_Click

End If

End If

End Sub

```

```

Private Sub cmdguardar_Click()

    Dim adicionar As String

    Dim consulta As String

    Set registro = New ADODB.Recordset

        consulta = "select * from dtc where codigo=" & _
        txtcodBDTC.Text & " "

    Set registro = conexion.Execute(consulta)

    If (validarBDTC(txtcodBDTC, txtdesBDTC, txtsolBDTC)) Then

        If registro.EOF Then

            adicionar = "insert into dtc (codigo, descripcion, solucion) VALUES (" &
txtcodBDTC.Text & ", " & txtdesBDTC.Text & ", " & txtsolBDTC.Text & " )"

            conexion.Execute (adicionar)

            frmMensaje.mensaje "Código DTC Ingresado Exitosamente.",
"ScannAmt06..."

```

```
cmdlimpiar_Click

Else

    modificar = " update dtc set descripcion=" & txtdesBDTC.Text & "
,solucion=" & txtsolBDTC.Text & " where codigo=" & txtcodBDTC.Text & ""

    conexion.Execute (modificar)

    frmMensaje.mensaje "Código DTC Actualizado Exitosamente.",
"ScannAmt06..."

    End If

End If
```

```
End Sub
```

```
Private Sub cmdlimpiar_Click()
```

```
txtcodBDTC.Text = ""
```

```
txtcodBDTC.SetFocus
```

```
txtdesBDTC.Text = ""
```

```
txtsolBDTC.Text = ""
```

```
End Sub
```

```
Private Sub cmdSalir_Click()
```

```
Me.Hide
```

```
End Sub
```

```
Private Sub Fom_Load()
```

```
    abrirBaseDato
```

```
End Sub
```

```
Private Sub txtcodBDTC_GotFocus()
```

```
    txtcodBDTC.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtcodBDTC_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txtcodBDTC_LostFocus()
```

```
    txtcodBDTC.BackColor = &HFFFFFF
```

```
End Sub
```

```
Private Sub txtdesBDTC_GotFocus()
```

```
    txtdesBDTC.BackColor = &HE0E0E0
```

```
End Sub
```

```
Private Sub txtdesBDTC_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

```
Private Sub txtDesBDTC_LostFocus()  
txtDesBDTC.BackColor = &HFFFFFF  
End Sub
```

```
Private Sub txtSolBDTC_GotFocus()  
txtSolBDTC.BackColor = &HE0E0E0  
End Sub
```

```
Private Sub txtSolBDTC_KeyPress(KeyAscii As Integer)  
KeyAscii = Asc(UCase(Chr(KeyAscii)))  
End Sub
```

```
Private Sub txtSolBDTC_LostFocus()  
txtSolBDTC.BackColor = &HFFFFFF  
End Sub
```

6. RECURSOS MINIMOS PARA IMPLANTAR EL SISTEMA

- COMPUTADOR CON PLATAFORMA OPERATIVA WINDOWS 2000/XP/...
 - MOTOR DE BASE DE DATOS MYSQL.
 - INSTALADORES DEL SISTEMA.
-

7. TERMINOLOGÍA TÉCNICA USADA

- ✓ Software:
- ✓ Taller Automotriz 1-A:
- ✓ Excel:
- ✓ Word:
- ✓ Requisitos (Requerimientos):
- ✓ Casos de uso:
- ✓ Modelador de Procesos:
- ✓ Diagrama de Flujo de Datos:
- ✓ Diccionario de Datos:
- ✓ Modelo Relacional:
- ✓ Modelo Entidad-Relación:

DISEÑO E IMPLEMENTACIÓN DE UN SCANNER AUTOMOTOR PARA EL
DIAGNÓSTICO DE LOS SENSORES DE OXIGENO, TEMPERATURA Y
CHOQUE PARA EL VEHÍCULO CHEVROLET BLAZER 1995 PARA EL TALLER
AUTOMOTRIZ 1-A.

YUNELIS GUTIÉRREZ SERRANO
LESLY MILAGRO RODRIGUEZ MONTAÑO
HÉCTOR RAFAEL TORRES CABALLERO

MANUAL DE USUARIO

UNIVERSIDAD SIMÓN BOLÍVAR
FACULTAD DE INGENIERÍA DE SISTEMA
INVESTIGACIÓN FORMATIVA IV
SEMESTRE X DIURNO
BARRANQUILLA
2006.



MANUAL DE USUARIOS

El sistema en su plantilla principal contiene siete módulos o pestañas que también se le puede llamar menú que están distribuidos de una manera muy sencilla; y cada uno de ellos tiene un submenú asociado, que será tema esencial de este manual.

Primero se mostrará la ventana o formulario de acceso al programa.

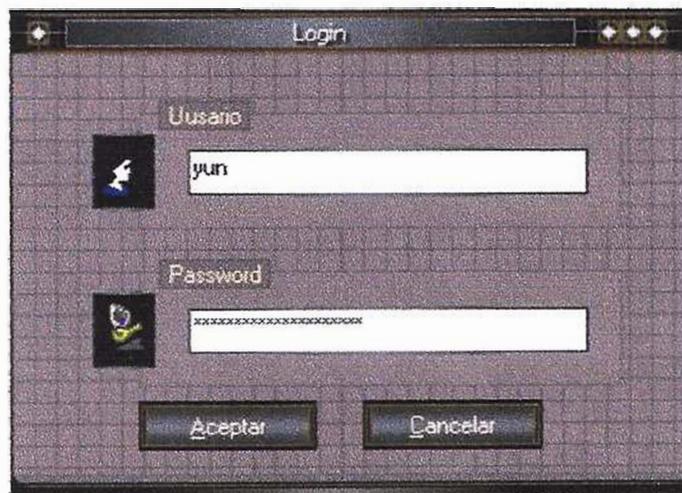


Figura 1. Ventana Login SCANNER AMT06

En esta ventana de acceso solicita los datos de usuario y contraseña almacenados en la base de datos, para poder entrar a la aplicación siguiente deberás estar registrado por alguno de los dos, si entras como Usuario sé deshabilita la opción de Usuario ya que solo el Administrador puede ingresar los

demás usuarios, y si entras como Administrador la opción Usuario estará habilitada. Contiene dos botones que en este caso es aceptar y cancelar.

Donde dice usuario al lado derecho se digitará el usuario y en la otra caja de texto digitará la clave asignada o password para poder entrar a la aplicación siguiente. Como también tiene una cruz en el lado superior este tiene como función cerrar el formulario y el signo menos (-) para minimizar la ventana

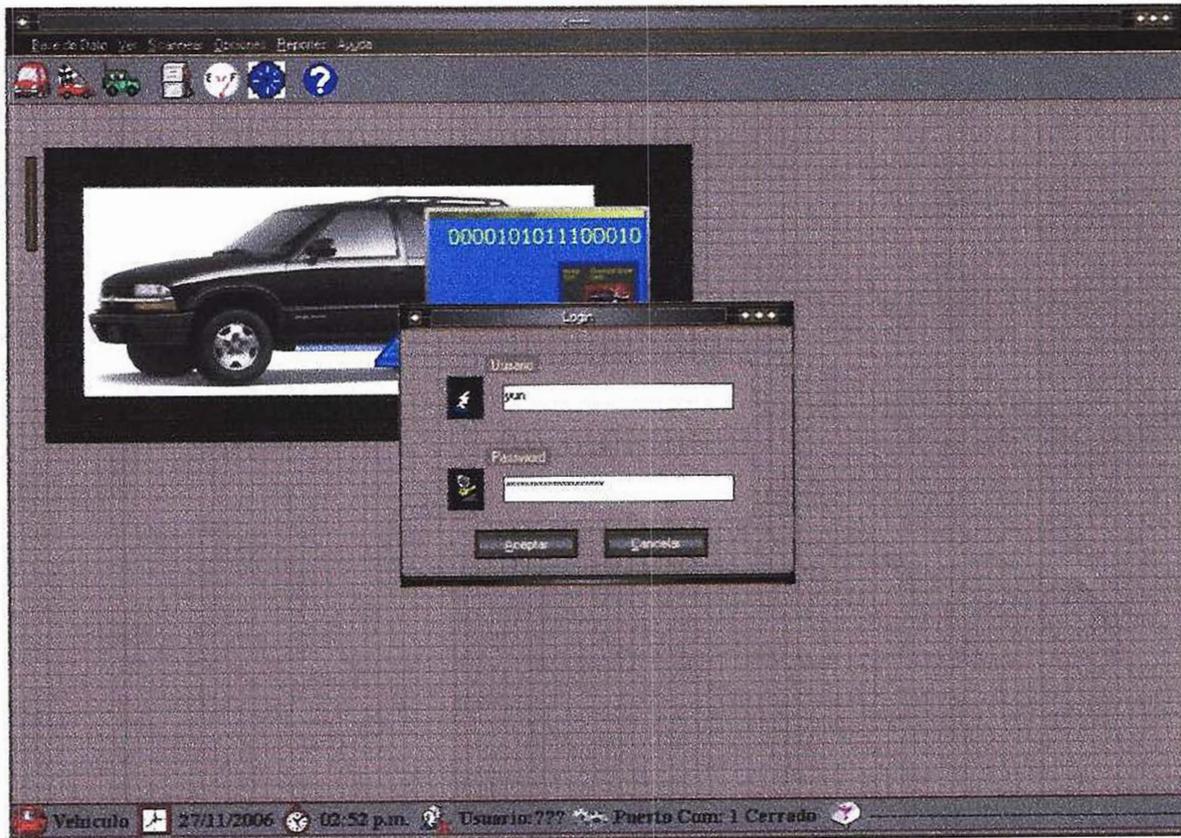


Figura 2. Ventana Principal SCANNER AMT06

Ventana Principal SCANNER AM0+6 Aquí encontrarás siete pestañas o menú los

cuales son, Base de Datos, Scanner, Opciones, Ayudas como también una imagen que fue insertada a gusto del diseñador del formulario e íconos, en la parte inferior estará la hora, fecha y el nombre del usuario que este utilizando la aplicación, lo que es fecha y hora son tomadas del sistema y el usuario de la base de dato.

