

01132
92



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**SISTEMA EXPERTO PARA UN ENLACE DE
MICROONDA PUNTO A PUNTO**

T E S I S

QUE, PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A N:

**MARLENE ~~SILVESTRE OCAÑA~~
EDUARDO HERNÁNDEZ JIMÉNEZ**



**DIRECTOR DE TESIS:
DR. ARTURO REINKING CEJUDO**

CIUDAD UNIVERSITARIA

2003

1



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

PAGINACIÓN

DISCONTINUA

Gracias . . .

a DIOS por permitirme llegar hasta el día de hoy y vivir al máximo este momento rodeada de los que amo y me aman.

a ti amado abuelito ELEAZAR, que en paz descanses. Por tu ejemplo de honestidad, entereza, y lucha por la vida. Me enseñaste que para alcanzar la dicha en esta vida hay que trabajar con todo las fuerzas y el corazón, de la mano de mi abuelita ALICIA que continúa mostrándome el camino del éxito. Te adoro, este logro también es de ustedes.

a ti MAMÁ por tu interminable constancia por hacer de mí una mejor persona en todos los aspectos, por tus consejos y sabiduría para guiarme hasta el final de mis metas y no permitir que claudicara cuando los malos momentos me invadían, eres todo un ejemplo a seguir, orgullosamente mi madre. Eres mi vida, te amo gordita.

a ti PAPÁ por tus invaluable esfuerzos y sacrificios para hacer realidad este sueño y por que los más bellos recuerdos que guardo en mi mente y corazón son de pequeña cuando pasábamos mas tiempo juntos y me enseñaste a creer en mi para ser autosuficiente, sembraste en mi esa semilla de esperanza, y ahora te entrego uno de sus frutos. Te amo gordito.

a mi hermana MARIFER por que estas llena de amor y siempre me esperabas para dormir con una bella sonrisa y un cálido abrazo a pesar de tu molestia por tener la luz de nuestra habitación encendida hasta la madrugada. Te adoro pequeña traviesa.

a mis padrinos MELY y CECI, mis segundos padres, por su ejemplo de lucha y humildad, gracias por apoyarme con más de lo que creo merecer y corregirme hasta en el más pequeño detalle para que pudiera cumplir mis metas. Los amo con todo mi corazón.

a mis queridas tías LIVIA y RAFAELA, por ser mujeres de lucha incansable. Me enseñaron que sólo trabajando honestamente se consigue la felicidad, pues ellas son el vivo ejemplo de estas palabras. Les entrego éste nuestro sueño, las amo tías.

a mi querida UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, por que eres y seguirás siendo mi máxima casa de estudios. Sabré hacerte honor.

a ti MARCO ANTONIO HERNÁNDEZ PÉREZ por que no importando las adversidades siempre llegaste en el momento justo para recordarme que hay más tiempo que vida siempre y cuando mantenga mi fe y trabaje con todas mis fuerzas para hacer realidad mis sueños. Este logro también es tuyo, te amo pequeño travieso.

TESIS CON
FALTA DE ORIGEN

a ustedes, ILIANA, LETY, EDUARDO, MARCO POLO, FERNANDO, ERICK, HERBERT, SOFIA, ALFONSO, MIRIAM, PEDRO, MARIAN, LIZ, ROSALBA, ARACELI, IRIS, GUILLERMO, MARCO ANTONIO, MAURICIO y a todos aquellos que posiblemente me faltaron nombrar pero que siempre estuvieron conmigo, gracias por brindarme su apoyo incondicional, sus sonrisas, sus regaños, sus consejos, sus enseñanzas por lo maravillosos que son cada uno de ustedes con su diversidad de caracteres, pero sobretodo por regalarme uno mis más bellos tesoros, ser mis amados y sinceros amigos. Orgullosamente pumas. Siempre los llevaré en mis pensamientos.

a ti LALO por compartir conmigo este sueño. Aprendí a tu lado que todo cuesta pero que la recompensa llegará tarde o temprano. Es una gran satisfacción trabajar a tu lado.

al Dr. ARTURO REINKING CEJUDO, por el tiempo y esfuerzo que invirtió en la elaboración de este trabajo, por sus consejos y recomendaciones mil gracias de corazón.

a todos y cada uno de los estudiantes y profesores que integran la FACULTAD DE INGENIERÍA por contribuir, sin saberlo, a mi formación profesional. En especial a los maestros que me enseñaron a valorar, admirar y ser agradecida con mi universidad.

TESIS CON
FALLA DE ORIGEN

MARLENE SILVESTRE OCAÑA

Dedicada a mis padres Elvira y A. Agustín quienes han puesto toda su confianza y fe en mí, y han sido muy pacientes a lo largo de mi carrera. A ustedes quienes me han ayudado en todo momento y ni aún con la culminación de ésta tesis, podré pagar el gran sacrificio que día a día hicieron para verme llegar a este momento. En especial a mi padre, quien ha sido el maestro de mi vida, mi amigo en todo momento, la persona en quien siempre puedo confiar y mi ejemplo a seguir. Gracias por todas tus enseñanzas en las buenas y en las malas. Ésto sólo es uno de los frutos de la semilla que sembraste en mí. Gracias Papá por todo. Los quiero y Amo mucho.

A mi hermano Armando, que siempre se mantuvo al margen de mi vida, ayudándome y a veces regañándome; pero más que regaños, yo sé que eran consejos para salir adelante y llegar a ser alguien de quien poder sentirse orgulloso.

A todos mis amigos y "compañeros" que de una u otra forma se involucraron para que yo saliera bien en mis estudios, en enorme agradecimiento a "La Bola": el grupo de amigos en el que siempre estuvimos ayudándonos y apoyándonos unos a otros sobre todo en la última etapa de la carrera. Me faltarian palabras para agradecerles de corazón toda la ayuda que me brindaron. Gracias Amigos.

A Sofía Martínez Araujo, mi mejor amiga, mi gran hermana, mi compañera de vida, la persona más especial en estos momentos. Gracias corazón por estar a mi lado y mil gracias por ser como eres. Gracias por todo tu apoyo, tu amor, tu cariño, tus sonrisas, y sobre todo por entrar en mi vida y hacerla muy feliz. Gracias a tu familia por permitirme entrar en sus corazones.

Al Dr. Arturo Reinking Cejudo por su apoyo, su tiempo, su confianza, sus enseñanzas y sus consejos para la realización de este proyecto. Gracias.

A mi gran amiga y compañera de Tesis Marlene Silvestre por dejarme compartir a su lado un proyecto tan importante como éste. Gracias Mar por tu amistad y cariño incondicional. Recuerda que conmigo siempre tendrás un excelente amigo en quien puedas confiar plenamente.

TESIS CON
FALLA DE ORIGEN

EDUARDO HERNÁNDEZ JIMÉNEZ

TESIS CON
FALLA DE ORIGEN

ÍNDICE

INTRODUCCIÓN	IV
CAPÍTULO I	
1. INTRODUCCIÓN A LOS SISTEMAS EXPERTOS	1
1.1 INTRODUCCIÓN	1
1.2 INTELIGENCIA ARTIFICIAL	1
1.2.1 PANORAMA GENERAL DE LA INTELIGENCIA ARTIFICIAL	2
1.2.2 RELACIÓN DE LOS SISTEMAS EXPERTOS CON LA INTELIGENCIA ARTIFICIAL	2
1.3 SISTEMAS EXPERTOS	3
1.3.1 DEFINICIÓN DE UN SISTEMA EXPERTO	3
1.3.2 RESEÑA HISTÓRICA	3
1.3.3 CARACTERÍSTICAS GENERALES DE LOS SISTEMAS EXPERTOS	4
1.3.4 PROBLEMÁTICA ATACADA POR LOS SISTEMAS EXPERTOS	5
1.3.5 PROBLEMAS Y LIMITACIONES DE LOS SISTEMAS EXPERTOS	5
1.4 ESTRUCTURA DE LOS SISTEMAS EXPERTOS	6
1.4.1 ARQUITECTURA DE UN SISTEMA EXPERTO TÍPICO	7
1.4.2 COMPONENTES DE UN SISTEMA EXPERTO	7
1.4.3 LA BASE DE CONOCIMIENTOS	8
1.4.4 LA BASE DE HECHOS	8
1.4.5 EL MOTOR DE INFERENCIA	9
1.4.6 MÓDULO DE COMUNICACIÓN	10
1.4.7 ADQUISICIÓN DE CONOCIMIENTOS	10
1.5 REPRESENTACIÓN DEL CONOCIMIENTO	11
1.5.1 LA LÓGICA COMO REPRESENTACIÓN DEL CONOCIMIENTO	11
1.5.2 REPRESENTACIÓN BASADA EN REGLAS	11
1.5.3 REPRESENTACIÓN BASADA EN MARCOS (FRAMES)	12
1.5.4 REDES SEMÁNTICAS	12
1.6 SISTEMAS BASADOS EN REGLAS (SBR)	13
1.6.1 COMPONENTES DE UN SISTEMA EXPERTO BASADO EN REGLAS (SEBR)	14
1.6.2 ESTRATEGIAS DE CONTROL. ENCADENAMIENTO HACIA DELANTE Y HACIA ATRÁS	15
1.7 TIPOS DE HERRAMIENTAS PARA SISTEMAS EXPERTOS	16
1.7.1 SELECCIÓN DE UNA HERRAMIENTA	17
1.8 CAMPOS DE APLICACIÓN DE LOS SISTEMAS EXPERTOS	18
CAPÍTULO II	
2. CLIPS	19
2.1 INTRODUCCIÓN	19
2.2 ¿QUE ES CLIPS?	19
2.3 REPRESENTACIÓN DEL CONOCIMIENTO EN CLIPS	19

TESIS CON
 FALLA DE ORIGEN

2.4 ESTRUCTURA DE UN PROGRAMA EN CLIPS	20
2.5 MEMORIA DE TRABAJO (MT)	20
2.5.1 ELEMENTOS BÁSICOS DE PROGRAMACIÓN	21
2.5.1.1 ELEMENTOS PRIMITIVOS DE DATOS.	21
2.5.1.2 FUNCIONES.	22
2.5.1.3 CONSTRUCTORES.	22
2.5.2 COMENTARIOS	22
2.5.3 HECHOS	22
2.5.3.1 HECHOS ORDENADOS	23
2.5.3.2 HECHOS NO ORDENADOS O PLANTILLAS (DEFTEMPLATE)	23
2.5.3.3 ATRIBUTOS DE LOS SLOT (CAMPOS)	24
2.5.3.4 DEFINICIÓN DE HECHOS INICIALES	26
2.5.3.5 VISUALIZACIÓN DE HECHOS INICIALES	26
2.5.3.6 BORRADO DE HECHOS INICIALES	27
2.5.4 OBJETOS	27
2.5.5 LIMPIEZA DE LA MEMORIA DE TRABAJO	27
2.6 BASE DE CONOCIMIENTO (BC)	28
2.6.1 REGLAS	28
2.6.1.1 ELEMENTOS CONDICIONALES (EC)	29
2.6.1.2 PROPIEDADES DE LAS REGLAS: SALIENCE Y AUTO-FOCUS	30
2.6.2 COMANDOS PREDEFINIDOS POR CLIPS	31
2.6.3 FUNCIONES PREDEFINIDAS POR CLIPS	32
2.6.4 FUNCIONES DEFINIDAS POR EL USUARIO	32
2.7 MOTOR DE INFERENCIA	32
2.7.1 PATTERN MATCHING	33
2.8 ENTORNO DE CLIPS CON WINDOWS	33
2.9 CLIPS ACTIVEX OCX	35
2.10 CLIPS Y VISUAL BASIC	35

