79
29.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DESARROLLO DE UN SISTEMA EXPERTO PARA DIAGNÓSTICO AUTOMOTRIZ

PRIMER NIVEL DE DIAGNÓSTICO

T E S I S

Que para obtener el Título de
INGENIERO EN COMPUTACIÓN

p r e s e n t a

SARA NOGUERÓN GALICIA

DIRECTOR DE TESIS:
M.I. EUGENIO MARIO LÓPEZ ORTEGA

México D.F. 1998

Este trabajo representa la culminación de una de las etapas más importantes y gratas en mi vida. En él se encuentra lo que he adquirido durante todo este tiempo, desde conocimientos hasta entereza. Para poder alcanzar esta meta han estado conmigo muchas personas, que de alguna manera me han impulsado a seguir adelante.

Doy mis más sinceros agradecimientos a: mis Padres, Elvia Galicia y Elías Noguerón, por su apoyo incondicional, por sus palabras de aliento en los momentos de desesperación, porque con su ejemplo me han enseñado a luchar hasta alcanzar mis sueños y porque me han dado una vida llena de felicidad que definitivamente ha contribuido a lograr lo que hoy tengo. A mis hermanas, Lydia, Areli, Jael y Dámaris, por todos los momentos que hemos pasado juntas, por apoyarme y por ser unas verdaderas amigas con las que sé que puedo contar en cualquier momento. A Hugo por su gran cariño, por su amistad, por sus palabras de aliento y por todo su apoyo para poder terminar lo que llama "mi obra maestra". A mi director de tesis, Ingeniero Eugenio López Ortega, por su asesoría, dedicación y apoyo para poder realizar este trabajo.

Agradezco a mis profesores, porque de cada uno de ellos he aprendido, no sólo conocimientos de la materia, sino también enseñanzas que me servirán para toda la vida. Agradezco a la Universidad Nacional Autónoma de México por darme la formación que me permitirá enfrentar lo que será mi carrera profesional. Por supuesto agradezco a Dios por permitirme estar aquí y por acompañarme siempre.

Con cariño y respeto a todos ustedes,
Gracias

Sara Noguerón Galicia.

ÍNDICE

Capítulo 1: Introducción

1.1 Un Sistema Experto para Diagnóstico Automotriz.

Cuando el dominio del conocimiento está concentrado en pocas personas, se necesita recurrir a ellas para solucionar problemas complejos. Pero hay ocasiones en que este conocimiento no está al alcance de todos, por lo que se tiene que obtener por otras medios, como consultar libros, manuales, o bien, información de Internet. Sin embargo muchas veces, el tener la información no es suficiente, hay que saber aplicarla.

Conocer la teoría no garantiza que se puedan resolver fácilmente los problemas complejos en la práctica. Es necesario familiarizarse con el funcionamiento real de los conceptos teóricos en combinación con los demás. Éste es el caso de la tecnología *fuel injection* en automóviles.

A principios de los años ochenta entró la tecnología *fuel Injection* en México. Esta tecnología se emplea para alcanzar altos niveles de rendimiento en automóviles de uso diario, mediante la inyección de combustible controlada electrónicamente. Por medio de esto, se logra introducir combustible y aire a las cámaras de combustión en la proporción más adecuada.

La tecnología *fuel injection* es relativamente nueva, lo que ocasiona una falta de capacidad técnica para diagnosticar y corregir fallas. Gran parte de los mecánicos que reparan autos con esta tecnología, lo hacen a partir de prueba y error, lo que los lleva a elevar el tiempo de entrega y los costos, porque en la mayoría de los casos cambian partes que están en perfecto estado.

Actualmente se pueden encontrar libros que explican esta tecnología, o bien, manuales, pero no es suficiente para aplicarla correctamente y poder hacer un diagnóstico. Es entonces cuando se requiere de una persona que esté completamente familiarizada con la teoría llevada a la práctica; es decir, se requiere de un *experto*. Desafortunadamente existen pocos expertos capaces de realizar un diagnóstico de un automóvil *fuel injection*; no es fácil contactar a uno, esto es porque el conocimiento está centralizado. Por esta misma razón sus servicios son muy costosos.

Dado que los automóviles se han convertido en elementos primordiales en la vida cotidiana del ser humano, es necesario que el conocimiento para su reparación esté disponible de forma amplia y dispersa. Es aquí donde juegan un importante papel los Sistemas Expertos, que son programas de computadora que simulan el conocimiento de un experto humano para resolver problemas específicos. Así es posible tener el conocimiento disponible sin necesidad de contar personalmente con el experto.

A pesar de que para su desarrollo y adquisición se requiera de fuertes inversiones, tanto de tiempo como de dinero, pueden ser distribuidos en muchos lugares cubriendo la necesidad de gente altamente capacitada para realizar el trabajo de diagnóstico.

Por otro lado una de las principales características de los automóviles con tecnología avanzada es que cuentan con una computadora, que controla principalmente sensores y actuadores; cuando estos presentan fallas, la computadora genera señales. En algunos casos se pueden ver directamente en el tablero del vehículo, en otros es necesario emplear un instrumento llamado scanner que se conecta a la computadora para poder detectarlas. El scanner lee la señal y la traduce a un código de falla que puede interpretar el mecánico con base en una guía.

Se puede observar que existen ya instrumentos que se encargan de detectar fallas, pero únicamente detectan fallas en los sensores y actuadores. Sin embargo un verdadero diagnóstico va mucho más allá de esto. Además existen condiciones de falla de algunos sensores que no pueden ser detectados oportunamente por la computadora. Por estas razones el contar con un scanner no es suficiente para hacer un diagnóstico, más bien la información que éste aporta es solamente una ayuda que permite al experto encontrar la falla.

Considerando estas limitaciones, es más concebible el desarrollo de un Sistema Experto. El Sistema Experto contendrá el conocimiento necesario y los mecanismos para aplicarlo con el fin de simular la labor del experto ante determinadas situaciones. Como es de imaginarse, esta información será extraida de un experto humano.

El Sistema Experto para Diagnóstico Automotriz, al que se ha denominado SEDA, no pretende suplir a los expertos. Por el contrario, necesita de ellos para poder simular su trabajo; sin embargo un mecánico no experto podrá hacer

reparaciones en a... tecnología *fuel injection* con ayuda del Sistema Experto, reducien... y costos.

La idea de c... un Sistema Experto que realice esta tarea surge ante el interés de un ing... erto en transmitir su conocimiento a nivel masivo empleando nuevos ... los tradicionales requieren de mucho tiempo y dedicación de su ... DA, el experto se ahorrará un sin número de consultas que inte... ...eres diarias.

El Sistema ... ser distribuido en talleres mecánicos y además de servir para diag... empleado con fines de capacitación, porque los mecánicos que lo utilicen podrán comprender mejor el funcionamiento de la tecnología *fuel inje*... hacer uso de su capacidad de aprendizaje para que con el tiempo se convi... ertos sin estar bajo la tutela directa de un humano.

El Sistema Experto indicará al mecánico dónde está la falla, pero finalmente será él quien hará la reparación y se familiarizará más con la tecnología hasta ll... tener un dominio sobre ella. De esta manera se tendrá el conocimiento más ... y disponible en cualquier momento (figura 1.1).

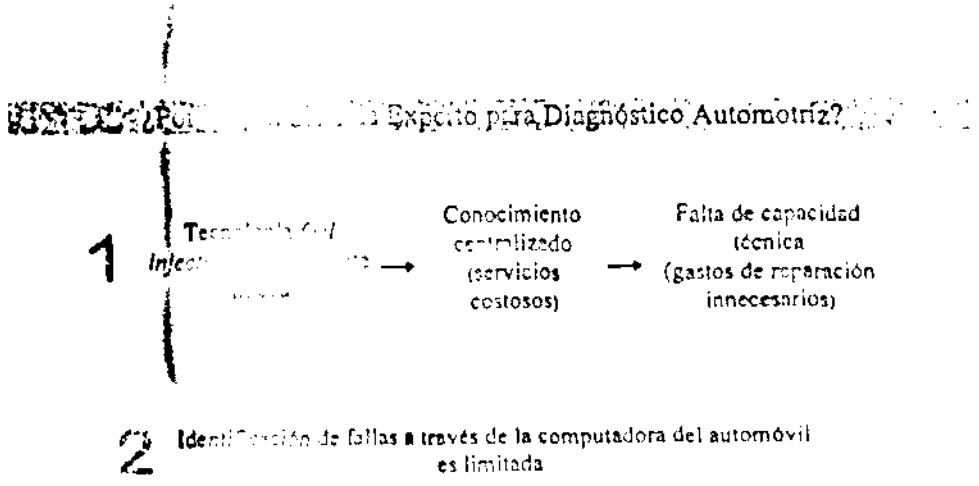


Figura 1.1 Razones de existir de un Sistema Experto para Diagnóstico Automotriz

1.2 Objeti[illegible]

El obje[illegible] diagnosticar las fallas, es decir las partes con sus problemas, q[illegible] automóvil *fuel injection* sin que se requiera de la presencia de u[illegible]no. Así, un mecánico no experimentado puede realizar el pro[illegible]do con el Sistema Experto que le indicará en dónde es[illegible] podrá solucionar el problema de tener centralizado el [illegible] necesario para hacer un diagnóstico en automóviles con tecnología [illegible]. Lo anterior permitirá hacer las reparaciones más rápido y no ne[illegible]r un experto.

Se pret[illegible]es que con todas las ventajas que SEDA puede representar, se comercialice dentro de talleres mecánicos no sólo como una herramienta para diagnóstico sino también como un importante e innovador medio de capac[illegible].

Un aut[illegible] sistema que se compone de muchas partes que interactúan entre [illegible] para hacerlo funcional. Para su estudio, todas estas partes se clasifican en sis[illegible]ces el automóvil puede ser dividido en subsistemas y estos a su ve[illegible] más pequeños; es decir que el automóvil puede analizarse desd[illegible]ra sistémica.

La tarea [illegible] será que a partir de la información que le proporcione el usuario, identifique cuál es la falla, esto lo hará en dos etapas o niveles, considerando la estructura sistémica del automóvil. El desarrollo de esta tesis abarca el primer nivel de diagnóstico en cuatro sistemas, como se verá más adelante.

El objetivo del primer nivel de diagnóstico es determinar en cuál de los subsistemas se [illegible]lla, con la finalidad de acotar las posibilidades y enfocarse más rápidamente a ella.

A partir de una serie de síntomas y algunas mediciones básicas, el primer nivel de diagnóstico determinará el o los subsistemas que presentan mayor posibilidad de falla, mediante la evaluación de todas las reglas de producción que componen a la base del conocimiento. En el segundo nivel, con mediciones específicas, se encontrará la parte – problema, que causa mal funcionamiento en el automóvil.

Existe una gran variedad de automóviles y entre ellos hay algunas diferencias en su estructura. Por lo que para el desarrollo de SEDA se consideraron únicamente automóviles General Motors modelo 86 en adelante, por ser de los más completos en su diseño.

Como se especificó anteriormente el automóvil tiene una estructura sistémica. Los cuatro grandes sistemas que lo componen son: el sistema motriz, el de suspensión, el de transmisión, y el de frenado. El primero es el más grande y complejo, en él se genera la energía que da movimiento al automóvil. Dada su importancia, SEDA se enfocará en este sistema.

El sistema motriz de un automóvil, se compone de ocho sistemas, esta tesis comprende a cuatro de ellos con sus respectivos subsistemas; (1) el sistema de alimentación de combustible que se divide en bombeo y conducción, inyección de combustible y admisión de aire; (2) el sistema de encendido que se divide en toma y elevación de voltaje, distribución, y generación de chispa; (3) el sistema de enfriamiento que se divide en bombeo y circulación del refrigerante y medición de temperatura y enfriamiento del refrigerante; y (4) el sistema eléctrico que se divide en alimentación y generación de energía eléctrica, arranque, y alimentación y servicios. En total se trabajará con once subsistemas que son los más extensos y suficientes para crear un prototipo del primer nivel de diagnóstico.

1.3 Metodología para el desarrollo de SEDA

En el siguiente capítulo se explicará cuáles son los pasos básicos a seguir, para el desarrollo de un Sistema Experto. Teóricamente, todos deben cumplir con ellos; sin embargo, no existe una metodología estricta porque dependiendo del autor, unos ponen más énfasis en algunas etapas dividiéndolas en otras más pequeñas según sus necesidades, creando así nuevas metodologías que se ajustan perfectamente al problema a solucionar.

Entonces, dependiendo del problema, se pueden crear etapas que permitan alcanzar el objetivo completamente, sin olvidar que cualquier metodología que se emplee debe pasar por las etapas básicas, que son: identificación, conceptualización, formalización, e implementación y prueba. En términos generales, las dos primeras equivaldrían a una fase de análisis, la segunda a una fase de diseño, la formalización e implementación, entrarían en una fase de desarrollo y las pruebas a una fase de instalación. Como un sistema experto puede

ser muy extenso, se consideran juntas la implementación y prueba porque se puede ir probando conforme se implementa el conocimiento experto.

La razón principal para crear metodologías que se ajusten al problema, es que en algunos casos las etapas toman más tiempo o interactúan varios factores y es mejor dividir las etapas para que el equipo de trabajo pueda organizarse mejor.

El equipo de trabajo para desarrollar el primer nivel de diagnóstico, está compuesto básicamente de tres personas: el experto, el director de tesis y la desarrolladora. El experto proporciona toda la información necesaria al director de tesis y desarrolladora, para que SEDA pueda imitar su comportamiento. La desarrolladora con ayuda del director de tesis analiza y organiza la información extraida del experto para posteriormente implementarla empleando un software especializado en la construcción de sistemas expertos.

Para llevar acabo este proceso, se pasó por las siguientes siete etapas; las dos primeras corresponden a la etapa de identificación, la siguiente a la etapa de conceptualización, del punto cuatro al seis equivalen a la etapa de formalización y se termina con la implementación:

1) Planteamiento del problema
2) Identificación de recursos
3) Análisis del proceso de diagnóstico del experto
4) Aprendizaje del software a utilizar
5) Diseño de estructura para representar el conocimiento experto
6) Adquisición del conocimiento experto
7) Implementación

1. Planteamiento del problema

Durante esta etapa, se define el problema intercambiando diferentes puntos de vista para identificar sus caracteristicas y los objetivos que se desean alcanzar; se da respuesta a preguntas como: ¿qué tan necesario es el uso de un Sistema Experto?, ¿qué tipo de conocimiento experto se requiere?, ¿cuáles son los conceptos básicos que intervienen en el problema? y otras que ayudan a visualizar completamente el problema.

En el caso de SEDA se trata de un Sistema Experto de diagnóstico

ser muy extenso, se consideran juntas la implementación y prueba porque se puede ir probando conforme se implementa el conocimiento experto.

La razón principal para crear metodologías que se ajusten al problema, es que en algunos casos las etapas toman más tiempo o interactúan varios factores y es mejor dividir las etapas para que el equipo de trabajo pueda organizarse mejor.

El equipo de trabajo para desarrollar el primer nivel de diagnóstico, está compuesto básicamente de tres personas: el experto, el director de tesis y la desarrolladora. El experto proporciona toda la información necesaria al director de tesis y desarrolladora, para que SEDA pueda imitar su comportamiento. La desarrolladora con ayuda del director de tesis analiza y organiza la información extraída del experto para posteriormente implementarla empleando un software especializado en la construcción de sistemas expertos.

Para llevar acabo este proceso, se pasó por las siguientes siete etapas; las dos primeras corresponden a la etapa de identificación, la siguiente a la etapa de conceptualización, del punto cuatro al seis equivalen a la etapa de formalización y se termina con la implementación:

1) Planteamiento del problema
2) Identificación de recursos
3) Análisis del proceso de diagnóstico del experto
4) Aprendizaje del software a utilizar
5) Diseño de estructura para representar el conocimiento experto
6) Adquisición del conocimiento experto
7) Implementación

1. Planteamiento del problema

Durante esta etapa, se define el problema intercambiando diferentes puntos de vista para identificar sus características y los objetivos que se desean alcanzar; se da respuesta a preguntas como: ¿qué tan necesario es el uso de un Sistema Experto?, ¿qué tipo de conocimiento experto se requiere?, ¿cuáles son los conceptos básicos que intervienen en el problema? y otras que ayudan a visualizar completamente el problema.

En el caso de SEDA se trata de un Sistema Experto de diagnóstico

necesario debido a la concentración de conocimiento que existe en un área que requiere del conocimiento en forma dispersa. Además hay interés por parte de propietarios de talleres mecánicos en el desarrollo de este tipo de herramientas, principalmente para la capacitación.

Para crear un prototipo de este Sistema Experto se considerarán únicamente cuatro de los sistemas del sistema motriz de los automóviles General Motors

2. Identificación de recursos

Aquí se identifican los medios con los que se cuenta para desarrollar el sistema, por ejemplo: los recursos humanos (entre ellos la fuente de conocimiento), el tiempo y el equipo de cómputo.

Se determina quiénes y de qué forma participarán en el desarrollo del sistema. Por lo general se debe contar con un equipo formado por ingenieros de conocimiento, analistas, desarrolladores y por lo menos un experto.

El tiempo para desarrollar un Sistema Experto es muy largo y éste se debe ir ajustando según el avance que se tenga con el experto, que será la fuente de conocimiento.

El experto debe comprometerse a invertir muchas horas de su tiempo para trasmitir su conocimiento y verificar si lo que hace el sistema es lo mismo que él haría ante una situación determinada.

Una vez que se habla con el experto queda más claro cuál es el tipo de información que se manejará por lo que entonces, se puede determinar el equipo que se empleará, así como el software.

Para el desarrollo de SEDA se cuenta con tres tipos de recursos humanos, el experto, el director de proyecto y los desarrolladores, estos dos últimos hacen en conjunto el papel de los analistas.

El experto que participó en el desarrollo del primer nivel de diagnóstico, es el Ingeniero Daniel Giner Cortés experto en tecnología *fuel injection*. Se emplea equipo y software proporcionado por el Instituto de Ingeniería de la UNAM.

Dado que el sistema está enfocado a usuarios que no tienen una formación especial en computo (mecánicos automotrices), se debe emplear equipo de fácil manejo y comprensión, como son las computadoras personales trabajando bajo un ambiente Windows. A pesar de que existen varios softwares para desarrollar sistemas expertos, no todos pueden cumplir con estas expectativas. Level 5 Object trabaja bajo ambiente Windows y permite la creación de interfaces de usuario muy amigables, lo que nos lleva a usarlo como lenguaje de programación principal para la creación de SEDA.

3. Análisis del proceso de diagnóstico del experto

En esta etapa se identifican los pasos que sigue el experto para realizar un diagnóstico y se determina específicamente qué hace en cada uno de ellos. Esto permite detectar el flujo de información que deberá tener el sistema experto e identificar los aspectos más importantes que intervienen en el problema para saber el conocimiento que se manejará.

En el caso de SEDA, este análisis se realiza durante varias sesiones con el experto en su taller mecánico. Mientras él hace un diagnóstico, se verifica su comportamiento y después de varios diagnósticos se determina cuál es el proceso que sigue. Esta tarea no es fácil porque el experto no está acostumbrado a ver su comportamiento como una serie de pasos, si no que simplemente actúa ante el problema, lo que dificulta el poder estructurar su comportamiento.

A raíz de ver el proceso de diagnóstico que sigue el experto, se decidió hacer el diseño de SEDA considerando dos niveles de diagnóstico. El primero determina en qué subsistema está la falla y el segundo identifica a la falla.

4. Aprendizaje del software a utilizar

Esta etapa junto con la de adquisición del conocimiento experto (etapa 6) es una de las más largas. En ella se emprende la tarea de aprender el lenguaje que se empleará para desarrollar el Sistema Experto, que en este caso es Level 5 Object.

El proceso de aprendizaje es largo principalmente porque se trata de programación orientada a objetos y no de programación estructurada como los

lenguajes más comunes, por lo que familiarizarse con su lógica lleva tiempo. Al ser un lenguaje no muy común, es difícil encontrar asesores en el área por lo que la forma de aprendizaje es mediante pequeños ejemplos que se aproximan cada vez más al sistema que se desea.

5. Diseño de estructura para representar el conocimiento experto

En la etapa tres se identifican los conceptos clave que intervienen en el problema. Para el primer nivel de diagnóstico estos conceptos son: subsistemas, partes, problemas, síntomas, lecturas del analizador de gases y códigos de falla arrojados por un scanner.

Durante esta etapa, todos esos conceptos se formalizan traduciéndolos a clases y objetos que permiten dar origen a las reglas de producción que forman a la base del conocimiento.

Considerando la información que se maneja y el flujo que debe tener ésta se determinó que el primer nivel de diagnóstico seguiría una metodología de inferencia de encadenamiento hacia adelante y el segundo nivel encadenamiento hacia atrás.

Una vez formalizada la información que se va a manejar y cómo se manejará, se plantean distintas formas para adquirir el conocimiento del experto.

6. Adquisición del conocimiento experto

En esta etapa se inicia la extracción de información del experto para dar origen a la base de conocimiento. Esta es la etapa más larga durante todo el proceso de desarrollo del Sistema Experto, porque es difícil tanto para el experto como para los ingenieros del conocimiento, entablar una comunicación entendible para ambos, debido a que cada uno tiene su propio vocabulario.

Otro de los factores que complica más esta tarea, es que para el experto resulta muy difícil transmitir su conocimiento en forma estructurada, porque está acostumbrado a hacerlo oralmente, traducirlo a un formato previamente diseñado es una labor complicada y algunas veces hasta tediosa.

Con las formas planteadas en la etapa anterior se inicia una serie de sesiones con el experto en las que se le plantean diversos ejemplos de problemas y él explica bajo qué circunstancias se puede presentar. Las formas originales se van modificando hasta encontrar la manera de entablar una mejor comunicación con el experto y agilizar el proceso de adquisición de conocimiento. Esto significa que las formas planteadas inicialmente pueden no ser las más adecuadas.

Para adquirir el conocimiento experto del primer nivel de diagnóstico, se diseñaron tres tipos de formatos. El primero es el formato parte-problema, el segundo el formato de síntomas por estado de operación, y el último fallas-síntomas. Estos formatos permiten obtener estructuralmente el conocimiento, facilitando la realización de reglas de producción.

La secuencia que se sigue para adquirir el conocimiento con ayuda de las formas antes citadas es la siguiente:

Se pide al experto que indique las partes que componen a determinado subsistema y los problemas que puede presentar cada una de ellas (detección de fallas). Esta información se registra en el primer formato.

A partir de las primeras fallas que se presentan se inicia una detección de síntomas, es decir, el experto indica cuáles son los síntomas más comunes, teniendo así la información para el segundo formato.

Cuando se tienen algunos síntomas, el experto formaliza la información relacionando las fallas con síntomas y mediciones, es decir que, indica bajo qué circunstancias se presentan fallas específicas, con esto se tiene la información del tercer formato.

Toda la información recabada se analiza y organiza para evitar redundancias o contradicciones.

En una nueva sesión, se verifica junto con el experto, que la información manejada esté correcta, si existen modificaciones se realizan y se vuelven a presentar al experto hasta que la información quede completamente lista para ser implementada.

Todo este proceso toma alrededor de cuatro a cinco sesiones. En algunas

ocasiones es difícil contactar al experto, por lo que también se realizan consultas vía telefónica y vía fax. Durante todo el proceso la información contenida en los formatos puede ser aumentada o modificada.

Es importante considerar que el experto no está dedicado solamente al desarrollo del Sistema Experto y sus ocupaciones no le permiten tener sesiones muy largas, por lo que la información se obtiene por bloques y en cada uno se repite el proceso. Para el primer nivel de diagnóstico cada subsistema implica una repetición del proceso.

7. Implementación

Es en esta etapa donde todo el conocimiento experto adquirido, se traduce a un lenguaje de programación que dará cuerpo al sistema experto.

En el desarrollo del primer nivel de diagnóstico, esta etapa se alterna con la adquisición del conocimiento experto, básicamente por dos razones. La primera es que la cantidad de información que se maneja es mucha, por lo que se va implementando a medida que se adquiere, y la segunda es que debido al tipo de inferencia que se sigue y el diseño realizado, cada una de las reglas de producción tiene asociado un factor de certidumbre que indica qué tan probable es que se presente una falla, y éste debe ser proporcionado o al menos aprobado por el experto.

Para obtener los factores de certidumbre el experto debe analizar las reglas por lo que es necesario tener la información previamente implementada. Este es otro tipo de información que también debe ser adquirida del experto.

Estas siete etapas permiten realizar de forma organizada todas las actividades necesarias para crear a SEDA. A lo largo de la presente tesis se explicará con detalle cada una de las actividades realizadas, quedando implícita la metodología empleada.

CAPÍTULO 2: ANTECEDENTES

En este capítulo se verán los conceptos básicos y las caracteristicas principales de los Sistemas Expertos, con el objeto de proporcionar las bases para comprender mejor el desarrollo de SEDA. También se verá la importancia que tienen los Sistemas Expertos en la industria automotriz y su participación en ella.

2.1 Los Sistemas Expertos

En la década de los cincuenta, empiezan a escribirse programas de computadora de tipo simbólico para la resolución automática de problemas. Es entonces cuando surge la Inteligencia Artificial (IA).

La Inteligencia Artificial es la solución de problemas complejos con el apoyo de una computadora, mediante la aplicación de procesos que son análogos al proceso de razonamiento humano.

La Inteligencia Artificial se divide en varias áreas, entre las que se encuentran:

- Solución heurística de problemas
- Redes neuronales
- Procesamiento del lenguaje natural
- Aprendizaje
- Sistemas Expertos
- Robótica
- Percepción y reconocimiento de formas

Una de las más exitosas de estas áreas, ha sido la de los Sistemas Expertos (SE), los cuales, a través de la aplicación de las técnicas de IA, captan el conocimiento básico que permite a una persona desempeñarse como experto frente a problemas complicados.

La investigación específica en SE realmente comenzó a mediados de los años sesenta. Varios sistemas se desarrollaron entre 1965 y 1970. La mayoría de ellos fueron de alcance muy limitado y se orientaron hacia juegos o temas altamente académicos e idealizados. Aunque el desarrollo de SE es relativamente nuevo, existen ya varios de estos, que se emplean en diversas organizaciones con diferentes aplicaciones

Los siguientes incisos, tienen como objetivo proporcionar un panorama general de lo que son los Sistemas Expertos, incluyendo su estructura y sus características principales.

2.1.1 ¿Qué son los Sistemas Expertos?

El área de Sistemas Expertos ha avanzado rápidamente en los últimos años; es una de las ramas de la inteligencia artificial con más popularidad desde su introducción comercial en los años ochenta. Hoy día podemos encontrar Sistemas Expertos, no sólo en el ámbito científico, sino también en otros campos más comunes como son ingeniería, manufactura, finanzas y economía, entre muchos otros.

Los Sistemas Expertos son básicamente mecanismos que simulan el comportamiento humano para resolver problemas específicos, como todos los que forman parte de la Inteligencia Artificial. Sin embargo, estos resuelven problemas complejos que sólo un experto humano podría resolver.

Un Sistema Experto, es un programa de computadora que emplea conocimientos y procedimientos de inferencia (figura 2.1), para resolver problemas que son los suficientemente complicados como para requerir de la experiencia humana para su solución. Tanto el conocimiento como la inferencia empleada, son extraídos de un experto humano. Entonces un Sistema Experto emula el comportamiento de un experto humano.

Figura 2.1 Elementos básicos de los Sistemas Expertos.

En esta definición, se encuentran tres conceptos importantes, *conocimiento, inferencia* y *experto*. A continuación se analizará cada uno de ellos:

Experto, es una persona que tiene conocimientos o habilidades especiales, que le permiten solucionar problemas que a los demás les resultarían imposibles de resolver.

El conocimiento que emplea un SE, es precisamente el que se obtiene de un experto y se le conoce como **conocimiento experto**. Para que el SE pueda desarrollar su tarea, es necesario primero representar el conocimiento, es decir, pasar todo el conocimiento adquirido del experto a formas que permitan hacer la implementación mediante algún software. Existen varias formas de hacerlo, las más comunes son:

- *Reglas de producción:* Son estructuras lógicas formadas de un conjunto de condiciones que deben cumplirse para dar una conclusión.

- *Conceptos estructurados o frames:* Son bloques de información relevante respecto a un objeto, representada en un formato preciso (como una base de datos). Esa información es básicamente la siguiente: el nombre del objeto, el o los géneros próximos del objeto, cualidades o atributos del objeto, y condiciones que permiten accesar a la información del objeto y/o modificarla, de manera que se pueda tener un control sobre ésta.

- *Redes semánticas:* Son una forma de representar el conocimiento mediante nodos y arcos, donde los nodos son los conceptos y los arcos las relaciones que hay entre ellos. Las redes semánticas sirven para ubicar el significado de un concepto dado en términos de sus relaciones con otros conceptos.

El conocimiento experto abarca solamente determinada área del saber, esto significa que los Sistemas Expertos como los expertos humanos, pueden resolver muy bien problemas que están dentro de un área especifica o dentro de un dominio, por ejemplo, medicina, finanzas, ingeniería , mecánica etc.

Se ha analizado lo que es un experto y el conocimiento experto, ahora se verá lo que es la **inferencia**. El experto al enfrentarse a un problema, infiere su solución, es decir, a través del razonamiento llega a ésta. De la misma manera el SE hace inferencias empleando el conocimiento que se le proporciona. Entonces, la forma en que el SE maneja el conocimiento proporcionado es la inferencia. En

el capítulo 2.1.2 se analizará detalladamente cómo es que el SE maneja el conocimiento y la inferencia.

Los Sistemas Expertos se emplean para realizar tareas complicadas que en el pasado solamente podían llevarse a cabo por personas expertas intensamente entrenadas. Ahora con la ayuda de un SE, es posible realizar esas tareas sin depender de la disponibilidad de un experto.

Existen diferentes tipos de sistemas expertos, de acuerdo a la tarea que realizan, éstos pueden ser clasificados de la siguiente manera:

De diagnóstico: A partir de determinados síntomas identifican la naturaleza y la causa del mal funcionamiento de un sistema.

De planeación: Diseñan estrategias para lograr un objetivo.

De predicción: Infieren las consecuencias de una situación determinada

De diseño: Desarrollan configuraciones para resolver problemas satisfaciendo sus restricciones.

De interpretación: Infieren significados de estados y situaciones como, por ejemplo, comprensión de mensajes, análisis de imágenes.

De control: Mediante la comparación de una situación real con una situación deseada, formulan un plan de corrección y monitorean su ejecución.

De monitoreo: Analizan observaciones del comportamiento de un sistema y detectan posibles desviaciones de la norma, para tomar medidas correctivas.

De instrucción: Ayudan en el proceso de aprendizaje, mostrando al usuario la información necesaria e indicando la manera óptima de adquirir el conocimiento y llevan un seguimiento del aprendizaje. También se les conoce como tutoriales.

De reparación: Desarrollan planes para corregir fallas, para posteriormente ejecutarlos.

De depuración: Describen soluciones correctivas para los casos en los que existen condiciones anómalas.

Estos diferentes tipos de SE pueden ser aplicados a cualquier área del conocimiento, por ejemplo, es común ver un SE de diagnóstico en el área médica. Un SE de interpretación en el área de química, o bien un SE de instrucción en electrónica. Cualquiera que sea su aplicación, todos los Sistemas Expertos tienen la misma estructura básica, que se verá a continuación.

2.1.2 Estructura de los Sistemas Expertos

Los Sistemas Expertos emplean una amplia variedad de estructuras específicas en sus sistemas, principalmente porque una estructura es más aplicable que otra cuando se considera una aplicación dada. A pesar de las diferencias significativas, la mayoría de las estructuras tienen muchos componentes en común.

Internamente, los SE se componen de dos elementos principales, la base de conocimientos y la máquina de inferencia, pero existen otros tres bloques funcionales que forman parte de su estructura general: el sistema de adquisición de conocimiento, la interfaz de usuario y la interfaz explicativa.

La base de conocimientos almacena los diferentes hechos y reglas que constituyen el conocimiento experto y que están siempre disponibles para el sistema. Este conocimiento está expresado de la misma manera que en un experto humano, con una serie de hechos conocidos que se presentan a partir de determinadas condiciones, plasmados en una *base de hechos* y un conjunto de relaciones cognoscitivas, referentes al dominio del SE, expresadas en forma de reglas de producción que configuran una *base de relaciones*.

La máquina de inferencia realiza el proceso de razonamiento, empleando la información contenida en la base de conocimientos. Así como un experto humano tiene conocimientos sobre un área específica y basándose en éstos, mediante su razonamiento llega a una conclusión para resolver un problema. Un SE puede hacer lo mismo usando la máquina de inferencia para manipular a los elementos que conforman a la base de conocimientos.

La forma en que hace el razonamiento, es mediante búsquedas partiendo de un patrón dado. Existen varias estrategias generales de control de las búsquedas.

las más conocidas son la de encadenamiento hacia adelante y la de encadenamiento hacia atrás, también se les conoce como método de inferencia.

El sistema de adquisición de conocimiento se utiliza para transferir el conocimiento, de uno o varios expertos, a la base de conocimientos. Éste es uno de los bloques más difíciles de efectuar, por la dificultad que puede tener extraer los conceptos de los cerebros humanos a estructuras de información que permitan su acceso y utilización a la máquina de inferencias.

La interfaz de usuario es el bloque encargado de establecer la comunicación entre el SE y el usuario. Normalmente, un experto humano necesita comunicarse con el usuario haciendo preguntas específicas; de la misma forma, el SE requiere de esa comunicación. Para esto, se apoya de una serie de recursos visuales especialmente estructurados para obtener información , tales como: ventanas, menús, hipertextos y preguntas selectivas entre otros.

La interfaz explicativa es la parte en la que el Sistema Experto da a conocer al usuario sus conclusiones , así como también puede dar justificaciones. De igual manera que un experto explica sus decisiones y da recomendaciones, un SE puede hacerlo mediante esta interfaz.

Otros elementos que se deben considerar, tal vez no como parte de la estructura, pero sí como elementos importantes para el desarrollo y funcionamiento del SE, son: *el experto* y *el usuario*. El primero proporcionará toda la información, para crear a la base de conocimientos y el segundo dará solución a sus problemas, sin requerir de la experiencia del primero, empleando al SE (figura 2.2).

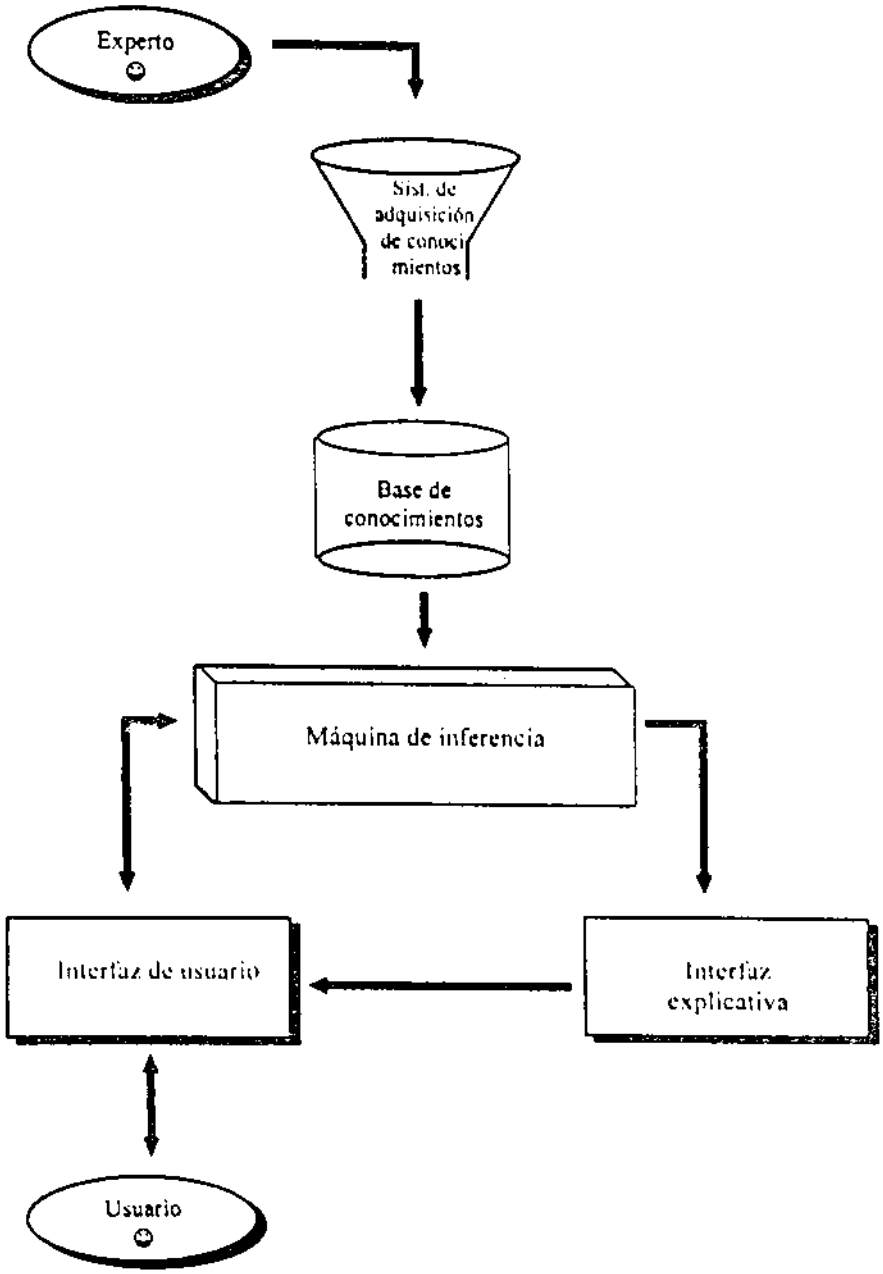


Figura 2.2 Estructura de un Sistema Experto.