Base de Datos este es una de las pestañas o una opción del menú que tiene el formulario principal como se puede mostrar en la parte superior: al desplegar la pestaña encontrarás tres opciones las cuales son propietarios, vehículos, y usuarios.

Este es la pestaña de base de datos



Figura 2.1. Pestaña Base De Datos SCANNER AMT06



Figura 2.2. Submenú Propietarios SCANNER AMT06

Al hacer click en propietarios saldrá el siguiente pantallazo

A screenshot of a software window titled 'Ingreso de Propietarios'. The window has a dark title bar and a light-colored background. At the top, there is a horizontal bar with five buttons: 'Guardar', 'Buscar', 'Eliminar', 'Limpiar', and 'Salir'. Below this bar, there is a section labeled 'Propietario' with four input fields: 'Cédula', 'Nombre', 'Dirección', and 'Teléfono'. Each field is a simple rectangular box with a thin border.

Figura 2.3. Ingreso de Propietarios SCANNER AMT06

El cual en la figura 2.3 se registrarán todos los distintos propietarios que llegaran

al taller con el fin de tener un historial del cliente. Donde se digitara la cedula, nombre, dirección y teléfono que serán guardados en la base de datos.

El usuario final le dará su respectivo uso a cada botón. Este formulario de ingreso de propietario se compone cinco botones:

- Guardar aquí se guardan los datos del propietario del vehículo, los cuales esos datos serán almacenados en la base de datos que contiene el programa,
- Buscar en este botón como su nombre lo indica buscara los datos del propietario que están guardados para consultar su historial o modificar cualquier dato.
- Eliminar este elimina cualquier registro o datos del cliente que en este caso es el propietario del vehiculo de la base de datos,
- Limpiar este limpia cada caja de texto dejando el formulario acto para ingresar o hacer cualquier función necesaria que en el momento lo requiera,
- Salir este permite salir del formulario Ingreso de Propietario donde a la vez existe la opción de ir o de abrir otra aplicación o formulario.

Al hacer click en la pestaña u opción Vehículos



Figura 2.4.Submenú Vehículos SCANNER AMT06

Se visualizará el Pantallazo de Ingreso de Vehículos

The screenshot shows a web application window titled "Ingreso de Vehículos". It features two main sections for data entry. The first section, "Datos del Automóvil", contains six input fields: "Placa" (text), "Marca" (dropdown), "Modelo" (text), "Año" (text), "Color" (dropdown), and "Estado" (text). The second section, "Propietario", contains four input fields: "Cedula" (dropdown), "Nombre" (text), "Dirección" (text), and "Teléfono" (text). To the right of these sections is a vertical column of six buttons: "Guardar", "Buscar", "Eliminar", "Limpiar", "Salir", and "Conectar".

Figura 2.5. Ingreso De Vehículos SCANNER AMT06

Aquí en la figura 2.5 se registrarán los datos del Automóvil como la Placa, Marca, Modelo, Año, Color y propietario los cuales serán guardados en la base datos con el fin de llevar un historial de cada Vehículo con sus respectivos datos del dueño, aquí se utilizaron cajas de textos y combo box, el combo box son unas opciones que se despliegan al dar click en la flechita que aparece al lado posterior, donde se debe escoger una de las opciones, y las cajas de textos

simplemente los digita el usuario final. Como en el formulario anterior aquí se utiliza también los cinco botones guardar, buscar, eliminar, limpiar y salir los cuales tienen las mismas funciones que habíamos mencionado anteriormente solo que aquí se guardaran los datos del vehículo.

Al hacer click en usuario la tercera opción



Figura 2.6. Submenú De Usuarios SCANNER AMT06

La opción usuario se encontrará inhabilitado en esta opción ya que al entrar al programa lo hizo como usuario y solo el administrador encontrará esta opción habilitada



Figura 2.7. Submenú De Usuario Deshabilitado SCANNER AMT06

De inmediato saldrá el Formulario de Registro de Usuario solo si entró al programa como administrador, ya que este tiene el privilegio de crear usuarios.



Registrar Usuarios

Datos de Login

Usuario

Password

Tipo

Datos del Cliente

Cédula

Nombre

Dirección

Teléfono

Guardar Buscar Eliminar Limpiar Salir

Figura 2.8. Registro de Usuarios SCANNER AMT06

Este formulario es en donde se registran los usuarios que tendrán acceso al sistema los cuales esos usuarios pueden ser administradores.

- Usuario: El nombre o siglas como deseas identificarte para ingresar al programa principal en este caso el formulario login.
- Password: La clave o contraseña personal que le corresponde a cada usuario para poder acceder al programa principal.
- Tipo: Este tiene dos opciones para ser registrado como administrador o usuario.

Los demás son datos personales como el nombre, cedula, dirección y teléfono ya

sea del administrador o usuario.

Contiene una serie de botones donde se podrá Guardar, Buscar, Eliminar, Limpiar y Salir.

La otra pestaña es Scanner al darle click saldrán tres opciones Leer Censor, Leer Código, Experto.

Al dar click en la primera opción

La otra pestaña de la figura 2. Ventana Principal SCANNER AM0+6 al darle click saldrán tres opciones Leer Censor, Leer Código, experto.

Al dar click en la cuarta opción



Figura 2.9. Submenú DTC SCANNER AMT06

Saldrá el Formulario de Ingreso de códigos DTC

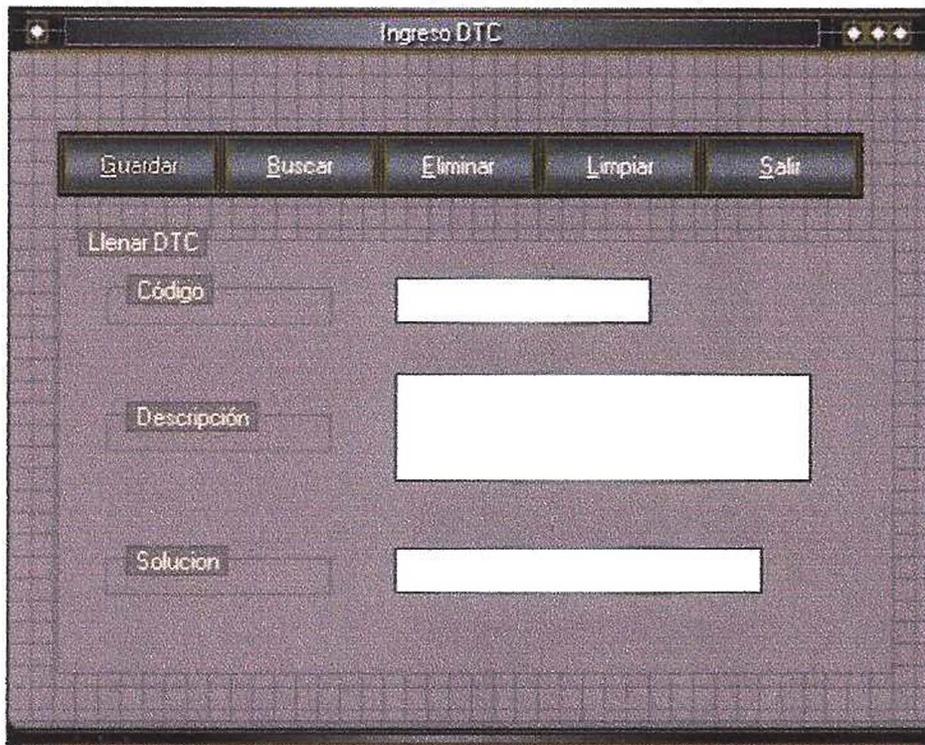


Figura 2.10. Ingreso DTC SCANNER AMT06

Sabiendo él numero del DTC (diagnostico de código de fallas) código que esta almacenado en la memoria de la computadora del vehículo. El cual diagnostica el comportamiento de los sensores de oxigeno, temperatura y choque respectivamente, detecta las fallas y las causas del problema del vehiculo.

En este formulario llenaremos el código que es el mismo DTC, una pequeña descripción y una posible solución que serán guardados en la base de datos para cuando se consulte el historial del vehiculo arroje la ultima falla..

Contiene cinco botones guardar, buscar, eliminar, limpiar, salir de la aplicación los cuales tienen las mismas funciones que habíamos mencionado anteriormente solo que aquí se guardaran los datos del ingreso de los DTC.