CAPÍTULO III

3. ENLACES MICROONDA	39
3.1 INTRODUCCIÓN	39
3.2 RADIOCOMUNICACIÓN	39
3.3 ¿QUÉ SON LAS MICROONDAS?	40
3.4 MICROONDAS TERRESTRES	40
3.5 ESTRUCTURA GENERAL DE UN ENLACE DE MICROONDAS	41
3.6 COMPONENTES DE UN ENLACE VÍA MICROONDAS	43
3.6.1 TRANSMISOR Y RECEPTOR EN LAS MICROONDAS	46
3.6.2 REPETIDORES	47
3.6.3 ZONA DE FRESNEL	48
3.7 TECNOLOGÍAS DE TRANSPORTE	49
3.7.1 MULTIPLEXACIÓN	49
3.7.2 SISTEMAS DE TRANSMISIÓN	49
3.7.3 ELEMENTOS DE MULTICANALIZACIÓN	49
3.7.4 JERARQUÍAS DE MULTIPLEXACIÓN	50
3.7.5 ESTRUCTURA DE TDM PARA 32 CANALES	51
3.7.6 OBJETIVO DE TDM	51
3.7.7 TRAMA DE UNA SEÑAL E1 DE 2.048 MBPS	52
3.7.8 ORDENES JERÁRQUICOS DE MULTIPLEXACIÓN EUROPEA	52
3.8 ENLACE PUNTO A PUNTO	53
3.8.1 ESTUDIO DE LÍNEA DE VISTA (LINE OF SIGHT- LOS)	53
3.8.2 ANÁLISIS PRELIMINAR	54
3.8.3 PERFIL DEL ENLACE	54

3.8.4	CÁLCULO DE ALTURA DE ANTENAS	55
3.8.5	ANÁLISIS DEL SITIO	55
3.8.6	INFORME	56
3.9	CÁLCULO DE DISTANCIA	56
3.10	CÁLCULO DE DISPONIBILIDAD Y CONFIABILIDAD	57
3.11	CÁLCULO DE ENLACE	58
3.12	ENLACE MULTIPUNTO	58

CAPÍTULO IV

4.	PLANTEAMIENTO DEL SISTEMA EXPERTO	59
4.1	INTRODUCCIÓN	59
4.2	PROBLEMA (DEFINICIÓN DEL SISTEMA)	59
4.2.1	ANÁLISIS DEL PROBLEMA	59
4.2.2	RECURSOS	60
4.2.2.1	RECURSOS DE HARDWARE	60
4.2.2.2	RECURSOS DE SOFTWARE	60
4.3	ANÁLISIS DEL SISTEMA	61
4.3.1	LIMITACIONES Y DELIMITACIONES	61
4.3.2	FACTIBILIDAD Y VIABILIDAD DEL SISTEMA EXPERTO	62
4.3.3	ALCANCE DEL SISTEMA EXPERTO	63
4.3.4	SELECCIÓN DE RECURSOS A UTILIZAR	63
4.4	DIAGRAMA DE FLUJO DE DATOS (DFD)	63
4.5	DIAGRAMA ENTIDAD-RELACIÓN DEL SISTEMA EMPP	74
4.6	ÁRBOL DE DECISIÓN	76
4.7	DISEÑO	77
4.7.1	CARTA ESTRUCTURADA	77
4.7.2	MÓDULOS DE LA CARTA DE ESTRUCTURA	77
4.8	CODIFICACIÓN	79

CAPÍTULO V

5.	EJECUCIÓN DEL SISTEMA EMPP	81
5.1	INTRODUCCIÓN	81
5.2	DESCRIPCIÓN DEL PROBLEMA	81
5.3	INTERACCIÓN DEL SISTEMA EMPP CON EL USUARIO	82
5.4	COMENTARIO FINAL	91

CONCLUSIONES	92
---------------------	-----------

BIBLIOGRAFÍA	93
---------------------	-----------

APÉNDICE A	94
-------------------	-----------

TESIS CON
FALLA DE ORIGEN

INTRODUCCIÓN

Gracias al alto desempeño que han tenido los sistemas expertos, es posible encontrarlos en un amplio campo de aplicaciones, ya que aportan mayor eficiencia y reducen el costo en la realización del trabajo en dicho campo.

Dentro del campo de las telecomunicaciones una parte muy importante son los enlaces punto a punto. Dichos enlaces han cobrado un gran auge dentro del área de la comunicación inalámbrica debido a la necesidad de transmitir datos en áreas metropolitanas.

Esta tesis esta fundamentada en los trabajos realizados por los Ingenieros Expertos de la empresa MAXCOM TELECOMUNICACIONES, quienes son los encargados de la realización de dichos enlaces y ante la falta de un software adicional para realizarlos, nace la idea de crear un sistema experto de Enlaces de Microonda Punto a Punto (EMPP) para ofrecer la solución mas factible en base a diferentes factores como distancia, disponibilidad del repetidor y número de E1's a instalar, entre otros.

Uno de los objetivos de esta tesis es la de construir un sistema que sea capaz de ofrecer la solución mas adecuada en la toma de decisiones para realizar un enlace punto a punto al tener a más de un repetidor como opción para efectuar el enlace.

El sistema EMPP se diseña con el cascarón CLIPS, el cual es una herramienta de libre distribución para el desarrollo y construcción de los Sistemas Expertos basados en reglas, que permite ser llamado desde otros lenguajes. De igual manera, la interfase del sistema EMPP se diseña con el lenguaje de programación Visual Basic, permitiendo la interacción de forma amigable con el Ingeniero Experto.

Los objetivos principales del sistema EMPP son:

Proponer la mejor solución de entre varias rutas posibles, así como la de sugerir algunos parámetros de validación e instalación del enlace microonda (frecuencia y polaridad por ejemplo).

Asignar la instalación para cada E1 solicitado en la(s) antena(s) correspondiente(s) al repetidor propuesto como mejor solución.

Gracias a que el sistema EMPP puede elegir la mejor ruta de entre varios posibles caminos, el ingeniero experto solo se enfocará a realizar los cálculos de enlace necesarios para esa ruta elegida. De resultar negativo el cálculo realizado, el ingeniero le informará al sistema que esa ruta no es factible y éste último le devolverá la segunda mejor opción. En caso de que tampoco ésta opción sea factible, el sistema dará la siguiente ruta de las asignadas. En caso de no contar con ninguna solución el sistema será capaz de detectar que no existe una solución optima para ese enlace y dará una recomendación al usuario de lo que debe hacerse.

Una función importante del sistema EMPP es la de tener un sistema de consulta referente a clientes, repetidores y antenas involucrados en cada enlace. También está previsto el mantenimiento a la base de datos del sistema con el fin de tener actualizada la disponibilidad e información general de toda la red de enlaces de la empresa de Telecomunicaciones.

Ésta tesis consta de cinco capítulos, en el primer capítulo se estudian de manera general los sistemas expertos, sus características, su estructura y la forma de representar el conocimiento

TESIS CON
FALLA DE ORIGEN

mediante reglas. En el segundo capítulo se estudia CLIPS, que como ya se mencionó es una herramienta para construir sistemas expertos. En el tercer capítulo se describen de manera general los enlaces microonda. En el cuarto capítulo se plantea, se analiza, se describe y se construye todo el sistema EMPP. El quinto capítulo contiene un ejemplo clásico de la manera en que el sistema EMPP interactúa con el usuario.

TESIS CON
FALLA DE ORIGEN

1. INTRODUCCIÓN A LOS SISTEMAS EXPERTOS

1.1 INTRODUCCIÓN

Este capítulo ofrece una visión general y básica de los sistemas expertos y presenta sus conceptos fundamentales, con el objetivo de contar con los antecedentes necesarios para tener una base sólida en cuanto al análisis, desarrollo y aplicación de los mismos. Partimos desde sus orígenes fundados en la Inteligencia Artificial para continuar con el avance que los Sistemas Expertos han tenido a lo largo del tiempo según las necesidades del hombre. También se plantean las estructuras y arquitecturas que se han considerado para que éstos sean cada día mas eficientes y útiles.

1.2 INTELIGENCIA ARTIFICIAL

Para entender la estructura interna y el funcionamiento de un Sistema Experto, es necesario entender los fundamentos conceptuales básicos de la solución de problemas con Inteligencia Artificial, dado que un Sistema Experto es una aplicación de técnicas de la Inteligencia Artificial.

La Inteligencia Artificial representa un gran volumen de conocimientos y técnicas que han sido desarrollados por muchos investigadores, por lo cual, han aparecido numerosas definiciones de Inteligencia Artificial, pero ninguna ha sido de aceptación general. Como punto de partida en este trabajo se entiende la siguiente definición.

La inteligencia Artificial se define como la ciencia que trata de la comprensión de la inteligencia y del diseño de máquinas inteligentes, es decir, el estudio y la simulación de las actividades intelectuales del hombre (manipulación, razonamiento, percepción, aprendizaje, creación, etc.).

La Inteligencia Artificial comprende el estudio y creación de sistemas computarizados que manifiestan cierta forma de inteligencia: sistemas que aprenden nuevos conceptos y tareas, sistemas que pueden razonar y derivar conclusiones útiles acerca del mundo que nos rodea, sistemas que pueden comprender un lenguaje natural o percibir y comprender una escena visual, y sistemas que realizan otro tipo de actividades que requieren de inteligencia humana.

A pesar de la diversidad de conceptos propuestos para la Inteligencia Artificial, en general todos coinciden en que la Inteligencia Artificial trata de alcanzar inteligencia a través de la computación.

Desde el punto de vista de los objetivos, la Inteligencia Artificial puede considerarse en parte como ingeniería y en parte como ciencia:

Como ingeniería, el objetivo de la Inteligencia Artificial es resolver problemas reales, actuando como un conjunto de ideas acerca de cómo representar y utilizar el conocimiento, y de cómo desarrollar sistemas informáticos.