2.1.3 Características de los Sistemas Expertos

En términos generales, los SE son diseñados y desarrollados de distinta forma que un programa convencional, porque el problema a tratar no tiene una solución algorítmica. Entonces, el SE se atiene a la inferencia para encontrar una solución razonable, esta es una de sus principales características.

Para lograr hacer las inferencias que lo llevan a la solución, un SE emplea *conocimiento heurístico*, es decir conocimiento adquirido de la experiencia, es éste otro aspecto que caracteriza a los SE.

El *conocimiento heurístico* puede ayudar en la búsqueda de solución de problemas, sin embargo no garantiza encontrarlas a diferencia de los sistemas convencionales, en los que siempre hay un algoritmo que nos lleva a una solución. A pesar de esto el conocimiento heurístico juega un papel muy importante en la solución de ciertos tipos de problemas, como los que encontramos en el campo de la medicina o la ingeniería. El conocimiento heurístico proporciona medios para llegar a una solución reduciendo tiempos y costos.

Durante el desarrollo de un SE existen varias etapas, estas son variables dependiendo del autor que se consulte. Por ejemplo J.P Sánchez y Beltrán en su libro "Sistemas Expertos, una metodología de programación", consideran las siguientes etapas:

1. Estudio de demanda
2. Análisis del problema
3. Elección de fuente de conocimientos
4. Presentación de soporte
5. Obtención del conocimiento
6. Selección de soporte
7. Construcción de prototipo
8. Validación de prototipo
9. Construcción del modelo operacional
10. Mantenimiento

Estas etapas pueden ser reducidas a las siguientes cuatro etapas básicas, por las que todo SE debe pasar y que los caracterizan:

❶ Identificación: en esta etapa se determinan las características del problema. El ingeniero del conocimiento[1] y el experto trabajan juntos para identificar el área del problema y definen el objetivo y el alcance; identifican a las personas que participarán durante el desarrollo del SE, como programadores y expertos; y determinan los recursos necesarios, como tiempo, equipo y software que se empleará para implementar el conocimiento experto.

❷ Conceptualización: en esta etapa, el experto y el ingeniero del conocimiento discuten los conceptos clave, las relaciones entre estos y las características del flujo de información, necesarias para describir el proceso de problema-solución. Es decir, encuentran los conceptos que representan al problema.

❸ Formalización: durante esta etapa se diseñan estructuras para organizar el conocimiento, mapeando a los conceptos y las relaciones en una representación formal; es decir, empieza el proceso de adquisición del conocimiento. El ingeniero del conocimiento considera el software a utilizar, para que en esta etapa las estructuras diseñadas puedan ser fácilmente implementadas con éste.

❹ Implementación: en esta etapa el ingeniero del conocimiento reorganiza el conocimiento adquirido y lo maneja para dar origen a reglas que cumplan con el flujo de información planteado. El resultado del conjunto de reglas y sus estructuras de control asociadas definen el prototipo del SE, listo para ser ejecutado y probado.

Una vez cumplidas estas cuatro etapas, podemos pasar a una etapa final que es la de **prueba**, en donde se validan las reglas que forman al Sistema Experto.

Se observa que las etapas de la primera propuesta, están inmersas en las etapas básicas anteriores. Así dependiendo de la aplicación y de los ingenieros del conocimiento estas etapas pueden ser divididas en otras.

Las características antes descritas, son características que presentan los SE por naturaleza, pero además de éstas, los SE deben tener otras para garantizar un funcionamiento exitoso. A continuación se verán algunas de ellas.

[1] Nombre que recibe la persona encargada del desarrollo del SE

⇨ Los Sistemas Expertos deben proporcionan soluciones que estén al mismo nivel que las que da un experto humano. Esto es, que deben ser lo suficientemente confiables como para no presentar errores que los conducen al desuso y además deben encontrar estas soluciones en un tiempo razonable, es decir, en el mismo o menor tiempo que un experto.

⇨ Los Sistemas Expertos deben ser capaces de mostrar en forma clara los pasos que siguieron en su razonamiento para llegar a la solución de la misma manera que un experto puede explicar el porqué de sus decisiones, esto permitirá que se facilite la verificación y corrección del conocimiento experto durante su desarrollo.

⇨ Debido a la gran cantidad de conocimiento que tiene un SE, es importante que cuente con un mecanismo eficiente para agregar, cambiar y borrar conocimiento, es decir que el sistema debe se flexible. La representación del conocimiento en forma de regla, permite esta flexibilidad porque son manejadas modularmente.

⇨ También los Sistemas Expertos deben listar las razones para llegar a una hipótesis y explicar las consecuencias de éstas.

Un problema con los SE, es que su experiencia está limitada a una área del conocimiento, por lo que no pueden generalizarlo usando analogías para nuevas situaciones, como lo hacen los humanos. A pesar de sus limitaciones, los SE han tenido un gran éxito en la solución de problemas reales que la programación convencional no puede resolver y presentan muchas ventajas, entre ellas se pueden considerar las siguientes:

* Están siempre disponibles para que se pueda hacer uso de su experiencia y encontrar soluciones.

* Reducen los costos de uso de experiencia, porque los honorarios causados por un experto son mucho más altos que los costos que produce un SE.

* En el caso de que se requiera del conocimiento experto en lugares muy accidentados no es necesaria la presencia de un experto humano.

- Puede contener experiencia de más de un experto.

- Es constante y no tiene emociones, ésta es una gran ventaja considerando que un experto, al ser humano, está sujeto a cambios en su estado de ánimo por tensiones, fatiga etc. que condicionan su capacidad de solucionar problemas.

2.2 Sistemas Expertos en la industria automotriz

La industria automotriz ha crecido grandemente y demanda nuevas tecnologías para cubrir a sus distintas áreas, desde la manufactura hasta el diagnóstico de automóviles. Es aquí donde los Sistemas Expertos juegan un papel muy importante por ser herramientas de software que sirven para facilitar el trabajo de un grupo de personas proporcionando resultados certeros y rápidos.

Desde hace ya varios años los Sistemas Expertos se han convertido en herramientas útiles para el diagnóstico automotriz. En los años ochenta se realizaron varios de ellos, quedando la mayoría en el campo experimental. En la actualidad las principales empresas automotrices están interesadas en contar con este tipo de sistemas, como elementos de gran ayuda, no sólo para el diagnóstico, sino también para la administración de empresas.

Tal es el caso de General Motors, que en 1995 lanzó un sistema experto llamado DCE (Dealer Consultant Expert) que analiza los reportes mensuales proporcionados por otro sistema encargado de procesar los estados financieros (CADAT - Computer Aided Dealership Analysis Tools), y genera un informe completo y claro del estado actual de la empresa.

En el área de diagnóstico, también se han desarrollado varios sistemas, sin embargo estos no cubren todas las necesidades que tiene la industria automotriz porque se enfocan a fallas simples y a casos específicos. Un ejemplo de esto, es el sistema experto llamado Automotive Expert System, que fue desarrollado en 1996 por el ISL (Intelligent System Laboratory). Este sistema fue hecho con CLISP un lenguaje de programación diseñado por NASA/ Johnson Space Center y diagnostica problemas muy simples que puede presentar un automóvil.

Es evidente que la realización de este tipo de sistemas no es algo simple, debido a que requiere de un gran equipo de trabajo entre expertos y

desarrolladores, además de un largo periodo de desarrollo. Por estas razones las compañías automotrices están solicitando la colaboración de grupos de investigadores para la creación de sistemas expertos para diagnóstico automotriz.

En Mayo de 1996 Chrysler contrató a Lockheed Martin Corporation para el desarrollo de un sistema experto que analizará a los sistemas claves del automóvil y buscará fallas. Para el desarrollo de este sistema el equipo de Lockheed Martin Defense Systems en Pittsfield, Mass., y Lockheed Martin Control System en Binghamton, N.Y., trabajarán conjuntamente con un equipo de mecánicos expertos de Chrysler para crear a lo que ellos han denominado como la "siguiente generación de software para diagnóstico".

Otro ejemplo de la gran necesidad de Sistemas Expertos para diagnóstico en la industria automotriz, es el sistema que se está desarrollando en la Universidad de Michigan conjuntamente con un equipo de ingenieros de Ford, para las pruebas e identificación de fallas en el proceso de manufactura de automóviles.

Con todos estos ejemplos podemos ver que la industria automotriz se ha convertido en un campo más para la explotación de Sistemas Expertos.

CAPÍTULO 3: ANÁLISIS DEL PROBLEMA

En el primer capítulo se vieron las razones por las que se requiere de un Sistema Experto para el diagnóstico de automóviles con tecnología *fuel injection* (*FI*). Ahora se analizará en qué consiste esta tecnología y cómo es la estructura general de los automóviles (estructura sistémica), así como el concepto falla automotriz como meta en un diagnóstico y, por lo tanto, en el Sistema Experto.

Posteriormente se visualizará la metodología que el experto sigue para hacer un diagnóstico y cómo lo hará el Sistema Experto. Todo esto permitirá tener una mejor perspectiva del problema.

3.1 El automóvil y el Sistema Experto

La tecnología *Fuel Injection* (Inyección de Gasolina, también conocida como Inyección de Combustible) es relativamente nueva . Muy pocas personas tienen un dominio sobre ella, lo cual hace que el diagnóstico eficiente en automóviles con sistema electrónico de inyección de gasolina sea difícil de obtener.

En la República Mexicana, existen cinco marcas principales en el mercado que manejan tecnología *fuel injection*. No obstante son similares en su forma de operación. La primera marca automotriz en incorporar un sistema de inyección electrónica en el país fue CHRYSLER en 1984, después GENERAL MOTORS en 1986 , seguido por FORD en 1988, posteriormente NISSAN y VOLKSWAGEN; pero son los modelos americanos los que incluyen una mayor tecnología electrónica en sus vehículos.

El sistema de inyección de gasolina en los motores de automóviles de uso diario, es un método de alta tecnología necesario para alcanzar niveles más altos de rendimiento. Este sistema no se había utilizado hasta hace pocos años, por dos razones principales. La primera es que para el control de la inyección de combustible se necesita una pequeña computadora y ésta tiene un alto costo; la

segunda es que la necesidad de aumentar el rendimiento del motor no había sido tan crítica como ahora.

La inyección de gasolina es el método de introducir combustible y aire a las cámaras de combustión (cilindros) en la proporción más adecuada. Este sistema sustituye al carburador, el cual no logra una mezcla tan precisa en cualquier condición de operación.

El carburador es un dispositivo diseñado para realizar la mezcla de gasolina-aire necesaria para la combustión dentro del motor. La operación eficiente de éste, depende del adecuado balance de la mezcla.

Existen diversas condiciones que tienden a modificar la mezcla gasolina-aire. Por ejemplo, las condiciones del ambiente, como temperatura del aire y presión atmosférica, modifican la cantidad de aire que entra al motor. Por otra parte, un motor funcionando en frío requiere una mayor cantidad de gasolina para alcanzar rápidamente la temperatura de operación, como también requiere de mucho más gasolina para la aceleración y menos para velocidades moderadas. Esto significa que hay diferentes condiciones de operación de un motor que demandan ajustes en la cantidad de gasolina y aire que entra a los cilindros para su combustión.

El desarrollo tecnológico del carburador, permitió que este dispositivo hiciera posibles diversos cambios en la mezcla gasolina-aire, de acuerdo a las condiciones de operación. Sin embargo, su ajuste era local y por lo tanto estaba sujeto a un mal funcionamiento, impidiendo la rápida identificación de la falla.

La tecnología *fuel injection* permitió centralizar los ajustes de la mezcla gasolina-aire necesarios para una operación eficiente del motor. Esta centralización se realiza a través del módulo de control del motor o ECM por sus siglas en inglés, *Engine Control Module*. Esto es a lo que se le conoce como computadora del automóvil.

A través de la instalación de un conjunto de sensores, el ECM recibe información de la cantidad de aire que entra al motor y calcula la cantidad de gasolina que debe suministrar. Dicho suministro se realiza por medio de actuadores llamados inyectores. Los inyectores responden a pulsos que envía el ECM para mantenerlos abiertos durante determinado tiempo, y así permitir el paso de la cantidad de gasolina adecuada.

Para el control de la mezcla gasolina-aire, el ECM recibe la siguiente información, proporcionada por los diversos sensores instalados en diferentes partes del auto:

- Ángulo de apertura de la mariposa de la garganta de admisión
- Temperatura del aire de entrada
- Masa del flujo del aire de entrada
- Presión barométrica
- Vacío dentro del múltiple de admisión
- Cantidad de oxígeno de los gases provenientes de la combustión

Cuando alguno de estos sensores no envía la información esperada, la computadora llega a registrar la falla y emite un código de falla a través del cual se puede identificar la presencia del problema.

La tecnología *fuel injection* ha incorporado la utilización de sensores y actuadores para el control de otras funciones. De esta manera algunas operaciones que antes se hacían mecánicamente, en la actualidad se realizan por medios electrónicos. Por ejemplo, la determinación del tiempo de encendido ya no se realiza mediante un distribuidor mecánico sino por medio de un sensor (sensor DIS), que recoge la señal de un disco colocado en el cigüeñal del motor.

La presencia cada día mayor de dispositivos electrónicos que controlan el funcionamiento del automóvil, ha ocasionado el problema de falta de capacidad técnica para diagnosticar una falla por parte del personal que labora en los talleres automotrices.

Por otra parte, la identificación de fallas a través de la computadora es limitada porque solamente determina el mal funcionamiento de sensores o actuadores, y no del resto de la partes que componen al motor. Además, existen condiciones de falla de algunos sensores y actuadores, que no pueden ser identificados plena y oportunamente por la computadora.

Por estas razones, se consideró oportuno aprovechar el conocimiento experto disponible para desarrollar un sistema experto para el diagnóstico automotriz que pudiera ser utilizado en muchos talleres automotrices y para la propia capacitación en tecnología *fuel injection*

3.1.1 Estructura Sistémica de un automóvil

La composición de un automóvil es muy compleja; contiene un gran número de partes que interactúan entre sí para realizar una tarea específica. Es por esto que para su mejor estudio y comprensión se ha dividido a todo el automóvil en grupos o sistemas. Estos a su vez se dividen en subsistemas. Cada subsistema se compone de partes que al operar conjuntamente realizan una tarea determinada.

El automóvil está compuesto de cuatro grandes subsistemas, que son:

- Sistema motriz
- Sistema de suspensión
- Sistema de transmisión
- Sistema de frenado.

El ***sistema motriz*** es el encargado de dar movimiento al vehículo a través del motor, el ***sistema de suspensión*** se encarga de proporcionar estabilidad en el movimiento, el ***sistema de transmisión*** se encarga de pasar el movimiento generado por el sistema motriz al sistema de suspensión, básicamente a las llantas, y finalmente el ***sistema de frenado*** se encarga de detener el movimiento del automóvil.

El Sistema Experto para Diagnóstico Automotriz, se enfocará a los sistemas que componen al sistema motriz , que son:

- Sistema de alimentación de combustible
- Sistema de encendido
- Sistema eléctrico
- Sistema de enfriamiento
- Sistemas de lubricación
- Sistema de escape
- Sistema de control
- Sistema mecánico

Todos los sistemas operan alrededor del ***sistema mecánico***. Este sistema corresponde al motor mismo. El motor es la fuente de energía del automóvil. Transforma el calor producido por la combustión de la mezcla gasolina-aire en el interior de los cilindros, en energía mecánica capaz de imprimir movimiento a las

ruedas, mediante un eje llamado cigüeñal que convierte el movimiento alternante de subida y bajada del pistón del cilindro, a movimiento rotatorio. En la siguiente figura se pueden apreciar todos estos componentes.

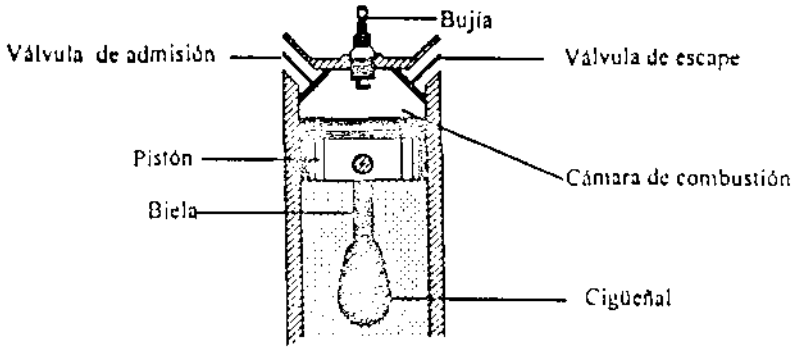


Figura. 3.1 Cilindro de motor.

Este es un cilindro visto transversalmente. Lo que se señala como cigüeñal es en realidad un eje al que se sujetan los demás cilindros del motor, como se observa en la figura 3.2.

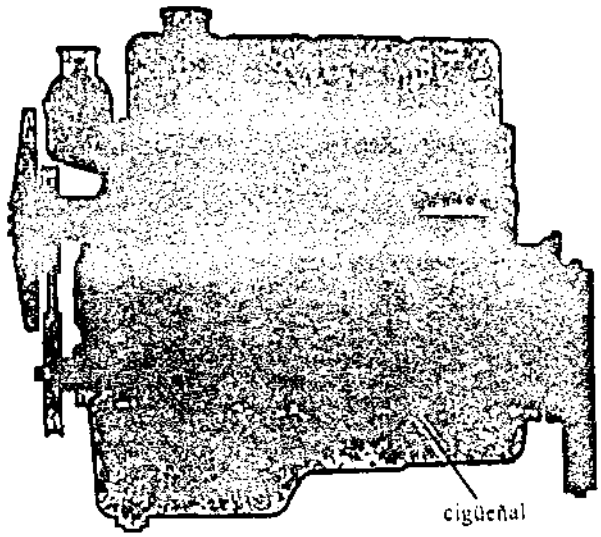
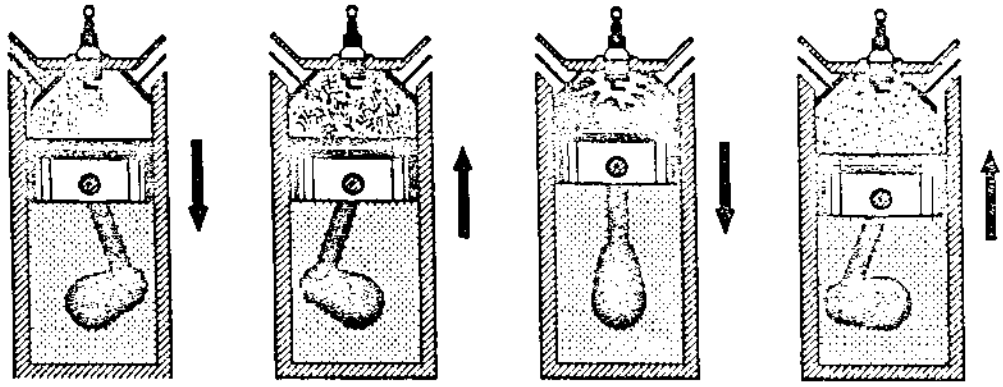


Figura 3.2 Motor de cuatro cilindros.

Todo el proceso que se describió anteriormente se efectúa en cuatro tiempos, como se observa en la siguiente figura.



1	2	3	4
Tiempo de admisión: La válvula de admisión se abre y la de escape permanece cerrada. El pistón desciende y aspira la mezcla.	**Tiempo de compresión:** Se cierra la válvula de admisión y la de escape continua cerrada. El pistón sube y comprime la mezcla.	**Tiempo de explosión:** Las dos válvulas permanecen cerradas. El gas comprimido se inflama por la chispa producida por la bujía. Al expandirse, el gas inflamado empuja el pistón, obligándolo a bajar	**Tiempo de escape:** Se abre la válvula de escape, mientras la de admisión permanece cerrada. El pistón sube y expulsa los gases quemados, al terminar se cierra la válvula de escape e inicia un nuevo ciclo.

Figura. 3.3 Ciclo de operación del cilindro.

Se observa que el movimiento de subida y bajada de los pistones, ocasiona un movimiento rotatorio del cigüeñal por medio de la biela.

El cigüeñal a su vez, mueve a otro eje llamado árbol de levas (esta pieza se puede apreciar en la figura 3.8), que se encarga de accionar las válvulas de admisión y de escape. El paso del movimiento del cigüeñal al árbol de levas se hace por medio de una banda llamada cadena de distribución. Se puede observar que existe retroalimentación a lo largo de todo el proceso.

Todas las partes que se han mencionado son los principales componentes del sistema mecánico. Para su mejor apreciación, se clasifican en dos subsistemas. El **subsistema motor** en el que intervienen las diferentes partes encargadas de la generación del movimiento, como los pistones, las bielas y el cigüeñal. Y el **subsistema auxiliar** en el que se consideran todas las demás partes que contribuyen a generar el movimiento, como las válvulas, el árbol de levas, la cadena de distribución, entre otros.

Para que el sistema mecánico pueda operar, se necesita de la intervención de los demás sistemas (ver figura 3.4), que le proporcionarán los elementos necesarios para su funcionamiento.

Como se observó anteriormente, el movimiento descendente del pistón es ocasionado por la explosión, dando movimiento rotatorio al cigüeñal, con el que se podrá efectuar posteriormente el movimiento ascendente del pistón de otro cilindro, dando origen a un movimiento cíclico. Sin embargo, para que empiece todo el proceso se necesita de un movimiento inicial, que es proporcionado por el *sistema eléctrico*.

Gracias al *sistema eléctrico* se logra el arranque del motor. Para ese momento el *sistema de alimentación de combustible* ya hizo su trabajo preparando la mezcla gasolina-aire y entregándola a las cámaras de combustión. Cuando la mezcla está comprimida, el *sistema de encendido* se encarga de generar la chispa en el momento preciso, para dar lugar a la explosión.

Todo este movimiento que se genera con base en la combustión, produce altas temperaturas en el motor. El *sistema de enfriamiento* se encarga de disipar gran parte de ese calor, mientras que el *sistema de lubricación* pone una capa de aceite a las partes móviles para impedir la fricción entre los metales y evitar desgaste o daños mayores.

De los gases que se generan por la combustión, se encarga el *sistema de escape*, expulsándolos hacia el medio ambiente o reciclándolos. Todas estas acciones son controladas por la computadora del automóvil gracias al *sistema de control*.

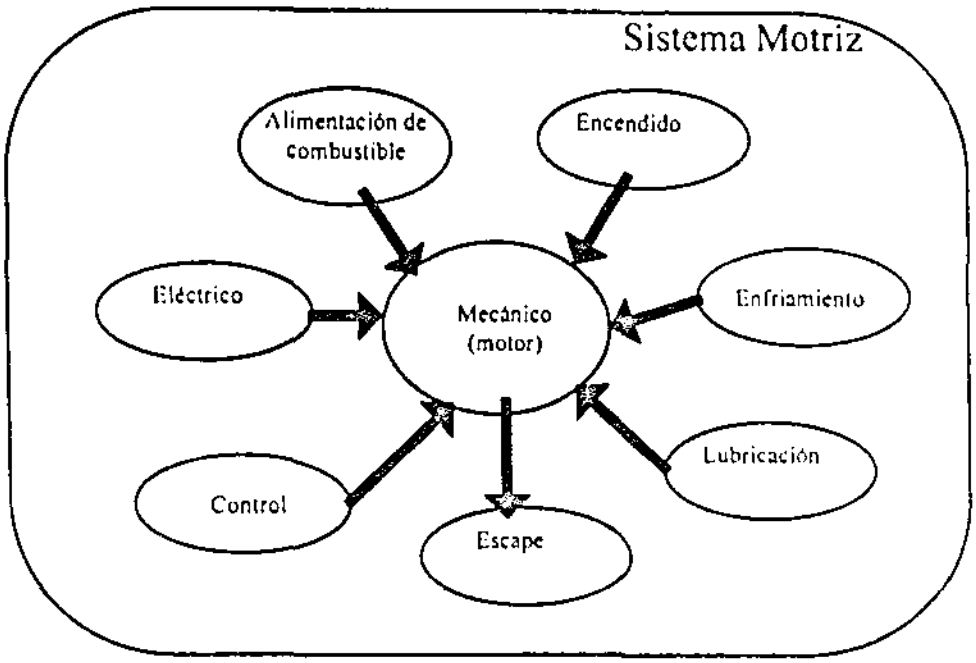


Figura 3.4 Sistemas que componen al sistema motriz.

Se ha visto la relación que hay entre el sistema mecánico y los demás que componen al sistema motriz. Ahora se explicará con más detalle cada uno de ellos.

El ***sistema de alimentación de combustible*** almacena la gasolina en forma líquida en un depósito de combustible, la bombea hasta el riel de inyectores y luego la mezcla con el aire que entra a través de los inyectores. Ésta es la mezcla que entra a los cilindros del motor durante la fase de admisión, donde es comprimida y encendida.

El sistema de alimentación de combustible debe variar las proporciones de aire y gasolina según las diversas condiciones de funcionamiento. Para un funcionamiento normal con un motor calentado, la proporción adecuada de la mezcla es aproximadamente 15 libras de aire por 1 libra de gasolina. Pero para el arranque inicial con un motor frío se necesita una mezcla mucho más rica; la mezcla no arde tan fácilmente en un motor frío, ni la gasolina se convierte en

vapor tan rápidamente. También cuando se acelera o durante el funcionamiento a alta velocidad o con carga completa (en alta) se requiere una mezcla mucho más rica, entendiendo por riqueza, mayor proporción de gasolina.

Entonces, este sistema es el encargado de suministrar una mezcla adecuada a los cilindros del motor por medio de los inyectores. Para hacer esto se divide en tres subsistemas:

- Bombeo y conducción
- Admisión de aire
- Inyección de combustible

El *subsistema de bombeo y conducción* bombea la gasolina y la lleva hasta el riel de inyectores, utilizando una bomba eléctrica y tuberías. La bomba eléctrica aspira del tanque más gasolina de la que se precisa inyectar y el sobrante vuelve al tanque a través de un regulador de presión, que evita la posibilidad de que se formen bolsas de aire y vapor de gasolina.

Al mismo tiempo, el subsistema de *admisión de aire* obtiene aire del medio ambiente pasándolo por un filtro. La cantidad de aire que entra depende de qué tanto esté acelerando el conductor, porque el acelerador cambia la posición de una mariposa que se interpone en el flujo del aire. Entonces, mientras más se acelere, la mariposa deja pasar más aire.

Este subsistema también se encarga de tomar algunas mediciones, por medio de algunos sensores (sensor de flujo de aire, MAF; sensor de presión absoluta, MAP; sensor de temperatura del aire, MAT; y sensor de posición del acelerador, TPS) y otros dispositivos (básicamente válvula IAC, control de aire en marcha ideal), que le permitirán a la computadora del automóvil saber las condiciones del motor y actuar de acuerdo a ellas.

Una vez que se tiene el aire y la gasolina, el *subsistema de inyección* permite que la gasolina que se encuentra presurizada en la parte superior de los inyectores, sea atomizada por la parte inferior y entre, junto con el aire en forma de spray, a la hora que se abre la válvula de admisión. La cantidad de gasolina que los inyectores entregan depende del tiempo que se mantengan abiertos. La computadora del automóvil (ECM) se encarga de indicar el momento en que se deben abrir y cerrar los inyectores. El cálculo del tiempo que se mantienen

abiertos lo hace a partir de la información proporcionada por el subsistema de admisión de aire.

En la siguiente figura se observan los componentes principales del sistema de alimentación de combustible.

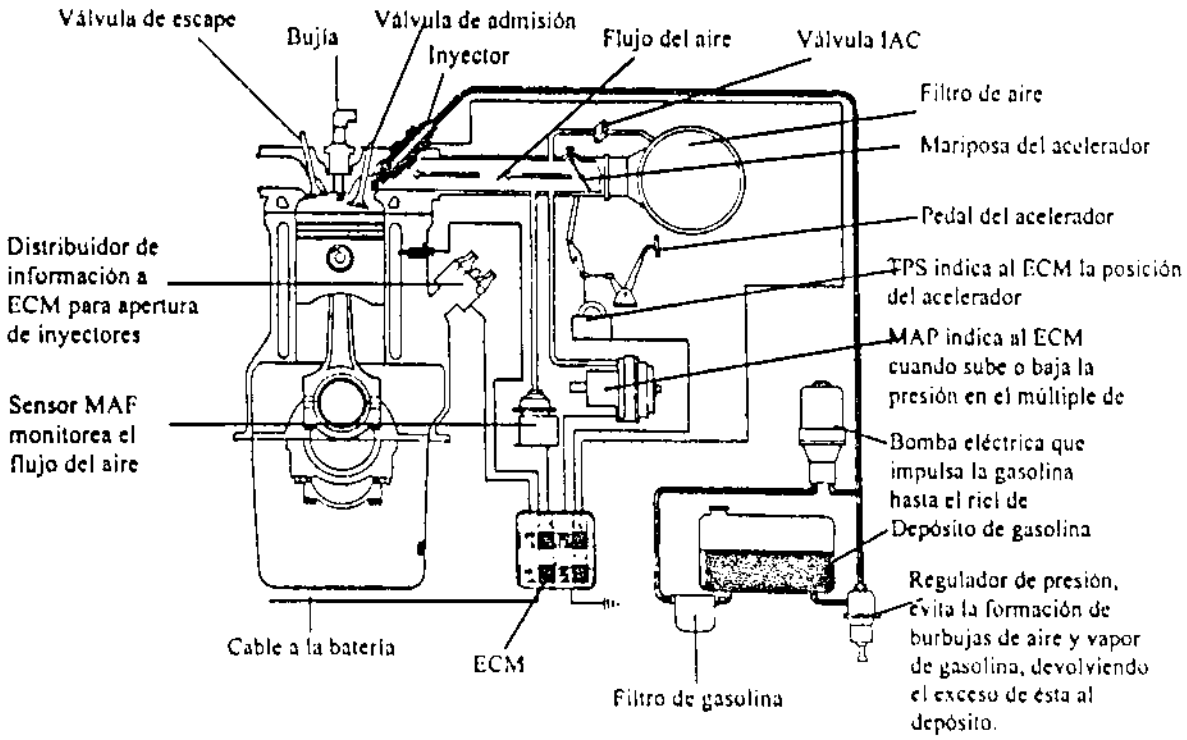


Figura 3.5 Sistema de alimentación de combustible

El ***sistema de encendido*** tiene como propósito producir pulsos de alto voltaje y enviarlos a la bujía indicada, en el momento preciso, para la generación de la chispa que ocasiona la combustión (mientras el switch de encendido mantenga el circuito cerrado de la batería al sistema).

Teóricamente, la chispa eléctrica debe generarse cuando el pistón está en el PMS (es decir la mezcla totalmente comprimida), pero en la práctica no es así, dado que al ser muy rápido el giro del motor, el pistón permanece una fracción de

tiempo muy pequeña en este PMS. Entonces el sistema de encendido debe ser capaz de mandar la chispa en el momento oportuno para evitar problemas, como picado de biela, que se presenta cuando la chispa se produce en el tiempo de compresión; o falta de potencia, que se presenta cuando la chispa se produce en el momento de descenso del pistón (ver figura 3.2).

El sistema de encendido realiza su labor mediante los tres subsistemas que lo componen, que son:

- Toma y elevación de voltaje
- Distribución
- Generación de chispa

El *subsistema de generación de chispa* está formado por las bujías y el cableado que les suministra voltaje. Las bujías son las encargadas de producir la chispa y se encuentra una en cada cilindro. Están compuestas por dos electrodos (elementos metálicos) que se alojan en la cámara de combustión. Cuando a la bujía llega un voltaje lo suficientemente elevado, la corriente salta entre los electrodos en forma de chispa ocasionando la combustión

El voltaje que se requiere para generar la chispa debe ser de por lo menos 14,000V pudiendo llegar hasta 30,000. La fuente de energía eléctrica que se emplea, es la batería del automóvil, la cual entrega de 6 a 12V. Es entonces donde interviene el *subsistema de toma y elevación de voltaje*, que adquiere el voltaje de la batería y lo eleva varios miles de veces por medio de una bobina.

Una vez producido el alto voltaje, debe ser distribuido adecuadamente hasta la bujía, en el momento indicado del ciclo de cuatro tiempos (fig. 3.2), para evitar los problemas antes mencionados. De esto se encarga el *subsistema de distribución*, transmitiendo la electricidad por turno a cada una de las bujías mediante el modulo DIS (Sistema de Ignición Directa por sus siglas en inglés *Direct Ignition Sistem*) para que la generación de chispa se haga justo en el momento oportuno.

El ***sistema eléctrico*** se encarga de abastecer de energía eléctrica al automóvil en dos aspectos. El primero para el arranque del motor y el segundo para los servicios del automóvil, como luces, radio, tablero, etc. Esto lo hace por medio de los tres subsistemas que lo componen:

- Alimentación y generación de energía eléctrica
- Arranque
- Alimentación y servicios

El *susbsistema de alimentación y generación de energía eléctrica*, se encarga de producir y acumular la energía eléctrica que soliciten los demás subsistemas. Está compuesto básicamente de dos partes, la batería, que acumula toda la energía eléctrica de la que se alimentan los demás componentes, y el alternador, que produce energía mecánicamente y restablece la que ya ha sido gastada mandándola a la batería.

En el sistema mecánico se explicó el funcionamiento de los pistones, y se dijo que para que pudiera empezar todo el ciclo se necesitaba de energía inicial. Esta energía proviene del motor de arranque, que es un motor eléctrico alimentado por la batería, que hace girar al cigüeñal imprimiendo un movimiento de acenso y descenso a las bielas y estas a su vez a los pistones. Una vez que se proporciona este movimiento inicial, el motor de arranque deja de mover al cigüeñal para que continúen con esta labor los pistones por sí solos. Ésta es la función del *subsistema de arranque*.

Cuando el motor ya está funcionando mecánicamente, la energía eléctrica se dedica a dar servicio al sistema de encendido, a la computadora, al tablero, luces etc. . El *subsistema de alimentación y servicios* se enfoca en las partes que requieren de energía eléctrica que no están contempladas dentro de otros subsistemas.

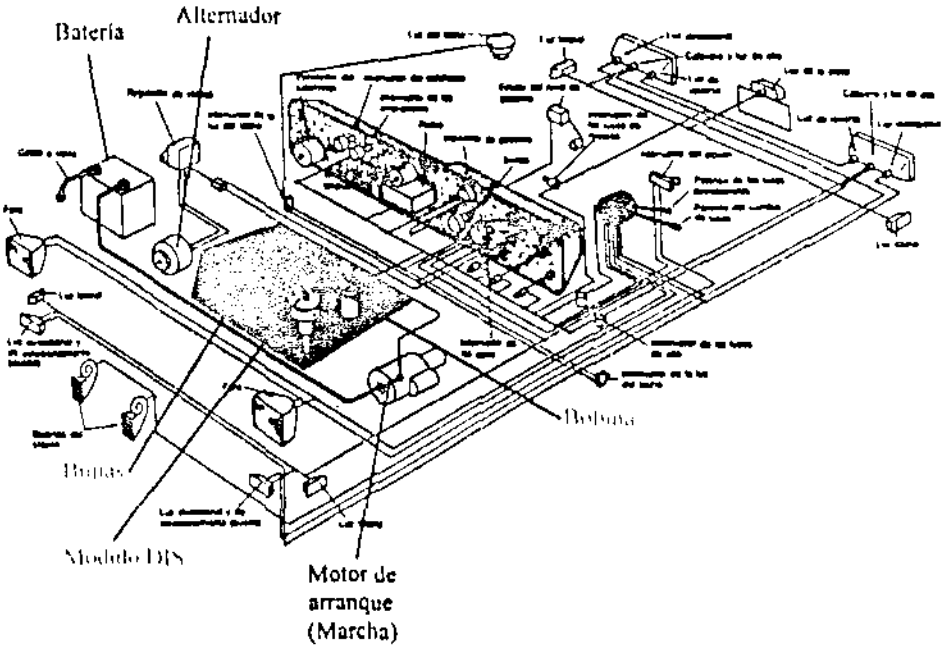


Figura 3.6 Sistema de encendido y sistema eléctrico

En la figura anterior se muestra las partes principales del sistema de encendido delimitado por la región gris. El resto del esquema muestra a la mayoría de los componentes que interviene en el sistema eléctrico.