Al hacer click en la quinta opción



Figura 2.11. Submenú De Sensores SCANNER AMT06

Saldrá el Formulario de Ingreso de Sensores

A screenshot of a software window titled "Ingreso De Sensores". At the top, there is a toolbar with five buttons: "Guardar", "Buscar", "Eliminar", "Limpiar", and "Salir". Below the toolbar, the main area is titled "Llenar Sensores" and contains eight input fields, each with a label to its left: "Código del Sensor", "Nombre Sensor", "Comando", "Bytes", "Acronimo", "Unidad", "Formula", and "Valor".

Figura 2.12. Ingreso de Sensores SCANNER AMT06



Este formulario tiene como función llenar los sensores como el código que es el mismo sensor, el nombre del sensor, comando que es lo que se envía al puerto serial, bytes el cual se ingresara cuantos bytes de información contiene, acrónimo es la abreviatura del sensor, unidad y fórmula.

Este contiene sus respectivos botones guardar, buscar, eliminar, limpiar, salir el cual se les dará su respectivo uso.

Al dar click en la sexta opción



Figura 2.13. Submenú Importar e Importar SCANNER AMT06

Al hacer click en la pestaña de Exportar sale el siguiente formulario

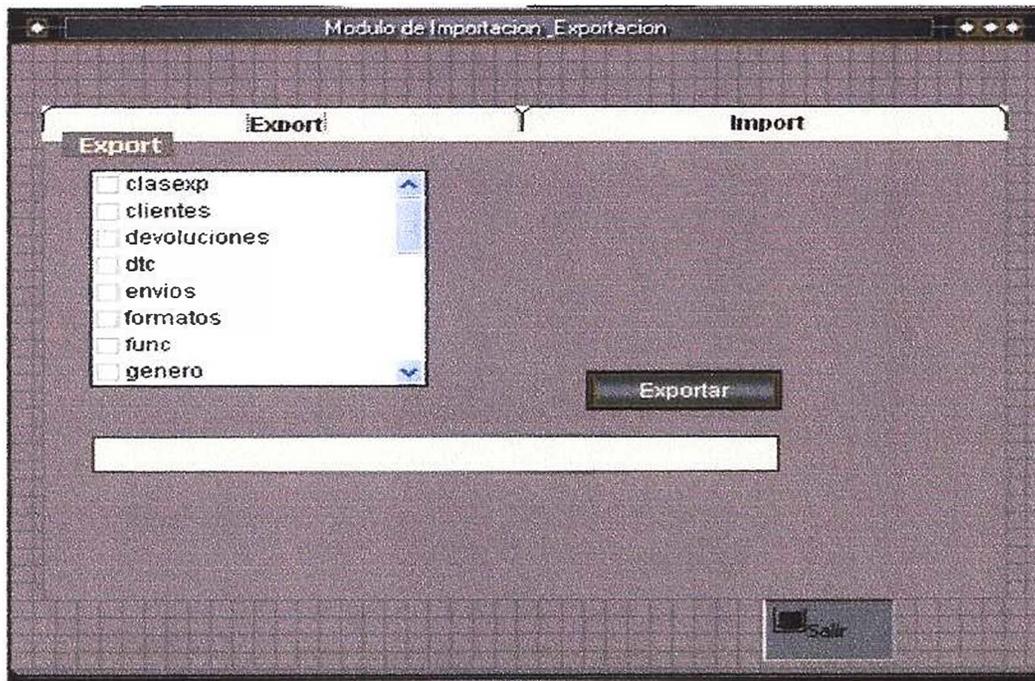


Figura 2.14. Exportar SCANNER AMT06

Este formulario nos permite enviar todos los datos de nuestra base de datos a un archivo de texto o considerado archivo plano, con el fin de hacer copias de la información de la base de datos en caso de que ella sufriera algún tipo de desperfecto

Al hacer click en la pestaña de Importar sale el siguiente formulario

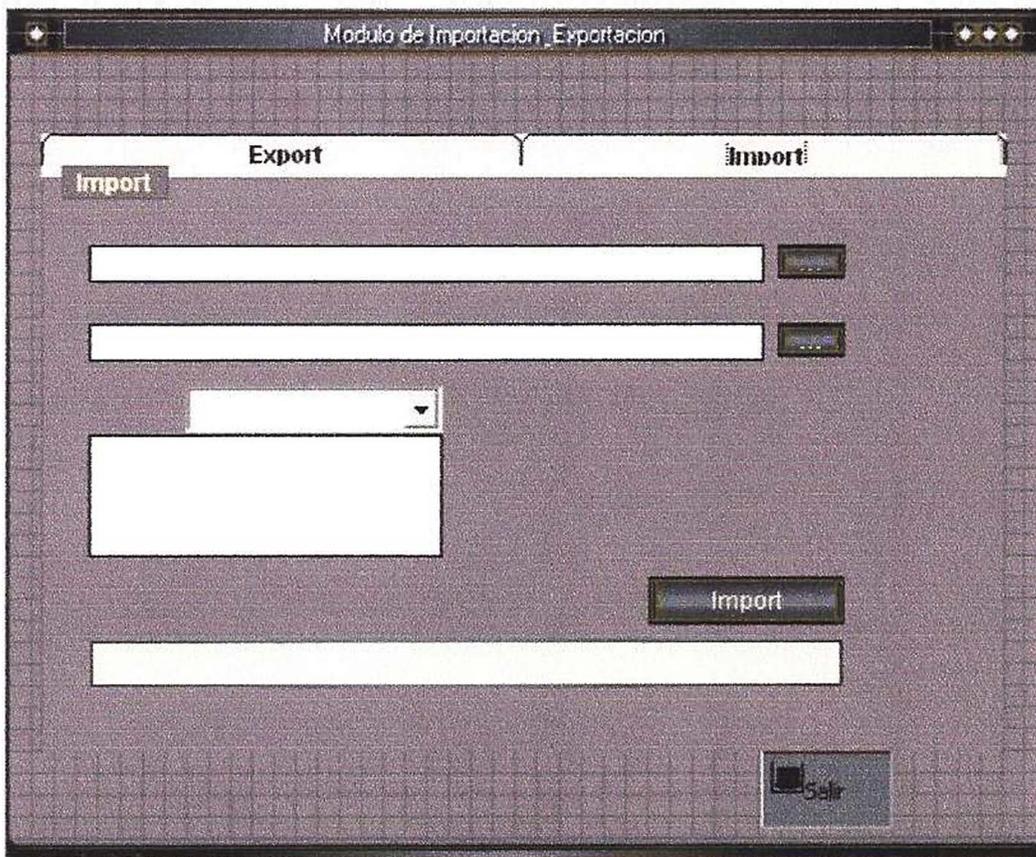


Figura 2.15. Importar SCANNER AMT06

Este formulario nos permite enviar todos los datos de nuestra base de datos a un archivo de texto o considerado archivo plano a nuestra base de datos

La siguiente pestaña del menú es Ver



Figura 3.0. Menú Barra De Estado SCANNER AMT06

Al dar click en barra de estado

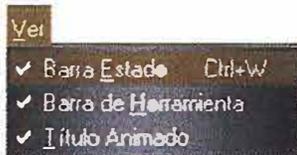


Figura 3.1. Submenú Barra De Estado SCANNER AMT06



Con Barra de estado

Figura 3.2. Barra De Estado Habilitado SCANNER AMT06

Este siempre toma la fecha del sistema para mostrarla en la barra de Estado.

La hora del sistema



Figura 3.5. Hora Del sistema SCANNER AMT06

Nombre del usuario



Figura 3.6. Nombre de Usuario SCANNER AMT06

Indica si el puerto Está Abierto o cerrado



Figura 3.7. Indicador De Puerto SCANNER AMT06

Al hacer click en la siguiente opción Barra de herramientas



Figura 3.8. Barra De Herramienta SCANNER AMT06

Saldrá el formulario con la barra deshabilitada de herramienta



Figura 3.9. Barra De Herramienta Deshabilitada SCANNER AMT06

Aquí deshabilitamos la barra de herramientas como se puede mostrar en la figura 3.9, el cual la barra de herramientas son iconos directos que al darle click te llevaran ala aplicación correspondiente de una manera más eficiente y rápida.

Barra de herramienta

 Ingreso de vehiculo



Figura 3.10. Icono Ingreso de Vehículos SCANNER AMT06



Ingreso de propietario



Figura 3.11. Icono Ingreso de Propietarios SCANNER AMT06

Ingreso de sensor



Figura 3.12. Icono Ingreso de Sensores SCANNER AMT06

Lectura de sensor



Figura 3.13. Icono Ingreso Lectura de Sensores SCANNER AMT06

Leer código



Figura 3.14. Icono Ingreso de Lectura De Códigos SCANNER AMT06

Experto



Figura 3.15. Icono Ingreso a Experto SCANNER AMT06



Figura 3.16. Icono Ingreso a La Ayuda SCANNER AMT06

Al hacer click en la siguiente opción de titulo animado sé deshabilitara



Figura 3.17. Submenú Título Animado SCANNER AMT06



Figura 3.18. Título Animado SCANNER AMT06



Figura 3.19. Titulo Deshabilitado SCANNER AMT06

En la figura 3.19 se muestra deshabilitado el título animado es una de las opciones de menú ver (figura 3.17) que al hacer click esta barra automáticamente quedara deshabilitada.