Como ciencia, el objetivo de la Inteligencia Artificial es buscar la explicación de diversas clases de inteligencia, a través de la representación del conocimiento y de la aplicación que se da a éste en los sistemas informáticos desarrollados.

1.2.1 PANORAMA GENERAL DE LA INTELIGENCIA ARTIFICIAL

La Inteligencia Artificial estudia las ideas con el propósito de integrarlas en máquinas, dichas máquinas podrán responder ante situaciones distintas de manera similar a la que un humano respondería, dándoles la capacidad humana de contemplar, juicio e intención. Cada una de estas máquinas deberá contar con un criterio y la capacidad de seleccionar una de las muchas opiniones de ella misma. Producidas por la labor y habilidad humana, estas máquinas deberán conducirse por ellas mismas de acuerdo con la vida, espiritualidad y sensibilidad, pero en realidad, solo son imitaciones.

Bajo esta perspectiva de la Inteligencia Artificial se han derivado muchos campos de estudio, como son: Sistemas de Lenguaje Natural, Sistemas Reconocedores de Imágenes, Robótica y Sistemas Expertos.

Sistemas de Lenguaje Natural. Estudia el uso del lenguaje natural (el que le es propio al hombre) como medio de comunicación con las máquinas (programas como las bases de datos, robots, etc.); es un problema complejo pues intervienen distintos procesos como son: la comprensión del lenguaje, la síntesis y el análisis de la voz, el resumen y la traducción.

La tarea de la Inteligencia Artificial estriba en disponer en la máquina de capacidades ligadas a la comprensión inteligente del lenguaje y a facilitar la comunicación entre el hombre y la máquina.

Sistemas Reconocedores de Imágenes. En estos sistemas debe reconocerse el significado de imágenes con ayuda de procesos exactamente definidos. Sin embargo aún no se cuenta con algún sistema reconocedor de imágenes con aplicación general, los procesos desarrollados hasta la fecha solo permiten aplicaciones avanzadas (por ejemplo, robótica, automatismo industrial, análisis de fotografías aéreas).

Se buscan procesos capaces de distinguir cantos de iluminación, cantos de orientación y cantos de reflexión, es decir, sombras, contornos y límites de color. Además existe también el objetivo de desarrollar un hardware especial con el que se pueda analizar posibles soluciones con gran cantidad de cálculo, por ejemplo, en la interpretación de la secuencia de imágenes.

Los dominios que lindan con el reconocimiento de imágenes son el reconocimiento de patrones (pattern recognition) y el tratamiento de imágenes (picture processing).

Robótica. Estudia las máquinas (robots) capaces de realizar procesos mecánicos repetitivos y tareas manuales de las cuales es capaz el hombre. Con el estudio de la Inteligencia Artificial se intenta que el comportamiento de los robots sea más "inteligente", reaccionando éste, ante fallas o resultados imprevistos, mediante el uso de técnicas de solución espacial de problemas (espacios tridimensionales y geometría de objetos), tiempo real (optimización de procesos de movimiento y coordinación), tratamientos de señales sensoriales (señales obtenidas por un sensor), sistemas de planificación (coordinación de varios robots en una producción en serie).

1.2.2 RELACIÓN DE LOS SISTEMAS EXPERTOS CON LA INTELIGENCIA ARTIFICIAL

Los Sistemas Expertos se emplean para ejecutar una variedad muy complicada de tareas que en el pasado solo podían llevarse a cabo por un número limitado de personas expertas intensamente entrenadas. A través de la aplicación de las técnicas de la Inteligencia Artificial, los Sistemas Expertos captan el conocimiento básico que permite a una persona desempeñarse como un experto frente a problemas complicados.

Un Sistema Experto es una expresión de los Sistemas Basados en el Conocimiento. Con la aplicación de técnicas de Inteligencia Artificial finaliza la transición del procesamiento de datos al procesamiento de conocimientos.

Los Sistemas Expertos se aplican por norma general en problemas que implican un procedimiento basado en el conocimiento. Un procedimiento de solución basado en el conocimiento comprende las siguientes capacidades:

- Utilización de normas o estructuras que contengan conocimientos y experiencias de expertos especializados.
- Deducción lógica de conclusiones.
- Capacidad de interpretar datos ambiguos.
- Manipulación de conocimientos afectados por valores de probabilidad.

Los Sistemas Expertos se están empleando en una amplia variedad de aplicaciones que comprenden, entre otras, diagnóstico, planeación, predicción, diseño, interpretación, control, monitoreo de estado e instrucción.

En el futuro, a medida de que se produzcan nuevas arquitecturas de equipos que soporten de una manera mas directa la ejecución de Sistemas Expertos y que se perfeccione la tecnología de la Inteligencia Artificial, es razonable esperar un desarrollo de sistemas que se aproximen al comportamiento humano en muchas áreas.

1.3 SISTEMAS EXPERTOS

Los Sistemas Expertos reproducen el proceso intelectual de un experto humano en un campo particular, pudiendo mejorar su productividad, ahorrar tiempo y dinero, conservar sus valiosos conocimientos y difundirlos más fácilmente. Por lo tanto este tipo de sistemas ofrecen un extenso campo de posibilidades en resolución de problemas y en aprendizaje. Su uso se esta extendiendo rápidamente, debido a su importante impacto sobre los negocios y la industria.

1.3.1 DEFINICIÓN DE UN SISTEMA EXPERTO

Los Sistemas Expertos (también conocidos por Sistemas Basados en el Conocimiento) son un conjunto de programas que hacen uso del conocimiento especializado para resolver problemas como un especialista humano. Éste es una persona que tiene experiencia desarrollada en cierta área, es decir, el especialista tiene conocimiento o habilidades especiales que la mayoría no conoce o de las que no dispone (el llamado dominio de conocimiento); el Sistema Experto puede resolver problemas que la mayoría no podría resolver o los resuelve con mucha mayor eficiencia y menor costo. El conocimiento de los Sistemas Expertos puede obtenerse por experiencia o consulta de los conocimientos que suelen estar disponibles en libros, revistas o personas capacitadas.

Dependiendo del dominio de conocimiento con el que cuente el Sistema Experto, éste razona o hace inferencias de la misma forma que un especialista humano inferiría la solución de un problema, es decir, dados unos hechos se infiere una conclusión.

Una definición mas universal sería: "Un Sistema Experto es un programa de computadora que simula la forma de razonar de un experto humano".

1.3.2 RESEÑA HISTÓRICA

La Tecnología representada por los Sistemas Expertos actuales, surge de las técnicas de la Inteligencia Artificial que han sido objeto de amplias e intensivas investigaciones desde finales de la década de 1950. Las investigaciones referidas comenzaron en el área de lenguajes para apoyar el razonamiento simbólico. El lenguaje de programación IPL, el primer lenguaje simbólico, para procesamiento de listas, se empleó ampliamente en las primeras implementaciones de la

Inteligencia Artificial. LISP, uno de los mas populares en la Inteligencia Artificial, fue desarrollado por John McCarthy en 1958.

La investigación específica en Sistemas Expertos realmente comenzó a mediados de los años sesenta. En ese período se creía que bastaban unas pocas leyes de razonamiento junto con potentes ordenadores para producir resultados brillantes. Un intento en ese sentido fue el llevado a cabo por los investigadores Alan Newell y Herbert Simon que desarrollaron un programa denominado GPS (General Problem Solver; solucionador general de problemas). Lo que no podía hacer el GPS era resolver problemas del mundo real, tales como un diagnóstico médico.

Algunos investigadores decidieron entonces cambiar por completo el enfoque del problema restringiendo su ambición a un dominio específico e intentando simular el razonamiento de un experto humano. En vez de dedicarse a computerizar la inteligencia general, se centraron en dominios de conocimiento muy concretos. De esta manera nacieron los sistemas expertos.

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. En la siguiente tabla se presentan varios sistemas que son de especial interés, principalmente porque forman gran parte de la base técnica e histórica de la tecnología de los Sistemas Expertos.

Primeros Sistemas Expertos

Sistema	Fecha	Autor	Tema
Dendral	1965	Stanford	Deduce información sobre estructuras químicas
Macsyma	1965	MIT	Realiza análisis matemático complejo
Hearsay	1965	Carnegie-Mellon	Interpreta en Lenguaje Natural un subconjunto del idioma
Mycin	1972	Stanford	Diagnóstico de enfermedades de la sangre
Telresias	1972	Stanford	Herramienta para transformación de conocimientos
Prospector	1972	Stanford	Exploración mineral y herramientas de identificación
Age	1973	Stanford	Herramienta para generar Sistemas Expertos
OPSS	1974	Carnegie-Mellon	Herramientas de desarrollo de Sistemas Expertos
Caduceus	1975	University of Pittsburg	Herramienta de diagnóstico para la medicina interna
R1	1978	Carnegie-Mellon	Configurador de equipos de computación para la DEC
Rosie	1978	Rand	Herramientas de desarrollo de Sistemas Expertos
CLIPS	1985	NASA	Herramientas de desarrollo de Sistemas Expertos

1.3.3 CARACTERÍSTICAS GENERALES DE LOS SISTEMAS EXPERTOS

Un Sistema Experto suele diseñarse para que tenga las siguientes características generales:

- Alto desempeño. El sistema debe tener la capacidad de responder a un nivel de competencia igual o superar al de un especialista de campo. Es decir, la calidad del consejo dado por el sistema debe ser muy alta.
- Tiempo de respuesta adecuado. El sistema debe actuar en un tiempo razonable, comparable o mejor al tiempo requerido por un especialista para tomar una decisión.
- Confiabilidad. El Sistema Experto debe de ser confiable y no propenso a fallos.
- Comprensible. El sistema debe ser capaz de explicar los pasos de su razonamiento. Este rasgo es muy importante debido a que en determinados momentos se puede depender de manera directa de la decisión tomada por el sistema. Además es una manera de asegurarse en la fase de desarrollo de que el sistema ha adquirido el conocimiento y lo está usando de manera correcta.
- Flexibilidad. Debido a la gran cantidad de conocimiento que puede albergar un Sistema Experto, es importante contar con un mecanismo eficiente para añadir, modificar y eliminar el conocimiento.
- Representación explícita del conocimiento.
- Manipulación y razonamiento sobre descripciones simbólicas.

TESIS CON
 FALLA DE ORIGEN

- Que puedan contemplar múltiples hipótesis en competición simultáneamente.
- Que puedan explicar por qué están formulando una pregunta.
- Que tengan la capacidad de explicar sus conclusiones y el proceso de razonamiento.