El ***sistema de enfriamiento*** (figura 3.7) está diseñado para disipar el calor del metal que rodea a la cámara de combustión. El calor desarrollado por la combustión puede ascender a más de 1000°C y aproximadamente la tercera parte de está energía calorifica se disipa por el sistema de enfriamiento. Si este calor no fuera disipado, las partes metálicas se expandirían y el aceite lubricante se quemaría. Cualquiera de esas condiciones causaría excoriaciones en el pistón y en los cilindros, y por consecuencia, el motor se pegaría.

El sistema de enfriamiento está compuesto por dos subsistemas:

- Bombeo y circulación del refrigerante
- Medición de temperatura y enfriamiento del refrigerante

El subsistema de bombeo y circulación del refrigerante, se encarga de distribuir al refrigerante por conductos internos del motor para que el calor se disipe, utilizando una bomba que fuerza la circulación del refrigerante a través del motor. Existen dos tipos de sistemas de refrigeración , el que es a base de aire y el que es a base de líquidos. El más común es el segundo, un líquido especial, que por sus propiedades permiten más fácilmente la disipación de calor.

Del monitoreo y control de la temperatura se encarga el *subsistema de medición de temperatura y enfriamiento del refrigerante*. Mediante un termostato, reduce la circulación del refrigerante hasta que el motor adquiere su temperatura normal de funcionamiento.

Con el sensor CTS (sensor de temperatura del refrigerante - *Coolant Temperature Sensor*), mide la temperatura e informa a la computadora para que pueda calcular la entrega de gasolina necesaria para mantener el motor andando a cualquier temperatura. También manda una señal al alcanzar los 110 °C para que se active el motoventilador (ventilador de enfriamiento eléctrico) que impulsa aire hacia el radiador. El radiador es como un panal por el que atraviesa el aire y enfría al refrigerante, que llega caliente desde el motor y que está circulando por él.

Con estos dispositivos se logra mantener la temperatura del motor dentro de un rango que le permita funcionar correctamente.

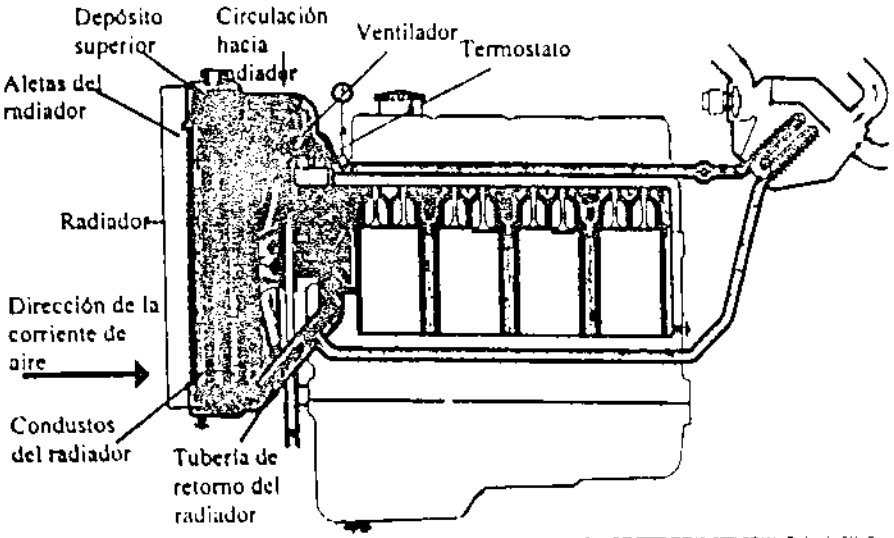


Figura 3.7 Sistema de enfriamiento

El ***sistema de lubricación*** (figura 3.8) ayuda a disminuir la fricción entre las partes móviles del motor, poniendo una pequeña película de aceite entre sus superficies. Las moléculas de un lubricante son muy pequeñas, flexibles y resbalosas; sin embargo se adhieren a la mayoría de las superficies. La película lubricante actúa como una capa de pelotitas que impide el contacto real entre las dos superficies metálicas.

Los aceites lubricantes del motor tienen cuatro funciones: Evitar el desgaste entre dos piezas móviles, ayudar a eliminar el calor del motor, limpiar las piezas del motor al lubricarlas, y formar un sello entre los anillos del pistón y las paredes del cilindro, para evitar que pasen a la parte inferior del motor (cárter) los gases de la combustión.

Si las piezas móviles fueran privadas de aceite se desgastarían o pegarían después de un período muy breve de funcionamiento del motor.

El sistema de lubricación está compuesto por dos subsistemas:

- Bombeo y conducción del lubricante
- Filtrado del lubricante.

Todos los componentes del sistema mecánico se encuentran dentro de un bloque de metal (monoblock), que es lo que se conoce como motor. La parte más baja de este bloque es el cárter y es aquí donde se encuentra almacenado el aceite.

A lo largo de todo el monoblock se encuentran pequeñas perforaciones que funcionan como venas por las que debe circular el aceite. Como el aceite está concentrado en la parte inferior, éste debe ser impulsado a presión por medio de una bomba y circular a través de las venas para lubricar las piezas del motor; el *sistema de bombeo y conducción del lubricante* realiza esta tarea.

La bomba impulsa varios litros de aceite por minuto, a una presión controlada por una válvula, pero antes de que lo expulse, lo hace pasar por un filtro que retiene las partículas más gruesas para conservar al aceite en buenas condiciones para su flujo. El subsistema de *filtrado del lubricante* se encarga de esta función.

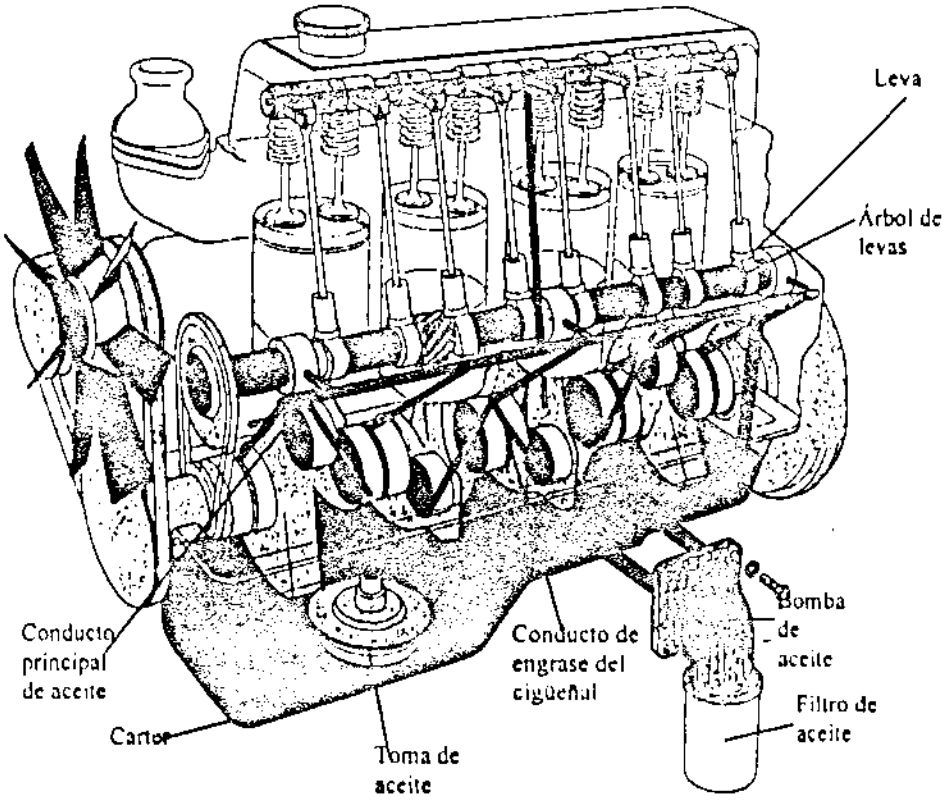


Figura 3. 8 Sistema de lubricación

El ***sistema de escape*** se encarga del manejo de los gases producidos a consecuencia de la combustión y se divide en dos subsistemas que son:

- Expulsión de gases
- Reciclado de gases

El subsistema de *expulsión de gases* tiene como función conducir los gases residuales calientes, producidos en el motor, hasta un lugar en el que puedan ser eliminados a la atmósfera sin peligro para los ocupantes del automóvil, pues estos gases contienen monóxido de carbono, que es inoloro y venenoso, y bióxido de carbono que puede causar asfixia.

Los gases expulsados por cada cilindro se expanden con gran fuerza y pasan a los colectores de escape que los guiarán hacia una salida. Si los gases no se eliminan con facilidad, se obstruirá la entrada de la mezcla gasolina-aire a las cámaras de combustión, además de que resultará contaminada por los gases residuales y se disminuirá el rendimiento del motor. Por eso los colectores de escape se proyectan de modo que no se interfieran los gases de escape que, por turno, salen de cada cilindro.

Cada vez que pasan gases al colector de escape (miles de veces por minuto), se forma una onda expansiva. Esta serie de ondas debe ser amortiguada, de lo contrario el ruido del motor sería inaceptable, para esto se emplea un dispositivo que frena las ondas de expansión y absorbe el ruido. Así, la misión del subsistema de expulsión de gases consiste en conducir los gases hacia el medio ambiente, obstaculizando lo menos posible su flujo y silenciando el ruido de escape.

Por otro lado, algunas veces la mezcla gasolina-aire resulta ser muy rica y parte de los gases que se expulsan pueden ser reutilizados para la combustión. El subsistema de *reciclado de gases* se encarga de capturar esos gases y conducirlos nuevamente a las cámaras de combustión para volver a ser quemados.

La gasolina que se emplea para la mezcla gasolina-aire, está contenida en un tanque. Con el movimiento del automóvil y el calor externo, esta gasolina empieza a evaporarse generando gas. El subsistema de reciclado de gases también se encarga de capturar este gas y llevarlo hacia las cámaras de combustión. Entonces, este subsistema permite aprovechar al máximo los elementos que contribuyen a la combustión.

En la siguiente figura se muestran el sistema de escape.

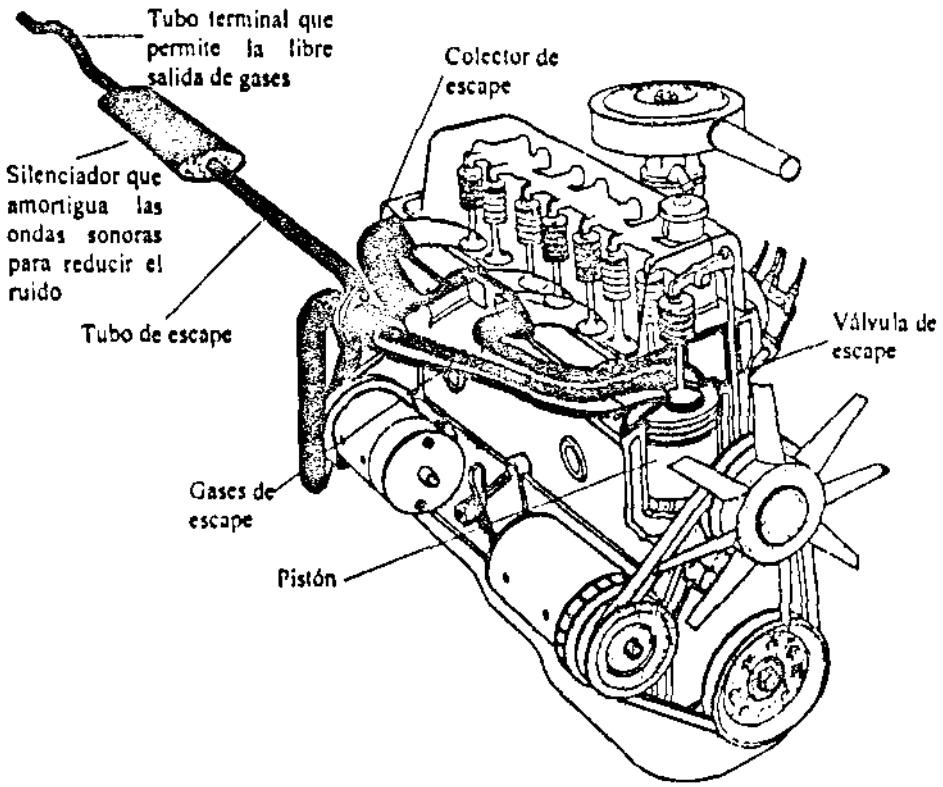


Figura 3.9 Sistema de escape

El ***sistema de control*** se encarga de monitorear todo el funcionamiento del automóvil por medio de dispositivos electrónicos (sensores). Todas las señales emitidas por estos sensores van a parar a la computadora del automóvil o ECM (Engine Control Module).

Asi, el sistema de control se divide en dos subsistemas:

- Sensores diversos
- Computadora (ECM)

Existen alrededor de 20 sensores, muchos de ellos se encuentran dentro de los otros subsistemas, porque realizan tareas clave dentro de ellos y si se clasificaran dentro del subsitema de *sensores diversos*, se perdería la secuencia del funcionamiento. Sin embargo los demás sensores que realizan tareas generales se encuentran dentro de este subsistema. Ejemplo de ellos son: el switch de presión de dirección hidráulica, el switch de palanca de cambios, el switch de freno, el switch de aire acondicionado y el sensor de velocidad. Todos estos proporcionan información a la computadora para que realice funciones determinadas, pero principalmente para que haga un mejor cálculo de las proporciones que debe llevar la mezcla gasolina-aire.

El subsistema de computadora, es el ECM en sí. El ECM es el cerebro de todo el sistema electrónico de inyección de gasolina (tecnología Fuel Injection). Está compuesto por un microprocesador que hace los cálculos y toma las decisiones y el resto de la computadora son básicamente memorias (RAM, ROM y PROM). Todos los sensores mandan la información a la computadora y le permiten determinar qué acción tomar para controlar la inyección y otras funciones del sistema motriz.

El ECM produce señales eléctricas que indican a los actuadores qué acción tomar. Los principales actuadores del sistema motriz son: inyectores, relevadores y válvula IAC.

Es de esta manera como los distintos subsistemas trabajan conjuntamente para dar mejor funcionalidad al automóvil.

Como se mencionó en el capítulo uno, para el desarrollo del prototipo se considerarán solamente, el sistema de alimentación de combustible, el sistema de encendido, el sistema eléctrico y el sistema de enfriamiento de un automóvil General Motors con tecnología *fuel injection*, por ser uno de los automóviles más completos en su composición.

3.1.2 Conceptualización de una falla automotriz

Al hacer un diagnóstico, se busca encontrar una falla a partir de los síntomas que se presentan. Ésta es la meta del mecánico automotriz al recibir un automóvil en mal estado, pero, ¿qué significa falla?

Cuando un automóvil presenta comportamientos fuera de lo normal en su funcionamiento, decimos que tiene una falla. En una **falla** están involucrados dos conceptos: la parte y el problema.

La **parte** es la pieza o el componente del automóvil, elemento de algún subsistema, que puede presentar un problema.

Un **problema** es el que origina el mal funcionamiento en determinada parte. De esta manera podemos asociar a cada parte uno o varios problemas que puede presentar, para dar origen a fallas.

Por lo tanto, una **falla** es la conjunción de una parte con su problema. Por ejemplo, si tenemos que la falla es: *Cedazo de la bomba tapado*, sabemos que la parte es el cedazo y el problema es que está tapado.

La falla es el elemento principal sobre el que trabajará el Sistema Experto para Diagnóstico Automotriz. Éste será su objetivo, detectar la falla o las fallas que tiene un automóvil. Obviamente no todas las partes presentan los mismos problemas, por lo que es necesario hacer un análisis de cada parte para ver qué problemas puede presentar y saber las fallas a las que deberá llegar el Sistema Experto.

3.2 Conceptualización del Sistema Experto

Las herramientas que el ser humano crea para ayudarse en sus labores cotidianas, tienen como principal objetivo disminuir esfuerzos al realizar su trabajo. Un sistema experto no es la excepción. En este caso se requiere de un sistema que emule el comportamiento de un mecánico experto al diagnosticar. Entonces, el sistema experto será de diagnóstico, es decir que identificará el origen del mal funcionamiento del automóvil a partir de síntomas .

A este *Sistema Experto para Diagnóstico Automotriz* se le asignará el nombre de *SEDA*. Su objetivo, es localizar de la manera más eficiente la falla que tiene un automóvil. Es decir, dar un diagnóstico certero imitando al experto, por lo que la participación de éste durante su desarrollo será indispensable.

Para dar origen a *SEDA*, es necesario analizar primero cuál es el comportamiento del experto para realizar un diagnóstico y posteriormente ver cómo lo hará el Sistema Experto.

3.2.1 Metodología del experto para diagnosticar

El experto no siempre sigue los mismos pasos para dar un diagnóstico. Su experiencia y conocimientos lo hacen llegar a una conclusión de una forma rápida, y a veces es difícil estructurarla. Sin embargo, se observó que de alguna manera, siempre efectúa los siguientes pasos.

Primero, al recibir un automóvil, verifica el año y la marca, esto es importante para él porque en cuanto empiece a pedir información al conductor, sabe de qué automóvil se trata y sabrá qué medidas tomar.

Posteriormente, pide al conductor que le indique los síntomas que presenta el automóvil y hace preguntas para irse enfocando al problema. Por ejemplo, si estos se dan al arrancar, al apagar, al estar operando, en frío o en caliente, etc.

A lo largo de todo el diagnóstico el experto puede hacer dos tipos de mediciones, las generales y las particulares. Las primeras le indican en términos generales el comportamiento del automóvil; éstas son proporcionadas por un analizador de gases, que le indica cómo se está llevando a cabo la combustión, y el scanner, que se conecta a la computadora del automóvil e identifica las fallas presentadas por los sensores, traduciéndolas a un código de falla. Las segundas son mediciones que hace por medio de un multímetro y otros instrumentos en partes específicas.

Después de adquirir los síntomas, se forma una idea de dónde puede estar la falla y, si lo cree necesario, toma mediciones generales. En algunas ocasiones, estas mediciones las toma cuando los síntomas por sí solos no le proporcionan la información suficiente y requiere de más datos, como las emisiones de gases y los códigos de falla. Cabe recordar que estos últimos no siempre son certeros y solamente están relacionados con los sensores, por lo que no es suficiente esta información para llegar a un diagnóstico.

Los datos obtenidos tanto del analizador de gases como del scanner, los compara con una tabla de emisiones y con una de códigos de falla, respectivamente, para sacar sus primeras conclusiones.

Toda esta información le ayuda a enfocarse solamente a una zona del automóvil (lo que conocemos como subsistema), y empieza a hacer mediciones más específicas empleando otros instrumentos, como son manómetro, voltímetro, amperímetro, etc., sobre las partes en las que él considera que pudiera estar el problema. Algunas mediciones lo llevan a hacer otras, y así va tomando una serie de mediciones que concluirán en la identificación de la falla.

En ciertos casos no necesita tomar muchas mediciones, con el simple comportamiento del auto sabe dónde está la falla, pero esto depende de su experiencia.

Las mediciones que hace lo ayudan a tomar un camino. Por ejemplo: si se enfoca en una parte y ésta marca una medición fuera de los parámetros normales (primera medición), y su buen funcionamiento depende de otra, esto lo obliga a hacer una medición en esa otra parte (segunda medición), para ver si el fallo está en la primera o su mal funcionamiento es consecuencia de la otra. Por otro lado, si la primera medición está correcta se olvida de esa parte y por consiguiente de la otra, es decir, ya no necesita hacer la segunda medición y pasa a hacer una nueva. De esta manera, no es necesario que revise todas las partes asociadas a un subsistema, si no que va escogiendo caminos por los que lo llevan las mediciones, hasta encontrar la falla y proporcionar un diagnóstico.

Entre más experiencia tenga el experto, más rápida será su búsqueda, debido a que puede ir acortando camino.

Todo este proceso de diagnóstico es muy similar al de un médico. Cuando se asiste con un médico lo primero que él hace es pedir información que le permita saber de qué paciente se trata, como la edad, el sexo (en un auto sería el año y la marca). Posteriormente pide al paciente que le indique cuáles son sus malestares o síntomas (en el caso del auto, el mecánico pide al automovilista los síntomas).

Después, si lo considera necesario, empieza a pesar al paciente, a medir su temperatura, su presión y su estatura (mediciones generales). Con esta información puede sacar sus primeras conclusiones y, en algunos casos, somete al paciente a análisis o bien lo manda con algún especialista, para adquirir datos más específicos proporcionados por un electrocardiograma, encefalograma, análisis de sangre, entre otros, según sea el caso (mediciones particulares). Finalmente, después de todos los estudios necesarios indica al paciente cuál es su padecimiento.

Esta comparación de un diagnóstico automotriz con un diagnóstico médico, permite tener más claro todo el proceso de diagnóstico que sigue el experto. A continuación se presenta de forma gráfica está metodología:

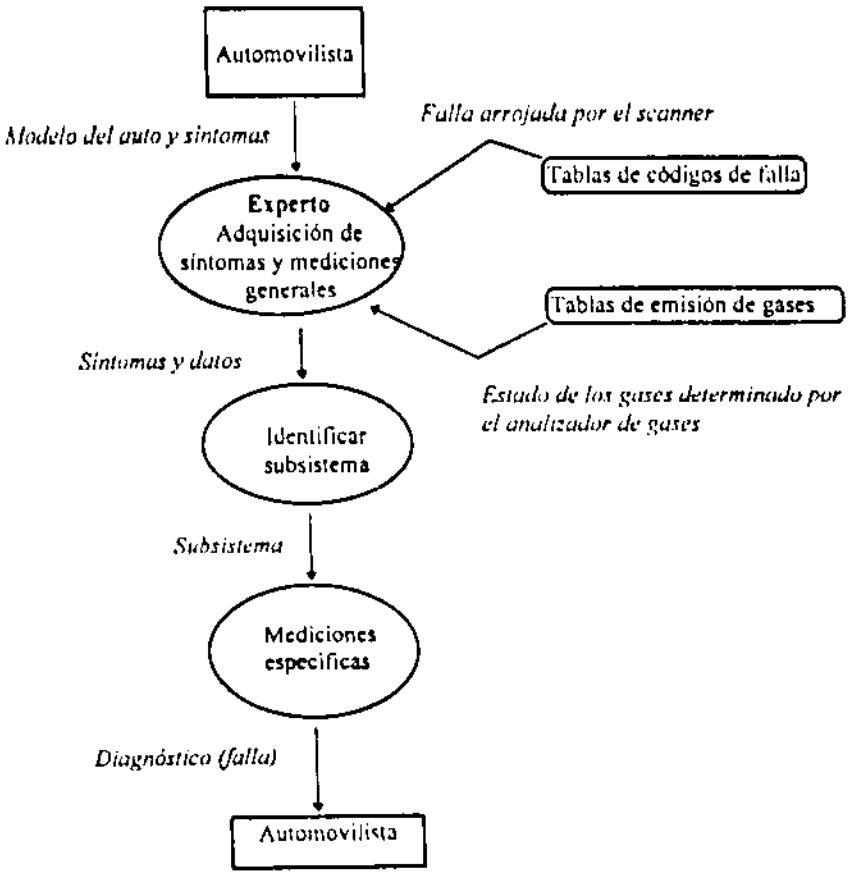


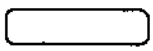
Figura 3.10 Diagrama de flujo de datos en el diagnóstico del experto.

El diagrama muestra el flujo de datos durante todo el proceso de diagnóstico del experto, separándolo en pequeños procesos. La simbología empleada es la siguiente:

 Representa al origen y destino de la información.

 Representa a los procesos independientes que transforman datos de entrada en datos de salida.

 Representa a los datos que fluyen durante el proceso.

 Representan un lugar físico para obtención o almacenamiento de datos.

3.2.2 Método de diagnóstico del Sistema Experto

Partiendo de la metodología que emplea el experto para diagnosticar, podemos darnos cuenta que lo hace acotando el problema. Primero a través de síntomas y algunas mediciones generales, se enfoca en la zona o subsistema en el que podría estar el problema y posteriormente hace mediciones más directas en partes específicas. De acuerdo a los resultados obtenidos durante las mediciones, va determinando por dónde esta la falla hasta llegar finalmente a ésta. Esta forma de diagnosticar le permite avanzar rápidamente sin tener que hacer mediciones innecesarias e incluso cambios de partes en buen estado.

El decidir en qué subsistema está la falla y hacer las mediciones correctas, son producto de su experiencia, pero ¿cómo se puede representar todo esto en un Sistema Experto?.

Para empezar es conveniente que, al igual que el experto, se haga el diagnóstico en dos partes a las que se llamarán **niveles**, el primero se encargará de identificar en qué subsistema está la falla y el siguiente, partiendo de los resultados del primer nivel, determinará cuál es ésta.

Primeramente el usuario indica la marca del automóvil, con esto el Sistema Experto identificará la base de conocimiento que deberá usar. Posteriormente indica el año del automóvil, con lo que obtiene los datos necesarios de las bases de datos.

Posteriormente, el usuario del SE introducirá los síntomas y mediciones generales. Para esto, el usuario seleccionará de una lista (que mostrará el SE) los síntomas que presenta el automóvil, los cuales están clasificados por el momento en que se presentan, para hacer más fácil su identificación. El SE también dará la opción de capturar los códigos de falla arrojados por el scanner y las lecturas del analizador de gases, que comparará con datos almacenados en una base de datos. Entre más datos se le proporcionen, más certero podrá ser el primer diagnóstico.

La base de conocimiento del SE estará formada por reglas construidas por los síntomas, y las fallas serán sus conclusiones. Las reglas que contengan a los síntomas que presenta el automóvil y cumplan con los datos adecuados en las mediciones generales (analizador de gases y códigos de falla) se convertirán en verdaderas.

Cada regla verdadera tendrá asociado un factor de confiabilidad (qué tan posible es que se de esa falla), que mediante una suma aritmética nos ayudará a calcular un factor de peso para cada subsistema, el cual determinará en qué subsistema hay mayor posibilidad de falla.

Así, el resultado final del primer nivel de diagnóstico, consiste en una calificación de cada uno de los dieciséis subsistemas que integran al sistema motriz del automóvil. El que tenga la calificación más alta será el de mayor posibilidad de falla.

El objetivo del primer nivel es acotar la gama de posibilidades y enfocarse sólo a la que sea más probable; con esto se evitará pedir que se hagan un gran número de mediciones, de las cuales muchas serían innecesarias y retardarían el proceso de diagnóstico.

Con base a la calificación obtenida, el usuario elegirá al subsistema con mayor puntuación. En este momento se inicia el segundo nivel de diagnóstico. En él se establece una secuencia de mediciones de acuerdo al subsistema en cuestión. Si en la primera medición se ubica a la falla, el diagnóstico termina. En caso contrario se pasa a la siguiente medición hasta que se determine la falla.

En algunos casos se presentará más de una falla, pero será porque son consecuencia de otras. Por ejemplo, puede decir "la bomba no funciona, porque el cedazo está tapado", podemos ver que son dos fallas pero la primera es consecuencia de la segunda.

Volviendo a la comparación con un diagnóstico medico, el primer nivel de diagnóstico de SEDA, equivale a que el médico identifique qué tipo de estudios necesita el paciente, porque no lo somete a todos los estudios existentes cuando por los síntomas, sabe que puede ser un problema cardíaco. De la misma manera que en el primer nivel de diagnóstico, el médico determina el subsistema de mayor posibilidad de ubicación de la enfermedad.

De la misma manera, el primer nivel determina primero el subsistema.

Después de los primeros estudios, si lo necesita, el médico somete al paciente a más estudios hasta que está seguro de su padecimiento. De la misma manera el segundo nivel de diagnóstico pide mediciones específicas hasta encontrar la falla.

Gráficamente el proceso de diagnóstico de SEDA es el siguiente:

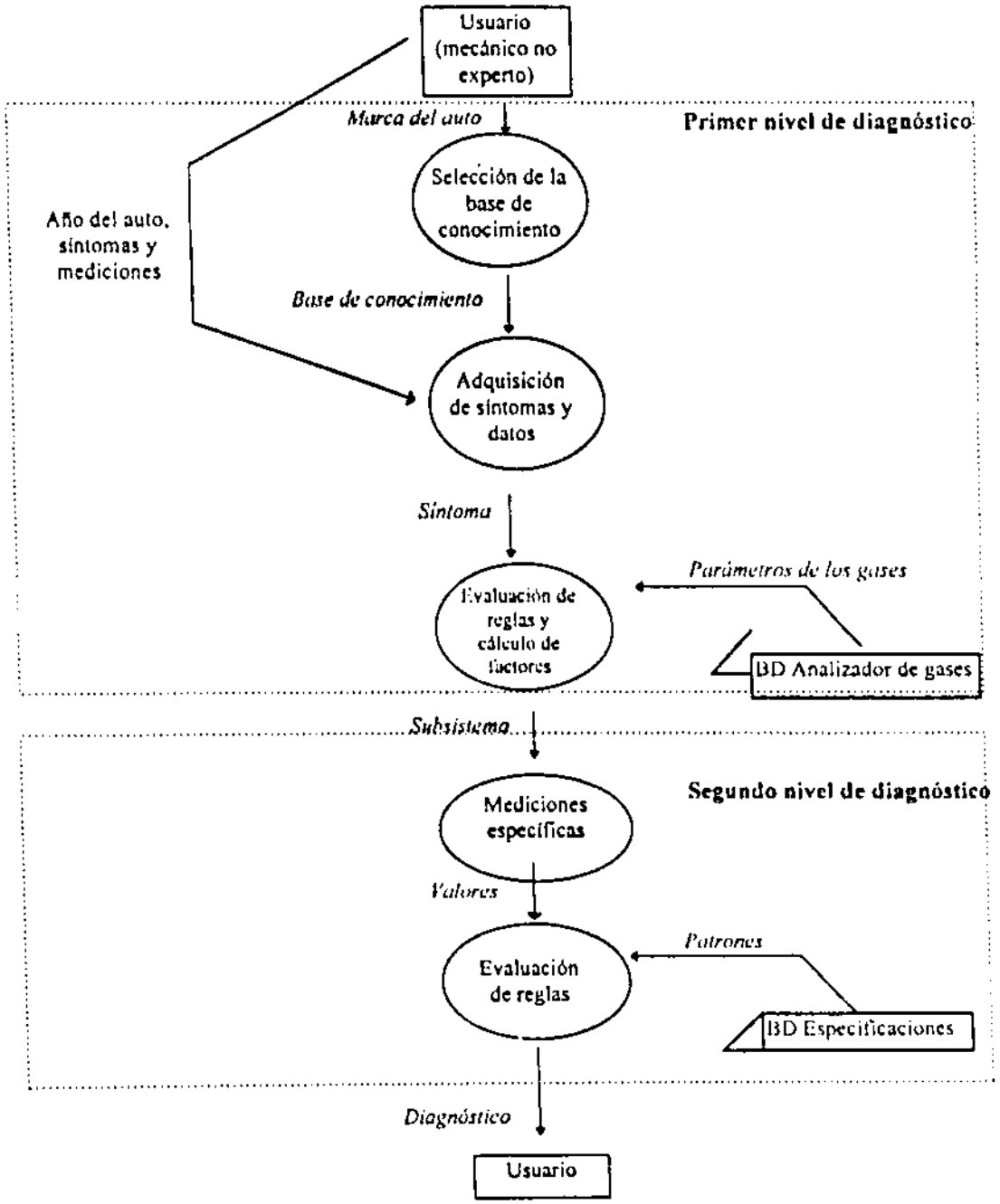


Figura 3.11 Diagrama de flujo de datos en el diagnóstico del Sistema Experto.

Este diagrama muestra cuál será el flujo de información, durante el proceso de diagnóstico con el Sistema Experto. Tiene la misma simbología que el de la figura 3.10, con la diferencia de que el lugar físico de almacenamiento es ahora una base de datos.

CAPÍTULO 4: DISEÑO DEL SISTEMA EXPERTO.

En este capítulo se definirán los elementos que intervienen en el diseño de SEDA. Se establecerán las características de la fuente de conocimiento y del software a utilizar. Posteriormente, se señalarán los métodos de inferencia y su utilización dentro de SEDA.

Finalmente se planteará la estructura de objetos y reglas de producción que se empleará para la creación de la base de conocimientos del primer nivel de diagnóstico, así como el manejo de certidumbre.

4.1 Fuente del conocimiento

Un factor muy importante durante el desarrollo de un Sistema Experto, es la fuente del conocimiento, que es de donde se adquiere la información para dar origen a la base de conocimientos. Esta fuente es un ser humano, experto en algún área del conocimiento.

Un experto es una persona práctica, hábil, que tiene experiencia en una ciencia o arte que le permite resolver con facilidad problemas que a la mayoría le resultan complejos y por tanto extraordinarios.

Estas personas se caracterizan porque hay un escaso número de ellas, porque su comportamiento es complejo frente a la manera de resolver los problemas en su área y además, porque necesitan de un largo tiempo de aprendizaje. El tiempo de formación de un experto se suele estimar de cinco a diez años, la forma más rápida de hacerlo es mediante el aprendizaje formal o académico, en un principio, y posteriormente un aprendizaje informal o práctico.

Para el desarrollo de un Sistema Experto debe haber por lo menos una fuente de conocimiento. Una persona que esté dispuesta a colaborar activamente en el proyecto. Lo ideal es que sólo se trate de una, porque en el caso de que hubiera varios expertos, se tendrían varias conclusiones ante una misma situación,

lo que originaría mayor complejidad en la construcción de la base de conocimientos. Sin embargo, muchas de la decisiones que se toman pueden ser desarrolladas por varias personas que son expertas en el área. No considerar las diferentes opiniones sería como ignorar gran parte de la experiencia usada.

La labor de recopilar el conocimiento experto, es una de las tareas más complicadas, porque el experto, al igual que todos los humanos, está sujeto a diversos factores que limitan su disponibilidad. Pueden ir desde una jubilación, hasta cambios en su estado de ánimo que le impiden transmitir su conocimiento de la manera más eficiente. También se debe considerar que la mayoría de los expertos tienden a ser ecléticos en la elección de información para tomar decisiones.

La actitud que tome el experto ante el desarrollo de un SE es crucial para la creación exitosa de la base de conocimientos. Debe mostrar interés y una gran participación para transmitir su sabiduría a los analistas, y que éstos la puedan codificar. Esto depende también de la forma en cómo se adquirirá el conocimiento. Es recomendable la creación de una metodología para hacerlo, la cual se ajustará conforme se vaya avanzando en el proyecto.

Existen varios métodos para obtener el conocimiento experto, entre los cuales se encuentran: la observación de la metodología de trabajo del experto; entrevistas en las que se obtiene el conocimiento mediante el planteamiento de preguntas y problemas; exámenes en los que se prepara una serie de pruebas y problemas para que el experto los resuelva y explique cómo lo ha hecho y con ello obtener los conocimientos, entre otros métodos.

Para la elaboración de SEDA, se cuenta con el apoyo de un ingeniero electrónico experto en la reparación de autos con tecnología *fuel injection*, dispuesto a colaborar durante su desarrollo proporcionando la información requerida mediante sesiones, tanto en el taller mecánico, como en nuestro lugar de trabajo. Es importante remarcar que el experto está interesado en el desarrollo de esta herramienta para sustituir las consultas vía telefónica que constantemente atiende y para su utilización con fines de capacitación.

Las sesiones en el taller mecánico, permiten tener una idea más clara de cómo el experto concibe el problema y así poder interpretar de mejor manera la información proporcionada. Las entrevistas a nivel teoría, son muy útiles para acomodar la información y de alguna manera guiar al experto para que la

proporcione de forma estructurada y así pueda ser codificada más rápidamente. Las sesiones en el taller mecánico fueron mínimas.