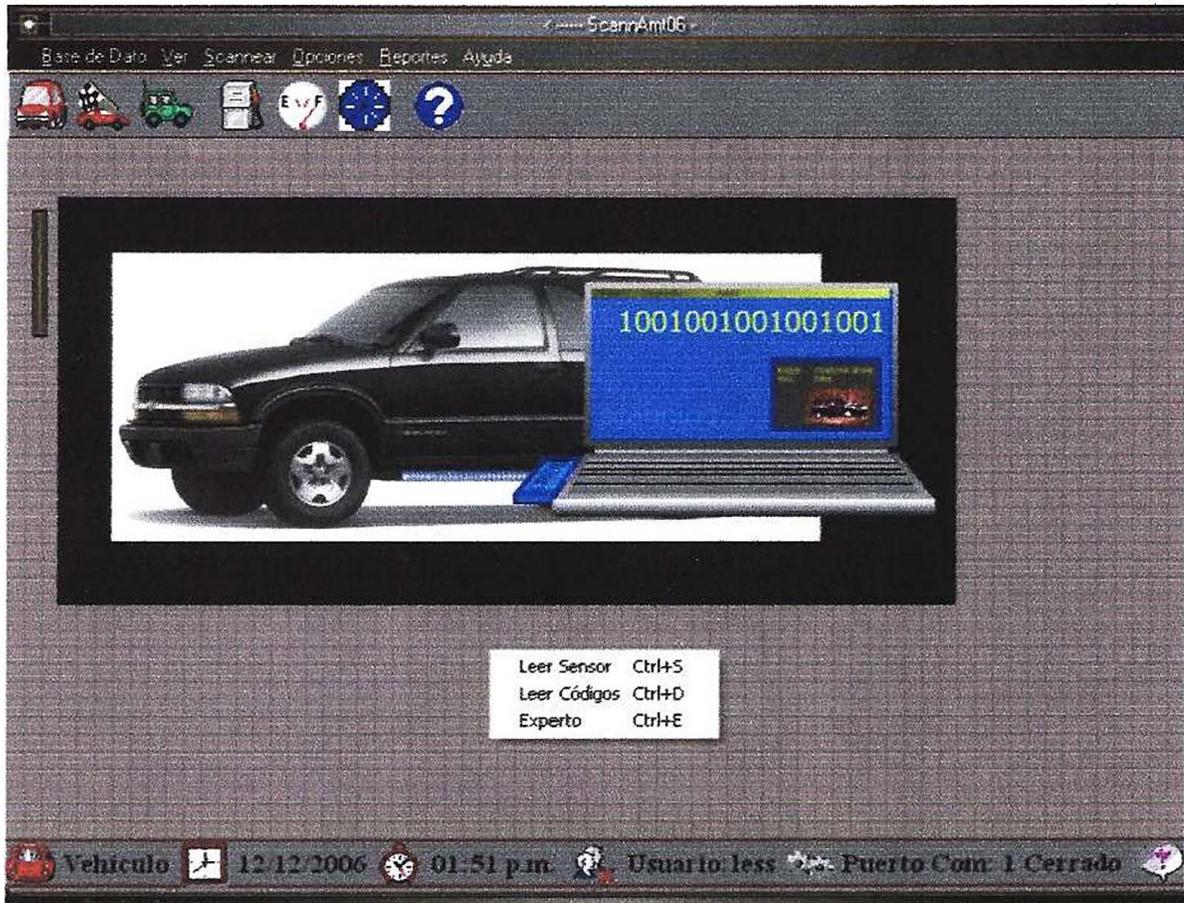


Figura 4. Click Derecho SCANNER AMT06

En este formulario le podemos dar click derecho en cualquier parte del formulario, para habilitar el menú contextual y escoger la opción leer sensor leer el sensor.

Tercera pestaña del menú Scanear



Figura 4.1. Pestaña Scanear SCANNER AMT06

Al hacer click en el menú Scannear saldrá un submenú

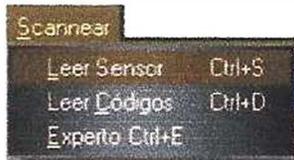


Figura 4.2. Submenú Leer Sensor SCANNER AMT06



Figura 4.3. Error SCANNER AMT06

Antes de leer Los sensores de un automóvil Deben estar conectados y si no lo están le dan click en el botón aceptar y les traerá el formulario de ingreso de vehículos, observe la figura 2.5

Una vez que los vehículos estén conectados se podrá acceder a la lectura de los sensores con el siguiente formulario

Formulario de leer sensor

The screenshot shows the 'Scanner AMT06' application window. At the top, there are three buttons: '<<' (left arrow), 'Posición abs. del estrangulador', 'Velocidad del motor RPM', 'Velocidad del vehículo', and '>>' (right arrow). Below this is a section titled 'Ningún Sensor Monitoreado'. To the right of this section is a checkbox labeled 'Simular'. The main configuration area contains several input fields: 'Acronimo' (Acronym) with 'Acronimo', 'Bytes De Información' (Information Bytes) with '1', 'Unidad' (Unit) with 'Unidad', 'Valor del Sensor' (Sensor Value) with '0', 'Comando' (Command) with 'ATI', 'Valor Decimal' (Decimal Value) with '0', 'Valor Mínimo Leído' (Minimum Value Read) with 'Min', 'Valor Máximo Leído' (Maximum Value Read) with 'Max', 'Valor promedio Leído' (Average Value Read) with 'Prom', and 'Formata del Sensor' (Sensor Format) with 'Formula'. To the right of these fields is a large empty rectangular area. On the far right, there is a vertical stack of buttons: 'Comenzar' (Start), 'Gráfico' (Graph), 'Limpiar' (Clear), 'Puerto Com' (COM Port), 'Cerrar' (Close), and 'Reporte' (Report). At the bottom, there is a status bar with four indicators: 'Puerto Abierto' (Port Open) with a red bar, 'Enviando' (Sending) with a green bar, 'Recibiendo' (Receiving) with a green bar, and 'Conectado' (Connected) with a green bar.

Figura 4.4. Lectura De Los Sensores SCANNER AMT06

En la figura 4.4 se muestra el formulario de leer los sensores del vehículo, para comenzar a escanear presione el botón Comenzar y después pase el Mouse sobre el sensor que desea leer, puede avanzar con el botón ">>" y retroceder con "<<". A medida que se escoge un sensor específico en las cajas de textos aparecerá información relevante del sensor.

Las cajas de Texto.

- Acrónimo: Aparecerán las siglas del sensor.

- Byte: Cuantos bytes necesarios de información.
- Unidad: En que unidades son los valores leídos del sensor.
- Valor del Sensor: Valor leído en tiempo real del sensor.
- Comando: Comando del sensor que se envía al vehículo.
- Valor decimal: Valor leído en decimal.
- Valor Mínimo: El valor mínimo leído del sensor durante el scaneo
- Valor Máximo: El valor máximo leído del sensor durante el scaneo
- Valor Promedio: El valor promedio leído del sensor durante el scaneo
- Caja de texto mayor: Aparecerá el valor leído en hexadecimal.

Los botones

- Comenzar/Detener: Comenzar o detener el escaneo del vehículo.
- Gráfico: Si existe un gráfico asociado al sensor que se está leyendo se desplegará un formulario con el gráfico.
- Limpiar: Limpia todas las cajas de textos del formulario.
- Puerto Com: Si desea configurar el puerto serial o escoger otro número de puerto.
- Cerrar: Cerrar este formulario.
- Reporte: Genera el reporte de los sensores leídos.