1.3.4 PROBLEMÁTICA ATACADA POR LOS SISTEMAS EXPERTOS

La aplicación de un Sistemas Expertos será adecuada en el lugar en donde los expertos dispongan de conocimientos complejos en un área estrechamente delimitada, donde no existan algoritmos elaborados (o donde los existentes no puedan solucionar algunos problemas) y no existan teorías completas. O también donde dichas teorías existen, pero resulta prácticamente imposible analizar todos los casos teóricamente imaginables mediante algoritmos y en un espacio de tiempo razonable.

En estas situaciones hace falta el conocimiento que el experto ha adquirido por experiencia, para llegar a una solución en un espacio de tiempo aceptable. Lo mencionado en el primer párrafo se caracterizan además por el hecho de que, aunque es posible la existencia de una o mas soluciones, la vía de solución no está previamente fijada. Sin embargo, el experto encuentra a menudo una solución al problema gracias a las informaciones que posee sobre el problema y a su experiencia. Mientras esta solución sea susceptible de repetición y el planteamiento del problema esté claro, existe un razonamiento que puede ser reproducido por un Sistema Experto.

Debido a que la estructuración e implementación del conocimiento del experto requiere una gran cantidad de trabajo, solo valdrá la pena realizar el esfuerzo de crear un Sistema Experto cuando un conocimiento sea válido durante un largo espacio de tiempo y vaya a ser utilizado por el mayor número posible de personas.

Por lo tanto los Sistemas Expertos ofrecen una solución y ayuda para:

- Evitar fallas en labores rutinarias complejas
- Ampliar de forma más rápida los conocimientos de los especialistas
- Diagnosticar las fallas con mayor rapidez
- Conseguir tareas de planificación más complejas y consistentes

En este sentido, el Sistema Experto supone una descarga del experto en el trabajo rutinario, y por lo tanto, la reducción de sus problemas. Además de que cuando la labor del experto no está tan sobrecargada, se reducen las decisiones erróneas y se aceleran los procesos de toma de decisiones. De tal forma que el experto se puede dedicar a tareas de mayor valor para su organización.

1.3.5 PROBLEMAS Y LIMITACIONES DE LOS SISTEMAS EXPERTOS

En contraste los sistemas expertos presentan también grandes carencias frente a los seres humanos:

- *Sentido común.* Para una computadora no hay nada obvio.
- *Flexibilidad.* La de un ser humano es ilimitada mientras que los sistemas expertos de la actualidad y del futuro próximo seguirán siendo terriblemente rígidos.
- *Lenguaje natural.* Todavía estamos muy lejos de tener un sistema que pueda formular preguntas flexibles y mantener una conversación informal con un usuario o con un paciente.
- *Experiencia sensorial.* Los sistemas expertos de la actualidad generalmente se limitan a recibir información mediante un *teclado* y un *ratón*. A excepción de la automatización industrial de procesos repetitivos en la línea, la cual consiste, principalmente, en diseñar sistemas programables que pueden operar independientemente del control humano,

mediante el uso de robots computarizados que están equipados con pequeños microprocesadores capaces de procesar la información que le proveen *sensores externos* y así es como el robot puede tomar, cambiar o mantener una operación en ejecución, a esto se le llama retroalimentación. La retroalimentación es esencial en cualquier mecanismo de control automático, ya que ayuda a controlar los factores externos que le afecten en la correcta ejecución de sus operaciones normales.

La mayoría de las industrias han sido automatizadas o utilizan la tecnología para automatizar algunas labores; se puede mencionar como ejemplo a la industria de la telefonía, en la cual la marcación, transmisión y facturación están completamente automatizadas.

- *Perspectiva global.* Un experto humano es capaz de detectar inmediatamente cuáles son las cuestiones centrales y cuáles son secundarias (separando los datos relevantes de los detalles insignificantes).

Además existen éstas otras limitaciones:

- *Falta de capacidad de aprendizaje.* Los expertos humanos son capaces de aprender de la experiencia.
- *Capacidad de manejar conocimiento no estructurado.* El experto humano organiza y usa la información y el conocimiento presentados de forma poco ordenada.
- *Funciones genuinamente humanas.* Por ejemplo, todo lo relacionado con el lenguaje natural, la formación de conceptos, el conocimiento de sentido común y la creación queda fuera de los Sistemas Expertos, al menos en el estado actual del conocimiento.

Por tanto, y en la actualidad, la idea de sustituir al humano por la máquina es aún irrealizable. Aunque el objetivo principal no es que la máquina duplique o sustituya al humano sino que colabore con él.

1.4 ESTRUCTURA DE LOS SISTEMAS EXPERTOS

Un programa tradicional esta compuesto por unas entradas y salidas de datos, unos datos (constantes o variables), unos algoritmos (lógicos, aritméticos) y unas sentencias de control, todo ello formando una misma unidad. Como resumen un programa clásico puede representarse de la forma:

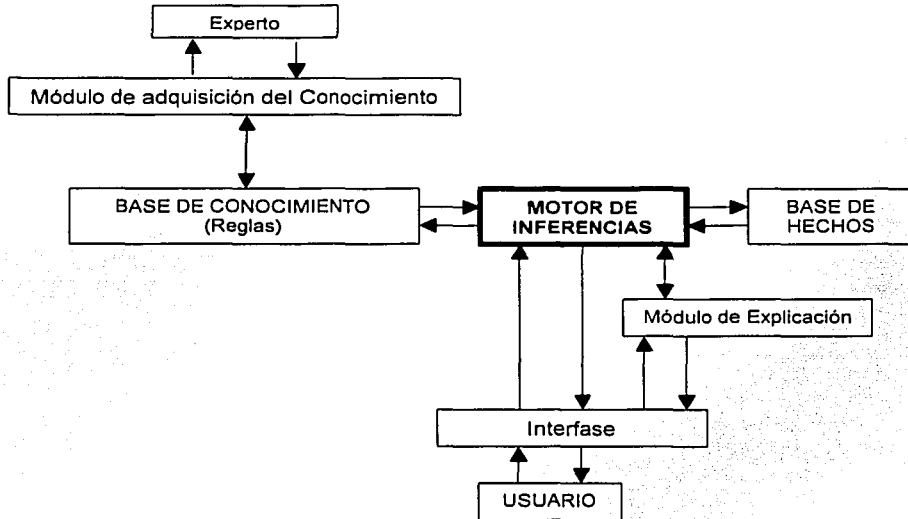
Datos + Algoritmos + Control + Entradas/Salidas = Programa

Por lo general en un Sistema Experto estos elementos son independientes unos de otros y forman unidades separadas. Los datos están agrupados en lo que se denomina "Base de Hechos" (BH), los algoritmos no existen y en su lugar se utilizan sistemas de representación del conocimiento de tipo declarativo que forman "La Base de Conocimientos" (BC), el control es independiente y se denomina "Motor de Inferencia" (MI), por último, la entrada y salida de datos es similar a los programas tradicionales en cuanto a los dispositivos empleados para recibir información y presentar los resultados (teclados, monitores, ratón, etc.).

Una forma de representar cómo está formado un Sistema Experto, podría ser:

- El Motor de Inferencias (M.I.)
- Los Módulos de Comunicación o de Entrada/Salida que se subdividen en:
 - El Módulo de Consulta o del Usuario
 - El Módulo de Trabajo o del Experto

1.4.1 ARQUITECTURA DE UN SISTEMA EXPERTO TÍPICO



1.4.3 COMPONENTES DE UN SISTEMA EXPERTO

Una característica decisiva de los Sistemas Expertos es la separación entre conocimiento (reglas, hechos) por un lado y su procesamiento por el otro. A ello se añade una interfase de usuario y un componente explicativo.

A continuación se muestra una breve descripción de cada uno de los componentes:

- **Módulo de Adquisición del Conocimiento.** No existe siempre. Mantiene la consistencia de la Base de Conocimientos, puede incluir un editor y corregir errores sintácticos.
- **Base de Conocimientos.** Contiene codificado en algún lenguaje el conocimiento que se tiene del sistema que involucra los hechos y experiencias de los expertos en un dominio determinado.
- **Base de Hechos.** Es una memoria de trabajo que contiene en cada momento toda la información relativa al problema que se está resolviendo.
- **Motor de Inferencias.** Combina los hechos ciertos de la Base de Hechos, con la información de la Base de Conocimientos, para establecer nuevos hechos ciertos. Es decir simula la estrategia de solución de un experto.
- **Módulo de Explicación.** Módulo que permite al programa explicar su funcionamiento al usuario, normalmente indicando los pasos que han llevado a dar la solución propuesta.

- **Interfase de Usuario.** Componente de comunicación Hombre-Máquina. Sirve para que el usuario pueda realizar una consulta en un lenguaje lo más natural posible.

1.4.4 LA BASE DE CONOCIMIENTOS

La Base de conocimientos contiene todos los hechos, las reglas y los procedimientos del dominio de aplicación que son importantes para la solución del problema.

Existen muchos modos de representar el conocimiento. La programación orientada a objetos se utiliza con frecuencia para crear la base de conocimientos. Puede darse el caso que determinados procesos y funciones puedan representarse con objetos. Cómo se lleva a cabo la clasificación en un grupo de las características y de los procedimientos alrededor de un objeto con las técnicas de programación y cómo deben ser las relaciones entre los objetos pueden variar mucho de aplicación a aplicación.

La base de conocimientos se crea utilizando dos tipos de conocimiento. El primero se conoce como "Conocimiento Basado en Hechos" que se basa en información y datos encontrados sobre el tema a tratar. El segundo se conoce como "Conocimiento Heurístico" el cual se basa en experiencias y prácticas. Es importante considerar que estos dos tipos de conocimiento muchas veces se utilizan en conjunto para poder crear una base de conocimientos mejor. Aunque estos conocimientos no son iguales dado que el conocimiento basado en hechos se basa más en los datos recopilados y el conocimiento heurístico en experiencias, en conjunto trabajan muy eficazmente. Estos dos tipos de conocimiento nos ayudan a recopilar la información de forma más completa. Basándonos en la información obtenida, ésta es generalmente representada mediante reglas de producción. Estas reglas son de la forma:

Si Premisas Entonces Conclusión y/o Acción

En la zona de las premisas se solicitan vinculaciones lógicas referentes a las cualidades de los objetos. En la zona de la conclusión se añaden nuevos hechos y cualidades a la base de conocimientos y/o se ejecutan acciones. Esto se define a menudo como programación orientada a reglas.

En la creación de la Base de Conocimientos se plantean las preguntas del tipo:

- ¿Qué objetos serán definidos ?
- ¿Cómo son las relaciones entre los objetos?
- ¿Cómo se formularán y procesarán las reglas?
- ¿La base de conocimientos hace totalmente referencia a la solución del problema?
- ¿La base de conocimientos es consistente?