En general, la forma en cómo se aprovechó al experto fue la siguiente: primero mediante la observación, se capta la manera en que el experto hace el diagnóstico; posteriormente se diseñan algunos formatos (que se verán con detalle en el capítulo 5.2) para el caso del primer nivel de diagnóstico, y árboles de inferencia para el segundo nivel de diagnóstico como se verá más adelante. Con base en éstos, se van planteando problemas al experto en donde él indica cómo se llega a cada uno de ellos a través de síntomas y mediciones. Se optó por esta metodología después de varias pruebas, y se comprobó que es de esta forma como se puede obtener mejor la información, además de que queda lista para ser implementada mediante un software.

4.2 Software a utilizar

En la actualidad, existe un gran número de herramientas para el desarrollo de Sistemas Expertos, algunas de ellas por sus características, son más adaptables a los problemas que otras. Podemos clasificarlas básicamente en cuatro grupos: los *lenguajes imperativos*, los *lenguajes funcionales*, los *lenguajes declarativos* y los *lenguajes orientados a objetos*.

Los *lenguajes imperativos* son los que permiten el desarrollo de sistemas de una forma procedural, es decir, que el control del programa pasa siempre a la siguiente línea, a menos que se le indique lo contrario. El programador indica en el programa el flujo de la ejecución de las instrucciones. Con este tipo de lenguajes, se crean sistemas muy largos y complejos, por lo tanto, difíciles de modificar. Un ejemplo de este tipo de lenguajes es **C**.

Los *lenguajes funcionales* o *aplicativos* son aquellos en los que se marca el flujo del programa por las necesidades que aparecen al evaluar una función. El programador únicamente tiene que indicar el orden de evaluación de las funciones, sin importar su posición física en el programa. En este tipo de lenguajes, el control de tipo secuencial es sustituido por cuatro pasos: analizar la función a evaluar, buscar la primera subfunción, evaluar la subfunción, retornar a la función anterior o regresar el valor final. Una de las ventajas que presentan

estos lenguajes es que es sencillo construir máquinas de inferencia; sin embargo, tienen la desventaja de que necesitan gran capacidad de memoria y tienen poca integración con otros lenguajes. Un ejemplo de lenguajes funcionales es **LISP**.

Los *lenguajes declarativos* son aquellos en los que hay que indicarle al programa el objeto que se quiere demostrar, especificando en el mismo, el universo sobre el que debe demostrarlo y las reglas que puede utilizar. Es decir, se especifica el "qué" pero no el "cómo". En estos lenguajes el control del proceso lo tiene la máquina de inferencia que es el propio interprete o compilador, por lo que requieren también de reglas de producción. Entre sus principales ventajas se encuentran las siguientes: son lenguajes muy compactos al no necesitar ninguna estructura de control; permiten la inferencia hacia atrás y hacia adelante y cuentan con un sistema sencillo de diálogo. Sin embargo, las operaciones de entrada y salida son complejas de programar, el motor de inferencia es fijo, son poco eficaces sobre arquitecturas tradicionales y poco fiables al no aparecer explícito el comportamiento del programa, porque la máquina de inferencia está de manera implícita. Un ejemplo de este tipo de lenguajes es **PROLOG**.

En los *lenguajes orientados a objetos*, los programas están formados por los objetos, los mensajes y los métodos; esta estructuración es la base fundamental de su gran modularidad. Las acciones que realizan estos programas son en realidad mensajes que se envían los objetos, de tal manera que cada objeto interpreta el mensaje que le llega. El control en este tipo de lenguajes es el siguiente: entran los datos, se proponen una serie de hipótesis, se verifican las hipótesis que pueden ser todas o las que se vayan planteando dependiendo del tipo de inferencia. Una vez terminado el proceso de verificación de hipótesis, se propone una solución con base en éstas. Una de sus principales ventajas, es que este sistema de control es fácilmente convertible a un máquina de inferencia y se trabaja bajo un marco de reglas de producción. Todo esto lo acerca mucho a las necesidades de un SE. Este tipo de lenguaje tiene como principal desventaja que presentan una metodología de programación completamente nueva. **LEVEL 5 OBJECT** es un ejemplo de este tipo de lenguaje.

Los Sistemas Expertos forman parte de los sistemas complejos[2] y una de las formas de enfrentar la complejidad, es por medio de la descomposición en

[2] Los sistemas complejos, son sistemas compuestos por varias partes que interactúan entre sí para cumplir varias funciones y que a su vez están compuestas de otras partes.(Apuntes de Inteligencia Artificial, Dr. Felipe Lara, FI)

subsistemas, la cual puede ser enfocada a las cosas (orientada a objetos) o enfocada a los procesos (algorítmica). La primera, permite un manejo más sencillo y eficiente de estos sistemas, por eso, para el desarrollo de SEDA, se ha optado por hacer un modelado por objetos, el cual se implementará a través de un lenguaje orientado a objetos.

4.2.1 La programación orientada a objetos

La programación orientada a objetos es un método mediante el cual los programas están organizados por colecciones de objetos que interactúan entre sí. Entendiéndose por objeto una forma abstracta de representar a un concepto que interviene en un problema determinado.

Los objetos son agrupados en clases y representan a un caso específico de estas. Las clases forman parte de una jerarquización dada por la herencia.

A partir de esto, se pueden remarcar tres puntos importantes: el primero es que la programación orientada a objetos emplea como estructura lógica fundamental objetos y no algoritmos; el segundo, que cada objeto es una instancia de una clase y, finalmente, que las clases se relacionan unas con otras por medio de la herencia.

La programación orientada a objetos proporciona claridad y flexibilidad esenciales para el desarrollo satisfactorio de sistemas complejos, proporciona entornos en los que los usuarios pueden comunicarse entre aplicaciones y navegar fácilmente por arquitecturas heterogéneas y distribuidas.

Para aquel que no es programador, orientación a objetos significa algo bastante familiar: considerar al mundo como un conjunto de entidades u objetos que están relacionados y se comunican con otro de ellos. Para el desarrollador de sistemas, la orientación a objetos es un nivel de abstracción más allá de los procedimientos y datos.

El hecho de que el tema central sean los objetos, marca una desviación significativa de los anteriores paradigmas de la programación (métodos procedurales), a pesar de ser intuitivo. Las siguientes líneas resumen la diferencia entre la programación procedural y la programación orientada a objetos:

Lea las especificaciones del software que desea construir. Subraye los verbos si persigue un código procedural, o los sustantivos si su objetivo es un programa orientado a objetos.
(Grady Booch, 1989)

Los mecanismos básicos de la programación orientada a objetos son: objetos, mensajes, métodos, clases, instancias y herencia

Un programa orientado a objetos consta solamente de objetos que contienen tanto los procedimientos, como los datos, a diferencia de un programa tradicional que consta de procedimientos y datos. Así, dentro de los objetos residen los datos de los lenguajes convencionales, como por ejemplo, números, matrices, cadenas de caracteres y registros, así como cualquier función, instrucción o subrutina que opere sobre ellos. Desde la perspectiva del programador, los objetos son módulos de una aplicación que interactúan entre sí para proporcionar una funcionalidad general.

Los objetos reciben, interpretan y responden a mensajes procedentes de otros objetos. Un mensaje es una solicitud que pide al objeto que se comporte de alguna forma. El conjunto de mensajes al que un objeto puede responder se le llama protocolo del objeto.

Los métodos residen en el objeto y determinan cómo actúa el objeto cuando recibe un mensaje.

Una clase es una descripción de un conjunto de objetos, conformada por las características (atributos) de éstos. A una ocurrencia de una clase se le denomina instancia y representa a un objeto.

Se ha hablado de tres conceptos claves en la programación orientada a objetos: *objeto*, *clase* e *instancia*, para comprenderlos mejor se analizará el siguiente ejemplo:

Imaginemos un sistema para una biblioteca, en ella hay libros, todos ellos tienen título, autor, tema, editorial y colocación, es decir que todos tienen estos mismos atributos, pueden tomar diferentes valores, pero todos los tienen.

Entonces nuestros objetos son los libros, para representar a estos objetos crearemos una clase que se llamará *libros*, que tendrá los siguientes atributos:

título, autor, tema, editorial, y *colocación.* Los atributos forman la estructura de la clase.

La clase objetos representa a todos los libros en general, pero si queremos hacer referencia a uno en particular, crearemos una instancia de esta clase, que se llamará *libro1*. Esta instancia tiene los mismos atributos de la clase *libros*, pero estos tomarán los siguientes valores:

Título: Cien años de soledad
Autor: Gabriel García Márquez
Tema: Literatura
Editorial: Diana
Colocación: QA76.13

Por cada libro que se desee manejar, se creará una instancia, de tal modo que las instancias representan a cada objeto (cada libro) y la clase representa al conjunto de estos.

Entonces, un objeto es la particularización de una clase (instancia); la clase representa a un concepto real involucrado en un problema específico y contiene los atributos de los objetos asociados a ésta. Es como si la clase fuera el molde o plantilla de los objetos que tienen características comunes.

Otro concepto de la programación orientada a objetos es la *herencia*, este es el término que se emplea para indicar que se pueden compartir de forma automática métodos y datos entre clases. La herencia permite la creación de nuevas clases a partir de una ya existente, agregando únicamente las diferencias; esto da lugar a la existencia de clases padres y clases hijas (subclases).

Regresando al ejemplo, vamos a suponer que existen libros de los cuales nos interesa manejar, además de los atributos de la clase *libro*, el idioma en el que están escritos y el año de publicación. Entonces se crearía una clase llamada *libros extranjeros* que heredaría los atributos de la clase *libros* y se le agregarían dos nuevos. Entonces la clase padre es *libros* y la clase hija es *libros extranjeros*, y quedaría conformada por los siguientes atributos: *título, autor, tema, editorial, colocación, idioma y año.*

Existen dos tipos de herencia: la herencia simple, que es cuando una subclase hereda de una clase, y la herencia múltiple, que es cuando una subclase hereda de más de una clase.

Para poder programar bajo una ideología orientada a objetos, primero se identifican los objetos que aparecen a lo largo del problema y en la solución; posteriormente se clasifican los objetos por sus semejanzas y diferencias (para concebir las clases); después se redactan los mensajes que relacionan a los objetos y finalmente se implantan los métodos o procedimientos en los objetos correspondientes.

La implementación de SEDA, se hizo con LEVEL5 OBJECT, un lenguaje orientado a objetos, que proporciona facilidades para el desarrollo de SE. A continuación veremos algunas de las ventajas que éste nos ofrece.

4.2.2 Level5 Object

LEVEL5 OBJECT es un ambiente orientado a objetos para el desarrollo de Sistemas Expertos. Proporciona una interfaz de programación con estructura de ventanas integrada a un lenguaje de reglas de producción (PRL por sus siglas en inglés *Production Rule Language*), éste se usa para crear la base de conocimientos.

En realidad, la programación con LEVEL5 OBJECT, pierde complejidad en cuanto se comprende y familiariza el programador con la orientación a objetos, porque la estructuración de la reglas de producción es relativamente sencilla.

Otras de sus principales ventajas son:

- ✓ Permite el manejo de objetos y reglas de producción, por lo que el conocimiento puede ser representado de una forma muy similar a la que maneja el ser humano en el mundo real.

- ✓ El conocimiento se encapsula en las reglas y los objetos, así, las reglas verifican patrones preestablecidos de objetos como hechos. Al mismo tiempo los objetos pueden operar independientemente de las reglas

- ✓ Como su interface de programación es mediante ventanas, facilita enormemente la construcción de las reglas de producción, de modo que casi no se necesita teclear líneas de programación.

- ✓ Las aplicaciones realizadas trabajan bajo ambiente Windows, por lo que es fácil interactuar con otras aplicaciones bajo el mismo ambiente. También acepta el manejo de bases de datos y hojas de cálculo.

- ✓ Se pueden desarrollar interfaces de usuario robustas, por lo que el uso de otro lenguaje de *Front-End* es prácticamente innecesario. De esta manera las aplicaciones son muy amigables para el usuario, porque permite el uso de hiperregiones, menús y botones que facilitan la navegación, entre otras herramientas.

- ✓ Cada aplicación creada cuenta con una serie de objetos propios del sistema (*system objects*) que son muy útiles durante su desarrollo, no sólo para su funcionamiento interno, sino también para la interface con el usuario.

- ✓ Existen ya varias aplicaciones creadas con LEVEL5 OBJECT, por lo que proporciona un alto grado de confiabilidad.

- ✓ Permite el manejo de factores de certidumbre. Uno de los conceptos que se emplean en el desarrollo de Sistemas Expertos es el de factor de certidumbre, LEVEL5 OBJECT, permite el uso de estos de una forma sencilla y es posible manipularlos según los requerimientos del problema. LEVEL5 OBJECT maneja tres tipos de factores de certidumbre que son:

  Factor desconocido (-2): se presenta cuando el valor se ha declarado desconocido por una fuente externa, o cuando la máquina de inferencia ha agotado todos los medios que pueden determinar el valor del atributo.

  Factor indeterminado (-1): quiere decir que el valor no ha sido asignado durante la sesión y por lo tanto no ha sido inicializado.

  Factores del 0 al 100: indican el grado de confiabilidad con el cual es conocido un valor. Si no se especifica un factor de confiabilidad, al hacerse verdadero el atributo por default adquiere el valor de 100.

Además los CF se manejan bajo un umbral, esto es, que sólo las reglas en las que su factor de certidumbre rebase el umbral, podrán ser consideradas como verdaderas.

✓ Una de las ventajas más importantes es que permite el desarrollo de aplicaciones siguiendo los dos tipos de inferencia más comunes: hacia adelante y hacia atrás. En el caso de SFDA se requiere de los dos tipos de inferencia, particularmente se emplea inferencia hacia adelante para el primer nivel y hacia atrás para el segundo.

En LEVEL5 OBJECT las reglas de producción reciben diferentes nombres según el tipo de encadenamiento. Para el encadenamiento hacia adelante se les llama DEMONS y para el encadenamiento hacia atrás RULES. Éstas últimas necesitan de una AGENDA en donde se indica, en forma de índice, el orden en que deben ejecutarse las reglas, ésta es la forma de implementar a los árboles de inferencia. En el siguiente inciso veremos en qué consisten estos dos métodos de inferencia.

4.3 Métodos de inferencia.

Lo que el Sistema Experto debe hacer es determinar cuál es el subsistema con mayor posibilidad de falla, mediante una evaluación para identificar a todas las posibles fallas y después con mediciones específicas llegar a la falla real. Partiendo de esto se decidió que el primer nivel seguiría una inferencia hacia adelante y el segundo una inferencia hacia atrás. Para poder justificar esto, es necesario primero explicar los dos tipos de inferencia, no sin antes recordar los elementos básicos de un Sistema Experto, que son la *base del conocimiento* y la *máquina de inferencia*.

La base de conocimientos se compone de objetos y reglas. Los objetos son una forma abstracta de representar a los elementos que intervienen en el problema, los cuales tienen atributos que definen sus cualidades. Las reglas representan al conocimiento experto, y mediante ellas podemos llegar a un objeto. Las reglas se forman de premisas y conclusiones.

Podemos decir entonces, que la base del conocimiento es una lista de objetos con sus reglas y atributos asociados.

Los objetos tienen que ser agrupados en clases según sus características. Por ejemplo:

> Pensemos en medios de transporte como una clase; los objetos que la componen podrían ser: automóvil, barco y avión. Cada uno de estos objetos tiene sus propias características; por ejemplo, vía que utilizan para trasladarse, número de llantas, combustible que utilizan, etc. Si estructuráramos una regla, sería de la siguiente manera: Si se transporta por vía terrestre, tiene cuatro llantas y emplea gasolina, entonces es un automóvil. Se observa que por medio de las reglas, empleando los atributos del objeto, podemos llegar a éste.

La máquina de inferencia es la parte del Sistema Experto que emplea un método de razonamiento para llegar a la solución del problema utilizando las reglas.

Para resolver el problema planteado, la máquina de inferencia utiliza el conocimiento experto, para inferir nuevos hechos y obtener las relaciones que dan la solución, sólo si el problema es soluble en este entorno.

Hay dos métodos básicos de inferencia: el método de encadenamiento hacia adelante y el método de encadenamiento hacia atrás.

El **método de encadenamiento hacia adelante** (*forward chaining* o *Data driven*), utiliza información proporcionada por el usuario para moverse a través de las reglas hasta encontrar puntos terminales. Las reglas que definen al objeto, crean el camino que lo conducen a él; en consecuencia, la única forma de llegar al objeto, es satisfaciendo todas sus reglas. El encadenamiento hacia adelante, empieza con alguna información y trata de encontrar al objeto que encaja con ésta.

Con el encadenamiento hacia adelante se parte de que no se sabe nada y se desea llegar a algo. Se emplea cuando tratamos de encontrar respuesta a una

pregunta, para lo que se nos proporciona información mediante la cual podemos llegar a una conclusión.

El método, arbitrariamente, inicia con la primer regla escrita en la base de conocimientos y trata de probarla. Solamente se cumplirá cuando todas sus premisas o condiciones sean verdaderas, posteriormente pasa a la siguiente regla. Las reglas que se van cumpliendo agregan sus conclusiones a una lista de verdades.

Las conclusiones de algunas reglas se convierten en premisas de otras, formando así el encadenamiento hacia adelante. El proceso termina una vez que se han verificado todas las reglas existentes en la base del conocimiento.

Una de las principales desventajas del encadenamiento hacia adelante, es que no existen bases para escoger entre una ruta y otra, por lo que es necesario hacer una búsqueda exhaustiva en la base del conocimiento antes de encontrar respuesta. Esto hace que el proceso sea muy largo y que al terminar se tenga una amplia gama de conclusiones.

Por lo tanto, el encadenamiento hacia adelante es más simple pero más largo, pues no es discriminativo, podemos decir, que va de lo particular a lo general, es decir, que dadas unas premisas busca en la base de conocimientos, probando todas las reglas, hasta encontrar a las que cumplan las premisas.

El **método de encadenamiento hacia atrás** (*backward chaining*), también conocido como *goal-driven* se diferencia principalmente del encadenamiento hacia adelante, porque empieza asumiendo una conclusión como verdadera y posteriormente utiliza las reglas para tratar de probarla.

Dicho método, parte de alguna conclusión y la prueba, demostrando la verdad de cada una de sus premisas, en un orden de izquierda a derecha. Para demostrar la veracidad de una premisa, se busca una regla que tenga esa misma premisa como una de sus conclusiones. Si una regla así es encontrada, se encadena hacia atrás de esa regla y se intenta probar demostrando la veracidad de cada una de sus premisas de la misma manera.

El encadenamiento hacia atrás resulta ser una buena técnica de inferencia cuando se puede adivinar cuál puede ser la conclusión por medio del razonamiento. Hay al menos dos razones para esto; la primera es que la cadena

lógica puede ser tan corta y directa como lo permita nuestra base de conocimientos; la otra, es que el cuestionamiento del usuario está centrado en el objetivo que está siendo probado.

El encadenamiento hacia atrás puede no ser más efectivo que el encadenamiento hacia adelante, cuando no se tiene idea de cuál puede ser la conclusión en una situación dada.

El encadenamiento hacia atrás va de lo general a lo particular, debido a que es mucho más discriminante y no verifica en toda la base de conocimientos, sino que sólo en una parte, dependiendo del camino que las reglas que se hacen verdaderas van trazando.

4.3.1 Primer nivel , encadenamiento hacia adelante

El experto a través de su experiencia, sabe que algunos síntomas son característicos de fallas en determinado subsistema, pero más bien asocia los síntomas y mediciones generales a un conjunto de falla. Si seguimos esta orientación tendremos una larga lista de posibles fallas, porque con los mismos síntomas se llega a varias de éstas.

Se observó que conociendo todas las posibles fallas, podríamos determinar cuál era el subsistema con mayor posibilidad de falla, debido a que si un subsistema presenta una larga lista de éstas, es señal de que contiene al problema. Ahora sólo faltaría determinar qué es lo que lo hace fallar o bien, exactamente en cuál de sus partes está el problema.

Las fallas estarán representadas por reglas de producción. Como nuestro interés es determinar todas las posibles fallas, es decir evaluar a todas las reglas, es inminente el uso de un encadenamiento hacia adelante. Entonces lo que podría ser una desventaja, para el caso del primer nivel de diagnóstico es una gran ventaja.

Lo que hará el primer nivel de diagnóstico es que, a partir de las entradas que proporcione el usuario, evaluará todas las reglas de la base de conocimientos,

buscando a las que tengan como premisas las entradas dadas y así identificar a todas las posibles fallas.

Como se vio en el capítulo 3.1.2, las fallas se forman de una parte y su problema. Las partes están agrupadas de tal manera que dan origen a los subsistemas. Entonces, cada una de las fallas está asociada a un subsistema, como consecuencia, al encontrar todas las posibles fallas, encontramos también al subsistema que presenta más de éstas.

De acuerdo a la forma en cómo el experto realiza el diagnóstico, las primeras entradas que le ayudan a determinar el subsistema en el que está la falla son: los síntomas, las mediciones del analizador de gases, códigos de falla y por supuesto la experiencia.

Esta última se refiere a la información que puede dar el experto en ciertos casos. Esto se debe a que algunas marcas de automóvil, cuando presentan determinados síntomas, sin más cuestionamiento se sabe el subsistema que está fallando, porque ese tipo de auto siempre presenta esa falla con esos síntomas. Entonces ya no es necesario evaluar más, sin embargo, se podrá hacer uso de esta entrada solamente cuando un mecánico relativamente experimentado empleé el Sistema Experto y sólo en algunos casos.

Para el prototipo desarrollado, no se considera esta entrada dado que solamente se trabajará con automóviles General Motors, por lo que no se pude hacer distinción entre marcas.

Como es de imaginarse, no todas las fallas aportan la misma información. Por ejemplo, dados unos síntomas se puede llegar a dos fallas, pero por los síntomas que se están dando es más probable que sea una que otra, esto lo determina el experto. Entonces a cada falla se le asociará un coeficiente de certidumbre o de confiabilidad que determina qué tan posible es que sea esa falla.

Se puede concluir que la forma en como actúa el primer nivel es el siguiente:

☑ Recibe los datos de entrada

☑ Evalúa todas las reglas de producción

☑ Determina las posibles fallas

☑ Finalmente, empleando un coeficiente de certidumbre, que dará un peso determinado a cada una de las fallas, se podrá saber en qué subsistema hay mayor posibilidad de falla. Esto permitirá hacer una acotación para posteriormente realizar solamente las mediciones mínimas necesarias para llegar a la falla real; de esto se encargará el segundo nivel. En el capítulo 4.5 se explicará el manejo de la certidumbre.

En este nivel se emplearán formatos (se explicarán con detalle en el capítulo 5.2) para la interpretación del conocimiento experto, estos nos ayudarán a representarlo fácilmente en reglas de producción.

4.3.2 Segundo nivel, encadenamiento hacia atrás

Una vez que el experto se sitúa en el subsistema con problemas, determina las mediciones que debe efectuar en partes específicas; cuando se topa con una medición que no coincide con los parámetros que sabe que permiten un buen funcionamiento, empieza a hacer mediciones alrededor de la parte que causa la medición errónea, olvidándose de otras partes. El Sistema Experto tratará de imitar este comportamiento.

El segundo nivel parte de que ya se sabe en qué subsistema está la falla y empezará a pedir algunos datos o mediciones para descubrir cuál es la parte que presenta problema.

Entonces se sabe por dónde iniciar la búsqueda; el segundo nivel sabe cuáles son las posibles fallas, que son las asociadas al subsistema indicado por el primer nivel, y solamente tratará de probarlas. Una vez que encuentre una, ya no es necesario seguir probando las demás, porque se ha detectado la falla. Es posible que sea más de una, debido a que un problema en determinada parte se

presente como consecuencia del problema en otra, por lo que se requiere hacer un chequeo hacia atrás hasta encontrar la falla real.

Esto es un típico comportamiento de un encadenamiento hacia atrás, por lo que el segundo nivel se desarrollará siguiendo este tipo de inferencia.

Entonces el segundo nivel inicia con una conclusión y pide alguna medición al usuario para comprobar sus premisas, las reglas estarán encadenadas de tal manera que a algunas sólo las verificará cuando otra lo requiera, formando así el encadenamiento hacia atrás. De esta manera, si la regla en cuestión es verdadera, pasa a probar la siguiente que está encadenada con ella, de lo contrario se olvida de ésta y verifica a la siguiente regla que no esté encadenada.

Para representar el conocimiento experto en este nivel, se emplearán reglas de producción originadas de árboles de inferencia, en los que los nodos serán las fallas, y las ramas las mediciones. Entonces se parte de un nodo inicial y conforme a los datos que se vayan introduciendo se irán recorriendo las ramas. Una vez que se encuentre algún resultado ya no regresará a verificar otras ramas.

Estos árboles se hicieron con ayuda del experto, analizando la relación que existe entre las partes del automóvil y los valores que deben de mantener (voltaje, presión, R.P.M., etc.).

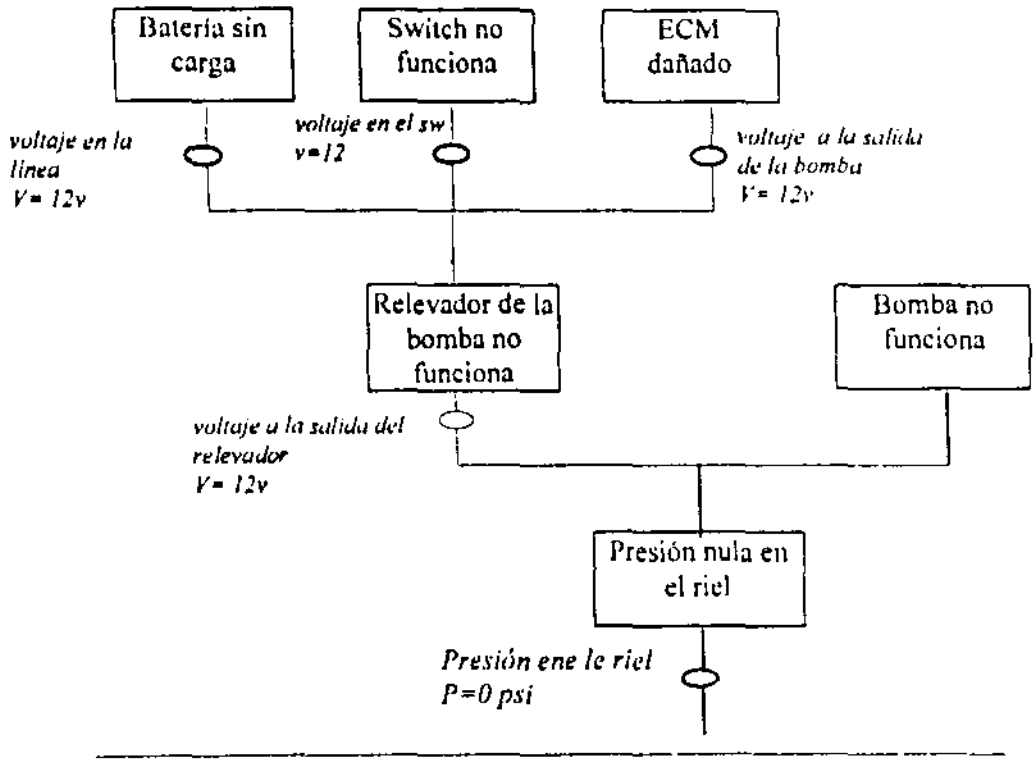


Figura. 4.1 Ejemplo de un árbol de inferencia para el segundo nivel de diagnóstico.

En la figura anterior se observa un árbol de inferencia. Los círculos indican una medición que se debe pedir al usuario y los cuadros, las fallas.

La primera medición corresponde al círculo situado más abajo; en el ejemplo se comienza por la solicitud de medición en el riel de inyectores y una respuesta igual a 0 PSI.

4.4 Los objetos y las reglas de producción en el primer nivel.

La forma más común de representar el conocimiento experto es mediante las reglas de producción, debido a su gran sencillez, por partir del principio de causalidad (causa - efecto).

Para el primer nivel de diagnóstico, se emplearán reglas de producción que se obtienen con la ayuda de formatos diseñados para recopilar el conocimiento experto, a diferencia del segundo nivel en el que nos apoyamos de árboles de inferencia para la construcción de las reglas.

Al experto se le dificulta expresar su conocimiento de forma concreta y a los analistas comprenderlo, por lo que se diseñaron tres formatos o tablas para extraer la información por etapas. Esto es, primero obtenemos las fallas y después los síntomas que nos llevan a ellas, de esta manera es más sencilla la comunicación entre el experto y los analistas. En el capítulo de adquisición del conocimiento experto (capítulo 5.2), se verán detalladamente estos formatos.

Una regla consta de un conjunto de acciones o efectos, que son ciertas cuando se cumple con un conjunto de condiciones o causas. Tanto los efectos como las causas pueden ser más de una. La fórmula general de una regla de producción es la siguiente:

SI < condiciones o premisas > ENTONCES <conclusiones>

En nuestro caso las premisas son los síntomas o mediciones o códigos de falla y las conclusiones son las fallas.

Por ejemplo:

SI *no da marcha* ENTONCES *la batería no retiene carga.*

SI *HC = 150* ENTONCES *las bujías están sucias.*

SI *el código de falla = 43* ENTONCES *el sensor de detonación está dañado.*

Estas son tres reglas de producción. La primera tiene como premisa un sintoma, la premisa de la segunda es una lectura del analizador de gases y de la tercera es un código de falla obtenido del scanner. Todas tienen como conclusión una falla.

Entonces los elementos básicos que definen el problema son los datos de entrada (los síntomas, las mediciones y los códigos de falla) y las fallas.

Estos elementos estarán representados por objetos, mismos que clasificaremos en clases.

La primera clase se empleará para agrupar a todos los síntomas, en ella encontraremos la larga lista de síntomas que se irán adquiriendo del experto conforme el desarrollo del SE.

Otra clase será para representar a las mediciones del analizador de gases que consideraremos como objetos; estos objetos podrán tomar diferentes valores según las mediciones obtenidas.

Una tercera clase servirá para representar a los códigos de falla, estos objetos operarán de forma similar a los anteriores.

Se necesita una clase para representar a las fallas. Las fallas están compuestas de partes y como vimos en el capitulo 3.1.1 las partes se clasifican en subsistemas, de tal forma que cada falla estará dentro de una clase que representa a un subsistema. Esta estructura permite tener a los elementos para armar la base de conocimientos, mejor clasificados y con un enfoque sistémico, que es lo que se necesita para el primer nivel de diagnóstico.

Entonces se emplearán once clases, una para cada subsistema, cada una contendrá las diferentes partes que los componen, así como sus problemas.

En el capitulo 4.2.1 se vio que un objeto se representa como instancias de una clase, entonces todas las instancias que se crean con la misma clase representan a objetos que tienen las mismas caracteristicas. Por ejemplo, los automóviles tienen como caracteristicas, color, modelo y marca. En cada caso estas caracteristicas pueden tomar valores diferentes como azul, 98 y Ford, respectivamente.

Las clases que se han planteado para el desarrollo de SEDA son muy particulares, por lo que la clase en sí representa al objeto y no es necesario el uso de instancias

Para comprender mejor lo anterior, se analizará la clase síntomas: ¿qué características pueden tener los síntomas?, solamente el momento en que se presentan, así, los atributos de esta clase serían los momentos y estos tomarían como valores a los síntomas. Esta estructura sólo se empleará una vez, es decir, para representar a los diferentes síntomas, no necesitamos de otra estructura como esta, por lo tanto no necesitamos instancias.

Esta misma situación se presenta en las demás clases. Por ejemplo en las clases de subsistemas, lo que caracteriza a cada subsistema son las partes, entonces los atributos de estas clases serán las partes y los valores que tomen los atributos serán las fallas. Considerando a la clase llamada *admisión de aire* (clase que representa a un subsistemas) dos de sus atributos son *filtro de aire* y *sistema sensor MAT*. Durante todo el proceso de diagnóstico, solo se hará referencia a ese filtro de aire y a ese sensor MAT del subsistema de admisión de aire, ningún otro objeto requerirá de esos mismos atributos; es entonces donde se ve que es innecesaria la creación de instancias, por lo que se trabaja con la clase como objeto.

Si se considerarán varias marcas de automóviles, se podrían crear instancias de las clases que representan a los subsistemas, porque cada instancia representaría a una marca. Regresando al ejemplo anterior, se tendría un filtro de aire de un automóvil General Motors y un filtro de aire de un automóvil Ford. Sin embargo aún contemplando la extensión del Sistema Experto para otras marcas, se considera mejor el planteamiento de clases como objetos evitando la creación de instancias, porque esto permitirá una estructura modular.

Lo anterior, significa que se podrán tener diferentes Sistema Expertos, cada uno con el conocimiento experto necesario para el diagnóstico de automóviles de una marca y estos serán independientes entre ellos. De esta manera SEDA se podrá adquirir por módulos dependiendo de las necesidades del usuario.

Por esta razón en el primer nivel de diagnóstico los objetos están representados por las clases y no por las instancias. Entonces los objetos son:

síntomas, lectura del analizador de gases, códigos de falla y los subsistemas. En el capítulo 5.3 veremos la estructura formal de las clases.

4.5 Manejo de certidumbre

La mayoría de las veces, las decisiones que el humano toma en su vida cotidiana, no son totalmente certeras o exactas, nunca sale de su hogar preparado para soportar un día lluvioso con una certeza de cien por ciento de que lloverá, o por ejemplo, un programador nunca está completamente seguro de que su programa correrá correctamente a la primera. Esto significa que el ser humano vive tomando decisiones con cierto grado de certidumbre y sólo en algunas ocasiones está completamente seguro de algo.

Los Sistemas Expertos, al intentar simular el comportamiento de un humano, requieren de considerar la certidumbre de los eventos que se involucran en él. Existen técnicas para efectuar una medición de certidumbre en los resultados arrojados por la máquina de inferencia, esta medición se realiza por medio de los factores de certidumbre o de confiabilidad, conocidos como **CF** por sus siglas en inglés *Certainty Factor o Confidence Factor*.

El factor de certidumbre es simplemente un factor de peso que se puede asociar a cada regla y sirve como una estimación de qué tan exacto puede ser el resultado arrojado por ésta. Entonces estamos hablando de una probabilidad que va de 0 a 100.

Los factores de certidumbre pueden ser calculados como un factor de peso basado en los aspectos específicos del problema a ser solucionado. La forma en cómo se manejan, varía mucho de una aplicación otra; su cálculo puede ir desde simples sumatorias, hasta métodos más elaborados que incluyen teoría de probabilidad y algoritmos matemáticos. Algunas veces son asignados arbitrariamente por el diseñador del sistema y posteriormente modificados para que proporcionen respuestas consistentes ante hechos reales. En cualquier caso la forma de manejarlos debe ser construida para cada problema.

Para el primer nivel de diagnóstico los CF soy muy importantes, porque con ellos se determinará en qué subsistemas hay mayor posibilidad de falla. Esto se hará de la siguiente forma: a cada regla se le asigna un CF, que indica qué tan

probable es que se dé esa falla con esos síntomas. Como cada falla está asociada a un subsistema, se puede ir agregando el CF de cada regla que se cumpla, a una variable que represente al subsistema. De esta manera se irán acumulando todos los CF correspondientes a las fallas de un subsistema y así, al terminar de verificar todas las reglas, tendremos en las variables un número que indica la posibilidad de falla en el subsistema.

Cuando la variable que representa a un subsistema tiene un número diferente de cero, indica que en ese subsistema se presentaron fallas. El subsistema que tenga el número mayor será en el que exista mayor posibilidad de fallas porque se presentaron muchas fallas en ese subsistema o bien porque las pocas fallas que se presentaron tienen CF muy grandes, haciendo así más notoria la presencia de la falla en ese subsistema.

El CF que se le asigna a cada regla representa el nivel de discriminación de dicha regla. Por ejemplo, si con los síntomas que se presenten se hacen verdaderas dos reglas de diferentes subsistemas, y es más probable que se presente una de ellas, entonces, a la regla más probable se le da un peso mayor, haciendo resaltar al subsistema al que pertenece y discriminando al otro.

El CF que se le asigna a cada regla y por lo tanto a cada falla, es parte del conocimiento experto. Es necesario que el experto determine ese factor de peso o bien encontrar una forma apropiada de calcularlo para que después sea ajustado por el experto. La manera en como se obtiene el CF para cada regla, se explicará con detalle en el capítulo de formulación de reglas de producción (capítulo 5.3.2)

Entonces la parte más importante del primer nivel es hacer una asignación correcta de CF, es por eso que se tiene una etapa de ajuste, para lo que se diseñaron pantallas en las que el experto pueda visualizar los CF, e indicar cuáles deben ser cambiados.