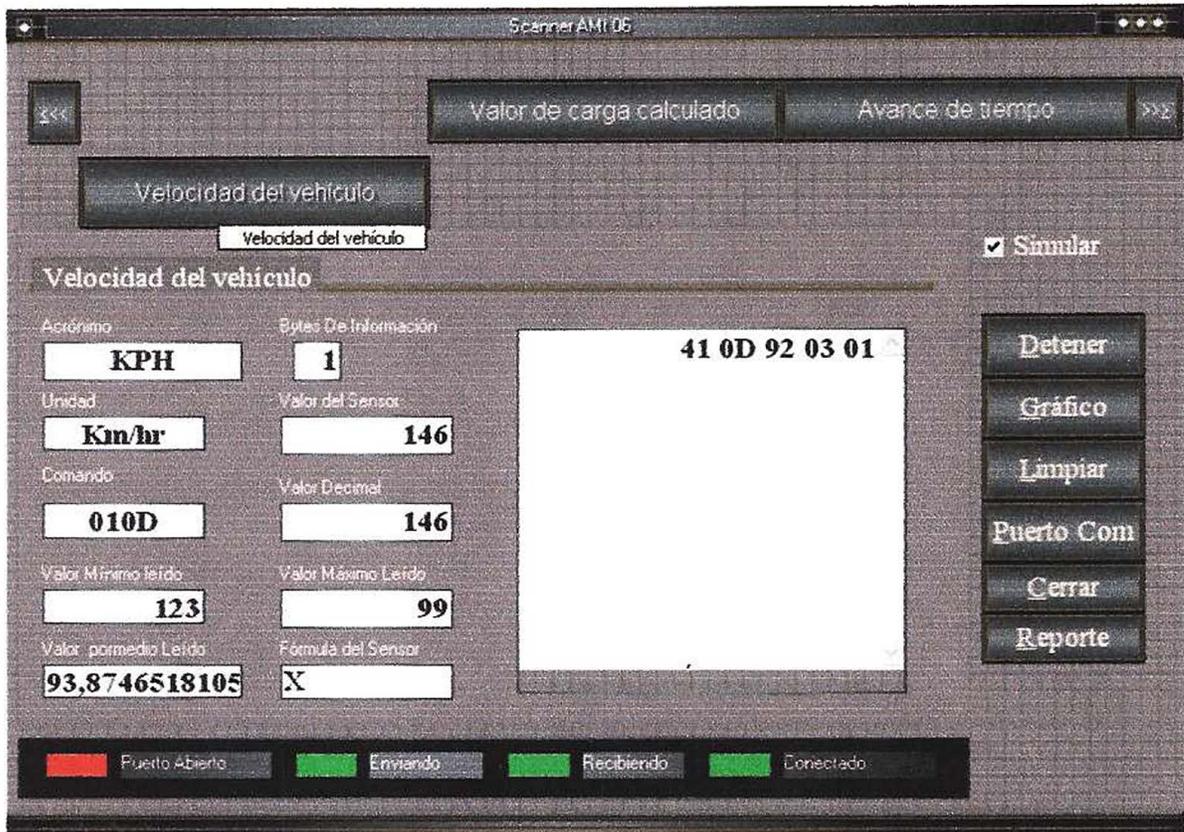


Figura 4.5. Simulación SCANNER AMT06

En esta figura si leemos la velocidad del motor RPM saldrá el siguiente formulario dando las revoluciones por minuto que arroja el automóvil. contiene dos botones el cual es detener y cerrar

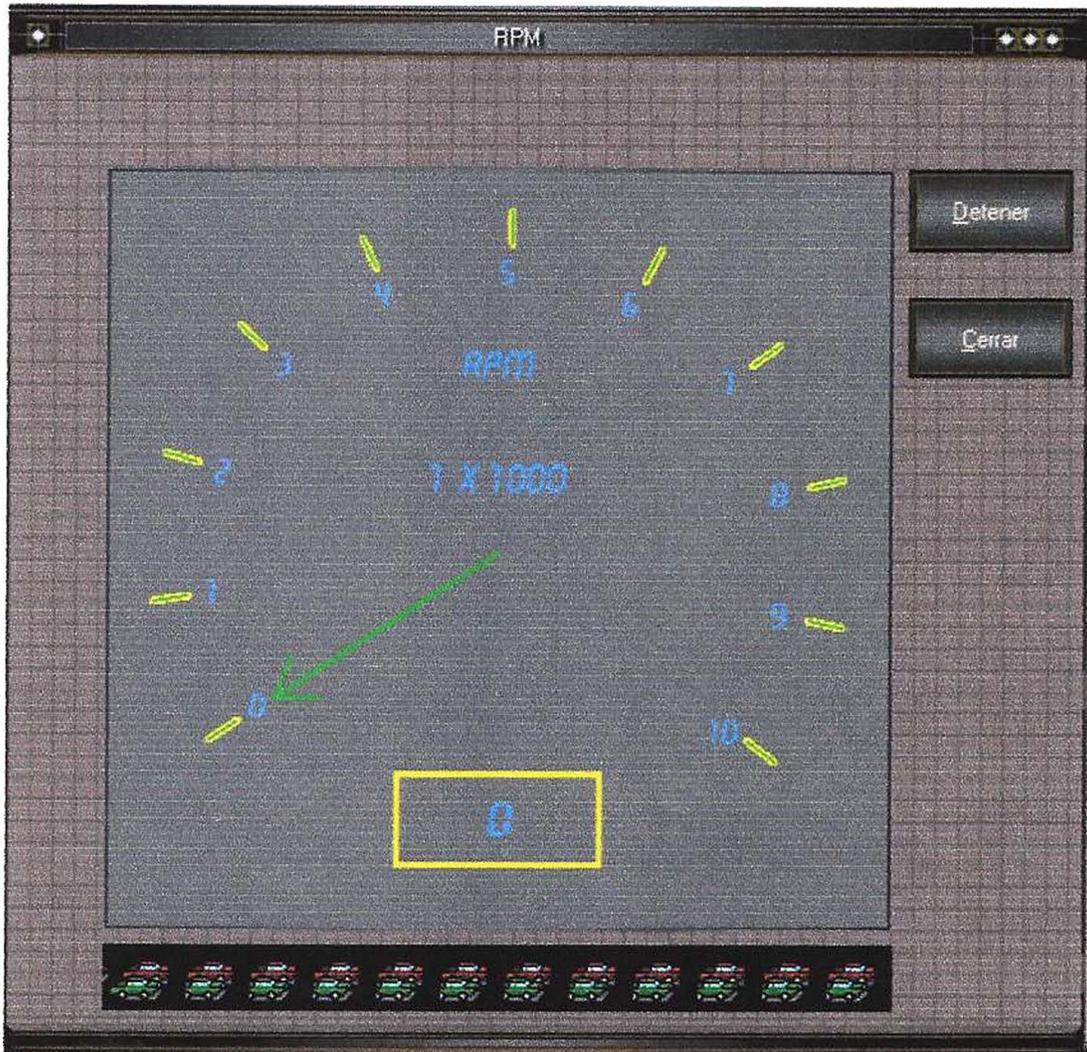


Figura 4.6. RPM SCANNER AMT06

Al hacer click en el botón grafico el cual esta en la figura 4.5 Simulación, saldrá el siguiente pantallazo

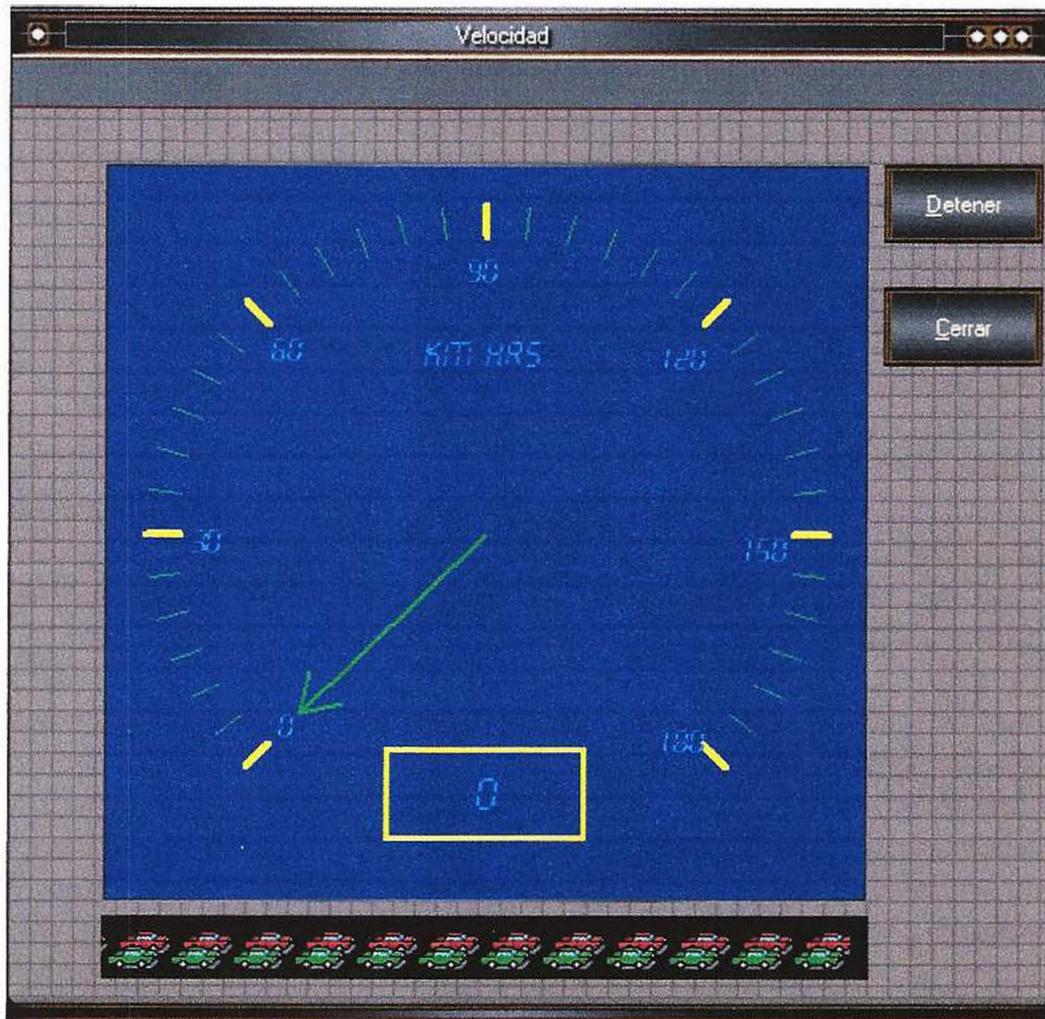


Figura 4.7. Velocidad SCANNER AMT06

Este formulario tiene como función leer las diferentes velocidades que el vehículo arroja realmente.

Al dar click en el botón que se encuentra en la figura 4.5 de reporte saldrá el siguiente formulario con el reporte específico del scanneo.