Las respuestas a estas preguntas son el trabajo del Ingeniero del Conocimiento junto con la colaboración de los expertos.

1.4.5 LA BASE DE HECHOS

La Base de Hechos, es el conjunto de información que conforma el universo del Sistema Experto, y en base a ellos (los hechos), y mediante la Base de Conocimientos, el Sistema Experto llega a la solución.

Se denominan *hechos* a la información que es invariable de una a otra resolución mientras que se suele denominar *datos* a aquella información que varía de una a otra resolución. Esta diferencia es importante pues los *hechos* pueden formar parte del Sistema Experto mientras que

los *datos* deben estructurarse en archivos independientes del programa principal que constituye el Sistema Experto.

La Base de Hechos es una *memoria auxiliar* que contiene a la vez los *hechos* iniciales que describen el enunciado del problema a resolver y los resultados intermedios a lo largo del procedimiento de deducción. Esta base (memoria temporal) no se conserva (salvo por necesidades del usuario) y depende exclusivamente de la situación estudiada.

Para algunos autores los *conocimientos* y los *hechos* aparecen conjuntamente en lo antes denominado Base de Conocimientos. También es habitual que a la Base de Hechos cuando está unida con una Base de Datos se denomine Memoria de Trabajo.

1.4.6 EL MOTOR DE INFERENCIA

El mecanismo de inferencia es la unidad lógica con la que se extraen conclusiones de la Base de Conocimientos, según un método fijo de solución de problemas que esta configurado imitando el procedimiento humano de los expertos para solucionar problemas.

Una conclusión se produce mediante aplicación de las reglas sobre los hechos presentes.
Ejemplo.

Una Regla es: Si p y q entonces r
Se dan los hechos: p y q

p y q son justo aquellos hechos que se mencionan en la cláusula "si" de la regla, es decir, las condiciones para la aplicabilidad de la regla. Aplicar la regla es: deducir de los hechos p y q el hecho r.

En un Sistema Experto existirá un hecho sólo cuando esté contenido en la base de conocimientos. Los hechos que constan en la cláusula "si" se llaman *premisas*, y el contenido en la cláusula "entonces" se llama *conclusión*. Cuando se aplica una regla sobre algunos hechos cualesquiera se dice que se *dispara*. El disparo de una regla provoca la inserción del nuevo hecho en la base de conocimientos.

Las funciones del mecanismo de inferencia son:

- Determinación de las acciones que tendrán lugar, el orden en que lo harán y cómo lo harán entre las diferentes partes del Sistema Experto.
- Determinar cómo y cuándo se procesarán las reglas, y dado el caso también la elección de qué reglas deberán procesarse.
- Control del diálogo con el usuario.

La decisión sobre los mecanismos de procesamiento de reglas, es decir, qué estrategias de búsqueda se implementarán, es de vital importancia para la efectividad del sistema en su conjunto.

Ante problemas o clases de problemas distintos se estructuran, como es lógico, diferentes mecanismos de inferencia. El mecanismo de inferencia debe de estar "adaptado" al problema a solucionar.

TESIS CON
FALLA DE ORIGEN

1.4.7 MÓDULO DE COMUNICACIÓN

En un Sistema Experto típico, existe el Módulo de Comunicación que se conoce como el *Módulo de Comunicación del Usuario*.

Un Sistema Experto por otra parte normalmente necesita comunicarse con otros programas (por ejemplo bases de datos) o dispositivos (por ejemplo sensores).

Un *Módulo de Comunicación* debe ser:

- Rápido, con el fin de que la comunicación sea rápida y que no sea pesada para el usuario.
- Potente, para que admita estructuras flexibles que sean cercanas al lenguaje natural.
- Sencillo, que no implique el estudio de complejas estructuras y extensos vocabularios.
- Adecuado, para que trabaje en el nivel de cada uno de los usuarios que tenga el sistema.

Con el fin de conseguir estos objetivos se utilizan: pantallas gráficas de alta resolución, selección y direccionamiento desde pantalla, sistemas multiventanas, ampliación de zonas de la pantalla (zoom), profundización incremental en las preguntas y respuestas, uso de iconos y color, animación, ayuda en línea, representación en árboles de la base de conocimientos, posibilidad de abreviación de vocablos, diálogos en lenguaje "pseudonatural", acceso a bibliotecas.

Módulo del Usuario. El Módulo de Comunicación con el Usuario debe permitir el diálogo de forma sencilla entre el usuario y el Sistema Experto, aproximándose lo mas posible al lenguaje natural. Las tareas mas importantes que desempeña este módulo son:

- Entrada de datos
- Entrada de opciones en procesos de iniciativa compartidas
- Salida de Explicaciones
- Salida de Justificaciones
- Salida de Soluciones

1.4.8 ADQUISICIÓN DEL CONOCIMIENTO

La adquisición del conocimiento a través de la comunicación con el Experto se puede dividir en varias etapas:

- Adquisición de Conocimientos. Permite la inclusión del conocimiento del experto en la Base de Conocimientos. Solamente se utiliza en la etapa de construcción del sistema Experto.
- Mantenimiento del Conocimiento. Este se mantiene durante toda la vida del Sistema Experto, con el fin de modificar e incrementar la Base de Conocimientos, de una forma sencilla.
- Validación y Depuración del Conocimiento. Permite detectar repeticiones, inconsistencias, errores, etc.
- Configuración del Sistema. Permite configurar el Motor de Inferencia y el Módulo de comunicaciones a los requerimientos del usuario. Si el Sistema Experto se desarrolla desde un lenguaje de alto nivel, este submódulo realmente es el editor del lenguaje.

1.5 REPRESENTACIÓN DEL CONOCIMIENTO

Para el procesamiento y la manipulación del conocimiento en Sistemas Expertos es necesario formalizar y estructurar dicho conocimiento. En su mayor parte, se dispone del conocimiento a través de entrevistas con los expertos en forma de descripciones de casos o en partes de su actividad.

Los métodos formales de representación del conocimiento son distintos aspectos de la lógica; por ejemplo, lógica de predicados, lógica modal, lógica multivaluada y lógica difusa. Estos métodos formales y matemáticos no son métodos apropiados para comunicarse con expertos ni tampoco ofrecen formas generales de representación del saber.

Para ello se han desarrollado procedimientos de representación del conocimiento que ofrecen un apoyo eficiente a la estructuración y al procesamiento del saber. Por mencionar los más utilizados: Lógica de Predicados, Reglas de Producción, Marcos, Redes Semánticas, etc.

1.5.1 LA LÓGICA COMO REPRESENTACIÓN DEL CONOCIMIENTO

El sistema de lógica formal más ampliamente llamada *Lógica de Predicados*, consiste en cuatro componentes principales: un *alfabeto*, un *lenguaje formal*, un *conjunto de enunciados básicos* llamados *axiomas* y un *conjunto de reglas de inferencia*. Cada axioma describe un fragmento de conocimientos y las reglas de inferencia se aplican a los axiomas para deducir nuevos enunciados verdaderos.

El alfabeto para el lenguaje formal consiste en los símbolos a partir de los cuales se construyen los enunciados. El alfabeto consta de predicados, variables, funciones, constantes, conjunciones, cuantificadores y delimitadores tales como paréntesis y comas.

Ejemplo. Formulando la siguiente expresión en lógica de predicados: "BOBY es un perro negro"

La estructura de esta frase se divide en dos oraciones: (1) "BOBY es un perro" y (2) "BOBY es negro". Un predicado, PERRO se usa para la primera frase y otro, NEGRO para la segunda. La fórmula bien definida es en consecuencia $PERRO(BOBY) \wedge NEGRO(BOBY)$.

1.5.2 REPRESENTACIÓN BASADA EN REGLAS

La regla es la forma más común de representar el conocimiento, debido a su gran sencillez y a ser la formulación más inmediata del principio de causalidad (causa-efecto).

Una regla consta de un conjunto de acciones o efectos (una o más) que son ciertas cuando se cumplen un conjunto de condiciones o causas (una o más). La potencia de una regla esta en función de la lógica que admita en las expresiones de las condiciones y de las conclusiones. Ejemplo:

Hecho: En mi jardín tengo una PLANTA PERENNE, ALTA Y CON EL TRONCO DESNUDO.

Regla 1: Un árbol es una planta perenne, alta y con el tronco desnudo.

Nuevo Hecho: En mi jardín tengo un árbol.

La fórmula general de una regla es la siguiente: Si < Condiciones > ENTONCES < Conclusión o Acciones >.

La conclusión suele referirse a la creación de un nuevo hecho válido, o la incorporación de una nueva característica a un hecho, mientras que la acción suele referirse a la transformación de un hecho. Para el ejemplo:

REGLA 1: < árbol > SI < planta Y perenne Y alta Y tronco desnudo >
 REGLA 2: < arbusto > SI < planta Y perenne Y NO alta Y tronco NO desnudo >
 REGLA 3: < alta > SI < altura ≥ 6 metros > ...

Existe una restricción de la sintaxis de las reglas de producción, y ésta consiste en que sólo existe una conclusión por regla y ésta conclusión no puede aparecer negada.

El conocimiento acerca de las reglas de producción que siguen esta misma estructura, se denominan Meta-reglas. Las Meta-reglas facilitan la resolución de los problemas, pues si la Base de Conocimientos está construida de forma modular, las Meta-reglas pueden inhibir parte de los módulos lo que hace más rápida la búsqueda de soluciones.

1.5.3 REPRESENTACIÓN BASADA EN MARCOS (FRAMES)

Un marco es la división de objetos, o también de situaciones, en sus componentes. Estos componentes son introducidos en ranuras (slots) correspondientes del marco. Las ranuras pueden contener uno o más enlaces (facets). Cada enlace tiene un valor asociado. Varios enlaces pueden ser definidos por cada ranura, por ejemplo:

- Rango. Es el conjunto de posibles valores para el slot.
- Valor. Es el valor del slot.
- Default. Es el valor a ser asumido si no se especifica alguno

Representación de un marco:

Nombre del Marco	Objeto 1	
Clase	Objeto 2	
Propiedad	Propiedad 1	Valor 1
	Propiedad 2	Valor 2
	Propiedad 3	Valor 3
	Propiedad 4	Valor 4

Nombre del Marco	Canario	
Clase	Pájaro	
Propiedades	Color	Amarillo
	Comida	Alpiste
	No. de alas	2
	Hambriento	desconocido

slot facet

Antes de su utilización, el marco es un armazón preestructurado de datos. La configuración del marco y las definiciones de los slots están ya fijadas. A lo largo del proceso se van rellenando los slots con contenidos. En este proceso puede haber varios marcos con la misma estructura pero diferente contenido. Los valores de un slot (slot-values) son heredables. De esta forma no hace falta modificar mas que el valor jerárquicamente superior en el slot y todas las instancias subordinadas del marco obtienen el nuevo valor.