CAPÍTULO 5: PRIMER NIVEL DE DIAGNÓSTICO.

En el capítulo 3.1.1 se señaló que el sistema motriz de un automóvil se divide en ocho sistemas y estos a su vez en subsistemas. El objetivo del primer nivel de diagnóstico consiste en determinar el subsistema en donde de localiza la falla. Con este resultado, el segundo nivel de diagnóstico solicitará mediciones específicas para identificar la falla exacta.

Para realizar el primer nivel de diagnóstico se requiere de tres tipos de datos proporcionados por el usuario: el primero son los síntomas que presenta el automóvil, el segundo las mediciones del analizador de gases y el tercero los códigos de falla. (Figura 5.1)

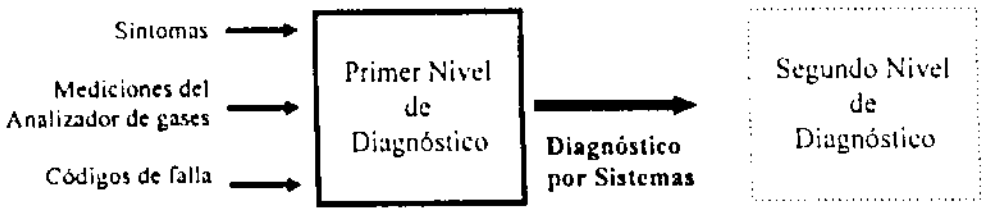


Figura 5.1 Diagrama de bloque del primer nivel de diagnóstico. Diagnóstico por sistemas → en qué subsistema se encuentra la falla

En este capítulo se presentan los principales aspectos asociados a la construcción del primer nivel de diagnóstico. En primer lugar se presenta una explicación de los tres tipos de datos que se requieren para ofrecer un diagnóstico. En segundo lugar se explica la manera estructurada en que se adquirió el conocimiento experto. Posteriormente se hace referencia a la forma en cómo se representó dentro del Sistema Experto el conocimiento adquirido. Finalmente se muestran los resultados que se obtienen con el primer nivel de diagnóstico, explicando la manera en la que el sistema acumula los valores de los CF

asignados a cada regla con el objeto de establecer el subsistema con mayor probabilidad de albergar a la falla.

5.1 Síntomas y mediciones.

Los síntomas y las mediciones del analizador de gases como del scanner, son la fuente de información que alimentan al primer nivel.

Los síntomas se clasifican por estado de operación. Esta clasificación se hace con la finalidad de facilitar su captura y evitar confusiones en el momento que el usuario suministra la información.

Las mediciones son de dos tipos, las proporcionadas por el analizador de gases y las que proporciona el scanner, que más que una medición es un código.

A continuación se explica con mayor detalle cada uno de los tipos de datos que se requieren en el primer nivel de diagnóstico.

5.1.1 Síntomas por estados de operación.

Los posibles síntomas que pueden presentar los automóviles son demasiados, por esta razón se les hace una clasificación de acuerdo al momento en que se presentan, para así facilitar su identificación. A estos diferentes momentos en los que se puede presentar el síntoma, se les conoce como *Estado de Operación.*

Los estados de operación en que se clasificaron los síntomas son: *Al dar marcha, en baja, al operar, en alta y al apagar*. En seguida se muestra la lista de síntomas que, de acuerdo al experto, se presentan en cada estado de operación.

Al dar marcha

Este es el estado de operación en el cual el automóvil entra en funcionamiento, es decir, desde que se da vuelta al switch hasta que el motor

empieza a funcionar. Los síntomas que se presentan en este momento son los siguientes:

- No da marcha
- No arranca
- Arranca después de un tiempo
- No se mantiene encendido
- Huele a gasolina
- Tarda en arrancar siempre
- Tarda en arrancar solo en frío (entendiéndose por frío que el motor no ha funcionado por un periodo determinado por lo que está a temperatura ambiente.)
- Arranca y se apaga
- Al poner switch no prenden focos indicadores
- Al poner switch no prende algún foco indicador
- A veces no da marcha
- Al arrancar se escucha ruido en la marcha
- Se queda pegado el switch de encendido

En baja

Este estado se conoce también como "ralentí" y se refiere al momento en el que el motor está funcionando pero sin que el vehículo esté en movimiento. En este momento el motor no está acelerado por lo que presenta de 700 a 800 RPM (revoluciones por minuto). Los síntomas que se pueden presentar son los siguientes:

- Al aplicar carga tiende a apagarse
- Se mantiene irregular (el motor no tiene estabilidad es decir que sube y baja su potencia)
- Permanece acelerado (el motor esta acelerado aún sin pisar el acelerador)
- Huele a gasolina en el escape
- Se ahoga (el motor tiene demasiada gasolina, por eso se apaga)
- Arroja humo negro
- Se escucha falla de cilindros en escape (se escuchan explosiones, en menor o mayor grado)
- No se mantiene en marcha mínima siempre (después de estar un momento encendido se apaga)

- No se mantiene en marcha mínima sólo en frío (mismo síntoma pero sólo a temperatura ambiente)
- La bomba suena al funcionar
- El ventilador suena al funcionar

En alta

Cuando el motor está siendo forzado, es decir, que se acelera a fondo sin movimiento del automóvil, se dice que está en "alta" o "crucero". En este momento el motor presenta de 2000 a 3000 RPM . Los síntomas que se pueden presentar son:

- No tiene potencia
- Se jalonea
- Al acelerar a fondo se apaga
- Al acelerar no desboca (al acelerar, el motor no aumentan sus revoluciones)
- Al acelerar cascabelea siempre (se escucha golpeteo de los pistones dentro del cilindro)
- Al acelerar cascabelea sólo con cambio de altura

Al operar

Este estado se presenta cuando el automóvil está en movimiento. En este momento el motor puede tener desde 900 hasta 5000 RPM. Los síntomas que se presentan son:

- Echa humo negro
- Huele a gasolina
- Consume mucha gasolina
- Al pisar acelerador no responde el motor
- Se apaga después de un tiempo
- Se apaga en caliente con poca gasolina en el tanque
- Se jalonea
- Se quedan prendidos focos indicadores
- Se queda prendido algún foco indicador
- Se descarga la batería
- No prende alguna luz exterior
- El motoventilador opera muy seguido
- El indicador de temperatura rebasa el rango normal

Al apagar

Se refiere al momento en el cual el motor deja de funcionar debido a que se cerró el switch. Los síntomas que se pueden presentar son:

- Se cierra el switch y el motor sigue operando
- Tira agua al piso
- Tira gasolina al piso
- Huele mucho a gasolina
- Huele a motor caliente

Los estados de operación se pueden observar en la siguiente gráfica:

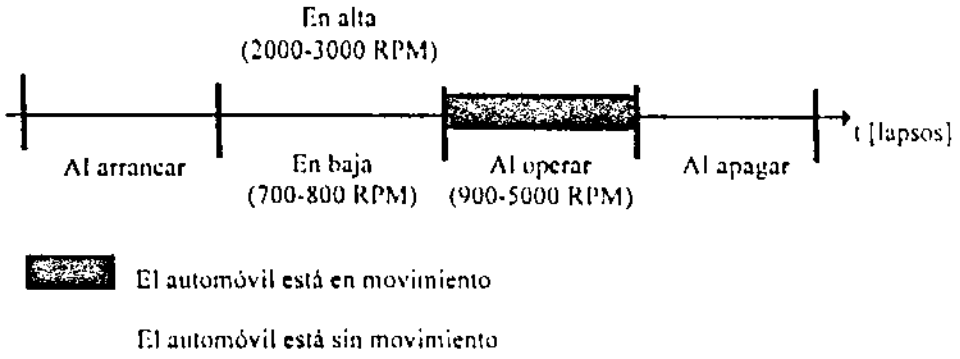


Figura 5.2 Estados de operación

Se tienen 50 síntomas y su manejo se facilita si están clasificados por estados de operación, pues de esta manera obtenemos cinco listas más pequeñas y el usuario es capaz de distinguir en qué momento se presentan las anomalías e identificar más rápidamente el síntoma requerido.

5.1.2 Analizador de gases

Un analizador de gases sirve para determinar la composición de los gases que salen del escape del automóvil. Este análisis proporciona gran información acerca de la mezcla de gasolina y aire necesario para la operación del motor.

Los gases que generalmente se analizan son los siguientes: monóxido de carbono (CO), bióxido de carbono (CO2), oxígeno (O2) e hidrocarburos (HC).

En condiciones eficientes de operación, un motor deberá quemar completamente el volumen de combustible introducido por los inyectores. Para ello se requiere que la mezcla de oxigeno -combustible, esté adecuadamente balanceada; dicha mezcla estequiométrica deberá ser de 14:1. Si esta mezcla contiene más combustible del necesario, se hablará de una mezcla rica. Por el contrario, si existe más oxígeno del requerido, se tendrá una mezcla pobre.

El análisis de los gases del escape, permiten identificar mal funcionamiento de algún sistema del motor. Por ejemplo:

El monóxido de carbono(CO) se produce por la insuficiencia de oxigeno en la mezcla, lo cual se puede deber, entre otros motivos, a un dato erróneo del sensor TPS:

La presencia de HC se debe a que no se consumió toda la gasolina durante la combustión, esto es producto de diversas fallas (desgaste de los inyectores, exceso de presión en el tren de inyectores, etc.).

El registro de oxigeno (O2) se produce por la presencia de una mezcla pobre, que también puede tener diferentes causas.

La combinación de diferentes lecturas de los gases señalados, indicará diferentes fallas del motor, por lo que el analizador de gases resulta un valioso instrumento para identificar el origen de la falla.

El analizador de gases se considera un factor necesario para el diagnóstico pues con estas mediciones se puede diagnosticar más certeramente debido a que la información con la que se cuenta es más, y en casos específicos, sólo con ésta se puede llegar fácilmente a la parte-problema. Sin embargo estas mediciones no son indispensables, es decir, el primer nivel de diagnóstico se puede llevar a cabo sin

ellas, como lo hacen muchos talleres mecánicos que no cuentan con este equipo. (figura 5.3)

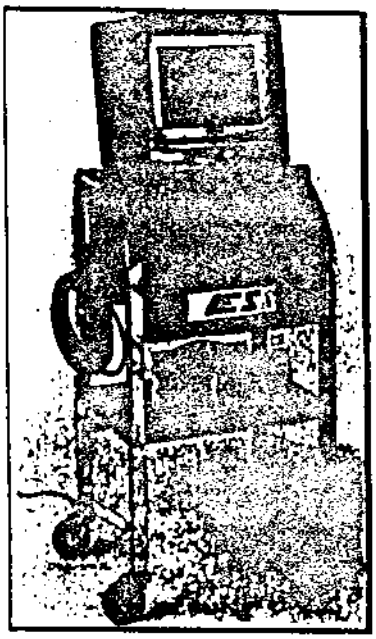


Figura 5.3 Analizador de gases (Emissions Testing - ES Tech / Pro Serie 2000)

5.1.3 Código de falla

Los automóviles *fuel injection* cuentan con una computadora ECM (unidad de control electrónico) que, con la información que le proporcionen los sensores, emite señales para la operación de actuadores, entre ellos los inyectores. Cuando se presenta una falla de un sensor, la computadora emite una señal de la que se obtiene un código de falla; este código indicará cuál sensor es el que falla.

La falla de un sensor, se da cuando su información se sale de un rango preestablecido como normal por un tiempo predeterminado en el programa de la computadora. En ese momento el mismo programa registra en la computadora un código de falla. Los manuales de servicio indican cómo recuperar el código o códigos grabados y a qué circuito se refiere cada uno de ellos, los cuales pueden llegar hasta cien diferentes según la marca y año. A la vez se enciende una luz indicadora en el tablero de instrumentos del vehículo para avisar de la ocurrencia

de una falla. Si la falla desaparece, el foco se apagará, pero esto no significa que el problema se haya corregido, puede tratarse de una falla intermitente, y en cualquier caso se debe corregir de inmediato.

Los códigos de falla son números de dos dígitos que hacen referencia a circuitos específicos y aislados cuyo sensor reporta un dato fuera de rango. Esto no significa que el sensor está defectuoso y sea necesario cambiarlo, dado que la falla puede estar también en el alambrado o en la computadora. De este modo el técnico debe revisar sensor y alambrado. Si ambos están correctos, la computadora está fallando y debe cambiarse.

Cada marca de automóviles maneja sus propios códigos de falla, sin embargo en la actualidad se está llegando a un acuerdo entre marcas para que dichos códigos sean estándar.

Para poder desplegar el código de falla, se emplea una herramienta llamada scanner, ésta se conecta a la computadora del automóvil y despliega en su pantalla la información requerida.

El scanner es como un multímetro, su utilidad es identificar el código de falla, cuenta con un teclado y una pantalla; tiene la habilidad de ahorrar tiempo en el diagnóstico y de evitar el remplazo de partes en buen estado. La clave para usar esta herramienta satisfactoriamente, radica tanto en la habilidad que tengan los técnicos para comprender los sistemas que intentan diagnosticar, como en comprender las limitaciones de la herramienta scanner.

Existen varios tipos de scanner, que varían según el fabricante, algunos permiten transmitir secuencias de comandos al automóvil y así controlar las revoluciones por minuto para poder ver la reacción de este y obtener códigos más específicamente, también verifican a los actuadores e inyectores, además contienen un multímetro digital y un osciloscopio. La utilidad de un scanner sofisticado es que se pueden observar las señales que envía cada sensor a la computadora. Con esta información, un experto puede identificar una falla de tipo electrónico, difícil de localizar sin esta ayuda. Es una herramienta muy poderosa y además muy fácil de operar , por su tamaño es muy práctica para hacer mediciones. Se utiliza en el segundo nivel de diagnóstico, en el primer nivel, sólo para los códigos de falla.

La información que proporciona el scanner es muy valiosa para hacer un diagnóstico. Cuando se cuenta con un código de falla, es más rápido encontrar la

parte problema, sin embargo muchas veces la parte que nos indica el código de falla no es la del problema, si no que es otra la que lo esta ocasionando, en estos casos los síntomas son muy importantes, es por esta razón que el código de falla forma parte de las fuentes de información para el primer nivel de diagnóstico. (figura 5.4)

Figura 5.4 Scanner (Mastertech - Multifuntion Tester / Ventronix)

5.2 Adquisición del conocimiento experto

Una de las partes fundamentales de un SE, es el conocimiento que puede proporcionar el ser humano como experto en determinada área. Este conocimiento debe ser obtenido de forma tal que pueda ser interpretado e implementado correctamente y así dar origen a un SE eficiente. Es muy importante diseñar un formato para la adquisición del conocimiento experto.

En el primer nivel de diagnóstico, el conocimiento experto se enfoca a los tres elementos que se acaban de analizar (síntomas, analizador de gases y códigos de falla), a las partes que componen a cada subsistema y a las fallas que se pueden presentar en cada una de ellas, así como su CF[3] .

Para iniciar con la adquisición de conocimiento, se le explicó al experto el orden y la estructura que se seguiría para hacerlo. Como ya se ha explicado, la forma de operar del primer nivel de diagnóstico es:

Se le dan síntomas, mediciones de gases y códigos de falla, y se obtiene el o los subsistemas que presentan mayor posibilidad de falla. Para lograr esto se requiere saber cuáles son las posibles fallas. De tal manera que, lo primero que se necesita conocer son: las partes que componen a cada subsistema, las fallas que pueden presentar y qué es lo que finalmente nos conducen a esa falla.

El proceso para la adquisición del conocimiento es largo, porque está sujeto a la disponibilidad del experto y a que se adapte a transmitir su conocimiento, no en una forma oral común, sino en una forma estructurada. Esto depende en gran medida, de los formatos que se diseñen para este fin. Después de una serie de pruebas se llegó a los formatos que a continuación se explican.

La información que se adquiera del experto quedará plasmada en una tabla con formato *Falla-síntomas* (entendiéndose por síntomas a las tres fuentes de información para el primer nivel) el cuál contiene las partes y sus problemas, y los síntomas y mediciones que nos llevan a ella. Este formato permite la creación de las reglas de producción. El formato *Falla-síntomas* se origina de otros dos formatos, el de *Parte- problema* y el de *Síntomas por estados de operación*.

[3] El término CF se definió en el capítulo 4.5

5.2.1 Formato *Parte-problema.*

Para dar origen a la tabla con formato *parte- problema*, se definieron los sistemas en los que se dividen los automóviles y se investigaron las partes que los componen, así se le fue entregando al experto, una tabla por sistema, que mostraba las partes del mismo, y que nos fue regresada con todos los posibles problemas que podía presentar cada una de las partes.

A continuación se muestra un fragmento de la tabla del sistema de encendido, como le fue mostrada al experto.

B. SISTEMA DE ENCENDIDO (incluye tres subsistemas)		
B.1 Toma y elevación de voltaje		
B.2 Distribución		
B.3 Generación de chispa		
	Parte	**Problema**
B.1.1	Bobina	
B.2.1	Sensor del cigüeñal	
B.2.2	Cable módulo DIS a ECM	
B.3.1	Bujías	

Tabla 5.1 Primera parte del formato *Parte-problema.*

De este modo el experto, con base en su conocimiento y experiencia, regresó la tabla como se muestra a continuación.

B. SISTEMA DE ENCENDIDO (incluye tres subsistemas)		
B.1 Toma y elevación de voltaje		
B.2 Distribución		
B.3 Generación de chispa		
	Parte	**Problema**
B.1.1	Bobina	*No funciona*
B.2.1	Sensor del cigüeñal	*No funciona*
B.2.2	Cable módulo DIS a ECM	*Dañado*
B.3.1	Bujías	*Dañada alguna*
		Mal calibrada o desgastada
		Sucia

Tabla 5.2 Formato *Parte - problema* terminado.

Cabe mencionar que fue necesario revisar con el experto, las tablas ya terminadas para que no hubiera confusión con la información plasmada ya en forma estructurada.

5.2.2 Formato *Síntomas por estados de operación*

Para crear este formato, se le pidió al experto que, conforme iba planteando los problemas, indicara los síntomas que permiten detectarlos, considerando la clasificación por estado de operación que vimos anteriormente. Así se tuvo una base de dónde partir para que el experto visualizara el objetivo, que era llegar a una falla por medio de síntomas. Estos fueron aumentando conforme se avanzó en la adquisición del conocimiento, incluso, algunos tuvieron que ser eliminados o modificados para facilitar su comprensión.

Como no todos los síntomas dan la misma información, fue necesario asignar un valor a la aportación de cada síntoma. Ésta fue dada en calificaciones de 5, 10, 15 y 20.

Tenemos que considerar que hay síntomas que nos acercan más a la posible falla que otros, por ejemplo:

Cuando en nuestra información figura el síntoma *Da marcha pero no arranca*, sabemos que es uno de los más comunes, pero si encontramos el síntoma *No da marcha* sabemos que no es tan común cómo el anterior; es por eso que al síntoma *Da marcha pero no arranca* le daremos una calificación de 5, en tanto que *No da marcha* tendrá una calificación de 20. Cuanto más común sea el síntoma, menor será la calificación asignada, ya que no es muy discriminante. En la asignación de las calificaciones se tomó en cuenta al experto, que las jerarquizó no sólo por esta regla, sino también con base en su experiencia.

Estas calificaciones nos ayudarán a determinar los CF, para indicar cuál de los subsistemas tiene mayor posibilidad de falla, como se explica en el capítulo 5.3.2.

Finalmente, después de tomados todos los elementos, la tabla de calificaciones quedó con la siguiente estructura:

Estado de operación	Síntomas	Calificación
Al dar marcha	No da marcha	20
	Da marcha pero no arranca	5
	No se mantiene encendido	10
En baja	Tiende a apagarse	15
	Se mantiene irregular	10
	Permanece acelerado	10
Al operar	Echa humo negro	10
	Huele a gasolina	10
	Consume mucha gasolina	5
En alta	No tiene potencia	10
	Se jalonea	10
	Al acelerar a fondo se apaga	15
Al apagar	Se cierra el switch y el motor sigue operando	20
	Tira agua al piso	10
	Tira gasolina al piso	10

Tabla 5.3 Fragmento del formato síntomas por estado de operación.

5.2.3 Formato *Falla - síntoma.*

Una vez obtenidos los formatos anteriores, se procedió a crear las tablas de *Falla-síntomas*. La información que contienen estas tablas es la que finalmente se interpreta para dar origen al sistema experto.

Este formato, como su nombre lo indica, está compuesto por fallas (parte-problema) por un lado y síntomas y mediciones por otro. Así, en una matriz, se pide al experto que marque los elementos que nos conducen a determinada falla. En la tabla 5.4 se puede observar cómo queda compuesta la matriz, y cómo es que el experto comienza a vaciar su conocimiento y experiencia en dicha tabla.

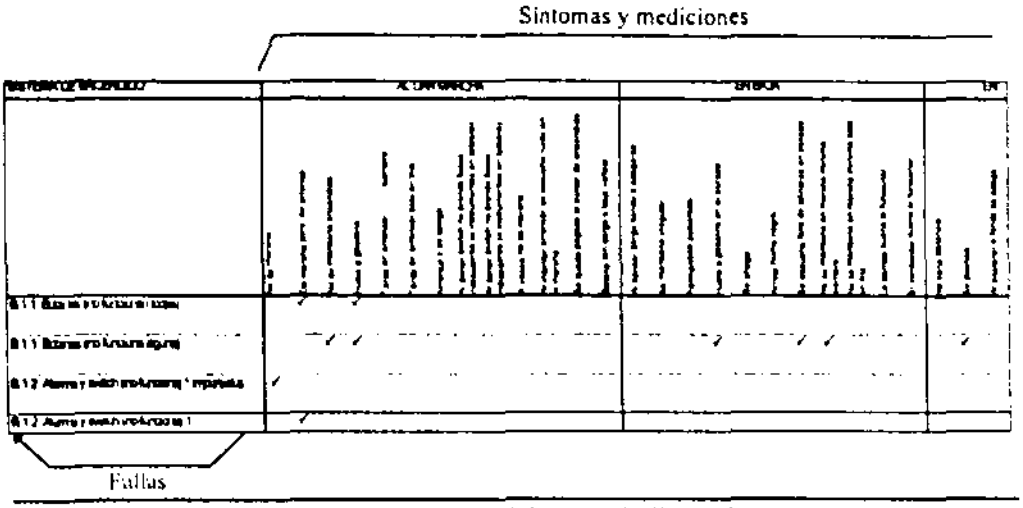


Tabla 5.4 Fragmento del formato *Fallas - Síntomas*.

Aún cuando las tablas resultaron ser muy extensas, después de varias pruebas para verificar cuál era la mejor manera de recopilar el conocimiento experto, se llegó a la conclusión de que este formato era el mejor, dado que permitía visualizar, en conjunto toda la información proporcionada, facilitando así su interpretación

5.3 Representación del conocimiento experto

La adquisición del conocimiento experto es una de las etapas más importantes durante el desarrollo de los sistemas expertos, pues de ella depende el poder hacer una buena interpretación y representación del conocimiento.

Una vez que se ha adquirido la información, se representará mediante un sistema experto, desarrollado en *Level 5 Object*, que pueda hacer la función del experto. Particularmente en este primer nivel, se empleará una metodología de encadenamiento hacia adelante.

Toda la información recabada del experto quedará plasmada en la *base de conocimiento* , la cual se compone de:

Conjunto de objetos: Es la manera de representar a todos los conceptos que intervienen en el problema.

Conjunto de reglas de producción: Es la parte que simula a la actividad del experto, relacionando a los objetos involucrados para comprobar o desechar hipótesis. Estas reglas de producción seguirán una inferencia hacia adelante, en *Level 5 Object* reciben el nombre de DEMONS.

5.3.1 Conjunto de objetos

En *Level 5 Object*, los objetos están representados por las clases, y mediante los atributos definen las características de éste, existen clases ya creadas propias de *Level 5 Object* , que representan a un grupo de objetos predefinidos que se crean automáticamente en todas las aplicaciones , y clases que se crearon para fines específicos.

Se pueden definir a los *objetos* como la parte que representa a los conceptos involucrados en el problema; los objetos están representados por las *clases*, que guardan la estructura de los mismos por medio de los *atributos*, que representan las características del objeto.

Se crearon tres tipos de objetos: Los que representan a los subsistemas, los que representan a las fuentes de información y los que sirven para presentar resultados.

Objetos para representar a los subsistemas

De la información que se obtuvo del experto, observamos que lo que caracteriza a los subsistemas del automóvil, son las partes que lo componen por la función que éstas efectúan (bobinas, bujías, sensores). Partiendo de esto, se creó una clase por subsistema, las cuales tienen como atributos a las partes. Éstas, pueden tomar diferentes valores dependiendo del problema que presentan, así los objetos con todo y su estructura y propiedades, quedaron representados por:

Clases, que son los subsistemas; atributos, que representan a las partes; y valores de los atributos, que son los problemas.

Debido a que una parte puede presentar más de un problema al mismo tiempo y para que esto se pueda representar, se definieron los atributos de tipo multicompound, que se emplean para atributos que pueden tomar más de un valor de un grupo de valores.

Las clases creadas para este fin son:

1. A Admisión de aire
2. A Bombeo y conducción
3. A Inyección de combustible
4. B Distribución
5. B Generación de chispa
6. B Toma y elevación de voltaje
7. C Alimentación y generación de energía
8. C Alimentación y servicios
9. C Arranque
10. D Bombeo y circulación del refrigerante
11. D Medición y enfriamiento temp refrigerante

Al nombre de cada uno de los subsistemas le antecede una letra, que indica el sistema al que pertenece, así el "A" significa que son del *sistema de*

Alimentación de combustible, "B" del *Sistema de Encendido*, "C" del *Sistema eléctrico* y "D" del *Sistema de enfriamiento.*

A Admisión de aire : Representa al sistema de admisión de aire, está formado por los siguientes atributos:

- Sistema sensor MAP
  puede tomar los siguientes valores:
    Lectura sólo de vacío
    Lectura sólo sin vacío
    Lectura sólo de vacío1
    Lectura sólo de vacío2
- Sensor barométrico
  puede tomar el valor de:
    Dañado
- Manguera de vacío
  puede tomar el valor de:
    Desconectada o rota
- Mecanismo de pedal
  puede tomar el valor de:
    Averiado
- Filtro de aire
  puede tomar los siguientes valores:
    Sucio o tapado
    Sucio o tapado1
    Sucio o tapado2
    Sucio o tapado3
- Sistema controlador de marcha mínima IAC
  puede tomar los siguientes valores:
    Abierto
    Cerrado
    Cerrado1
    Cerrado2
- Sistema sensor TPS
  puede tomar los siguientes valores:
    Dañado alto
    Dañado bajo
- Sistema sensor MAT
  puede tomar el valor de:

Desconectado

- Sistema sensor MAF
  puede tomar los siguientes valores:
  Dañado
  Dañado1
  Dañado2

Podemos observar que dentro de las listas de valores que pueden tomar los atributos, se tienen algunos con un número al final, estos se crearon para representar a los casos en los que diferentes síntomas nos llevan a la misma falla. Entonces, el número representa los casos que se dan para cada falla. A cada grupo numerado le corresponde un valor sin numerar que los representará. Por ejemplo, el sistema sensor MAP puede tener una lectura sólo de vacío, y se llega a esa conclusión ya sea por el caso lectura sólo de vacío1 o lectura sólo de vacio2 o bien por ambos. Para saber por cuál se dio, se analizan por separado para finalmente llegar al diagnóstico "Sensor MAP lectura sólo de vacío", representado por el valor lectura sólo de vacío.

De igual manera, siguiendo la misma estructura, se crearon las otras clases. Así, *A Bombeo y conducción* representa al subsistema de bombeo y conducción y *A Inyección de combustible* al subsistema de Inyección de combustible, ambos pertenecientes al sistema de Alimentación de combustible. *B Distribución, B Generación de chispa y B Toma y elevación de voltaje* representan a los tres subsistemas del sistema de Encendido. *C Alimentación y generación de energía, C Alimentación y servicios y C Arranque*, representan a los subsistemas del sistema eléctrico. *D Bombeo y circulación del refrigerante* representa al subsistema de bombeo y circulación de refrigerante y *D Medición y enfriamiento temp refrigerante* al subsistema de medición y enfriamiento de la temperatura del refrigerante, ambos del sistema de enfriamiento.

Podemos observar todas estas clases completas con sus respectivos elementos, en el editor de objetos de *Level 5 Object*.

Objetos para representar a las fuentes de información

Las fuentes de información son: los síntomas, lectura de analizador de gases y códigos de falla; se creó una clase por cada fuente.

Los síntomas se pueden dar por estado de operación, asi que se creó una clase ***Síntomas***, que tiene como atributos a los estados de operación y que pueden tomar diferentes valores, que representan al síntoma en si.

A continuación se presentan los atributos que componen a esta clase con los diferentes valores que pueden tomar.

- Al dar marcha

  No da marcha
  Da marcha pero no arranca
  No se mantiene encendido
  Huele a gasolina
  Arranca y se apaga
  Al poner switch no prenden focos indicadores
  Al poner switch no prende algún foco indicador
  A veces no da marcha
  Al intentar arrancar se escucha ruido en la marcha
  Se queda pegado el switch de encendido
  Tarda en arrancar siempre
  Tarda en arrancar sólo en frio

- En baja
  Al aplicar carga tiende a apagarse
  Se mantiene irregular
  Permanece acelerado
  Se ahoga
  Arroja humo negro
  Se escucha falla de cilindros en escape
  Huele a gasolina en el escape
  La bomba suena al funcionar
  El ventilador suena al funcionar
  No se mantiene en marcha minima siempre
  No se mantiene en marcha minima sólo en frio

- Al operar
  Echa humo negro
  Huele a gasolina
  Consume mucha gasolina
  Al pisar acelerador no responde el motor
  Se jalonea
  Se quedan prendidos focos indicadores
  Se queda prendido algún foco indicador
  Se descarga la batería
  No prende alguna luz exterior
  El motoventilador opera muy seguido
  El indicador de temperatura rebasa el rango normal
  No prende ventilador
  Se queda prendido foco indicador de carga
  Se apaga después de un tiempo
  Se apaga en caliente con poca gasolina en tanque

- En alta
  No tiene potencia
  Se jalonea
  Al acelerar a fondo se apaga
  Al acelerar no desboca
  Al acelerar cascabelea siempre
  Al acelerar cascabelea sólo con cambio de altura

- Al apagar
  Se cierra el switch y el motor sigue operando
  Tira agua al piso
  Tira gasolina al piso
  Huele mucho a gasolina
  Huele a motor caliente

Esta clase representa parte de la información que se obtuvo con el formato *síntomas por estado de operación* que se describió anteriormente.

Para representar a la información que se obtiene del analizador de gases se creó la clase ***Lectura del analizador de gases*** , en la cual se declararon atributos que hacen la función de variables, que guardan los valores para posteriormente hacer cálculos con ellos. Los atributos son los siguientes:

- Año
- Busca datos
- Año leido
- HC minimo en baja
- HC maximo en baja
- CO minimo en baja
- CO maximo en baja
- O2 minimo en baja
- O2 maximo en baja
- CO2 minimo en baja
- CO2 maximo en baja
- HC en baja
- CO en baja
- O2 en baja
- CO2 en baja
- HC minimo en alta
- HC maxmo en alta
- CO minimo en alta
- CO maximo en alta
- O2 minimo en alta
- O2 maximo en alta
- CO2 minimo en alta
- CO2 maximo en alta
- HC en alta
- CO en alta
- O2 en alta
- CO2 en alta

La utilidad que tienen es la siguiente:

- ***Año*** : es una variable numérica que guarda el año del vehículo que se comparará con el valor de una base de datos.

- ***Busca datos*** : es una variable simple, que sirve como bandera para dar la señal de buscar información en una base de datos.

- *Año leido*: guardará el valor del año que introduce el usuario y determinará a qué año se debe hacer referencia en la base de datos mediante la variable *Año*.

- De *HC minimo en baja*, a *CO2 maximo en baja:* son variables numéricas que guardarán las lecturas introducidas por el usuario, cuando el automóvil está en baja. Después, mediante un método que se verá más adelante, se determinará cómo es el estado del gas, si es alto, muy alto, bajo, muy bajo, normal, inestable o pobre o muy pobre en el caso del oxígeno. Dicho estado se guardará en las variables *HC en baja, CO en baja, O2 en baja o CO2 en baja* según el caso. Estas variables son de tipo cadena.

De las misma manera se trata a las variables restantes, con la diferencia de que estos son valores tomados cuando el automóvil está en alta.

Para poder simular el comportamiento del experto, se emplea una base de datos (**AG.DBF**), que sustituye a la tabla de valores, que el experto consulta para sacar conclusiones de los resultados arrojados por el analizador de gases. Así, el sistema consulta a la base de datos y obtiene conclusiones. Esta base de datos se pasa a *Level 5 Object* dando origen a una clase con el nombre de ***dB3 AGI*** con dos tipos de atributos, los primeros son generados automáticamente por *Level 5 Object* y son propios del manejador de base de datos; los otros, son los campos de la base de datos.

La información almacenada en la base de datos es la que se muestra en la tabla 5.5 y más adelante se explicará cómo se maneja esta información.

Año	HC bajo	HC min	HC max	HC alto	CO bajo	CO min	CO max	CO alto	O2 min	O2 max	O2 alto	CO2 min	CO2 max
1986 -1992	35	70	350	1050	0.35	0.7	3.25	10.5	1	3	6	7	18
1993	30	60	300	900	0.25	0.5	2.75	7.5	1	3	6	7	18
1994	10	20	150	300	0.1	0.2	1.5	3	1	3	6	7	18

Tabla 5.5 Información almacenada en la base de datos. Representa a la guía de emisiones que emplea el experto.

La clase ***Código de falla*** cuenta con cuatro atributos numéricos, que servirán para representar los códigos de falla leídos con el scanner . Sólo se declararon cuatro atributos, porque por la experiencia del experto se sabe que a lo máximo se toman en cuenta únicamente cuatro códigos de falla. La clase quedó conformada por los siguientes atributos:

- cofa1
- cofa2
- cofa3
- cofa4

Objetos para presentar resultados

La manera en que se da el diagnóstico es mediante la evaluación de todas las posibles fallas a partir de los datos proporcionados. Los resultados se muestran utilizando un código de colores que indica la posibilidad de falla en cada subsistema, este código se obtiene a partir del número arrojado por los CF de cada regla que se hizo válida; para ello se involucraron tres clases:

Colores arreglos. Contiene los siguientes arreglos con longitud 30 :

- factores
- numeros
- colores

Los dos primeros son de tipo numérico y el último es de tipo color (los valores almacenados en éste son colores expresados en niveles de rojo, verde y azul).

Colores índices y vars. Contiene las variables necesarias para manejar a los arreglos y desplegar resultados. Estas variables son:

- k
- y
- j
- numero
- EX

Todas numéricas excepto EX que es de tipo simple.

domain, Es una clase creada automáticamente por *Level 5 Object*, de propósito general, en la cual se pueden poner los atributos deseados, para cualquier fin, con la particularidad de que esta información se puede compartir con otras aplicaciones desarrolladas en *Level 5 Object*. Los atributos que se declararon en esta clase, al igual que las anteriores actúan como variables, son de tipo numérico y guardan los resultados arrojados por los CF para cada subsistema, así la clase queda conformada por:

- Fallas bombeo y conduccion
- Fallas inyeccion
- Fallas admision de aire
- Fallas toma y elevacion de voltaje
- Fallas distribucion
- Fallas generacion de chispa
- Fallas bombeo y circulacion de refrigerante
- Fallas medi y enfriamiento refrigerante
- Fallas alim y gen de energia electrica
- Fallas arranque
- Fallas alimentacion y servicios

Cada una de estas variables contendrá un valor que representa la posibilidad de falla en su respectivo subsistema. Estas variables, contienen los resultados del primer nivel y los cuales pasan a las variables de la clase *colores arreglos*, mediante las variables de la clase *colores indices y vars*, para finalmente mostrar los resultados no sólo en forma numérica, sino también con la ayuda de un código de colores que hará mas fácil su identificación. Más adelante se explicará con detalle cómo se realiza este proceso.