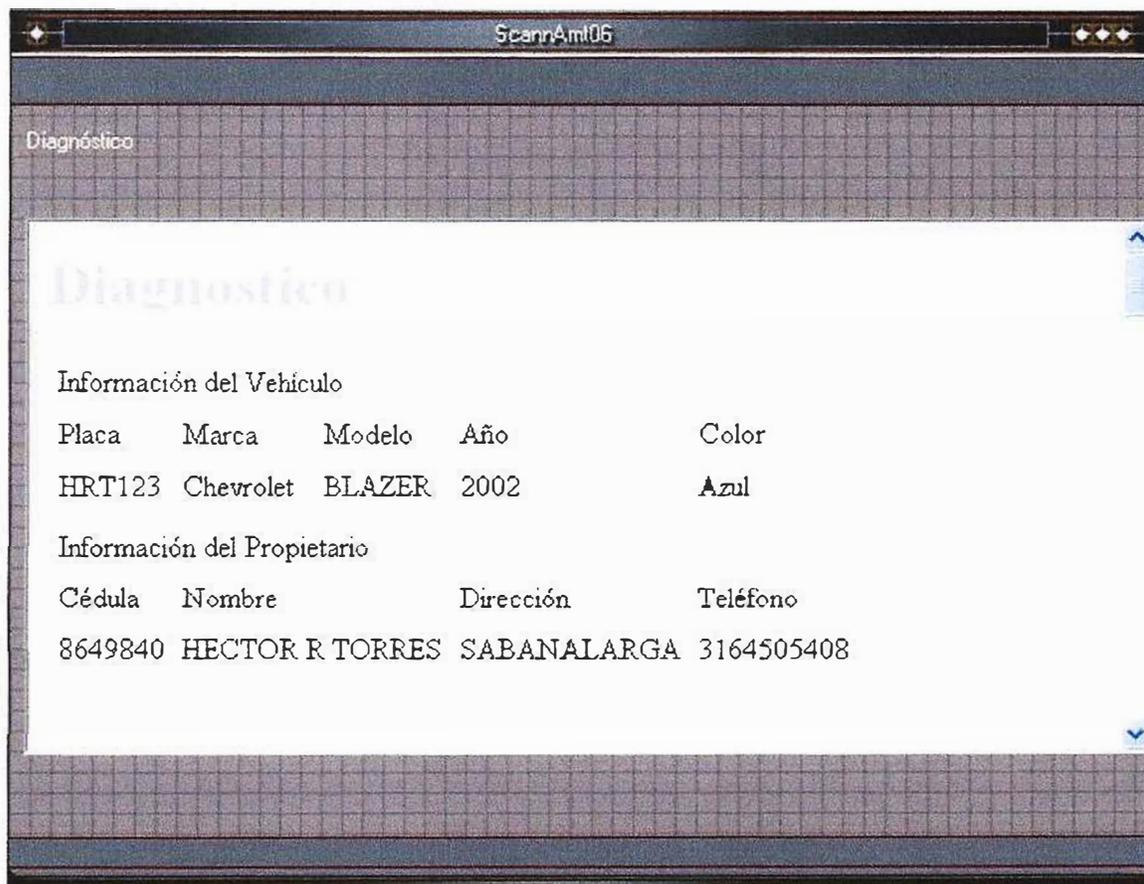


Figura 4.8. Diagnostico SCANNER AMT06



Figura 4.9. Submenú Leer Códigos SCANNER AMT06

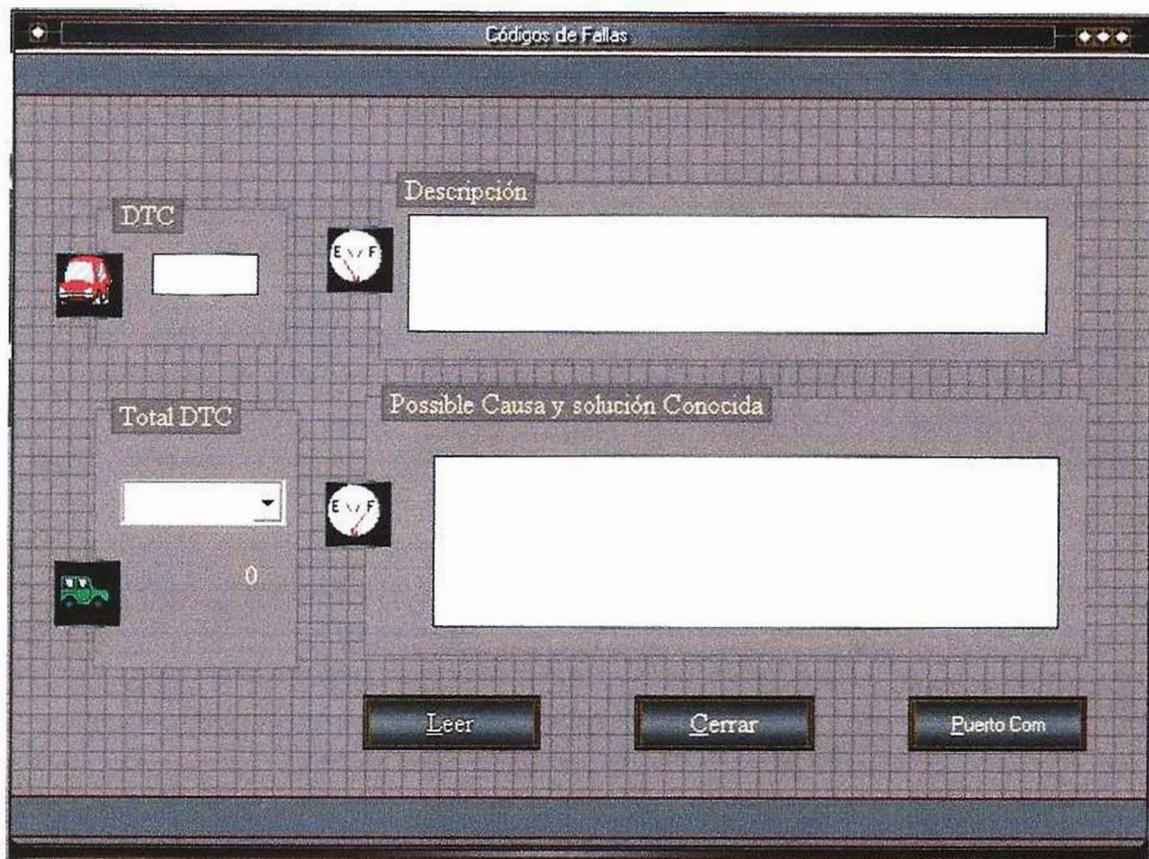


Figura 4.10. Códigos De Fallas SCANNER AMT06

Los botones

- Leer: Abre la conexión con el vehículo y comienza a leer los códigos de fallas.
- Cerrar: Cierra esta Ventana.
- Puerto Comm: Permite configurar el puerto serial y escoger un número de puerto válido.

Cajas de textos.

- DTC: El DTC actual leído.
- Total DTC: En número total de DTC presentes en el vehiculos.



- Descripción: Descripción del código de falla.
- Solución: Solución sugerida para el DTC actual.



Figura 4.11. Submenú Experto SCANNER AMT06

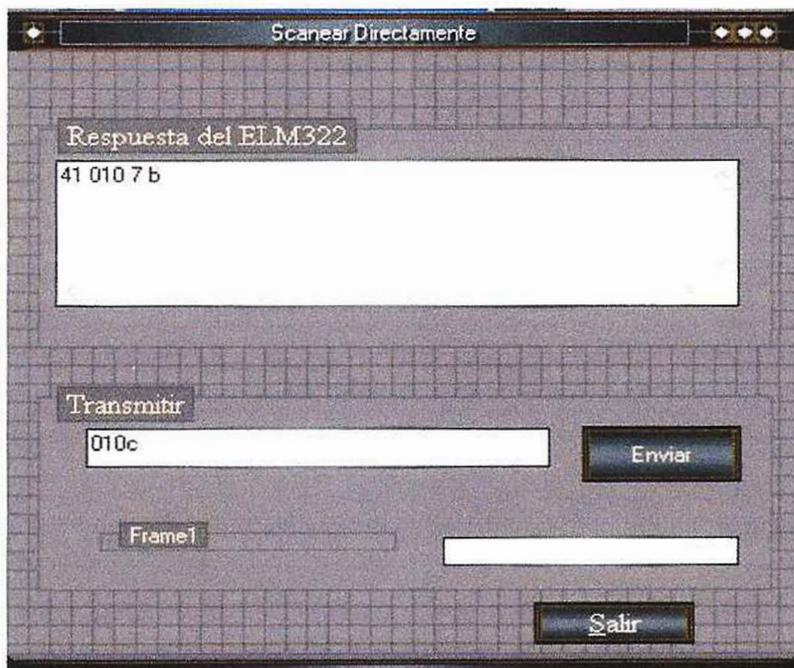


Figura 4.12. Experto SCANNER AMT06

Este formulario es para los más avanzados y quieren monitorear de cerca el estado del vehículo.

En la caja de Texto “Transmitir” se escribe el comando para enviar al vehículo, para luego presionar el botón enviar y se espera en la respuesta en la caja de texto mayor.

Al hacer click en el menú de opciones inmediatamente se desplegara un submenú con tres opciones.



Figura 5. Pestaña Opciones SCANNER AMT06



Figura 5.1. Submenú Apariencia SCANNER AMT06

Este submenú de apariencia tiene como función cambiar la apariencia de cada uno de los formularios el cual es llamado skin son ocho opciones de apariencia que el usuario final puede utilizar, en este caso utilizamos la opción que es chuleada en la figura 5.1 Galaxy.