1.5.4 REDES SEMÁNTICAS

Las Redes Semánticas son un método de representación del conocimiento sobre las relaciones de los objetos. Los nodos de una red semántica corresponden a los objetos y los arcos describen las relaciones entre los objetos. Por lo tanto podríamos tomar un arco con sus dos nodos como una sola unidad de conocimiento.

Con la red semántica no se da información sobre el proceso de la red, las reglas de inferencia deben de estar expresadas en forma explícita. Una red semántica ofrece una buena visión general sobre las relaciones y dependencias de un área de conocimientos y es muy apropiada

para la estructuración del conocimiento y verificación del experto. Sin embargo, los enunciados de las relaciones mencionadas en los arcos deben ser formuladas fuera de la red.

En la red semántica puede haber relaciones unidireccionales (dependencia de una relación entre objetos) y relaciones bidireccionales (relaciones entre objetos). Las relaciones unidireccionales se representan con una flecha en la dirección del objeto. Ejemplo.

Establecemos las siguientes relaciones:

Temp(sangre caliente, mamíferos) Los mamíferos son de sangre caliente; *Temp* es la relación que describe la temperatura corporal.

Es_un(perro, mamífero) La relación de *Es_un* es una relación unidireccional, cada perro es un mamífero, pero no cada mamífero es un perro.

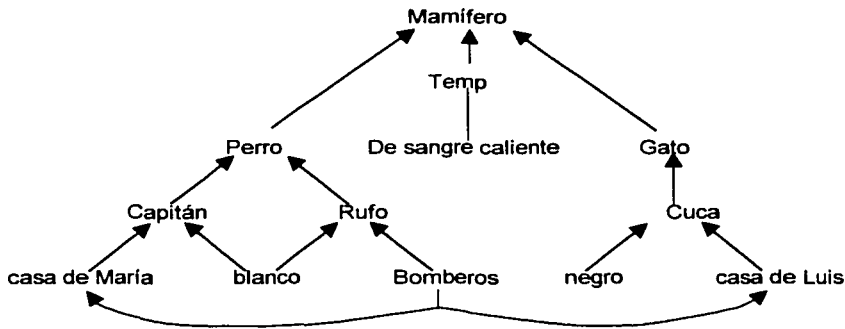
Es_un(gato, mamífero) ; *Es_un*(Capitán, perro); *Es_un*(Rufo, perro); *Es_un*(Cuca, gato)

Lugar(casa_de_María, Capitán) Con la relación de *Lugar* se crea un enunciado sobre le lugar donde vive Capitán; vive en casa de María.

Lugar(Bomberos, Rufo); *Lugar*(casa_de_Luis, Cuca)

Color(blanco, Capitán); *Color*(blanco, Rufo); *Color*(negro, Cuca).

Entre(casa_de_María, Bomberos, casa_de_Luis) Bomberos esta entre la casa de María y la casa de Luis.



1.6 SISTEMAS BASADOS EN REGLAS (SBR)

Un Sistema Experto basado en reglas es un programa capaz de procesar información específica contenida en memoria con un conjunto de reglas, contenidas en la *Base de Conocimientos*, mediante un *Motor de Inferencia* para deducir nueva información.

Actualmente son los más populares para codificar los conocimientos debido a que los expertos tienden a expresar la mayoría de sus técnicas de solución de problemas en términos de un conjunto de reglas "situación - reacción", y ésta es una de las razones para que se adopte tan frecuentemente este tipo de representación.

Estos SBR presentan las siguientes características:

- Incorporan conocimientos prácticos en reglas del tipo : " *Si ... – ENTONCES ...* "

- Sus capacidades, en principio, aumentan al aumentar el tamaño de sus Bases de Conocimiento. Puesto que cada regla representa un "átomo" de conocimiento, estos pueden acumularse gradualmente, aumentando la potencia del sistema.
- Pueden resolver problemas de cierta complejidad seleccionando reglas relevantes, y combinando los resultados de manera adecuada.
- Determinan adaptativamente la mejor secuencia para ejecutar las reglas.
- Explican sus conclusiones por medio de un seguimiento de la cadena de razonamiento ejecutada, y traduciendo a lenguaje natural la lógica de cada regla aplicada.

Los conocimientos representados en las reglas pueden ser, por ejemplo:

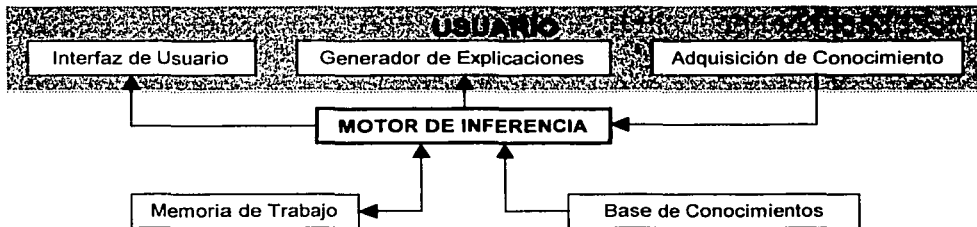
- Inferencias específicas que se guían de observaciones específicas.
- Abstracciones, generalizaciones y categorizaciones de datos dados.
- Condiciones necesarias y suficientes para conseguir algún objetivo.
- Lugares más prometedores para buscar información pertinente.
- Estrategias preferidas para eliminar incertidumbre y disminuir otros riesgos.
- Probables consecuencias de situaciones hipotéticas.
- Probables causas de síntomas.

Los sistemas basados en reglas difieren de la representación basada en la lógica en las siguientes características principales:

- Son en general no-monoatómicos, es decir hechos o afirmaciones derivadas, pueden ser retractados, en el momento en que dejen de ser verdaderos.
- Pueden aceptar incertidumbre en el proceso de razonamiento.

1.6.1 COMPONENTES DE UN SISTEMA EXPERTO BASADO EN REGLAS (SEBR)

Una característica decisiva de los Sistemas Expertos es la separación entre el conocimiento (reglas, hechos) por un lado y su procesamiento por el otro. A ello se añade una interfase de usuario y un componente explicativo.



Esta estructura no se modifica para el caso de los SEBR, por lo que conservan los componentes y módulos de un Sistema Experto Típico. Sin embargo, para los SEBR hay que tomar en cuenta las siguientes consideraciones:

- Para la *Adquisición de Conocimientos* las reglas deben de introducirse de la forma mas sencilla posible.
- En cuanto a la *Base de Conocimientos*, cuando los conocimientos almacenados han quedado obsoletos, o cuando se dispone de nuevos conocimientos, es relativamente fácil añadir reglas nuevas, eliminar las antiguas o corregir errores en las existentes. No es necesario reprogramar todo el sistema. Las reglas suelen almacenarse en alguna

secuencia jerárquica lógica, pero esto no es estrictamente necesario. Se pueden tener en cualquier secuencia y el *Motor de Inferencia* las usará en el orden adecuado que necesite para resolver un problema.

- Por último, el *Motor de inferencia* utiliza procesos de búsqueda o estrategias de control, básicamente de dos formas:
 1. Se puede partir considerando todos los datos conocidos y luego ir progresivamente avanzando hacia la solución. Este proceso se le denomina *guiado por los datos* o de **encadenamiento progresivo** (*forward chaining, hacia delante, deductivo o modus ponens*).
 2. Se puede seleccionar una posible solución y tratar de probar su validez buscando evidencia que la apoye. Este proceso se denomina *guiado por el objetivo* o de **encadenamiento regresivo** (*backward chaining, hacia atrás, inductivo, modus tollens*).

1.6.2 ESTRATEGIAS DE CONTROL. ENCADENAMIENTO HACIA DELANTE Y HACIA ATRÁS

Encadenamiento hacia delante. En este caso se comienza con los hechos disponibles en la Base de Datos, y se buscan reglas que satisfagan esos datos, es decir, reglas que verifiquen la parte *Si*. A este enfoque se le llama también *guiado por datos*, por que es el estado de la Base de Datos el que identifica las reglas que se pueden aplicar. Cuando se utiliza este método, el usuario comenzará introduciendo datos del problema en la Base de Datos del sistema. Los pasos que lleva a cabo esta estrategia son:

Paso 1 . Interacción (Matching). En este paso, en las reglas en la *Base de Conocimientos* se prueban los hechos conocidos al momento para ver cuáles son las que resultan satisfechas. Para decir que una regla ha sido satisfecha, se requiere que todas las premisas o antecedentes de la regla resuelvan a verdadero.

Paso 2. Resolución de Conflictos. Es posible que en la fase de unificación resulten satisfechas varias reglas. La resolución de conflictos involucra la selección de la regla que tenga la más alta prioridad de entre el conjunto de reglas que ha sido satisfechas.

Paso 3. Ejecución. El último paso en la interpretación de reglas es la ejecución de la regla. La ejecución puede dar lugar a uno o dos resultados posibles: nuevo hecho o hechos pueden ser derivados y añadidos a la Base de Hechos, o una nueva regla o reglas pueden ser añadidas a la Base de Conocimiento que el sistema considera para ejecución.

En esta forma, la ejecución de las reglas procede de una forma progresiva (hacia adelante) hacia los objetivos finales.

Un conjunto de aplicaciones adecuadas al razonamiento progresivo incluye supervisión y diagnóstico en sistemas de control de procesos en tiempo real, donde los datos están continuamente siendo adquiridos, modificados y actualizados. Estas aplicaciones tienen dos características importantes:

- Necesidad de respuesta rápida a los cambios en los datos de entrada.
- Existencia de pocas relaciones predeterminadas entre los datos de entrada y las conclusiones derivadas.

Encadenamiento hacia atrás. Se le conoce también como *guiado por objetos*, ya que, el sistema comenzará por el objetivo (parte acción de las reglas) y operará retrocediendo para ver cómo se deduce ese objetivo partiendo de los datos. Esto se produce directamente o a través de

conclusiones intermedias. Lo que se intenta es probar una hipótesis a partir de los hechos contenidos en la Base de Datos y de los obtenidos en el proceso de inferencia.

El sistema comienza con un conjunto de hechos conocidos el cual típicamente está vacío. Se proporciona una lista ordenada de objetivos (o conclusiones), para los cuales el sistema trata de derivar valores. El encadenamiento hacia atrás utiliza esta lista de objetivos para coordinar su búsqueda a través de las reglas de la *Base de Conocimientos*. Ésta búsqueda consiste de los siguientes pasos:

Paso 1. Conformar una pila inicialmente compuesta por todos los objetivos prioritarios definidos en el sistema.