La calificación que se le asocia a cada síntoma da origen a un CF determinado. Estos CF requieren de ser ajustados con ayuda del experto, dado que en algunos casos no se puede seguir una regla estricta, así que se crearon pantallas para poder trabajar este ajuste con el experto en las cuales se visualizan los CF obtenidos; para ello se crearon las siguientes clases:

desplegar cfs a. Muestra los CF del subsistema de arranque.

desplegar cfs aa. Muestra los CF del subsistema de admisión de aire.

desplegar cfs agee Muestra los CF del subsistema de alimentación y generación de energía eléctrica..

desplegar cfs as Muestra los CF del subsistema de alimentación y servicios.

desplegar cfs bc Muestra los CF del subsistema de bombeo y conducción.

desplegar cfs bcr Muestra los CF del subsistema de bombeo y circulación del refrigerante.

desplegar cfs dis Muestra los CF del subsistema de distribución.

desplegar cfs gc Muestra los CF del subsistema de generación de chispa.

desplegar cfs ic Muestra los CF del subsistema de inyección de combustible.

desplegar cfs metr Muestra los CF del subsistema de medición y enfriamiento del refrigerante.

desplegar cfs tev Muestra los CF del subsistema de toma y elevación de voltaje.

Todas estas clases contienen variables numéricas que almacenan a los valores de CF obtenidos para cada regla de producción .

5.3.2 Formulación de reglas de producción

Las reglas de producción, simulan el comportamiento del experto, evaluando una serie de síntomas para saber cuál es la falla, empleando los objetos definidos. Dado que se empleará un razonamiento hacia adelante, las reglas de producción reciben el nombre de DEMONS y tienen una estructura IF-THEN-ELSE.

La información obtenida con el formato *Fallas-Síntomas* se representó mediante estas reglas, teniendo como premisas a los síntomas, mediciones del analizador de gases y códigos de falla obtenidos por el scanner; y como conclusiones a las fallas. Así, cada falla tiene asociada por lo menos una regla, porque considerando que no en todos los talleres mecánicos cuentan con el equipo necesario, las reglas que tiene como premisas, mediciones de analizador de gases o de scanner se realizaron por separado de los síntomas.

El resultado que se obtiene en el primer nivel de diagnóstico es el subsistema que tiene mayor posibilidad de falla, esto se logra a partir de los CF como se explicó en el capítulo 4.5. **La obtención del CF**, que se le asigna a cada regla, se realiza de la siguiente manera:

> En el formato *Síntomas por subsistema* (capítulo 5.2.2) asignamos una calificación a cada síntoma según su aportación. Haciendo una sumatoria de las calificaciones de los síntomas que se involucran en una regla obtenemos el CF para cada regla. Cabe aclarar que en algunos casos el valor del CF fue ajustado por el experto.

Una vez obtenido el CF de una regla y tomando en cuenta que cada regla está asociada a un subsistema, se van agregando los CF de las reglas que se hacen válidas a las variables que representan a los subsistemas (atributos de la clase *domain*). Así, obtenemos finalmente un número asociado a cada subsistema, de tal manera que el subsistema que tenga el mayor número, ya sea por que presentó más posibles fallas o porque las pocas fallas que presentó tienen un CF muy grande, será el que tenga mayor posibilidad de contener a la falla.

Es importante remarcar la relevancia que tienen los CF para este primer nivel, pues el número que arrojen nos indicará en qué subsistema está la falla y, en el segundo nivel de diagnóstico solamente se efectuarán mediciones para ese subsistema. Si los resultados arrojados nos llevan a un subsistema que no es el indicado, la búsqueda de la falla real será ineficiente.

Entonces la parte mas importante del primer nivel es hacer una asignación correcta de CF, es por eso que se tiene una etapa de ajuste en la que participa el experto. Para esto, se diseñaron pantallas en las que él puede visualizar los CF e indicar cuales deben ser cambiados.

La estructura general de las reglas es:

IF *Síntomas*
THEN Falla (parte-problema) en subsistema CF ##
AND Fallas subsistema (posibilidad de falla) = Fallas subsistema + CF
ELSE Falla (parte-problema) en subsistema CF 0

De esta manera se asigna a cada conclusión un CF, el cual se va acumulando junto con los de las demás reglas que se cumplan, pertenecientes al mismo subsistema.

Se crearon dos tipos de DEMONS; los que representan directamente a las fallas (reglas de producción) y los que sirven para hacer algún proceso, como ver CF, agregar CF máximos y ejecutar código de colores, que se irán explicando junto con los otros.

Los DEMOS que representan a las fallas pueden tener diferentes estructuras según sea el caso que están representando. Pueden representar a **fallas sencillas**, es decir que solamente tienen como premisas síntomas; **fallas múltiples**, para las que varios síntomas nos conducen al mismo diagnóstico y tiene como premisas síntomas; **fallas con analizador de gases**, que tiene como premisas mediciones del analizador de gases y **fallas con cofa**, que tiene como premisas a los códigos de falla.

Por ejemplo, cuando se diagnostica "cable módulo DIS a ECM dañado", podemos llegar a ésta conclusión por las lecturas del analizador de gases, para lo cual se hace una regla que se llama *Cable modulo DIS a ECM AG*; por los códigos de falla proporcionados, para lo que se crea una regla llamada *Cable*

modulo DIS a ECM ***COFA*** ; o bien por los síntomas, para lo que se crearía una regla llamada simplemente *Cable modulo DIS a EC*. Si se considera que para llegar a este diagnóstico, hay varias combinaciones diferentes de síntomas que nos conducen a él, entonces en lugar de éstas se crea una regla por cada caso con los nombres de : *Cable modulo DIS a ECM 1, Cable modulo DIS a ECM 2, Cable modulo DIS a ECM 3.*

Así, se pueden tener fallas que se representan por medio de varias reglas, dependiendo de qué nos conduce al diagnóstico.

Un ejemplo de **fallas sencillas** es el caso de diagnóstico de *banda de la bomba de agua rota*, que se representa mediante el siguiente DEMON:

DEMON Banda de la bomba de agua rota
IF selected OF pushbutton Diagnostica IN
AND Al operar OF Síntomas IS Se queda prendido algún foco indicador
AND Al operar OF Síntomas IS No prende ventilador
THEN Banda de la bomba OF D bombeo y circulacion de refrigerante IS Rota CF 25
AND Fallas bombeo y circulacion de refrigerante := Fallas bombeo y circulacion de refrigerante + CONF(Banda de la bomba OF D bombeo y circulacion de refrigerante IS Rota)
ELSE Banda de la bomba OF D bombeo y circulacion de refrigerante IS Rota CF 0

La estructura que llevan este tipo de reglas es: se ponen como premisa los síntomas que conducen a la falla. En la conclusión se da la falla y se le asigna un CF, sumando las calificaciones de los síntomas que se muestran en la tabla 5.3. Se agrega el CF a la variable que representa al subsistema correspondiente, en este caso al de bombeo y circulación de refrigerante. En caso de que no haga verdadera la regla, se le asigna un valor de cero al CF, para poderlo desplegar en la pantalla de ajuste de CF y saber que la regla no se ejecutó, por lo tanto no existe falla.

Para poder ver en pantalla los CF, se hizo un DEMON por cada subsistema con el nombre de "ver CFs de " y el nombre del subsistema, de tal manera que el CF obtenido en el DEMON *Banda de la bomba de agua rota* pasa al DEMON *ver CFs bombeo y circulacion refrigerante*

DEMON 3 ver CFs bombeo y circulacion refrigerante
IF selected OF pushbutton cf bombeo y conduccion ref
THEN bombaagu desgastada OF desplegar cfs ber := CONF(Bomba de agua OF E bombeo y circulacion de refrigerante IS Desgastada)

AND bandabomba rota OF desplegar cfs bcr := CONF(Banda de la bomba OF D bombeo y circulacion de refrigerante IS Rota)
AND bandabomba se patina OF desplegar cfs bcr := CONF(Banda de la bomba OF D bombeo y circulacion de refrigerante IS Se patina)
AND refrigerante pocovol OF desplegar cfs bcr := CONF(Refrigerante OF D bombeo y circulacion de refrigerante IS Poco volumen)
AND refrigerante fuera esp OF desplegar cfs bcr := CONF(Refrigerante OF D bombeo y circulacion de refrigerante IS Fuera de especificaciones)
AND manguerasradiador confuga OF desplegar cfs bcr := CONF(Mangeras de conexión con radiador OF D bombeo y circulacion de refrigerante IS Con fuga)
AND manguerasaccesorios confuga OF desplegar cfs bcr := CONF(Mangeras de conexión accesorios OF D bombeo y circulacion de refrigerante IS Con fuga)

Se puede observar que la función de este DEMON es pasar a una variable (declaradas en la clase *desplegar cfs bcr*) el CF obtenido en los de las reglas asociadas a un mismo subsistema, para que ésta pueda ser desplegada en pantalla por medio de los displays (*pantallas que maneja Level 5 Object*).

Este tipo de DEMONS se crearon considerando que la versión completa del sistema (primero y segundo nivel) no necesita mostrar los CF, porque esta información es transparente al usuario, así que se hicieron por separado para que fuera más sencilla la eliminación de esta parte en un momento dado; sin embargo, en el primer nivel es muy importante tenerla por su utilidad en el ajuste de CF.

Un ejemplo de **fallas múltiples**, es decir, en las que diferentes grupos de síntomas nos llevan a la misma falla, es el de *Filtro de aire sucio o tapado*, que puede darse por tres casos diferentes.

Primer caso:
DEMON Filtro de aire sucio o tapado 1
IF selected OF pushbutton Diagnostica IN
AND En baja OF Sintomas IS No se mantiene en marcha minima siempre
AND En alta OF Sintomas IS No tiene potencia
AND En alta OF Sintomas IS Al acelerar no desboca
THEN Filtro de aire OF A Admision de aire IS Sucio o tapado1 CF 30
ELSE Filtro de aire OF A Admision de aire IS Sucio o tapado1 CF 0

Segundo caso:
DEMON Filtro de aire sucio o tapado 2
IF selected OF pushbutton Diagnostica IN
AND En alta OF Sintomas IS No tiene potencia
THEN Filtro de aire OF A Admision de aire IS Sucio o tapado2 CF 10

ELSE Filtro de aire OF A Admision de aire IS Sucio o tapado2 CF 0

Tercer caso:
DEMON Filtro de aire sucio o tapado 3
IF selected OF pushbutton Diagnostica IN
AND En alta OF Sintomas IS Al acelerar no desboca
THEN Filtro de aire OF A Admision de aire IS Sucio o tapado3 CF 10
ELSE Filtro de aire OF A Admision de aire IS Sucio o tapado3 CF 0

Como se puede observar, se llega al mismo diagnóstico ya sea porque se presenten los síntomas *no se mantiene en marcha mínima siempre, no tiene potencia* y *al acelerar no desboca*; o bien que se presente el síntoma *no tiene potencia*; o bien el síntoma *al acelerar no desboca*. Estos DEMONS tienen la misma estructura que los de *fallas sencillas*, con la diferencia de que en éstas no se agrega el CF a la variable correspondiente al subsistema; esto es porque se tienen tres CF.

Si se analizan bien las reglas se observa que el segundo y tercer caso contienen síntomas involucrados en el primero. Entonces la primera regla representa al peso de las otras más un valor agregado, de tal manera que si se llegarán a hacer válidas dos o incluso las tres reglas, se está redundando la información si se agregara el CF de cada una de ellas. Es por esta razón que se agrega solamente el CF mayor. Si no se cumple alguna de las reglas, se le agrega un CF igual a cero y pasaría únicamente el CF de la regla que sí se cumplió (por ser la mayor).

Para agregar estos CF, se creó un DEMON, con el nombre de "*Agregar cfs maximos a las sumas de fallas*", que obtiene el CF mayor de las *fallas múltiples* (empleando la función MAX de *Level 5 Object*) y lo asigna a un valor general que representa a esas fallas y es este CF el que se agrega a la variable que representa al subsistema.

Continuando con el ejemplo de *Filtro de aire sucio o tapado*. Se recordará que las clases que representan a los subsistemas, tienen como atributos a las partes, la cuales pueden tomar diferentes valores, que son los problemas. Así en la clase *A Admision de Aire*, se tiene el atributo *Filtro de aire* , que puede tomar los siguientes valores:

Sucio o tapado (valor general que representa a los tres siguientes)

Sucio o tapado1 (para el primer caso)
Sucio o tapado 2 (para el segundo caso)
Sucio o tapado 3 (para el tercer caso)

Cada una de las reglas que diagnostican *Filtro de aire sucio o tapado*, asigna un CF al valor *sucio tapado 1, sucio o tapado 2 y sucio o tapado 3* respectivamente, y posteriormente el mayor de estos CF se asigna al CF del valor sucio o tapado. La razón por la que se crean valores que representen a los diferentes casos de una falla múltiple, es para guardar todos los posibles CF y posteriormente seleccionar a uno.

Para asignar el mayor CF de los tres casos se emplean las siguientes líneas del DEMON *Agregar cfs maximos a las sumas de fallas*:

CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado) := MAX(CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado1), CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado2), CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado3))
AND **Fallas admision de aire :=** Fallas admision de aire + CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado)

Como se puede ver en estas líneas, primero se asigna el CF máximo al atributo *Filtro de aire sucio o tapado* y posteriormente se agrega este CF a la variable *Fallas admisión de aire*, que es la que representa al subsistema.

En éste DEMON se emplean las siguientes dos funciones:

- La función MAX que tiene la sintaxis MAX(valor1, valor2, ..., valor N), y regresa como resultado el máximo valor de los parámetros.

- La función CONF cuya sintaxis es CONF(atributo) y hace referencia al CF del atributo.

En el apéndice se pude observar el código completo del DEMON *Agregar cfs maximos a las sumas de fallas*.

Las **fallas con analizador de gases**, son las que tiene como premisas lecturas del analizador de gases; un ejemplo de éstas es el de inyector sucio.

DEMON Inyector sucio AG
IF selected OF pushbutton Diagnostica IN
AND CO en baja OF Lectura del Analizador de gases = "MUY BAJO"
AND HC en baja OF Lectura del Analizador de gases = "MUY ALTO"
AND CO en alta OF Lectura del Analizador de gases = "MUY BAJO"
AND HC en alta OF Lectura del Analizador de gases = "MUY ALTO"
AND O2 en alta OF Lectura del Analizador de gases = "MUY POBRE"
THEN Inyector OF A Inyeccion de Combustible IS Sucio CF 30
AND Fallas inyeccion := Fallas inyeccion + CONF(Inyector OF A Inyeccion de Combustible IS Sucio)
AND inyec sucio AG OF desplegar efs ic := CONF(Inyector OF A Inyeccion de Combustible IS Sucio)

Se observa que las premisas no son propiamente las mediciones que proporciona el usuario, sino que son los estados en los que se encuentran los gases (alto, muy alto, bajo, normal, etc). Dichos estados están determinados por un métodos WHEN CHANGED que, como su nombre lo indica, se ejecutan cuando el atributo asociado a él cambia de valor. A continuación se analizará uno de ellos, el WHEN CHANGED del atributo ***CO maximo en alta OF Lectura del Analizador de gases.***

```
WHEN CHANGED
 BEGIN
  IF CO minimo en alta OF Lectura del Analizador de gases < co_min OF dB3 AG 1
    AND CO minimo en alta OF Lectura del Analizador de gases >= co_bajo OF dB3 AG1
  THEN
    CO en alta OF Lectura del Analizador de gases := "BAJO"
  IF CO minimo en alta OF Lectura del Analizador de gases < co_bajo OF dB3 AG 1
  THEN
   CO en alta OF Lectura del Analizador de gases := "MUY BAJO"
  IF CO maximo en alta OF Lectura del Analizador de gases > co_max OF dB3 AG 1
    AND CO maximo en alta OF Lectura del Analizador de gases <= co_alto OF dB3 AG 1
  THEN
   CO en alta OF Lectura del Analizador de gases := "ALTO"
  IF CO maximo en alta OF Lectura del Analizador de gases > co_alto OF dB3 AG 1
   THEN CO en alta OF Lectura del Analizador de gases := "MUY ALTO"
  IF CO minimo en alta OF Lectura del Analizador de gases >= co_min OF dB3 AG 1
    AND CO maximo en alta OF Lectura del Analizador de gases <= co_max OF dB3 AG1
  THEN
   CO en alta OF Lectura del Analizador de gases := "NORMAL"
  IF CO minimo en alta OF Lectura del Analizador de gases < co_min OF dB3 AG 1
    AND CO maximo en alta OF Lectura del Analizador de gases > co_max OF dB3 AG 1
   THEN
   CO en alta OF Lectura del Analizador de gases := "INESTABLE"
```

END

Este método determina si el valor proporcionado por el usuario es bajo, alto, muy bajo, muy alto, normal o inestable, comparándolo con los valores almacenados en la base de datos, representados por la clase **dB3 AG 1**. Para determinar el estado del gas cuando el automóvil está en baja, se emplean dos variables, *CO minimo en baja y CO maximo en baja*, porque los valores que se leen del Analizador de Gases pueden ser cambiantes, por lo que se pide el mínimo y el máximo leído. Entonces, cada que cambie cualquiera de las dos, se debe correr el mismo método, para determinar el estado del gas. Esto se hizo pensando en que el usuario puede modificar los datos en cualquier momento y si el método se asociaba sólo a una de las dos, podría no arrojar resultados verídicos.

De la misma manera cuando el automóvil está en alta se emplean otras dos variables *CO minimo en alta OF Lectura de Analizador de gases* y *CO maximo en alta OF Lectura de Analizador de gases*, las cuales tiene asociadas un método que tiene la misma función que los anteriores, pero ahora considerando datos de *en alta*

Así cada uno de los gases tiene asociadas cuatro variables; valores máximos y mínimos en alta, y valores máximos y mínimos en baja; las cuales al cambiar corren un método. Por lo tanto cada uno de los gases depende de cuatro métodos para determinar su estado. Estos WHEN CHANGED se pueden observar en el listado anexo al final de este ejemplar, o bien en la pantalla de *Methods/ Rules / Demons* de *Level 5 Object*, y se identifican por estar asociados a una variable de la clase *Lectura del Analizador de gases*.

Una vez que se han determinado los estados de los gases, se ponen en los DEMONS como premisas en lugar de poner las mediciones.

Regresando al *DEMON Inyector sucio AG* , observamos que si se cumplen las premisas, se le asigna un CF de 30 (así es para todas las reglas de AG) , inmediatamente se agrega a la sumatoria de fallas del subsistema correspondiente, porque si se ejecuta otra regla asociada a este mismo atributo, se perderá; por la misma razón se pasa también inmediatamente el valor de CF a una variable para desplegarlo, es decir, que las reglas de AG no pasan valores a los DEMONS *desplegar Cfs*. Lo mismo sucede con la regla que diagnostica *Inyector sucio* por medio de síntomas, como estamos trabajando con el mismo atributo, se tiene que

agregar y desplegar el CF en la misma regla, antes de que se pierda el valor, como se puede observar a continuación:

DEMON inyector sucio
IF selected OF pushbutton Diagnostica IN
AND En baja OF Sintomas IS Se mantiene irregular
AND En alta OF Sintomas IS No tiene potencia
THEN Inyector OF A Inyeccion de Combustible IS Sucio CF 20
AND Fallas inyeccion := Fallas inyeccion + CONF(Inyector OF A Inyeccion de Combustible IS Sucio)
AND inyec sucio OF desplegar cfs ic := CONF(Inyector OF A Inyeccion de Combustible IS Sucio)
ELSE Inyector OF A Inyeccion de Combustible IS Sucio CF 0
AND inyec sucio OF desplegar cfs ic := CONF(Inyector OF A Inyeccion de Combustible IS Sucio)

Estas dos reglas *DEMON inyector sucio* y *DEMON inyector sucio AG*, están relacionadas, debido a que concluyen sobre un mismo atributo; es como si estuvieran en una sola, por lo que el ELSE Inyector OF A Inyeccion de Combustible IS Sucio CF 0, sólo es necesario en una de ellas, y como la primera que se ejecuta es la de *síntomas*, ésta es la que lo lleva, cabe mencionar que las variables asociadas a *AG* se inicializan con cero, para desplegar valores correctos en las pantallas de ajuste (variables de las clase desplegar cfs).

Las fallas con Códigos de Falla, son las que tienen como premisas lecturas del scanner, que manejamos como COFA . Un ejemplo de éstas es:

DEMON Sensor de detonacion dañado COFA
IF selected OF pushbutton Diagnostica IN
AND (cofa1 OF Codigo de Falla = 43 OR cofa2 OF Codigo de Falla = 43 OR cofa3 OF Codigo de Falla = 43 OR cofa4 OF Codigo de Falla = 43)
THEN Sensor de detonación OF B Distribución IS Dañado CF 40
AND Fallas distribucion := Fallas distribucion + CONF(Sensor de detonación OF B Distribución IS Dañado)
AND sendetona dañado COFA OF desplegar cfs dis := CONF(Sensor de detonación OF B Distribución IS Dañado)

Se puede observar, como condición que si cualquiera de las cuatro variables destinadas para leer códigos de falla, es igual a 43, entonces se presenta la falla *Sensor de detonación dañado*, y se le asigna un *CF* igual a 40 (que es el

valor que se asigna a todas las reglas con códigos de falla) al atributo *Sensor de detonación OF B Distribución IS Dañado.*

Al igual que en las regla de analizador de gases, se está trabajando con el mismo atributo que con las reglas de los síntomas, por lo tanto este *CF* debe ser agregado y desplegado en la misma regla antes de que pierda su valor. Y tampoco requiere del *ELSE Sensor de detonacion OF B Distribución IS Dañado CF 0* porque ya está contenido en el *DEMON Sensor de detonacion dañado*, que es le que tiene síntomas como premisas.

DEMON Sensor de detonacion dañado
IF selected OF pushbutton Diagnostica IN
AND Al dar marcha OF Sintomas IS Tarda en arrancar siempre
AND En baja OF Sintomas IS Al aplicar carga se apaga
AND En alta OF Sintomas IS No tiene potencia
THEN Sensor de detonacion OF B Distribucion IS Dañado CF 35
AND Fallas distribucion := Fallas distribucion + CONF(Sensor de detonacion OF B Distribucion IS Dañado)
AND sendetona dañado OF desplegar cfs dis := CONF(Sensor de detonacion OF B Distribucion IS Dañado)
ELSE Sensor de detonacion OF B Distribucion IS Dañado CF 0
AND sendetona dañado OF desplegar cfs dis := CONF(Sensor de detonacion OF B Distribucion IS Dañado)

Al igual que con las de AG, las variables de COFA en las clases *desplegar cfs* se inicializan con cero.

Existen casos en los que se combinan las cuatro estructuras de fallas, como es el de *Cable modulo DIS a ECM dañado*, que podemos llegar a este diagnóstico, por tres combinaciones de síntomas diferentes, por analizador de gases o por códigos de falla. En este caso se maneja todo igual, con la excepción, de que la acción de agregar y desplegar *CF* para las reglas de síntomas, no se hace en éstas, sino que se hace en el ***DEMON 1 Agregar cfs maximos a las sumas de fallas***.

Así es como queda integrado el conjunto de reglas de producción; como podemos observar los DEMONS *ver cfs, agregar cfs maximos a la suma de fallas* y *ejecutar codigo de colores*, se crearon solamente para ayudar a presentar los resultados.

Este último DEMON tiene como función ejecutar un WHEN CHANGE, que mostrará el código de colores a partir de los resultados obtenidos después de la ejecución de todas las reglas. Este método se explicará más adelante.

5.4 Diagnóstico por Sistemas

Como ya se había planteado, el diagnóstico por sistema, es el resultado que se obtiene del primer nivel de diagnóstico. Éste indica qué subsistemas tienen mayor posibilidad de falla. Para conseguirlo se asocia un valor a cada subsistema que nos indica esta posibilidad. El subsistema que tenga un valor mayor será en el que muy probablemente se encuentre la falla buscada.

Estos resultados se mostrarán al usuario empleando un código de colores, para facilitar su identificación, puesto que para él será más sencillo guiarse por un color, que estar verificando qué subsistema tiene un número mayor asociado.

Para esto, se emplea el WHEN CHANGED de la variable *EX OF Colores indices y vars*, la cual cambia de valor con el *DEMON 2 Ejecutar codigo de colores* que se muestra a continuación

DEMON 2 Ejecutar codigo de colores
IF selected OF pushbutton Diagnostica IN
THEN EX OF Colores indices y vars := TRUE

Al hacerse verdadera la variable *EX OF Colores indices y vars* se ejecuta el WHEN CHANGED cuya función es ordenar los resultados obtenidos de mayor a menor para jerarquizar a los subsistemas y asignarle un color según el lugar que ocupen, así el mas probable tiene un color verde, el que le sigue un amarillo y así sucesivamente. En el apéndice se muestra el código de este WHEN CHANGED.

Con esta forma de presentar los datos no se busca obligar al usuario a aprenderse un código de colores, sino que identifique el subsistema deseado rápidamente guiándose por el color. Pero si no está familiarizado con los colores, tendrá confusión, por lo que junto con el color se muestra un número, no el obtenido por los *CF*, sino el que representa en qué lugar quedó el subsistema en la jerarquización; así mientras se familiariza podrá guiarse por este número.

Con esta información se construyó una pantalla en donde se muestran todos los subsistemas en forma de botón, que al oprimirse se pasa al segundo nivel de diagnóstico, como se verá en el capítulo 6.4.

Otro métodos empleados para el funcionamiento general del sistema experto son:

- WHEN CHANGED Al dar marcha OF Sintomas
- WHEN CHANGED En baja OF Sintomas
- WHEN CHANGED En alta OF Sintomas
- WHEN CHANGED Año leido OF Lectura del Analizador de gases
- WHEN CHANGED Busca datos OF Lectura del Analizador de gases

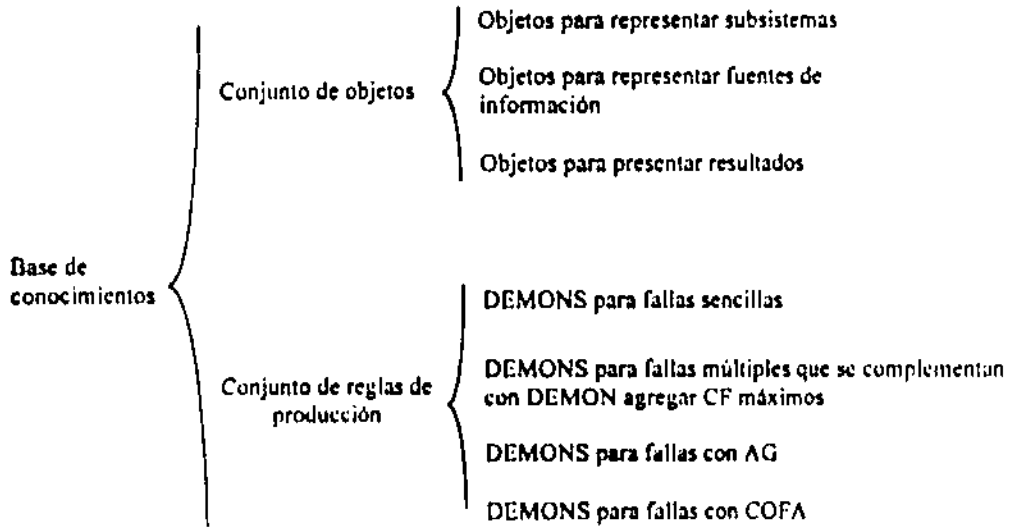
Los tres primeros se emplean para verificar que la combinaciones de síntomas sean adecuadas, en caso de que se detecte una inconsistencia, se mostrará un mensaje de error, como es el caso de los síntomas *no da marcha* y *da marcha pero no arranca* en el que no se pueden dar al mismo tiempo.

Cada uno de ellos estos WHEN CHANGED se activará cuando cambie la variable a la que están asociados, es decir que, cada que se introduzca un síntoma verificará que las combinaciones efectuadas sean congruentes. Dichas combinaciones sólo se verifican en cada estado de operación y no entre ellos, porque hay ocasiones en los que el funcionamiento del automóvil en determinado estado de operación es aleatorio, es decir, algunas veces funciona en un estado de operación, y no vuelve a funcionar en el mismo una segunda vez. En el apéndice se muestra el código de estos WHEN CHANGED.

El WHEN CHANGED Año leido OF Lectura del Analizador de gases, se encarga de asignar el año proporcionado por el usuario a un año compatible con los existentes en la base de datos. Su código se presenta en el apéndice.

El WHEN CHANGED Busca datos OF Lectura del Analizador de gases, se encarga de alistar la base de datos para poder ocupar la información que contiene(ver apéndice).

En el siguiente cuadro se muestra cómo queda finalmente construida la base de conocimientos:



Los DEMONS y métodos que se emplean para procesos auxiliares son:

- DEMON ver Cfs (diez DEMONS, uno por subsistema)
- DEMON ejecutar código de colores
- WHEN CHANGED para combinaciones no validas (Al dar marcha OF Sintomas, En baja OF Sintomas, En alta OF Sintomas)
- WHEN CHANGED para asignar año a base de datos (Año leído OF Lectura del Analizador de gases)
- WHEN CHANGED para determinar estado del gas (cuatro por cada uno de los gases)
- WHEN CHANGED para abrir la base de datos (Busca datos OF Lectura del Analizador de gases)
- WHEN CHANGED para código de colores (EX OF colores indices y vars)

La manera en cómo interactúan el conjunto de reglas para proporcionar el diagnóstico de primer nivel, se muestra gráficamente en la siguiente figura:

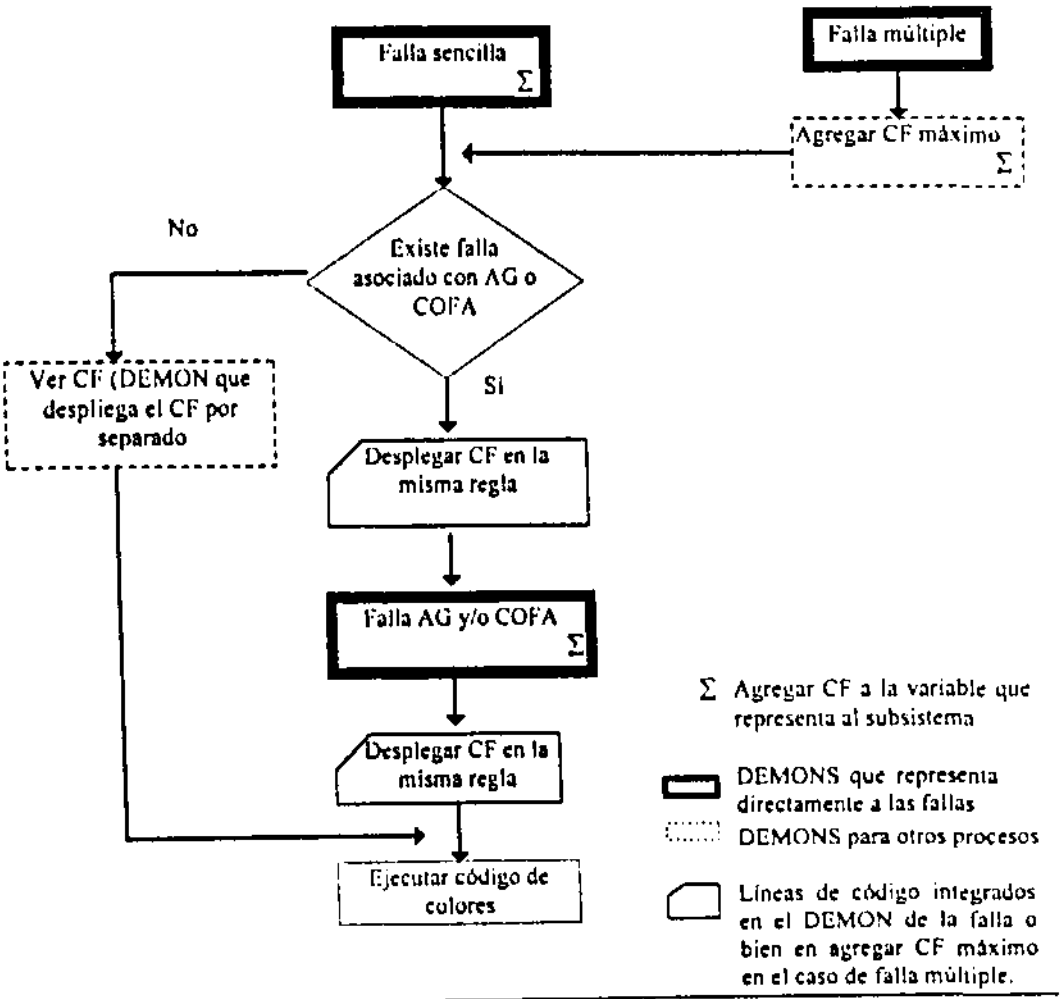


Figura 5.5 Diagrama de flujo de las reglas.

En el siguiente capítulo se explica cómo efectuar un diagnóstico.

CAPÍTULO 6: NAVEGACIÓN

Este capítulo muestra el funcionamiento de SEDA a nivel usuario. SEDA opera bajo un ambiente Windows, lo cual facilita su manejo. La navegación a través de él es muy sencilla y amigable, maneja seis tipos de ventanas: las de presentación , el menú, las de captura de información, la de diagnóstico por sistemas y presentación de resultados. En este capítulo se explicará en qué consisten y cuál es su función.

6.1 Cómo ejecutar SEDA

Para usar SEDA se debe contar con los siguientes requisitos de hardware y software, para asegurar un funcionamiento eficiente.

Hardware

- Una computadora con microprocesador 80486 o posteriores (por lo menos 66 MHZ.)
- 8 Megabytes de RAM.
- 5 Megabytes en disco duro disponibles.
- Monitor SVGA (configurado con resolución media, 800 x 600 pixeles)

Software

- Microsoft Windows 3.1 o posterior.
- Level 5 Object Run-Only System. Es una versión reducida de Level 5 Object que contiene los elementos necesarios para únicamente ejecutar una aplicación (extensión KNB). Es decir, que no contiene elementos de edición ni compilación. Los 4 MB de disco duro que se especifican anteriormente ya consideran la existencia de este software.

Una vez que se cuenta con los medios necesarios se puede instalar SEDA. Para instalarlo se deben seguir los siguientes pasos:

➔ Crear un directorio "seda" en el directorio raíz.

c:\seda

➔ Dentro de este directorio crear un subdirectorio con el nombre de "runonly".
c:\seda\runonly

➔ Copiar los archivos contenidos dentro del archivo **seda.zip** dentro del directorio **seda**.

➔ Copiar los archivos contenidos en el archivo **runonly.zip** dentro del subdirectorio **runonly**.

Con los archivos copiados se tiene la aplicación (SEDA_1N.KNB) y el software para ejecutarla (L5RO.EXE)

Es importante que las rutas sean creadas como se especifica, porque éstas se manejan internamente y al no detectarse los archivos se generarán errores.

Para poner a funcionar SEDA se necesita ejecutar primero el RUNONLY (c:\seda\runonly\L5RO.EXE) y dentro de este indicar el nombre de la aplicación que se desea ejecutar (c:\seda\SEDA_1N.KNB). Para hacer transparente este proceso, se creó una aplicación en Visual Basic con carácter de ejecutable, de tal forma que solo se necesita ejecutar un archivo (c:\seda\SEDA_1N.EXE) para poder iniciar la navegación por SEDA.

Como ya se mencionó SEDA trabaja bajo un ambiente Windows y contiene características propias de éste, como son botones, barra de herramientas y menús.

Durante la explicación de la navegación se emplearán términos como:

- *Dar clic*, que significa presionar el botón izquierdo del mouse.
- *Colocar cursor*, quiere decir que se coloca el puntero del mouse donde se desea escribir y se da clic, entonces aparecerá el cursor parpadeando lo que indica que ya se puede proporcionar la información.
- *Pulsar botón*, indica que se debe dar clic sobre alguno de los botones que se visualizan en la ventana.

6.2 Inicio

Una vez que se está dentro de SEDA, se visualizarán dos ventanas iniciales. La primera muestra los datos de la Tesis y se le denominó *ventana de presentación* (figura 6.1).