Figura 5.2. Submenú Configurar Puerto SCANNER AMT06



Figura 5.3. Configurar Puerto SCANNER AMT06



Figura 5.4. Submenú Habilitar Sonido SCANNER AMT06



Figura 6. Pestaña Reportes SCANNER AMT06

Al hacer click en el submenú de reportes dinámicos que se encuentra en la figura anterior 6. Saldrá el siguiente pantallazo.

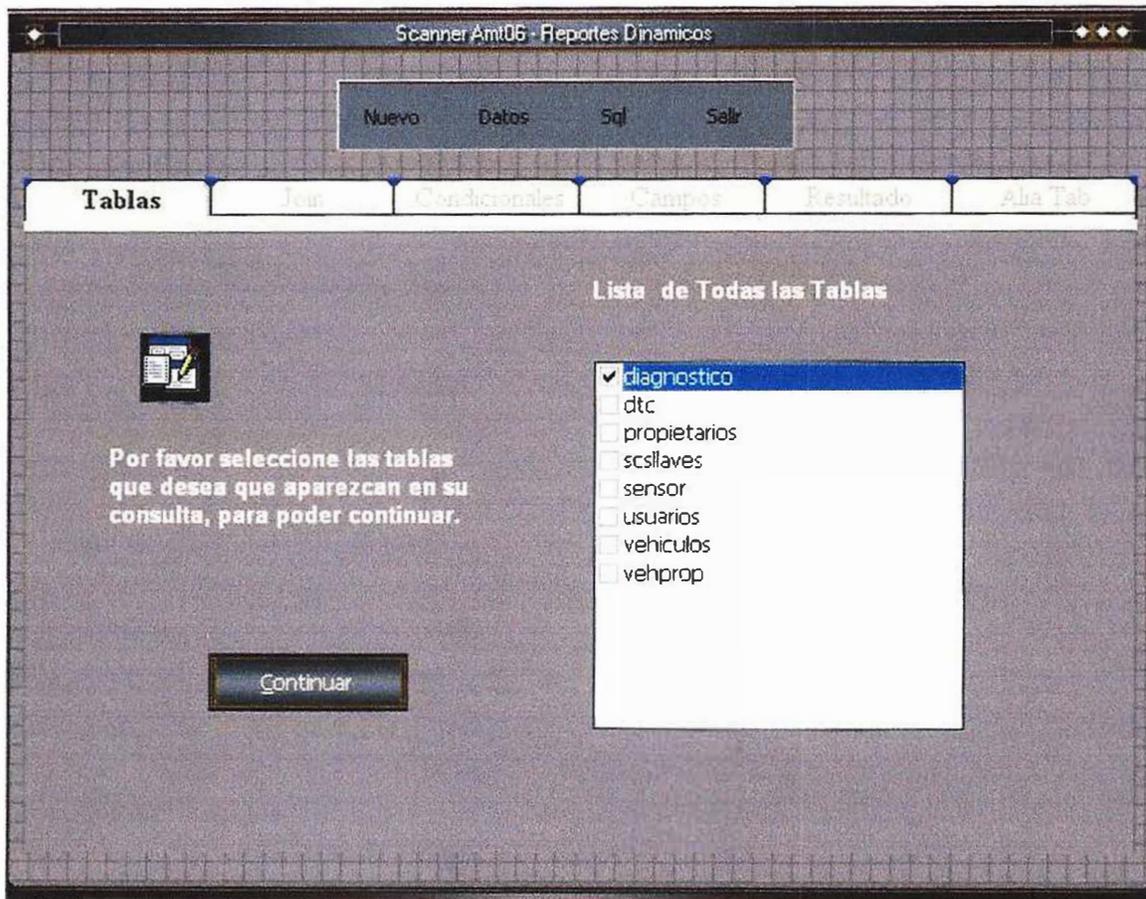


Figura 6.1. Tablas SCANNER AMT06

En este formulario en la pestaña tablas aparecerán una serie de campos de chequeo donde el usuario podrá escoger una o varias opciones, al finalizar dar click en el botón continuar.

Después aparecerá la pestaña Alia Tab

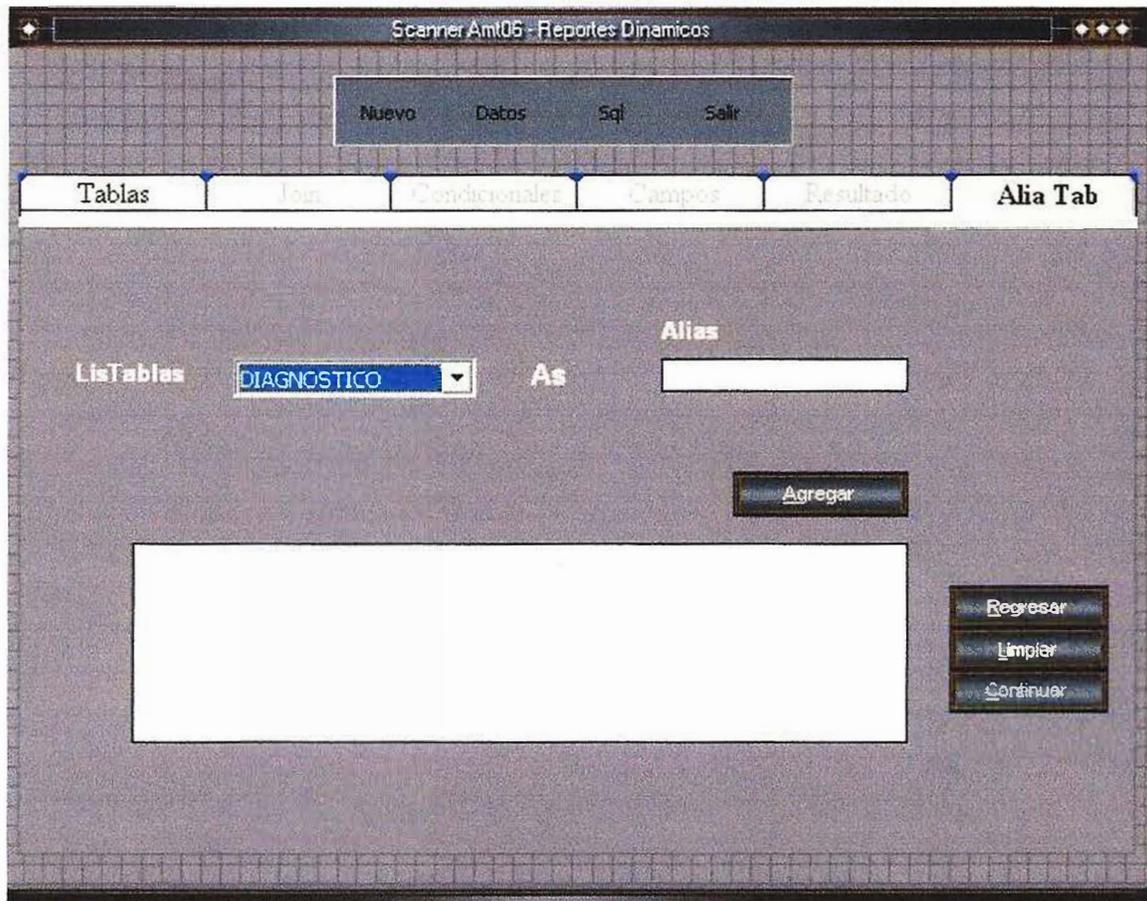


Figura 6.2. Alia Tab SCANNER AMT06

En el cual aparecerá en un combo las tablas seleccionadas anteriormente el cual a cada tabla deberá colocar un alias que se identifique con la tabla, después se le da click al botón agregar y luego continuar.