Paso 2. Considerar el primer objetivo de la pila. Determinar todas las reglas capaces de satisfacer este objetivo, es decir aquellas que mencionen al objetivo en su conclusión.

Paso 3. Para cada una de estas reglas examinar en turno sus antecedentes:

- a. Si todos los antecedentes de la regla son satisfechos (esto es, cada parámetro de la premisa tiene su valor especificado dentro de la Base de Datos), entonces ejecutar esta regla para derivar sus conclusiones. Debido a que se ha asignado un valor al objetivo actual, removerlo de la pila y retornar a al Paso 2.
- b. Si alguna premisa de la regla no puede ser satisfecha, buscar reglas que permitan derivar el valor especificado para el parámetro utilizado en esa premisa.
- c. Si en el paso (b) no se puede encontrar una regla para derivar el valor especificado para el parámetro actual, entonces preguntar al usuario por dicho valor y añadirlo a la Base de Datos. Si este valor satisface la premisa actual entonces continuar con la siguiente premisa de la regla. Si la premisa no es satisfecha, considerar la siguiente regla.

Si todas las reglas que pueden satisfacer el objetivo actual se han probado y no han podido derivar un valor, entonces este objetivo quedará indeterminado. Por lo tanto hay que removerlo de la pila y retornar al Paso 2. Si la pila esta vacía parar y anunciar que se ha terminado proceso.

Esta estrategia de control es mucho mas adecuada para aplicaciones que tienen mucho mayor número de entradas. Su razonamiento es mas eficiente que el encadenamiento hacia delante. Una excelente aplicación es el diagnóstico, donde el usuario dialoga directamente con el sistema basado en conocimiento y proporciona los datos a través del teclado

1.7 TIPOS DE HERRAMIENTAS PARA SISTEMAS EXPERTOS

Para el desarrollo de un Sistema Experto se han utilizado diversos tipos de "herramientas", que se pueden clasificar en 5 categorías:

- Lenguajes tradicionales, que son orientados a procesos numéricos (Pascal, C, etc...).
- Lenguajes orientados a objetos (Smalltalk, Eiffel, C++, etc...).
- Lenguajes orientados a manipulación simbólicas como LISP.
- Lenguaje de "programación lógica", PROLOG.
- Herramientas dedicadas para Sistemas Expertos ("Expert System Shell" como Rosie, Cadeceus, OPS5, Age, CLIPS, etc...).

Las herramientas dedicadas para Sistemas Expertos, a su vez se caracterizan, por ejemplo, como:

- *Inductivas*. Que tienen la capacidad de inducir reglas a partir de ejemplos.
- *Basadas en reglas simples*. Que no organizan sus reglas dentro de módulos.
- *Basadas en reglas estructuradas*. Que organizan sus reglas en grupos.

- *Híbridas.* Que utilizan variadas estructuras de datos.
- *Específicas.*

1.7.1 SELECCIÓN DE UNA HERRAMIENTA

Los principales factores que se deben considerar en el momento de seleccionar una herramienta de desarrollo (Shell) para implementar un Sistema Experto son:

- Que tenga suficiente potencia en cuanto a la capacidad para manejar una Base de Conocimientos del tamaño que nos interesa y con una buena velocidad de respuesta.
- Que sea flexible para que el trabajado de desarrollo se pueda realizar ágilmente.
- Que las estructuras que usan para representar el conocimiento sean adecuadas para el tipo de problema que deseamos resolver.
- Que los métodos de razonamiento que utiliza también sean adecuados al tipo de problema que enfrentamos.
- Que las interfaces que ofrece ante el desarrollador y ante el usuario sean de fácil manejo y contribuyan a facilitar tanto el desarrollo como la explotación, en vez de hacerlos difíciles.
- Que ofrezca las interfaces necesarias con otros sistemas como: procedimientos escritos en lenguajes de programación convencional y Bases de Datos.
- Que el proveedor respalde el producto con entrenamiento, asesoría y soporte.
- Que el software funcione en la plataforma de hardware que tenemos disponible para el desarrollo y la posterior explotación del sistema.
- Que los costos estén dentro del presupuesto destinado a este proyecto.

1.8 CAMPOS DE APLICACIÓN DE LOS SISTEMAS EXPERTOS

Sector Aplicación	Banca Seguros	Industria	Comercio Servicios	Encargos Estatales
Control de procesos, supervisión	<ul style="list-style-type: none"> • Observación de tendencias 	<ul style="list-style-type: none"> • Control de procesos • Gobierno de procesos • Aviso de estados de excepción 	<ul style="list-style-type: none"> • Observación de tendencias 	<ul style="list-style-type: none"> • Control de centrales nucleares o de grandes redes (agua, gas)
Diseño		<ul style="list-style-type: none"> • Configuración • Instalaciones fabriles • Diseño de productos 	<ul style="list-style-type: none"> • Requisitos de productos 	<ul style="list-style-type: none"> • De redes de distribución (correos, energía)
Diagnóstico	<ul style="list-style-type: none"> • Concesión de créditos • Comprobación de hipotecas • Análisis de siniestros 	<ul style="list-style-type: none"> • Motivo de fallo • Mantenimiento 	<ul style="list-style-type: none"> • Concesión de créditos • Cálculo de riesgos 	<ul style="list-style-type: none"> • Diagnóstico Médico (Hospitales) • Diagnóstico Técnico • Economía Energética
Planificación	<ul style="list-style-type: none"> • Análisis de riesgos 	<ul style="list-style-type: none"> • Funciones Lógicas de Proyectos 	<ul style="list-style-type: none"> • Análisis de riesgos 	<ul style="list-style-type: none"> • Planificación de inversiones

	<ul style="list-style-type: none"> • Gestión de valores • Planificación de Inversiones 	<ul style="list-style-type: none"> • Proyectos 	<ul style="list-style-type: none"> • Análisis e mercado 	<ul style="list-style-type: none"> • Planificación de emergencias • Planificación de la distribución
Asesoramiento	<ul style="list-style-type: none"> • Asesoramiento de clientes 	<ul style="list-style-type: none"> • Asesoramiento de clientes 	<ul style="list-style-type: none"> • Asesoramiento de clientes • Servicios especiales 	<ul style="list-style-type: none"> • Asesoramiento de clientes
Formación	<ul style="list-style-type: none"> • Formación de colaboradores • Formación del servicio exterior 	<ul style="list-style-type: none"> • Formación de colaboradores 	<ul style="list-style-type: none"> • Formación de colaboradores • Formación del servicio exterior 	<ul style="list-style-type: none"> • Formación interna en cuestiones jurídicas

2. CLIPS

2.1 INTRODUCCIÓN

En este capítulo damos a conocer uno de tantos shells que existen para la construcción de sistemas expertos. También se da a conocer un panorama general de la programación con CLIPS. Se mencionan los elementos y conceptos básicos para la construcción de un Sistema Experto en particular con CLIPS. No se pretende dar a conocer todo un manual sobre la programación con CLIPS pero si un panorama general.

2.2 ¿QUE ES CLIPS?

CLIPS son las iniciales de C Language Integrated Production System y es una herramienta para el desarrollo y construcción de Sistemas Expertos Basados en Reglas (SEBR) y/u objetos creada por la Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center. Los orígenes de CLIPS se remontan al año de 1984, cuando éste se diseñó para facilitar el desarrollo de software que modele el conocimiento humano.

CLIPS es una herramienta de libre distribución para el desarrollo de los Sistemas Expertos y las plataformas que soporta son: MS-DOS, Windows, Unix y Macintosh.

La Versión mas reciente para plataforma Windows es la 6.2. Ésta versión soporta los paradigmas heurísticos, procedurales y orientado a objetos de la programación para representar el conocimiento.

Otras características importantes de CLIPS son las siguientes:

- Con CLIPS se puede desarrollar software formado sólo por reglas, sólo por objetos, o combinación de ambos.
- CLIPS se ha diseñado para poder ser integrado con otros lenguajes como C, ADA y FORTRAN. Puede llamarse desde otros lenguajes para que CLIPS desarrolle la función y retorne la salida y el control al programa que lo llamó, el conocimiento procedural puede definirse como funciones externas, ser llamadas por CLIPS y retornar la salida y el control a CLIPS.
- CLIPS usa un mecanismo de encadenamiento hacia adelante, aunque puede también emular el encadenamiento hacia atrás pero de forma restringida.
- El motor de inferencias incluye un sistema de mantenimiento de verdad, adición dinámica de reglas, y diferentes estrategias de resolución de conflictos.

2.3 REPRESENTACIÓN DEL CONOCIMIENTO EN CLIPS

Como se mencionaba anteriormente, CLIPS soporta los paradigmas heurísticos, procedurales y orientado a objetos.

A) Conocimiento heurístico: reglas

Este tipo de conocimiento se expresa mediante reglas que se utilizan para representar el conocimiento del experto humano utilizando un conjunto de acciones a realizar para llegar a una situación dada. El creador del Sistema Experto define una colección de reglas que, en conjunto resuelven un problema.

Se puede pensar que las reglas son como sentencias IF-THEN de lenguajes procedurales como C. Sin embargo, las reglas actúan más bien como sentencias SIEMPRE QUE-ENTONCES.

B) Conocimiento procedural: funciones y objetos.

Este tipo de conocimiento se expresa mediante funciones definidas por el usuario (*defunctions*), funciones genéricas y la programación orientada a objetos. CLIPS soporta las características generalmente aceptadas en la programación orientada a objetos: clases, mensajes, abstracción, encapsulamiento, herencia y polimorfismo.

2.4 ESTRUCTURA DE UN PROGRAMA EN CLIPS

Como se menciona en el capítulo anterior, un sistema experto consta de cuatro elementos básicos: Base de Hechos, Base de Conocimientos, Motor de Inferencia y Módulos de Entrada/Salida. El *shell* de CLIPS que es la parte que realiza inferencias o razonamientos provee los primeros tres elementos básicos de un sistema experto:

1. Memoria de Trabajo o Base de Hechos (MT): contiene a la vez los *hechos* iniciales que describen el enunciado del problema a resolver y los resultados intermedios a lo largo del procedimiento de deducción hasta llegar a la meta.

2. Base de Conocimiento (*knowledge base*): contiene todos los hechos, las reglas y los procedimientos del dominio de aplicación que son importantes para la solución del problema.

3. Motor de Inferencia (*inference engine*): controla la ejecución global de las reglas: decide qué reglas deben ejecutarse y cuándo.

Un SEBR escrito en CLIPS es un programa dirigido por los datos (*data driven*), es decir, dirigido de acuerdo a los hechos y objetos que se vayan generando. Las reglas pueden interactuar (*match*) con objetos y hechos para que satisfagan sus condiciones, aunque los objetos pueden usarse por sí solos (mediante el envío de mensajes) sin utilizar el motor de inferencia.