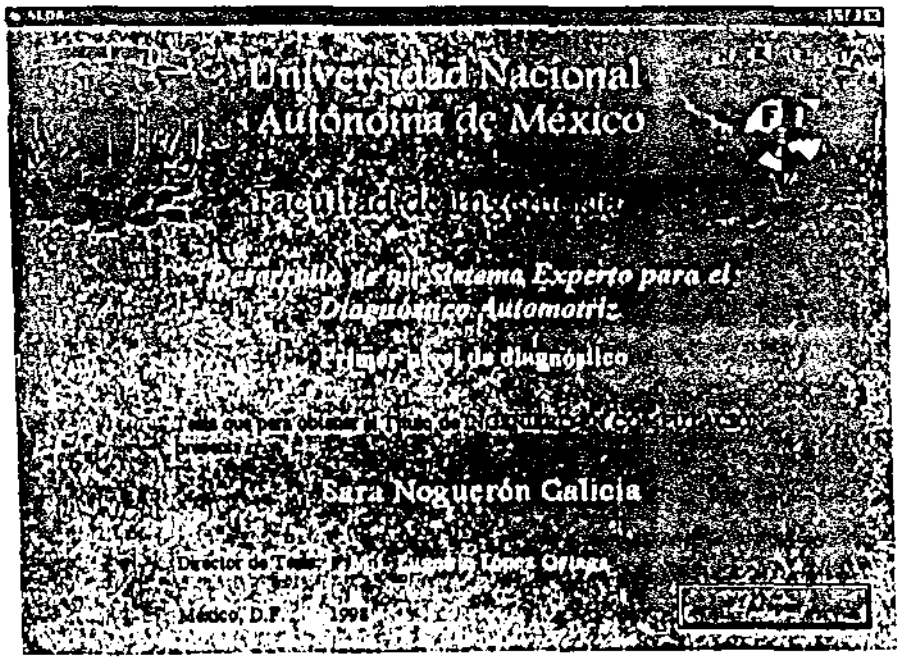


Figura 6.1. Ventana de presentación.

En la esquina inferior derecha de esta ventana se encuentra el botón **Aceptar** que al ser pulsado conduce a la segunda ventana inicial (*ventana principal*)

SEDA realiza el primer nivel de diagnóstico siguiendo la metodología que se describió en el capítulo 3.2.2 (ver figura 3.5). Primero se introducen los datos del automóvil, en este caso el año porque la marca es un dato fijo, dado que este SE sólo es para autos General Motors. Posteriormente se introducen los síntomas y mediciones generales, para después realizar el diagnóstico.

La siguiente ventana, a la que se le llamó *ventana principal* (figura 6.2), sirve para introducir los datos del automóvil. En ella se distinguen dos recuadros, en el del lado izquierdo se tiene la presentación del sistema y en el del lado derecho se encuentran los datos del automóvil.

En el pequeño recuadro que está frente a "del Año :" se debe teclear el año del automóvil que se va a diagnosticar. Si este dato no se proporciona, la información que se extrae de la base de datos no se actualizará, por lo que el diagnóstico que se arroje puede ser erróneo.

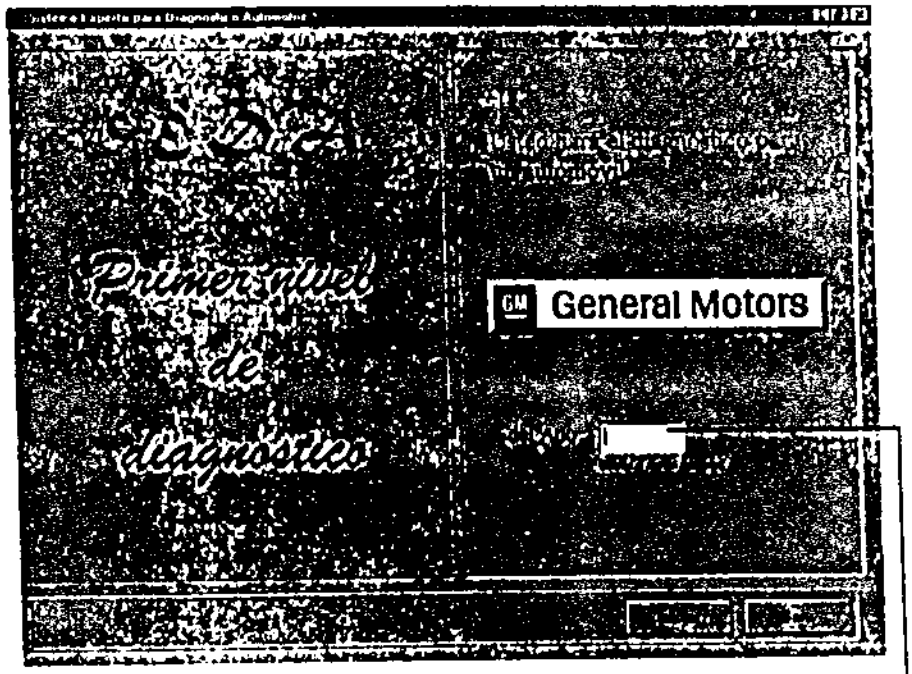


Figura 6.2 Ventana Principal (segunda ventana de presentación)

En la parte inferior derecha de esta ventana se encuentran dos botones. Al pulsar el botón de **iniciar** se pasa a la siguiente ventana . El botón **salir** interrumpe el proceso cerrando la aplicación.

6.3 Captura de síntomas

Cuando se pulsa el botón **iniciar** de la *ventana principal* se pasa a la *ventana de menú*. Esta ventana contiene una barra de menús, una barra de herramientas y un conjunto de botones que dan ordenes al sistema (figura 6.3)

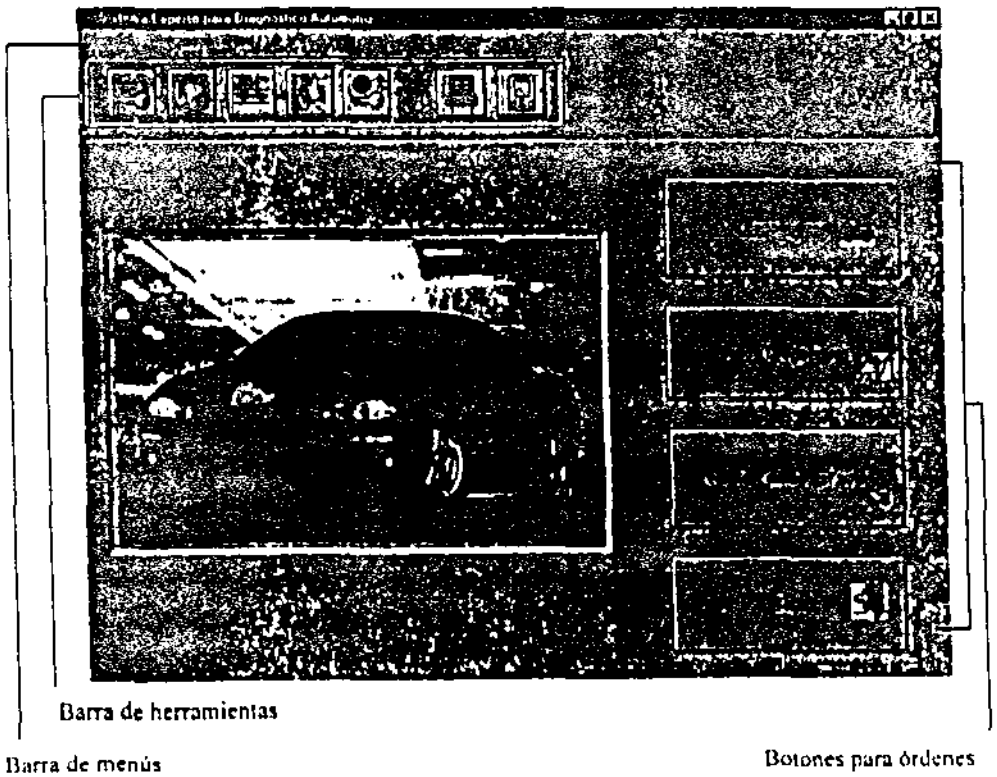


Figura 6.3. Ventana menú

La *ventana de menú* tiene varios propósitos. Es la ventana central que permite dirigirse a las ventanas restantes, las acciones que se pueden realizar en ella son las siguientes:

- ✓ Iniciar la captura de síntomas y mediciones generales.
- ✓ Iniciar el diagnóstico de primer nivel.
- ✓ Hacer un nuevo diagnóstico.
- ✓ Verificar los datos del automóvil.

✓ Salir del sistema.

✓ Para iniciar la captura de síntomas y mediciones generales se emplea la barra de herramientas o bien las opciones del menú **Información**. Los botones que componen a la barra de herramientas representan a los diferentes estados de operación, y a las mediciones generales. De izquierda a derecha los botones son los siguientes:

Estados de operación:

 **Al dar marcha**

 **En baja**

 **Al operar**

 **En alta**

 **Al apagar**

Mediciones generales:

 **Analizador de gases**

 **Códigos de falla**

Este mismo orden se conserva para el menú **información** (figura 6.4).

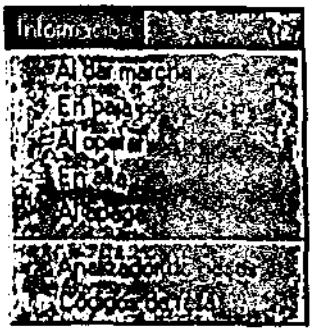


Figura 6.4 Menú **Información** de ventana de menú

Al pulsar algunos de estos botones (u opciones en el caso del menú **información)** se llegará a una ventana nueva en donde se dan opciones al usuario para que introduzca la información necesaria para realizar un diagnóstico.

✓ Para iniciar un diagnóstico, primero se debe capturar la información para después pulsar el botón **diagnosticar** o la opción **diagnosticar** del menú **diagnóstico** (figura 6.5) y SEDA iniciará el proceso de diagnóstico.

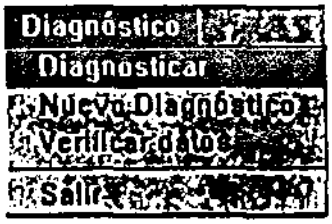


Figura 6.5 Menú **Diagnóstico** de ventana de menú.

✓ Para hacer un nuevo diagnóstico se pulsa el botón **Nuevo diagnóstico** que conduce a la ventana principal, borrando toda la información que se manejó con el diagnóstico anterior. Esta misma acción se puede realizar usando la opción **Nuevo diagnóstico** del menú **Diagnóstico**.

✓ Para verificar los datos del automóvil se pulsa el botón **Verificar datos** o bien usando la opción del mismo nombre en el menú **Diagnóstico**, ambos conducen a la ventana principal conservando toda la información que se está manejando en el diagnóstico presente (año, síntomas y mediciones).

✓ Para salir del sistema se pulsa el botón **Salir** que termina la ejecución del Sistema Experto.

Las ventanas que se emplean para la captura de información son siete y tienen la siguiente estructura:

♦ Para los estados de operación:

Contienen la misma barra de herramientas de la *ventana de menú*, con la finalidad de que el usuario no se vea forzado a regresar a ésta para proporcionar más información. Estas ventanas no cuentan con barra de menús, en su lugar está la inscripción **OK!**, que al ser pulsada, indica al sistema que se ha finalizado la captura de información y regresa a la *ventana menú*. La única forma de salir de estas ventanas es mediante el **OK!**.

En la figura 6.6 se observa que bajo la barra de herramientas, la ventana se divide en dos partes. La parte izquierda indica el estado de operación del que se trata (al igual que un recuadro que rodea al botón correspondiente de la barra de herramientas) y proporciona una pequeña explicación, señalando bajo qué condiciones se considera que el automóvil está en dicho estado de operación.

La parte de la derecha muestra los síntomas que se dan en ese estado de operación, de los cuales el usuario deberá elegir los que presente el automóvil que está diagnosticando, dando clic sobre ellos. Al hacer esto, los síntomas seleccionados se marcarán con una cruz dentro del pequeño cuadro que se encuentra a su izquierda.

Algunos síntomas no se pueden presentar simultáneamente, por ejemplo: *no da marcha* y *da marcha pero no arranca* . Si estos síntomas se dan al mismo tiempo, hay una contradicción, es por eso que el sistema despliega un mensaje como el de la figura 6.7, informando que se ha dado un error de este tipo. Se debe entonces descartar uno de los dos síntomas. Para esto se da clic nuevamente sobre el síntoma que ya no se desea considerar y la cruz del cuadro desaparecerá.

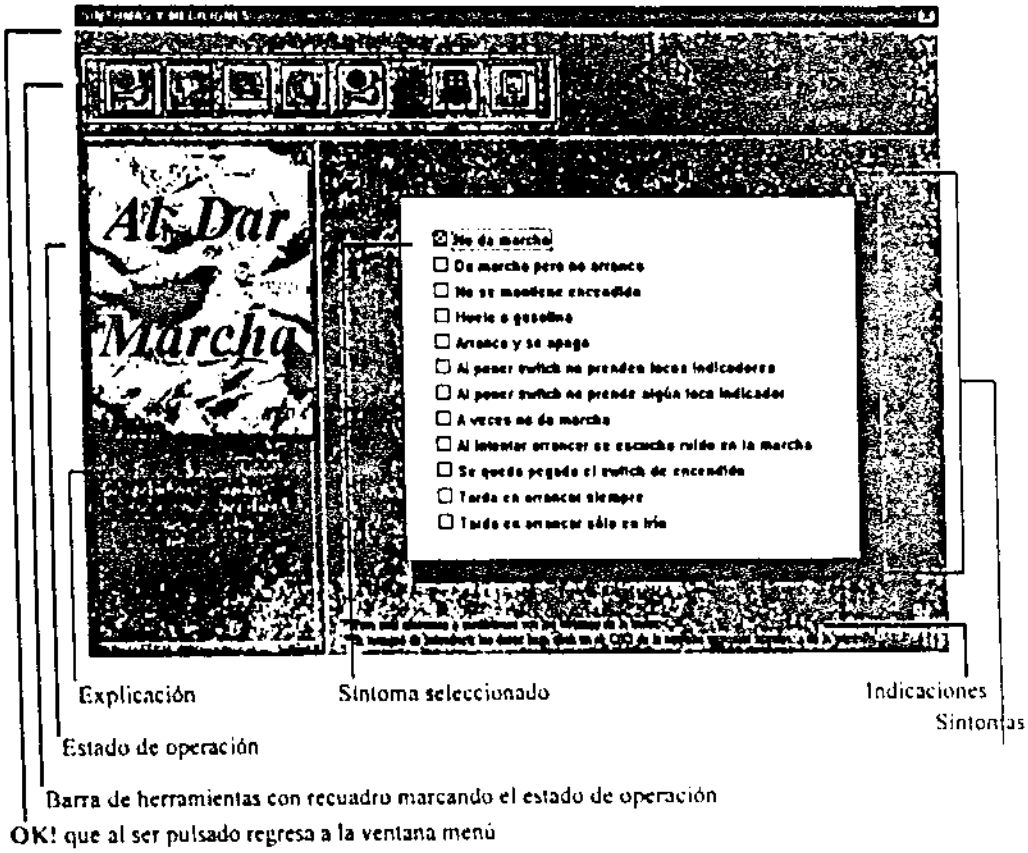


Figura 6.6 Formato de las ventanas de síntomas. Ventana al dar marcha.

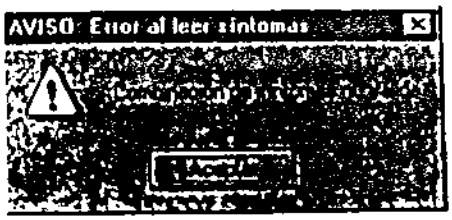


Figura 6.7 Mensaje de error al leer síntomas.

Las cinco ventanas que representan a los síntomas por estado de operación tienen el mismo formato que se ha descrito. Cambiando solamente el nombre del estado de operación con su explicación y los síntomas que se presentan.

♦ Para las mediciones generales

La estructura es básicamente la misma. Contienen la misma barra de herramientas, la inscripción **OK!** y las indicaciones de las ventanas anteriores, sólo que en estas ventanas no se dan opciones, sino que se presentan espacios en los que se deben teclear los valores obtenidos con una analizador de gases o con un scanner. En la figura 6.8 se muestra la *ventana de analizador de gases.*

Para introducir los datos en los distintos espacios, se puede usar la tecla ENTER o la tecla TABS hasta llegar al espacio deseado, o bien posicionar el cursor con ayuda del mouse.

Al introducir los datos en los espacios correspondientes, el sistema despliega un letrero indicando cual es la condición del gas en cuestión (alto, muy alto, bajo, muy bajo, irregular o inestable).

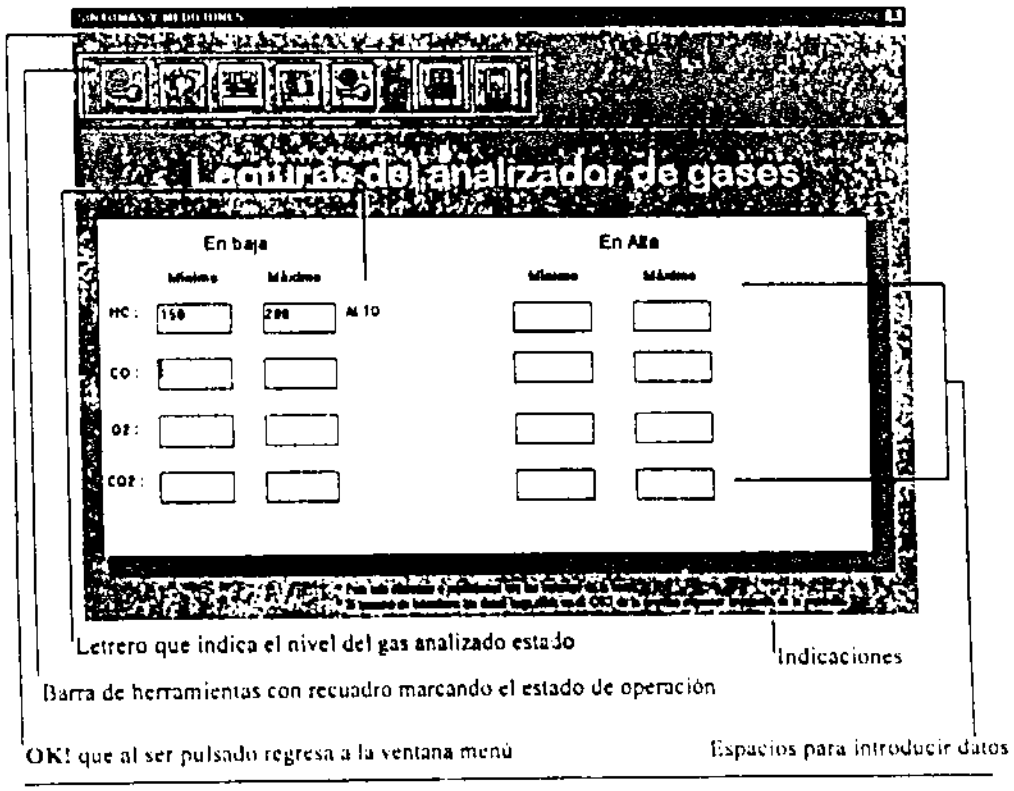


Figura 6.8 Ventana de analizador de gases

Cuando se termina de introducir la información se pasa a otra ventana, usando la barra de herramientas si se desea proporcionar más información o con el **OK!** en caso contrario.

En la figura 6.9 se muestra la *ventana de códigos de falla*. En algunas ocasiones el scanner arroja más de un código de falla, por eso esta ... da la opción de capturar más de uno. La captura de datos se hace de la misma manera que en la *ventana de analizador de gases*.

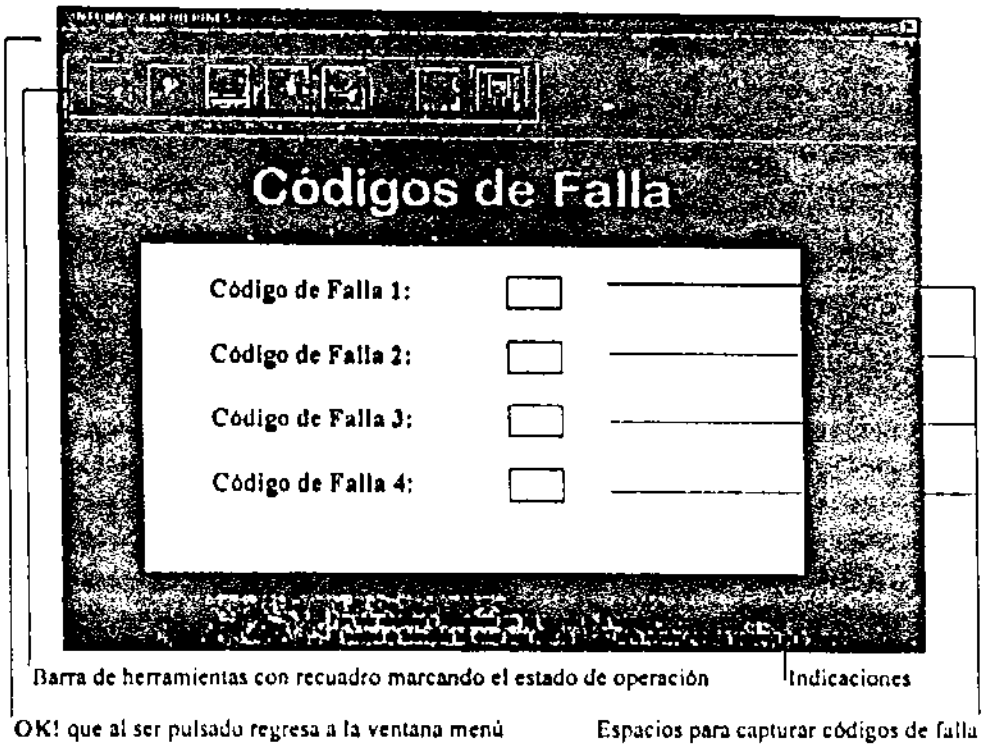


Figura. 6.9 Ventana de códigos de falla.

Hasta ahora se han señalado las características que tienen las primeras ventanas, así como la función de sus componentes. Con esto ya es posible hacer el diagnóstico. El mecanismo que se sigue para efectuar un diagnóstico utilizando SEDA, es similar al que se sigue cuando se hace una compra por catálogo en algún centro de autoservicio. Primero se identifican los elementos que se desea que contenga el paquete mediante un catálogo, que es equivalente a navegar por

las siete ventanas para introducir información. Cuando se han seleccionado las opciones se hace la orden, que en SEDA sería pulsar el botón **Diagnosticar** de la *ventana menú*.

Para hacer un diagnóstico se deben seguir los siguientes pasos:

➔Al entrar a la *ventana principal* se introduce el año del automóvil que se someterá a diagnóstico y se pulsa el botón **Iniciar**, lo que conducirá a la *ventana menú*.

➔Dentro de la *ventana menú*, se pulsa alguno de los botones de la barra de herramientas (o opciones equivalentes del menú **Información**) para situarse en una de las siete pantallas para la captura de información.

➔Si se está en alguna de las ventanas para indicar los síntomas (*al dar marcha*, *en baja*, *al operar*, *en alta* o *al apagar*) se da clic sobre las opciones que mejor describan los síntomas que presenta el automóvil. Si se desea proporcionar síntomas en otro estado de operación, se emplea la barra de herramientas para navegar a través de las distintas ventanas que sirven para el mismo fin.

➔Si se está en la *ventana de analizador de gases* o en la *ventana de códigos de falla*, en las que no hay opciones, el usuario debe colocar el cursor en los espacios correspondientes y teclear los valores obtenidos con el analizador de gases o con el scanner según sea el caso.

➔Cuando se ha concluido la captura de información se pulsa la inscripción OK! situado en el lugar de la barra de menús, con lo que se regresará a la *ventana de menú*, en donde se debe pulsar el botón **Diagnosticar** para hacer el diagnóstico.

Con esta última acción se pasa a una nueva ventana en donde se muestra el diagnóstico del primer nivel.

6.4 Diagnóstico

Al pulsar el botón **Diagnosticar** de la ventana menú, SEDA inicia el proceso de diagnóstico y al concluirlo proporciona los resultados obtenidos, empleando la *ventana de diagnóstico por sistemas* (figura 6.10). Esta ventana muestra los once subsistemas y su calificación asociada que indica qué subsistema tiene mayor posibilidad de falla. Para facilitar su identificación se emplea un código de colores que jerarquiza a los subsistemas.

Para que el usuario no se vea obligado a aprenderse el código de colores, a cada color se le relaciona con un número del uno al once, de esta manera las primeras veces que se utilice el sistema, el usuario se guiará por el número, pero después de un tiempo, identificará mejor al subsistema por medio de los colores. A los primeros seis sistemas se les asigna un color diferente al blanco, después del sexto, si existe posibilidad de falla en algún sistema, esta será muy baja en comparación con los primeros por lo que se les asigna el color blanco. A los subsistemas que no presenten posibilidad de falla (calificación igual a cero) no se les asigna ningún color.

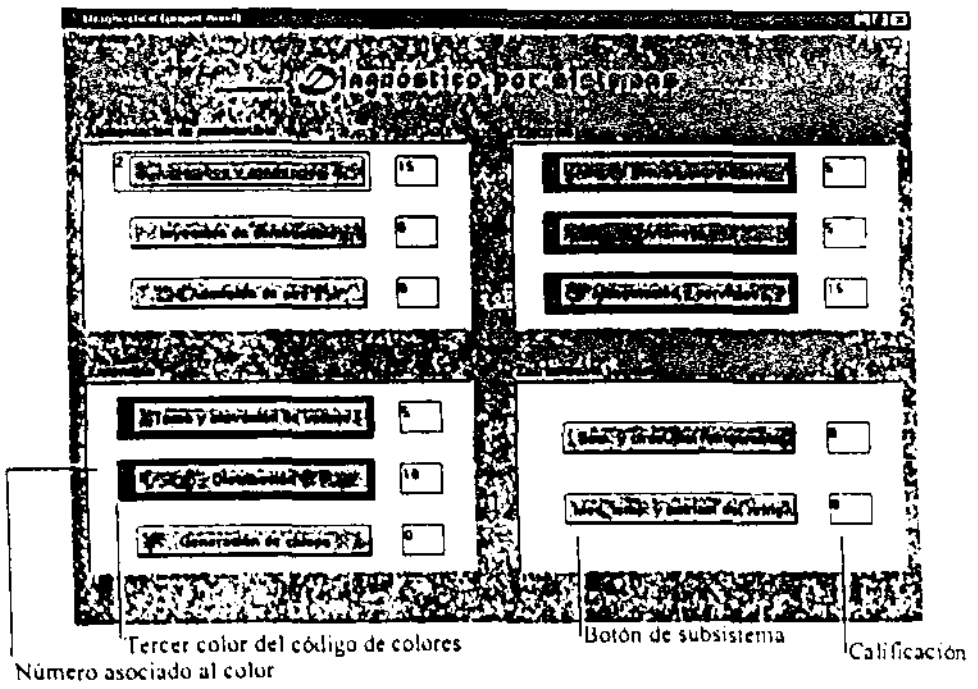


Figura 6.10. Ventana de diagnóstico por sistemas

Esta ventana al igual que la *ventana menú* contiene el menú **Diagnóstico** , pero presenta distintas opciones como se observa en la figura 6.11.

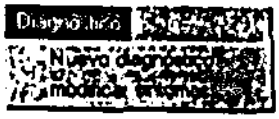


Figura 6.11 Menú Diagnóstico de la ventana diagnóstico por sistemas

La opción **Nuevo diagnóstico** conduce a la *ventana principal* para iniciar un nuevo diagnóstico, desechando toda la información manejada en el diagnóstico previo.

La opción **Modificar síntomas...** conduce a la *ventana menú* para seleccionar, con la barra de herramientas, el estado de operación en el que se desea marcar más síntomas o bien descartar alguno.

Si se consideraran los dos niveles de diagnóstico, con esta ventana se concluiría el primero y al pulsar alguno de los botones de los subsistemas se pasaría al segundo nivel. Antes de pasar a dicho nivel es necesario, asegurarse de que los resultados arrojados por el primer nivel son correctos. Para esto se diseñaron dos tipos de ventanas de presentación de resultados con las que se trabaja con el experto para hacer los ajustes necesarios. Estas ventanas no se desarrollaron para ser vistas por el usuario, por lo que se les denomina *ventanas auxiliares*.

Las *ventanas auxiliares* despliegan información de las reglas que se hicieron verdaderas, es decir las posibles fallas, de dos formas: en la primera, mediante la *ventana de fallas*, se muestran las partes y los problemas; y en la segunda, mediante la *ventana de CF*, se muestran las partes con sus problemas y además los CF asociados a cada falla.

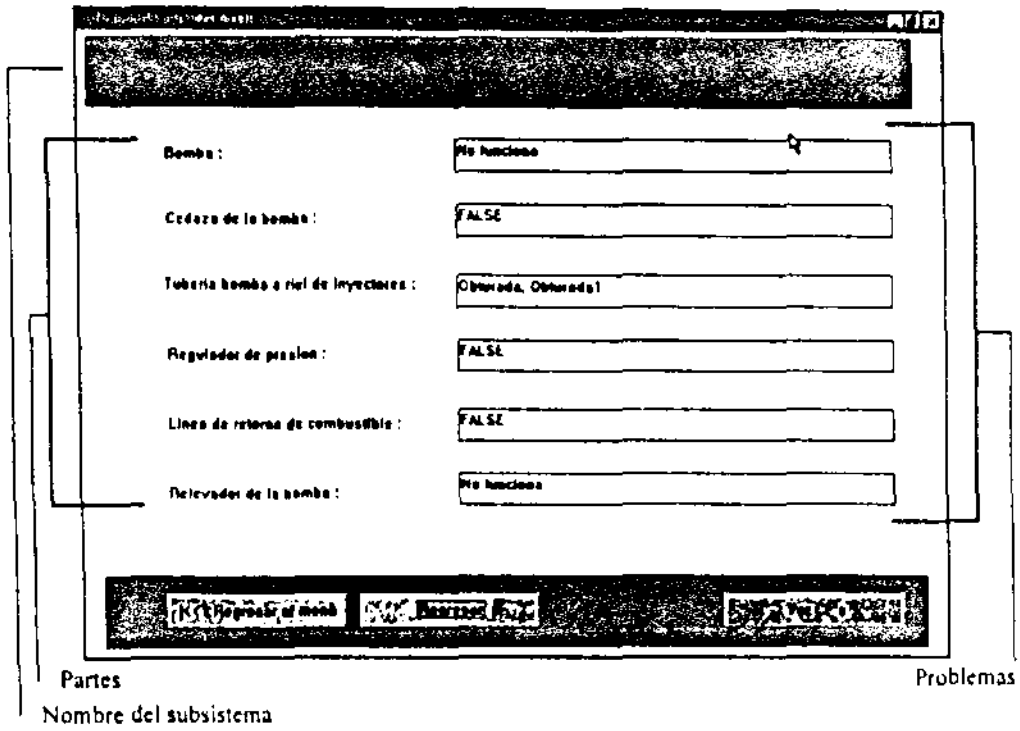


Figura 6.12. Ventana de fallas del subsistema bombeo y conducción.

En la *ventana de fallas* (figura 6.12) se observan, en la parte inferior, tres botones que realizan las siguientes acciones:

Regresar al menú : regresa a la *ventana menú.*
Regresar: regresa a la *ventana diagnóstico por sistemas.*
Ver CF'S : conduce a la *ventana de CF* correspondiente al subsistema del que se trate.

En la figura 6.13 se observa la *ventana de CF* para el subsistemas de bombeo y conducción. En ella se muestran las partes y los problemas representados de otra forma. En la parte superior de un recuadro se encuentra el nombre de una parte y dentro de éste todos los problemas que puede presentar. Para identificar a los problemas que se presentan de acuerdo a la información

proporcionada por el usuario, se marcan con una cruz dentro del pequeño cuadro que esta a su izquierda.

Cada uno de los problemas que están dentro de los recuadros representan a una regla de producción. Entonces, con esta pantalla se pueden ver las reglas que se hicieron verdaderas y los CF que se les asignó. El CF asignado puede ser por síntomas, por analizador de gases o por códigos de falla.

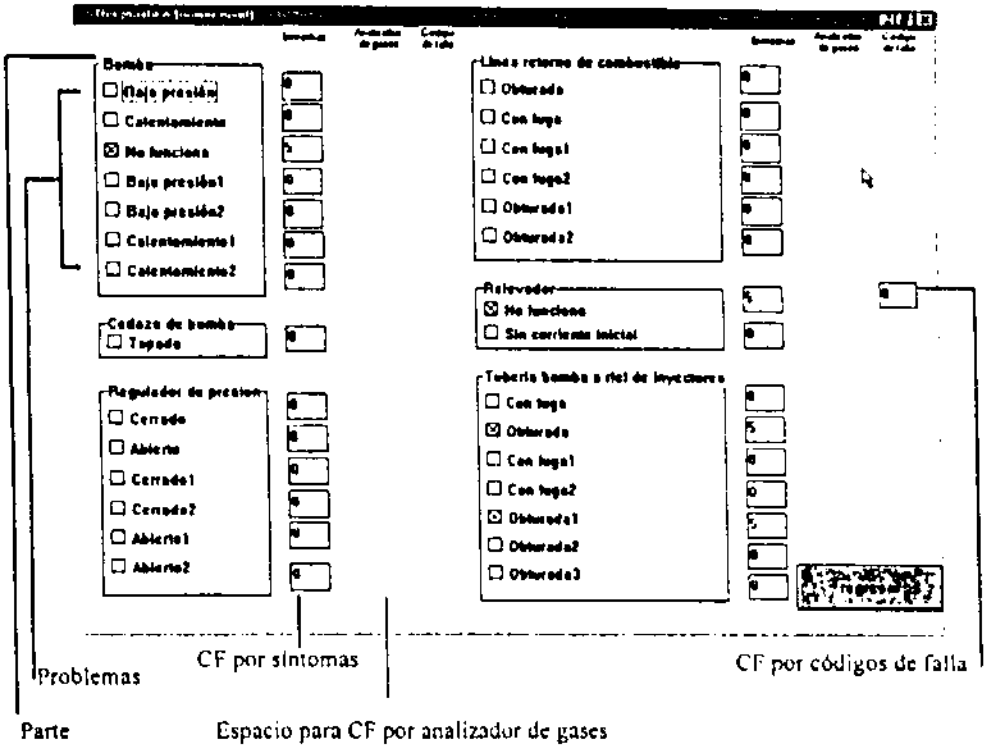


Figura 6.13 Ventana de CF del subsistema bombeo y conducción

En la esquina inferior derecha de estas pantallas se localiza el botón **regresar,** que conduce a la ventana de fallas del subsistema correspondiente.

Por cada uno de los subsistemas hay una *ventana de fallas* y una *ventana de CF*, y por medio de ellas se pueden analizar los resultados obtenidos.

A continuación se muestra de forma gráfica la relación entre las pantalla.

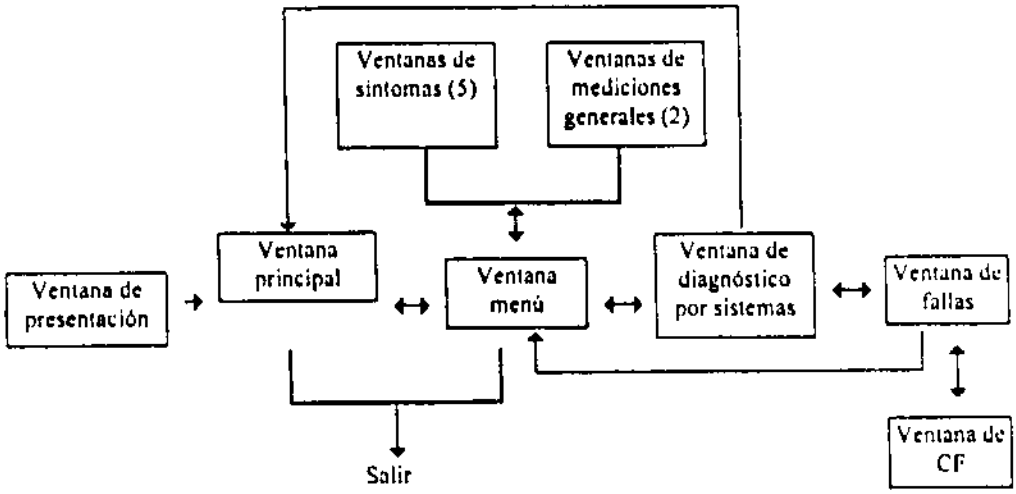


Figura 6.14 Navegación del primer nivel de diagnóstico

CAPÍTULO 7: CONCLUSIONES.

Los Sistemas Expertos (SE) son una herramienta muy poderosa para solucionar problemas, en los el conocimiento está centralizado en unos cuantos expertos, que toman sus decisiones basándose en factores generalmente ambiguos y totalmente dinámicos.