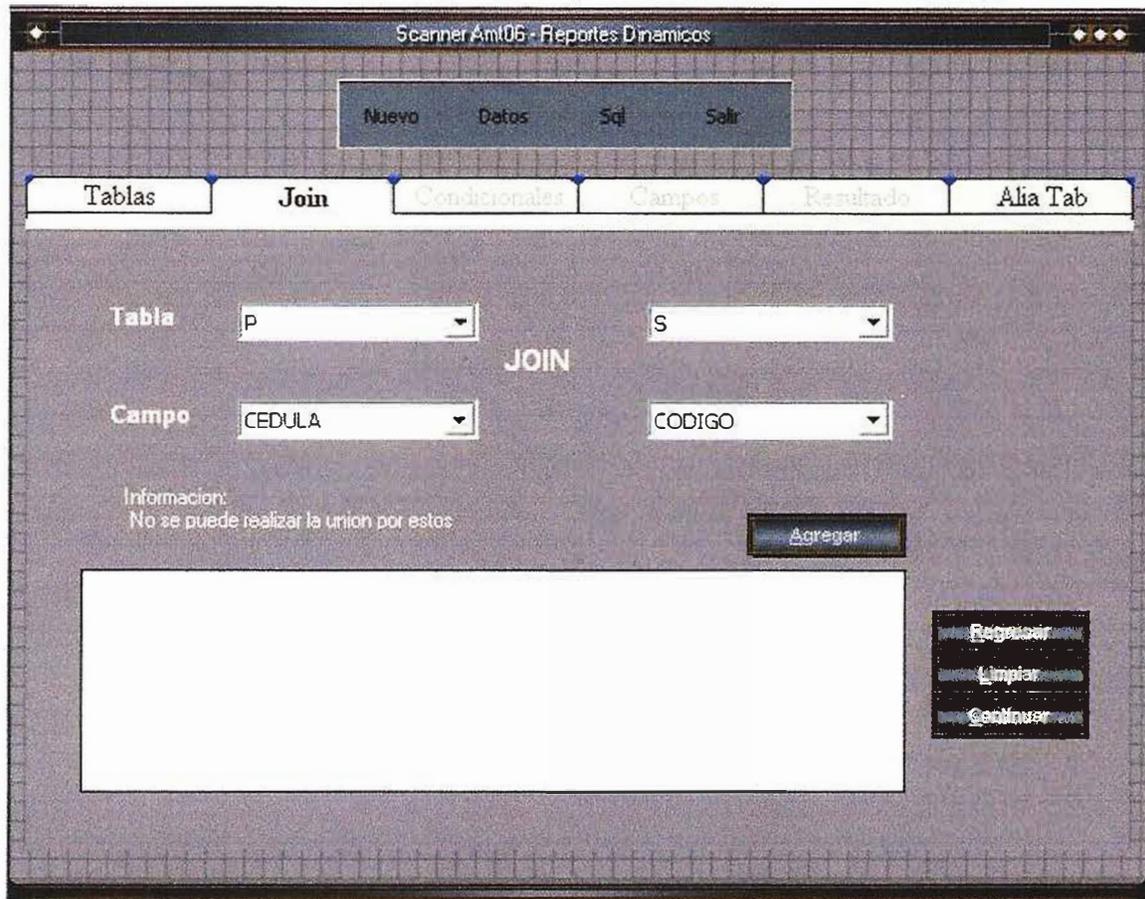


Figura 6.3. Join SCANNER AMT06

Después si el usuario a seleccionado varias tablas saldrá seleccionado la pestaña Join en el cual escogerá el campo que le corresponda a cada tabla y agregamos y luego continuar.

Scanner Amt06 - Reportes Dinamicos

Nuevo Datos Sql Salir

Tablas Filtros **Condicionales** Campos Resultados Alia Tab

Tabla D Operador Logico

Campo Operador Val1 Val2

PLACAVEH = hrt123 And

Informacion:
Campo: Valor

Agregar

Regresar
Limpiar
Continuar

Figura 6.4. Condicionales SCANNER AMT06

En este formulario en la pestaña condicionales relacionamos el campo de la tabla con el valor respectivo le damos click en el botón agregar y luego en el botón continuar.

Scanner Amt06 - Reportes Dinamicos

Nuevo Datos Sql Salir

Tablas Join Condicionales **Campos** Resultado Alia Tab

Tabla D

Funcion NINGUNO Campo PLACAVEH Alias As pv

Informacion:
Campo: Valor Alfanumerico - Longitud

Agregar

D.PLACAVEH AS pv

Agrupado Por

Regresar
Limpiar
Continuar

Figura 6.5. Campos SCANNER AMT06

En este formulario en la pestaña CAMPOS seleccionamos la tabla, el campo de la tabla y su respectivo Alias le damos click en el botón agregar y luego en continuar.

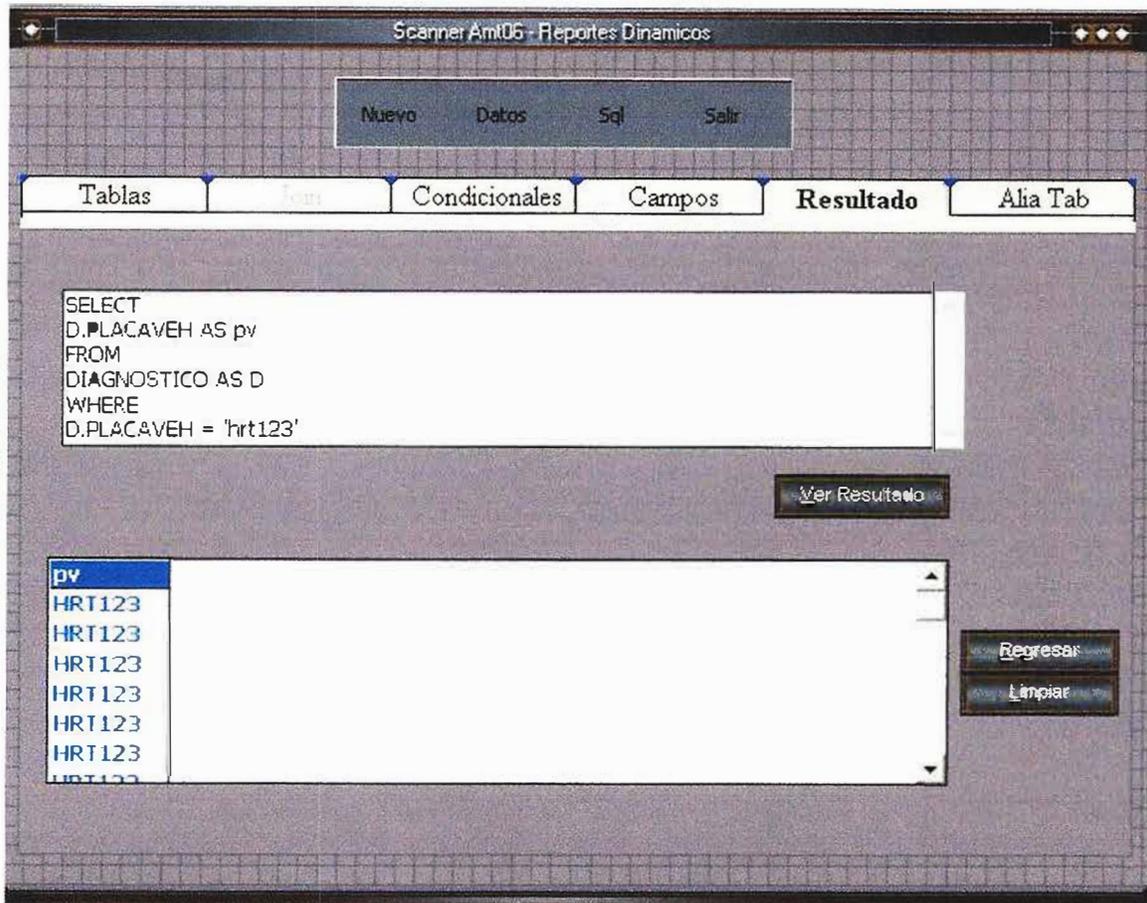


Figura 6.6. Resultados SCANNER AMT06

En este formulario aparecerá la consulta en una caja de texto de todos los procesos que fueron realizados anteriormente, le damos click en el botón resultado y en la segunda caja de texto aparecerá una serie de columnas con distintos resultados de acuerdo con la consulta elaborada anteriormente.

Y finalmente estos son los pasos para obtener un reporte dinámico.



Figura 7. Pestaña Ayuda SCANNER AMT06

Esta figura es un menú de ayuda el cual despliega un submenú con seis opciones, este menú de ayuda es muy importante para este programa es un sistema de guía.



Figura 7.1. Submenú Contenido SCANNER AMT06

Este submenú de contenido



Figura 7.2. Submenú Índice SCANNER AMT06



Figura 7.3. Submenú Búsqueda SCANNER AMT06



Figura 7.4. Submenú Cerrar Ayuda SCANNER AMT06



Figura 7.5. Submenú Acerca SCANNER AMT06





Figura 7.6. Submenú Tip SCANNER AMT06

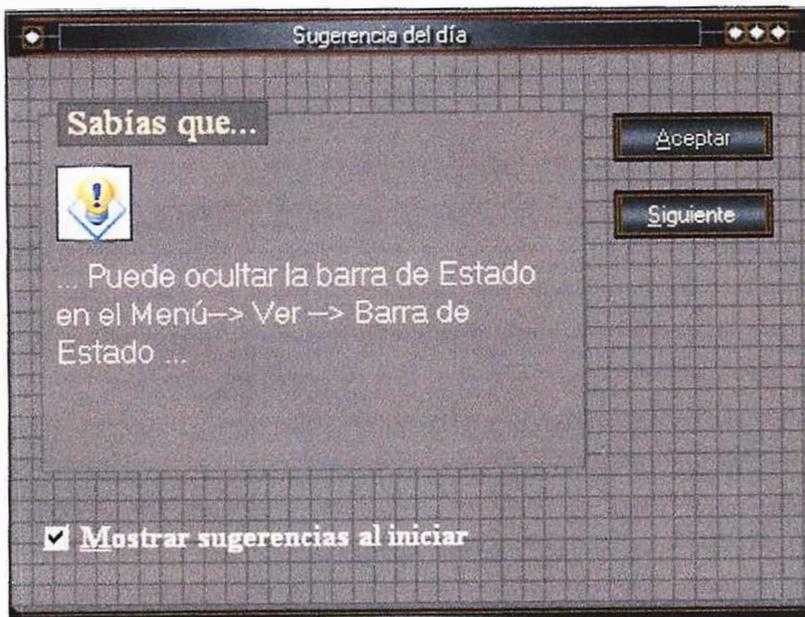


Figura 7.7. Submenú Sugerencia SCANNER AMT06