Los SE imitan el comportamiento de los expertos, permitiendo así, tener el conocimiento para la solución de problemas específicos, disponible en cualquier momento, a través de una computadora, sin tener la necesidad de contar con una persona especializada para la toma de decisiones.

En la mayoría de los casos, los Sistemas Expertos requieren de un largo tiempo de desarrollo y SEDA no es la excepción. Una de las tareas más complicadas durante la elaboración de SEDA, ha sido la adquisición del conocimiento, porque no es fácil para el experto transmitirlo y con frecuencia se desvía del problema que se le plantea. Esta situación ocasiona que el proceso de adquisición de conocimiento sea tardado y su estructuración compleja.

SEDA será empleado por personal de talleres mecánicos, que no cuenten con una amplia capacitación y experiencia para la reparación de automóviles con tecnología *fuel injection*. Tiene dos propósitos: emplearse para la capacitación y emplearse para el diagnóstico.

La presente tesis tiene como objetivo desarrollar el primer nivel de diagnóstico, considerando el alcance que se planteó en el capítulo 1.2, obteniendo así la base para la elaboración del resto de los sistemas expertos que conformarán a SEDA.

Una parte importante del primer nivel de diagnóstico son los factores de certidumbre, de ellos depende la correcta identificación del subsistema en el que se encuentra la falla. Entonces el conocimiento experto no se limita solamente a los síntomas y mediciones que conducen a una falla, sino que también involucra grados de certidumbre.

Las aportaciones que se hacen con la elaboración de esta tesis son:

Se plantean los conceptos básicos de los Sistemas Expertos, con lo que se logra brindar una introducción a está rama de la Inteligencia Artificial.
Se abre un panorama sobre la gran aplicación que tienen los Sistemas Expertos en la industria automotriz.

Se presenta una explicación clara y sistémica de la estructura y funcionamiento de un automóvil.

Se brinda una introducción a la programación orientada a objetos explicando sus conceptos principales, disipando así la idea de complejidad a la que generalmente se le asocia, por las diferencias que presenta ante los métodos convencionales de programación.

Se resalta la utilidad que tiene la programación orientada a objetos en el desarrollo de Sistemas Expertos.

Se plantea una metodología para el desarrollo de SEDA efectiva y flexible, que se acopla muy bien a las necesidades de éste y que será muy útil para su continuación y para problemas que tengan el mismo enfoque.

Se plantea el diseño para ambos niveles de diagnóstico. Se identifican los tipos de encadenamiento para cada nivel y la estructura de objetos y reglas de producción que se necesitan.

Se hace uso de un concepto muy importante dentro de la teoría de los Sistemas Expertos, los *factores de certidumbre*, para modelar adecuadamente el diagnóstico de un experto.

Se proporciona un Sistema Experto desarrollado con un lenguaje orientado a objetos que realiza el primer nivel de diagnóstico formando la base de SEDA. A partir de éste la creación de los bloques restantes para su culminación será más sencilla y se brindará una herramienta muy poderosa a la industria automotriz, para la realización de diagnósticos.

Se proporciona un software destinado a mecánicos automotrices que no cuentan con una amplia formación en cómputo, por lo que tiene una interfaz de usuario elaborada bajo ambiente Windows lo que lo hace muy amigable y de fácil navegación.

Las ventajas se que ofrecen con la elaboración de la presente tesis son:

Se tiene un Sistema Experto que determina el subsistema en el que se encuentra la falla, acotando así el camino a seguir para concluir un diagnóstico.

Es un Sistema Experto muy amigable que contiene una etapa de ajuste en donde se pueden ver los resultados obtenidos, que normalmente para el usuario serán transparentes, pero que para el experto y los desarrolladores resulta de gran utilidad
Con la etapa de presentación de resultados para el ajuste se puede monitorear fácilmente el comportamiento del Sistema Experto y tomar medidas ante resultados erróneos o inesperados.

Con la elaboración de este Sistema Experto se obtiene la estructura básica que se deberá seguir para su crecimiento.

Con el diseño que se propone se logra obtener un enfoque modular respecto a marcas. Esto significa que, se podrán crear varios Sistemas Expertos bajo la misma estructura, y cada uno representará a una marca de automóviles diferente, lo que permitirá su comercialización por partes, dependiendo de los requerimiento del cliente.

Con el diseño planteado para el segundo nivel de diagnóstico, es posible desarrollar los módulos que lo compondrán (un módulo por subsistema).

SEDA proporcionará ayuda para el diagnóstico y subsanará la necesidad de expertos, ayudando a la capacitación técnica de personal para poder realizar diagnósticos.

Las principales limitaciones a las que se enfrenta el Sistema Experto desarrollado son:

Se tiene solamente la primera etapa del diagnóstico con la que no se puede llegar a una falla concreta.

La base de conocimiento no incluye el conocimiento experto para todos los subsistema que componen al automóvil.

No cuenta con un sistema de ayuda muy amplio, que podría ser necesario para algunos usuarios, a pesar de que su uso es sencillo.

Para obtener una base de conocimientos completa se necesita de más tiempo, ya que ésta requiere de mucha información. Además es necesario que los desarrolladores tengan conocimientos de programación orientada a objetos y de mecánica automotriz básica.

Se necesita de un largo periodo de prueba para detectar posibles comportamientos alejados de la realidad y es mejor realizarlas una vez que se tiene concluida la base de conocimientos.

El conocimiento experto que hasta ahora se ha recabado e implementado forma los cimientos de un proyecto muy amplio que sin lugar a dudas dejará muy buenos frutos. Para concluirlo se necesita aún varios aspectos. Las perspectivas de desarrollo son:

Introducir la información de los subsistemas restantes para construir completamente la base de conocimientos del primer nivel de diagnóstico.

Diseñar un sistema de ayudas más amplio y que vaya acorde con el tipo de usuarios.

Conjuntar el primer nivel de diagnóstico con el segundo y realizar las pruebas pertinentes para asegurar el funcionamiento eficiente de SEDA.

Hacer los ajustes necesario para que este Sistema Experto, sea adecuado para otras marcas de automóviles y no solamente para General Motors.

Para la integración de los dos niveles de diagnóstico y de los Sistema Expertos para las otras marcas se recomienda diseñar una interfaz en otro lenguaje de programación (empleado como *Front-End*) con la finalidad de crear una unión robusta, ya que Level5 Object no soporta la integración de varias aplicaciones con bases de conocimiento muy grandes.

Diseñar una interfaz de hardware para la captura de información directamente de los instrumentos de medición y de la computadora del automóvil.

Como se puede apreciar, para la llegar a la culminación de SEDA faltan varios elementos. Sin embargo, se considera que con el prototipo desarrollado se tiene ya la mayor parte, porque se han trabajado los subsistemas más complejos tomando como base automóviles General Motors que tiene la estructura más completa. Además, se cuenta ya con todo el diseño de SEDA tanto para el primer nivel como para el segundo.

BIBLIOGRAFÍA Y REFERENCIAS.

Capítulo 1

Hayes-Roth F. , Waterman D., Lenat P.
Building Expert System, an overview of expert system
Addison Wesley
London 1983

Sistema Práctico de aprendizaje Fuel Injection
MEX-FEY SA de CV

Capítulo 2

Hayes-Roth F. , Waterman D., Lenat P.
Building Expert System, an overview of expert system
Addison Wesley
London 1983

Lara, F. y Gelman, J.
Métodos y modelos de la ingeniería del conocimiento para Sistemas Expertos
Informe técnico, Instituto de Ingeniería, UNAM
México 1989

David W. Rolson
Principios de Inteligencia Artificial y Sistemas Expertos
Mc-Graw Hill

Efraim Turban
Expert System and applied artificial intelligence
Macmillan

Giarratano, J. and Riley, G.
Expert System principles and programming
PWS Publishing Company
Segunda Edición, 1994

J. P Sánchez y Beltrán
Sistemas Expertos una metodología de programación
Macrobit, 1990

Robert I. , Diane E. Drang, Barry Edelson
A Comprenhensive Guide to AI and Expert Systems
Mc-Graw Hill 1987

Dr. Felipe Lara Rosano
Apuntes de la materia: Inteligencia Artificial.

Manual de datos técnicos para motores a gasolina (fuel injection)
TF Victor, S.A. de C.V.
Décima edición

Sitios WEB:

http:// seraphim.csee.usf.edu ➔ ISL Intelligent Systems Laboratoty

http:// www.swi.psy.uva.nl/mailing-lists/kaw/archives/0439.html ➔ Expert Systems 96

http:// www.best.com/~worktree/g/90/121g.html ➔A distributed Fuzzy System Model For Automotive Diagnosis

http:// www.lmco.com/lmtoday/0496/chrysler.html ➔ Lockheed Martin Corporation

Capítulo 3

Manual de motores y vehículos Motorizados
Ministerios del ejercito y la fuerza aerea de los EUA
Ed. Continental SA de CV México
Décimo segunda impresión 1987

Frederick C. Nash
Fundamentos de Mecánica automotriz
Diana

F. Aparicia Izquierdo
El mecánico de Automóviles
Paraninfo
Cuarta edición
España 1992

DJ Leeming y M. Howart
El motor del automóvil, conocimientos básicos
Publicaciones marcombo, SA
México 1988

Sistema Práctico de aprendizaje Fuel Injection
MEX-FEY SA de CV

El libro del automóvil
Selecciones del Reader's Digest
Reader's Digest México S.A. de C.V.

En marcha! Servicio y reparación de su automóvil
Selecciones del Reader's Digest
Reader's Digest México S.A. de C.V.

Manual de datos técnicos para motores a gasolina (fuel injection)
TF Victor, S.A. de C.V.
Décima edición

Capítulo 4

Robert Keller
Expert System Technology, development & application
Yourdon Press
1987

J. P Sánchez y Beltrán
Sistemas Expertos una metodología de programación
Macrobit, 1990

Ann L. Winblad, Samuel D. Edwards y David R. King
Software Orientado a Objetos
Ed. Addison Wesley

Keith Weiskamp, Terry Hengl
Artificial Intelligence Programming with turbo Prolog
John Wiley & sons, Inc. 1988

Peter Jackson
Introduction to Expert Systems
Addison Wesley

Reference Guide
Level5 Object for Microsoft Windows
Information Builders
1994

Dr. Felipe Lara Rosano
Apuntes de la materia: Inteligencia Artificial.

Capítulo 5

Lic. Guillermo Alamilla Esquivel
Manual de taller para inyección electrónica de combustible (fuel injection)
Alamilla Editores
Saltillo, Coah. México

Manual de datos técnicos para motores a gasolina (fuel injection)
TF, Victor S.A de C.V.
Décima edición

Instruction Manual
Four Gas Infrared Analyzer

Mastertech, multifuncional tester
Vetronix Corporation

Sistema Práctico de aprendizaje Fuel Injection
MEX-FEY SA de CV

Reference Guide
Level5 Object for Microsoft Windows
Information Builders
1994

Capítulo 6

Reference Guide
Level5 Object for Microsoft Windows
Information Builders
1994

APÉNDICE

A continuación se lista el código de los DEMONS y métodos, que permiten hacer operaciones auxiliares, como desplegar el código de colores, abrir la base de datos, agregar CF máximos, etc. Por la función que realizan no tienen conocimiento experto involucrado, por lo tanto no forman parte de las reglas de producción.

El siguiente DEMON se emplea para obtener el CF mayor de las fallas múltiples y agregarlo a la variable correspondiente, según el subsistema al que pertenezca la falla.

DEMON 1 Agregar cfs maximos a las sumas de fallas
IF selected OF pushbutton Diagnostica 1N
THEN CONF(Bomba OF A Bombeo y conduccion IS Baja presion) := MAX(CONF(Bomba OF A Bombeo y conduccion IS Baja presion1), CONF(Bomba OF A Bombeo y conduccion IS Baja presion2))
AND Fallas bombeo y conduccion := Fallas bombeo y conduccion + CONF(Bomba OF A Bombeo y conduccion IS Baja presion)
AND CONF(Bomba OF A Bombeo y conduccion IS Calentamiento) := MAX(CONF(Bomba OF A Bombeo y conduccion IS Calentamiento1), CONF(Bomba OF A Bombeo y conduccion IS Calentamiento2))
AND Fallas bombeo y conduccion := Fallas bombeo y conduccion + CONF(Bomba OF A Bombeo y conduccion IS Calentamiento)
AND CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado) := MAX(CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado1), CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado2), CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado3))
AND Fallas admision de aire := Fallas admision de aire + CONF(Filtro de aire OF A Admision de aire IS Sucio o tapado)
AND CONF(Inyector OF A Inyeccion de Combustible IS Obstruido) := MAX(CONF(Inyector OF A Inyeccion de Combustible IS Obstruido1), CONF(Inyector OF A Inyeccion de Combustible IS Obstruido2))
AND Fallas inyeccion := Fallas inyeccion + CONF(Inyector OF A Inyeccion de Combustible IS Obstruido)

AND CONF(Linea retorno de combustible OF A Bombeo y conduccion IS Con fuga) := MAX(CONF(Linea retorno de combustible OF A Bombeo y conduccion IS Con fuga1), CONF(Linea retorno de combustible OF A Bombeo y conduccion IS Con fuga2))
AND Fallas bombeo y conduccion := Fallas bombeo y conduccion + CONF(Linea retorno de combustible OF A Bombeo y conduccion IS Con fuga)
AND CONF(Linea retorno de combustible OF A Bombeo y conduccion IS Obturada) := MAX(CONF(Linea retorno de combustible OF A Bombeo y conduccion IS Obturada1), CONF(Linea retorno de combustible OF A Bombeo y conduccion IS Obturada2))
AND Fallas bombeo y conduccion := Fallas bombeo y conduccion + CONF(Linea retorno de combustible OF A Bombeo y conduccion IS Obturada)
AND CONF(Regulador de presion OF A Bombeo y conduccion IS Abierto) := MAX(CONF(Regulador de presion OF A Bombeo y conduccion IS Abierto1), CONF(Regulador de presion OF A Bombeo y conduccion IS Abierto2))
AND Fallas bombeo y conduccion := Fallas bombeo y conduccion + CONF(Regulador de presion OF A Bombeo y conduccion IS Abierto)
AND CONF(Regulador de presion OF A Bombeo y conduccion IS Cerrado) := MAX(CONF(Regulador de presion OF A Bombeo y conduccion IS Cerrado1), CONF(Regulador de presion OF A Bombeo y conduccion IS Cerrado2))
AND Fallas bombeo y conduccion := Fallas bombeo y conduccion + CONF(Regulador de presion OF A Bombeo y conduccion IS Cerrado)
AND CONF(Sistema controlador de marcha minima IAC OF A Admision de aire IS Cerrado) := MAX(CONF(Sistema controlador de marcha minima IAC OF A Admision de aire IS Cerrado1), CONF(Sistema controlador de marcha minima IAC OF A Admision de aire IS Cerrado2))
AND Fallas admision de aire := Fallas admision de aire + CONF(Sistema controlador de marcha minima IAC OF A Admision de aire IS Cerrado)
AND CONF(Sistema sensor MAF OF A Admision de aire IS Dañado) := MAX(CONF(Sistema sensor MAF OF A Admision de aire IS Dañado1), CONF(Sistema sensor MAF OF A Admision de aire IS Dañado2))
AND Fallas admision de aire := Fallas admision de aire + CONF(Sistema sensor MAF OF A Admision de aire IS Dañado)
AND CONF(Sistema sensor MAP OF A Admision de aire IS Lectura solo de vacio) := MAX(CONF(Sistema sensor MAP OF A Admision de aire IS Lectura solo de vacio1), CONF(Sistema sensor MAP OF A Admision de aire IS Lectura solo de vacio2))
AND Fallas admision de aire := Fallas admision de aire + CONF(Sistema sensor MAP OF A Admision de aire IS Lectura solo de vacio)

AND CONF(Tuberia bomba a riel de inyectores OF A Bombeo y conduccion IS Con fuga) := MAX(CONF(Tuberia bomba a riel de inyectores OF A Bombeo y conduccion IS Con fuga1), CONF(Tuberia bomba a riel de inyectores OF A Bombeo y conduccion IS Con fuga2))
AND Fallas bombeo y conduccion := Fallas bombeo y conduccion + CONF(Tuberia bomba a riel de inyectores OF A Bombeo y conduccion IS Con fuga)
AND CONF(Tuberia bomba a riel de inyectores OF A Bombeo y conduccion IS Obturada) := MAX(CONF(Tuberia bomba a riel de inyectores OF A Bombeo y conduccion IS Obturada1), CONF(Tuberia bomba a riel de inyectores OF A Bombeo y conduccion IS Obturada2), CONF(Tuberia bomba a riel de inyectores OF A Bombeo y conduccion IS Obturada3))
AND Fallas bombeo y conduccion := Fallas bombeo y conduccion + CONF(Tuberia bomba a riel de inyectores OF A Bombeo y conduccion IS Obturada)
AND CONF(Alarma y switch OF B Toma y elevación de voltaje IS No funciona) := MAX(CONF(Alarma y switch OF B Toma y elevación de voltaje IS No funciona 1), CONF(Alarma y switch OF B Toma y elevación de voltaje IS No funciona 2))
AND Fallas toma y elevacion de voltaje := Fallas toma y elevacion de voltaje + CONF(Alarma y switch OF B Toma y elevación de voltaje IS No funciona)
AND CONF(Cable modulo DIS a ECM OF B Distribución IS Dañado) := MAX(CONF(Cable modulo DIS a ECM OF B Distribución IS Dañado1), CONF(Cable modulo DIS a ECM OF B Distribución IS Dañado2), CONF(Cable modulo DIS a ECM OF B Distribución IS Dañado3))
AND DIS a ECM dañado OF desplegar cfs dis := CONF(Cable modulo DIS a ECM OF B Distribución IS Dañado)
AND Fallas distribucion := Fallas distribucion + CONF(Cable modulo DIS a ECM OF B Distribución IS Dañado)
AND CONF(Batería OF C Alimentacion y generacion de energia IS No retiene carga) := MAX(CONF(Batería OF C Alimentacion y generacion de energia IS No retiene carga 1), CONF(Batería OF C Alimentacion y generacion de energia IS No retiene carga 2), CONF(Batería OF C Alimentacion y generacion de energia IS No retiene carga 3), CONF(Batería OF C Alimentacion y generacion de energia IS No retiene carga 4), CONF(Bateria OF C Alimentacion y generacion de energia IS No retiene carga 5))
AND Fallas alim y gen de enrgia electrica := Fallas alim y gen de enrgia electrica + CONF(Batería OF C Alimentacion y generacion de energia IS No retiene carga)

AND CONF(Conexión batería marcha OF C Arranque IS Interrumpida) := MAX(CONF(Conexión batería marcha OF C Arranque IS Interrumpida1), CONF(Conexión batería marcha OF C Arranque IS Interrumpida2))
AND Fallas arranque := Fallas arranque + CONF(Conexión batería marcha OF C Arranque IS Interrumpida)
AND CONF(Motor de arranque OF C Arranque IS No funciona) := MAX(CONF(Motor de arranque OF C Arranque IS No funciona1), CONF(Motor de arranque OF C Arranque IS No funciona2))
AND Fallas arranque := Fallas arranque + CONF(Motor de arranque OF C Arranque IS No funciona)
AND CONF(Motor de arranque OF C Arranque IS Arrastra) := MAX(CONF(Motor de arranque OF C Arranque IS Arrastra1), CONF(Motor de arranque OF C Arranque IS Arrastra2))
AND Fallas arranque := Fallas arranque + CONF(Motor de arranque OF C Arranque IS Arrastra)
AND CONF(Instrumentos en el tablero OF C Alimentacion y servicios IS En corto circuito) := MAX(CONF(Instrumentos en el tablero OF C Alimentacion y servicios IS En corto circuito1), CONF(Instrumentos en el tablero OF C Alimentacion y servicios IS En corto circuito2))
AND Fallas alimentacion y servicios := Fallas alimentacion y servicios + CONF(Instrumentos en el tablero OF C Alimentacion y servicios IS En corto circuito)
AND CONF(Switch de encendido OF C Alimentacion y servicios IS No conduce corriente) := MAX(CONF(Switch de encendido OF C Alimentacion y servicios IS No conduce corriente1), CONF(Switch de encendido OF C Alimentacion y servicios IS No conduce corriente2))
AND Fallas alimentacion y servicios := Fallas alimentacion y servicios + CONF(Switch de encendido OF C Alimentacion y servicios IS No conduce corriente)
AND CONF(Cable de alimentación al switch OF C Alimentacion y servicios IS Interrumpido) := MAX(CONF(Cable de alimentación al switch OF C Alimentacion y servicios IS Interrumpido1), CONF(Cable de alimentación al switch OF C Alimentacion y servicios IS Interrumpido2))
AND Fallas alimentacion y servicios := Fallas alimentacion y servicios + CONF(Cable de alimentación al switch OF C Alimentacion y servicios IS Interrumpido)

El siguiente método se ejecuta con ayuda del "DEMON 2 Ejecutar código de colores" que le asigna el valor de verdadero a una variable una vez que se han verificado todas las reglas de producción. A continuación se listan ambos:

DEMON 2 Ejecutar codigo de colores
IF selected OF pushbutton Diagnostica IN
THEN EX OF colores indices y vars := TRUE
AND title OF main window := "Diagnóstico (primer nivel)"

WHEN CHANGED *EX OF Colores indices y vars*
BEGIN
!ver "DEMON 2 Ejecutar codigo de colores" , mediante este demon se ejecuta este código, esto se hizo !para asegurar que siempre ejecute este *wh ch* después de evaluar todas las reglas y así pasar los valores !correctos al arreglo factores
! * En esta parte se pasan los coeficientes de cada variable al arreglo
! factores[ ], para después hacer la ordenación sobre éste.
factores[1] OF Colores arreglos := Fallas bombeo y conduccion
factores[2] OF Colores arreglos := Fallas inyeccion
factores[3] OF Colores arreglos := Fallas admision de aire
factores[4] OF Colores arreglos := Fallas toma y elevacion de voltaje
factores[5] OF Colores arreglos := Fallas distribucion
factores[6] OF Colores arreglos := Fallas generacion de chispa
factores[7] OF Colores arreglos := Fallas bombeo y circulacion de refrigerante
factores[8] OF Colores arreglos := Fallas medi y enfriamiento refrigerante
factores[9] OF Colores arreglos := Fallas alim y gen de enrgia electrica
factores[10] OF Colores arreglos := Fallas arranque
factores[11] OF Colores arreglos := Fallas alimentacion y servicios
! ** En esta parte se inicia la ordenación, poniendo la jerarquía
! correspondiente en el arreglo números [ ], usando a las variables i, j y k como índices.
FOR (k OF Colores indices y vars := 1 TO 11)
BEGIN
i OF Colores indices y vars := 1
FOR (j OF Colores indices y vars := 2 TO 11)
BEGIN
IF factores[i OF Colores indices y vars] OF Colores arreglos <= factores[j OF Colores indices y vars] OF Colores arreglos THEN
i OF Colores indices y vars := j OF Colores indices y vars

```
      END
     numero OF Colores indices y vars := numero OF Colores indices y vars + 1
     numeros[ i OF Colores indices y vars] OF Colores arreglos := numero OF
Colores indices y vars
     factores[ i OF Colores indices y vars] OF Colores arreglos := -1
! el -1 es para descartar el número en la siguiente iteración, si ponemos 0 no la
hace bien porque ese valor existe como factor cf
   END
! FIN de la ordenación y llenado del arreglo números

! *** En esta parte se hace la asignación de colores a cada valuebox dependiendo
de la jerarquía que tenga,
!      excepto cuando la variable asociada (fallas de ...) es cero. Si es así, se pone
blanca la caja sin importar la jerarquía que tenía, debido a que la ordenación
considera a todos los ceros y les asigna un valor,
!      pero si es cero no tiene fallas , por lo cual el color blanco indicará justamente
eso.
!-------------------Para bombeo y conducción
  IF Fallas bombeo y conduccion = 0 THEN
   BEGIN
    fill color OF valuebox 134 := 255,255,255
    pen color OF valuebox 134 := 255,255,255
    frame OF valuebox 134 := FALSE
   END
  ELSE
   BEGIN
    fill color OF valuebox 134 := colores[ numeros[ 1] OF Colores arreglos] OF
Colores arreglos
    pen color OF valuebox 134 := 0,0,0
    frame OF valuebox 134 := TRUE
   END
!-------------------Para inyección
  IF Fallas inyeccion = 0 THEN
   BEGIN
    fill color OF valuebox 135 := 255,255,255
    pen color OF valuebox 135 := 255,255,255
    frame OF valuebox 135 := FALSE
   END
  ELSE
```

```
    BEGIN
     fill color OF valuebox 135 := colores[ numeros[ 2] OF Colores arreglos] OF
Colores arreglos
     pen color OF valuebox 135 := 0,0,0
     frame OF valuebox 135 := TRUE
    END
!--------------------Para admisión de aire
   IF Fallas admisión de aire = 0 THEN
    BEGIN
     fill color OF valuebox 136 := 255,255,255
     pen color OF valuebox 136 := 255,255,255
     frame OF valuebox 136 := FALSE
    END
   ELSE
    BEGIN
     fill color OF valuebox 136 := colores[ numeros[ 3] OF Colores arreglos] OF
 Colores arreglos
      pen color OF valuebox 136 := 0,0,0
      frame OF valuebox 136 := TRUE
     END
 !--------------------Para toma y elevación
    IF Fallas toma y elevacion de voltaje = 0 THEN
     BEGIN
      fill color OF valuebox 137 := 255,255,255
      pen color OF valuebox 137 := 255,255,255
      frame OF valuebox 137 := FALSE
     END
    ELSE
     BEGIN
      fill color OF valuebox 137 := colores[ numeros[ 4] OF Colores arreglos] OF
  Colores arreglos
       pen color OF valuebox 137 := 0,0,0
       frame OF valuebox 137 := TRUE
      END
  !--------------------Para distribución
     IF Fallas distribucion = 0 THEN
      BEGIN
       fill color OF valuebox 138 := 255,255,255
       pen color OF valuebox 138 := 255,255,255
```

```
      frame OF valuebox 138 := FALSE
    END
   ELSE
    BEGIN
      fill color OF valuebox 138 := colores[ numeros[ 5] OF Colores arreglos] OF
Colores arreglos
      pen color OF valuebox 138 := 0,0,0
      frame OF valuebox 138 := TRUE
    END
!-------------------Para generación de chispa
   IF Fallas generacion de chispa = 0 THEN
    BEGIN
      fill color OF valuebox 139 := 255,255,255
      pen color OF valuebox 139 := 255,255,255
      frame OF valuebox 139 := FALSE
    END
   ELSE
    BEGIN
      fill color OF valuebox 139 := colores[ numeros[ 6] OF Colores arreglos] OF
 Colores arreglos
      pen color OF valuebox 139 := 0,0,0
      frame OF valuebox 139 := TRUE
    END
 !-------------------Para bombeo y circulación de refrigerante
    IF Fallas bombeo y circulacion de refrigerante = 0 THEN
     BEGIN
      fill color OF valuebox 143 := 255,255,255
      pen color OF valuebox 143 := 255,255,255
      frame OF valuebox 143 := FALSE
     END
    ELSE
     BEGIN
      fill color OF valuebox 143 := colores[ numeros[ 7] OF Colores arreglos] OF
  Colores arreglos
      pen color OF valuebox 143 := 0,0,0
      frame OF valuebox 143 := TRUE
     END
  !----------------------Para medición de refrigerante
    IF Fallas medi y enfriamiento refrigerante = 0 THEN
```

```
    BEGIN
     fill color OF valuebox 156 := 255,255,255
     pen color OF valuebox 156 := 255,255,255
     frame OF valuebox 156 := FALSE
    END
   ELSE
    BEGIN
     fill color OF valuebox 156 := colores[ numeros[ 8] OF Colores arreglos] OF
Colores arreglos
     pen color OF valuebox 156 := 0,0,0
     frame OF valuebox 156 := TRUE
    END
!------------------Para alimentación de energia eléctrica
   IF Fallas alim y gen de enrgia electrica = 0 THEN
    BEGIN
     fill color OF valuebox 150 := 255,255,255
     pen color OF valuebox 150 := 255,255,255
     frame OF valuebox 150 := FALSE
    END
   ELSE
    BEGIN
     fill color OF valuebox 150 := colores[ numeros[ 9] OF Colores arreglos] OF
 Colores arreglos
     pen color OF valuebox 150 := 0,0,0
     frame OF valuebox 150 := TRUE
    END
 !------------------Para arranque
   IF Fallas arranque = 0 THEN
    BEGIN
     fill color OF valuebox 151 := 255,255,255
     pen color OF valuebox 151 := 255,255,255
     frame OF valuebox 151 := FALSE
    END
   ELSE
    BEGIN
     fill color OF valuebox 151 := colores[ numeros[ 10] OF Colores arreglos]
  OF Colores arreglos
     pen color OF valuebox 151 := 0,0,0
     frame OF valuebox 151 := TRUE
```

```
    END
!------------------Para alimentación y servicios
   IF Fallas alimentacion y servicios = 0 THEN
    BEGIN
     fill color OF valuebox 152 := 255,255,255
     pen color OF valuebox 152 := 255,255,255
     frame OF valuebox 152 := FALSE
    END
   ELSE
    BEGIN
     fill color OF valuebox 152 := colores[ numeros[ 11] OF Colores arreglos]
OF Colores arreglos
     pen color OF valuebox 152 := 0,0,0
     frame OF valuebox 152 := TRUE
    END
  END
```

Los siguientes tres métodos se emplean para identificar cuando se presenta una combinación no válida entre los síntomas por estado de operación, y desplegar un mensaje de error en caso de que suceda.

WHEN CHANGED Al dar marcha OF Sintomas
BEGIN
!Este WCH verifica que no haya combinaciones incompatibles al leer síntomas
IF Al dar marcha OF Sintomas IS Da marcha pero no arranca AND Al dar marcha OF Sintomas IS No da marcha AND Al dar marcha OF Sintomas IS Tarda en arrancar siempre AND Al dar marcha OF Sintomas IS Tarda en arrancar solo en frio THEN
ASK mensaje error al leer síntomas
ELSE
IF Al dar marcha OF Sintomas IS Da marcha pero no arranca AND Al dar marcha OF Sintomas IS No da marcha THEN
ASK mensaje error al leer síntomas
ELSE
IF Al dar marcha OF Sintomas IS Da marcha pero no arranca AND Al dar marcha OF Sintomas IS Tarda en arrancar siempre THEN
ASK mensaje error al leer síntomas
ELSE
IF Al dar marcha OF Sintomas IS Da marcha pero no arranca AND Al dar marcha OF Sintomas IS Tarda en arrancar solo en frio THEN
ASK mensaje error al leer síntomas
ELSE
IF Al dar marcha OF Sintomas IS No da marcha AND Al dar marcha OF Sintomas IS Tarda en arrancar siempre THEN
ASK mensaje error al leer síntomas
ELSE
IF Al dar marcha OF Sintomas IS No da marcha AND Al dar marcha OF Sintomas IS Tarda en arrancar solo en frio THEN
ASK mensaje error al leer síntomas
ELSE
IF Al dar marcha OF Sintomas IS Tarda en arrancar siempre AND Al dar marcha OF Sintomas IS Tarda en arrancar solo en frio THEN
ASK mensaje error al leer síntomas
END

WHEN CHANGED En alta OF Sintomas
BEGIN
IF En alta OF Sintomas IS Al acelerar a fondo se apaga AND En alta OF Sintomas IS Se jalonea THEN
ASK mensaje error al leer sintomas
ELSE
IF En alta OF Sintomas IS Al acelerar a fondo se apaga AND En alta OF Sintomas IS No tiene potencia THEN
ASK mensaje error al leer sintomas
ELSE
IF En alta OF Sintomas IS Al acelerar cascabelea siempre AND En alta OF Sintomas IS Al acelerar cascabelea solo con cambio de altura THEN
ASK mensaje error al leer sintomas
END

WHEN CHANGED En baja OF Sintomas BEGIN
IF En baja OF Sintomas IS No se mantiene en marcha mínima siempre AND En baja OF Sintomas IS Al aplicar carga tiende a apagarse AND En baja OF Sintomas IS Se mantiene irregular THEN
ASK mensaje error al leer sintomas
ELSE
IF En baja OF Sintomas IS No se mantiene en marcha mínima siempre AND En baja OF Sintomas IS Al aplicar carga tiende a apagarse THEN
ASK mensaje error al leer sintomas
ELSE
IF En baja OF Sintomas IS No se mantiene en marcha mínima sólo en frío AND En baja OF Sintomas IS Al aplicar carga tiende a apagarse THEN
ASK mensaje error al leer sintomas
ELSE
IF En baja OF Sintomas IS No se mantiene en marcha mínima sólo en frío AND En baja OF Sintomas IS Se mantiene irregular THEN
ASK mensaje error al leer sintomas
ELSE
IF En baja OF Sintomas IS No se mantiene en marcha mínima siempre AND En baja OF Sintomas IS Se mantiene irregular THEN
ASK mensaje error al leer sintomas
ELSE

IF En baja OF Sintomas IS No se mantiene en marcha mínima siempre AND En baja OF Sintomas IS No se mantiene en marcha mínima sólo en frío THEN

ASK mensaje error al leer síntomas

ELSE

IF En baja OF Sintomas IS Al aplicar carga tiende a apagarse AND En baja OF Sintomas IS Se mantiene irregular THEN

ASK mensaje error al leer síntomas

END

El siguiente método asigna el año proporcionado por el usuario a un año de la base de datos:

```
WHEN CHANGED Año leido OF Lectura del Analizador de gases
 BEGIN
!***Esta parte verifica que el año que se esta dando en dprincipal sea valido, de acuerdo con los datos que se tienen en la
!    base de datos.
  IF Año leido OF Lectura del Analizador de gases < 1980 OR Año leido OF Lectura del Analizador de gases > 1998 THEN
    ASK mensaje error al leer el año
!    Error al leer año OF Errores := "Error, año no válido"
!  ELSE
!    Error al leer año OF Errores := ""
!***Debido a que en la base se metieron años y no intervalos, es nesesario verificar el año leido y asignarlo a un año  especifico
!    que si este en la base
  IF Año leido OF Lectura del Analizador de gases >= 1980 AND Año leido OF Lectura del Analizador de gases <= 1986 THEN
    Año OF Lectura del Analizador de gases := 1986
  IF Año leido OF Lectura del Analizador de gases >= 1987 AND Año leido OF Lectura del Analizador de gases <= 1993 THEN
    Año OF Lectura del Analizador de gases := 1993
  IF Año leido OF Lectura del Analizador de gases >= 1994 AND Año leido OF Lectura del Analizador de gases <= 1998 THEN
    Año OF Lectura del Analizador de gases := 1994
 END
```

El siguiente método asigna el año proporcionado por el usuario a un año de la base de datos:

```
WHEN CHANGED Año leido OF Lectura del Analizador de gases
 BEGIN
!***Esta parte verifica que el año que se esta dando en dprincipal sea valido, de acuerdo con los datos que se tienen en la
!    base de datos.
   IF Año leido OF Lectura del Analizador de gases < 1980 OR Año leido OF Lectura del Analizador de gases > 1998 THEN
     ASK mensaje error al leer el año
!     Error al leer año OF Errores := "Error, año no válido"
!   ELSE
!     Error al leer año OF Errores := ""
!***Debido a que en la base se metieron años y no intervalos, es nesesario verificar el año leido y asignarlo a un año  especifico
!    que si este en la base
   IF Año leido OF Lectura del Analizador de gases >= 1980 AND Año leido OF Lectura del Analizador de gases <= 1986 THEN
     Año OF Lectura del Analizador de gases := 1986
   IF Año leido OF Lectura del Analizador de gases >= 1987 AND Año leido OF Lectura del Analizador de gases <= 1993 THEN
     Año OF Lectura del Analizador de gases := 1993
   IF Año leido OF Lectura del Analizador de gases >= 1994 AND Año leido OF Lectura del Analizador de gases <= 1998 THEN
     Año OF Lectura del Analizador de gases := 1994
 END
```

El siguiente método prepara a la base de datos para que proporcione la infromación requerida:

```
WHEN CHANGED Busca datos OF Lectura del Analizador de gases
 BEGIN
!***Este WCH abre la base y nos coloca en el renglon correspondiente usando como llave
!     el año. El valor de Año OF Lectura del Analizador de gases lo da el WCH Año leido OF Lectura del Analizador de gases
!    No se puede hacer una relación directa por lo de los intervalos.
   FIND dB3 AG 1
    LIMIT 1
    WHERE  Año OF Lectura del Analizador de gases = year OF dB3 AG 1
    FIND END
  END
```