2.5 MEMORIA DE TRABAJO (MT)

La Memoria de Trabajo es la parte donde temporalmente se almacenan las inferencias y aseveraciones tanto iniciales como las generadas a lo largo del procedimiento de deducción. Los datos en la Memoria de Trabajo son los que permiten cumplir las condiciones de las reglas y dispararlas para que las acciones de éstas modifiquen, añadan o quiten elementos de esta Memoria de Trabajo.

Todos los elementos de la Memoria de Trabajo son entidades que corresponden o bien a hechos o bien a instancias de una clase de objetos. La Memoria de Trabajo consta de una lista de hechos (*fact-list*) y de una lista de instancias (*instance-list*).

Un hecho es una forma básica de alto nivel para representar información. Es la unidad de datos fundamental utilizada en las reglas. Un hecho se compone de una serie de campos. Un campo es un lugar, con o sin nombre, que puede llevar asociado un valor.

Dentro de la Memoria de Trabajo se desarrollan todos los elementos básicos de programación en CLIPS, como son: los tipos de datos, las funciones y los constructores.

2.5.1 ELEMENTOS BÁSICOS DE PROGRAMACIÓN

CLIPS proporciona tres elementos básicos para escribir programas: algunos tipos de datos primitivos, funciones para manipularlos y constructores para añadirlos a la Base de Conocimiento.

2.5.1.1 ELEMENTOS PRIMITIVOS DE DATOS

Los tipos de datos que proporciona CLIPS son: reales (float), enteros (integer), símbolos (symbols), cadenas (strings), direcciones externas (external-address), direcciones de hechos (fact-address), nombres de instancias (instance-name) y direcciones de instancias (instance-address).

Un **número** consta de los siguientes elementos: los dígitos (0-9), un punto decimal (.), un signo (+ ó -) y, opcionalmente una notación exponencial (e) con su correspondiente signo. Los números se almacenan en CLIPS como reales o enteros. Un número se interpreta que es entero si consta de un signo (opcional) seguido de sólo dígitos. Cualquier otro número se interpreta como un real.

Un **símbolo** en CLIPS es cualquier secuencia de caracteres que no sigue exactamente el formato de un número. Más concretamente empieza con cualquier carácter ASCII imprimible y termina con un delimitador (caracteres ASCII no imprimibles). Los caracteres no imprimibles son: espacios, tabulaciones, retornos de carro, "line feeds", doble comillas, '(', ')', '&', '|', '<' y '~'. El carácter ';' también actúa como delimitador y es entendido por CLIPS que lo que se encuentre a partir de ahí hasta el final de línea es un comentario. CLIPS también distingue entre mayúsculas y minúsculas.

Una **cadena** es un símbolo que empieza y termina por dobles comillas. En el caso de que se desee que el string contenga las dobles comillas, antes se colocará la diagonal inversa [backslash (\)].

Una **dirección externa** es la dirección de una estructura de datos externos devuelta por una función escrita en C por ejemplo y que ha sido integrada con CLIPS. Este tipo de datos sólo puede importarse mediante la llamada a la función. La impresión de una dirección externa es de la forma <Pointer-XXX> donde XXX es la dirección externa.

Un **hecho** es una lista de uno o más valores que o bien son referenciados por su posición (para hechos ordenados) o por un nombre (para hechos no ordenados). La impresión de la **dirección de un hecho** es de la forma <Fact-XXX> donde XXX es el índice que ocupa el hecho en la memoria de CLIPS.

Una **instancia** es un caso particular de un objeto de CLIPS. Los objetos de CLIPS son números, símbolos, strings, valores multivariado, direcciones externas, direcciones de hechos e instancias de una clase definida por el usuario. Una clase definida por el usuario se crea mediante el constructor defclass. Y una instancia de la clase construida se hace con la función makeinstance.

Un **nombre de instancia** [instance-name] se construye encerrando entre corchetes un símbolo. Así, los símbolos puros no pueden empezar con corchetes. Es importante indicar que los corchetes no forman parte de la instancia, sino que los corchetes sólo se utilizan para indicar que el símbolo que se encuentra en su interior es la instancia.

Las **direcciones de las instancias** (instance-address) se presentan en el formato <Instance-XXX> donde XXX es el nombre de la instancia.

Por campo o casilla se entiende cualquier lugar que puede tomar un valor (de los tipos de datos primitivos) en una sentencia. De esta forma, atendiendo al número de campos que aparece en una sentencia se distinguen dos tipos de valores:

- Valores uni-campo: los formados por tipos de datos primitivos. En particular una constante es un valor uni-campo que no varía y está expresado como una serie de caracteres.

- Valores multi-campo: los formados por una secuencia de cero o más valores uni-campo.

Cuando CLIPS muestra los valores multicampo, éstos se muestran entre paréntesis. Así, no es lo mismo el valor uni-campo hola que el valor multicampo (hola).

2.5.1.2 FUNCIONES

Una función es la codificación de un algoritmo identificado con un nombre que puede o no devolver valores útiles a otras partes del programa. Existen dos grandes grupos de funciones y distinguiremos de entre ambos tipos de funciones denominando comandos o acciones a aquellas funciones que no devuelven valores, pero que generalmente tendrán algún efecto lateral útil.

- a) funciones (predefinidas) del sistema: Definidas internamente por el entorno de CLIPS.
 b) funciones definidas por el usuario: De este tipo de funciones podemos distinguir las siguientes:
- funciones externas: escritas en otro lenguaje (C, Ada) y *ligadas* con el entorno de CLIPS.
 - funciones definidas directamente en CLIPS: Utilizan sintaxis de CLIPS (*deffunctions*).
 - funciones genéricas: son funciones que permiten ejecutar diferentes fragmentos de código dependiendo del tipo y número de parámetros.

2.5.1.3 CONSTRUCTORES

Son estructuras sintácticas identificadas por una palabra reservada del lenguaje que permiten definir funciones, reglas, hechos, clases, etc., que alteran el entorno de CLIPS añadiéndolas a la base de conocimiento. Los constructores no devuelven ningún valor. Los más importantes son:

- **deffunction**: Para definir funciones
 - **defglobal**: Para definir variables globales
 - **deftemplate**: Para definir plantillas
 - **deffacts**: Para definir hechos
 - **defrule**: Para definir reglas
 - **defmodule**: Para definir módulos
 - **defclass**
 - **definstances**
 - **defmessage-handler**
 - **defgeneric**
 - **defmethod**
- } Para Objetos

Algunos de los constructores anteriores se mencionarán con más detalle a lo largo del capítulo

2.5.2 COMENTARIOS

CLIPS también permite comentar el código. Todos los constructores (excepto uno llamado *defglobal*) permiten incorporar en su definición un comentario directamente entre comillas (" "). En las demás partes, los comentarios pueden intercalarse con el código usando el punto y coma (;). CLIPS entiende que a partir del punto y coma hasta el final de línea es un comentario.

2.5.3 HECHOS

Un hecho (fact) representa un trozo de información que se almacena en la llamada lista de hechos (fact-list). A cada hecho en la lista se le asocia un identificador (fact identifier) que no es más que un índice asociado a ese hecho en la lista. Cuando a CLIPS se le pide que muestre el identificador

de un hecho, éste se muestra de la forma f -XXX, donde XXX denota al índice asociado. Por ejemplo, f -3 se refiere al hecho que tiene el índice 3. Se distinguen dos tipos de hechos: los hechos ordenados y los hechos no ordenados.

2.5.3.1 HECHOS ORDENADOS

Son los formados por campos y estos pueden ser de cualquier tipo primitivo de datos, excepto el primero que debe de ser un símbolo seguido de cero o más campos separados por espacios y todos ellos delimitados por paréntesis.

Para entender mejor este tipo de hechos podemos interpretar el primer campo, que es un símbolo, como una relación y el resto de los campos como los términos que se relacionan para dicha relación. Como se sabe, en una relación el orden de los términos que intervienen es importante ya que no es lo mismo decir que Manuel es el padre de Luis, que decir Luis es el padre de Manuel. Un modo de representar estos dos hechos es como sigue:

(padre "Manuel" "Luis") (padre "Luis" "Manuel")

Para tener una identidad o coincidencia (*matching*) con una regla, sus campos deben aparecer en el mismo orden que indique la regla. Por eso es importante remarcar que los hechos ordenados codifican la información según la posición de los campos, por lo que el usuario, cuando accede a la información, no sólo debe saber qué datos están almacenados, sino también qué campos contienen esos datos y esto lo vuelve un poco complicado a la hora de realizar consultas. Para evitar este problema, se recurre a los hechos no ordenados o plantillas.

2.5.3.2 HECHOS NO ORDENADOS O PLANTILLAS (DEFTEMPLATE)

Un hecho no ordenado es una secuencia de cero o más campos con nombre separados por espacios delimitados por paréntesis. La diferencia con los hechos ordenados es que ahora cada campo contiene un nombre y un valor que toma ese campo

Por ejemplo, la situación de parentesco que se mencionó anteriormente (Manuel es el padre de Luis) puede representarse como:

(parentesco (padre "Manuel") (hijo "Luis"))
(parentesco (hijo "Luis") (padre "Manuel"))

Debemos de hacer notar que ahora el orden de los campos en una plantilla no es importante, ya que los dos hechos anteriores son equivalentes.

Antes de empezar a definir el primer constructor, debemos entender la simbología usada para representar la sintaxis de los comandos. Todos los comandos en CLIPS comienzan y terminan en paréntesis. Sin embargo, cada comando presenta una sintaxis propia y una serie de opciones. Para poder utilizar una notación general para todos ellos se utilizará el siguiente convenio.

Los corchetes, [] indicarán que lo que se encuentra en su interior es opcional.

Lo que se encuentre entre los símbolos < >, indica que debe de sustituirse necesariamente, incluyendo los mismos símbolos, por algún valor del tipo especificado

El símbolo * se asocia a un tipo, e indica que la descripción puede reemplazarse por cero o mas ocurrencias del tipo especificado. En general se presenta de la forma <tipo>*. Así, <entero>* indica que debe sustituirse por cero o más valores enteros.

El símbolo + se asocia a un tipo, e indica que la descripción puede reemplazarse por uno o mas ocurrencias del tipo especificado. En general se presenta de la forma <tipo>+. Así <entero>+ indica que debe sustituirse por uno o mas valores enteros.

La barra vertical | indica que debe hacerse una elección entre los elementos que se encuentran separador por la barra. En general se presenta de la forma opción-1 | opción-2 | ... | opción-n. Así, se puede elegir entre la opción1 o la opción2 o la opción que se desee.

El constructor `deftemplate` crea una plantilla o patrón que se usa para acceder, por su nombre, a los campos (*slots*) de un hecho no ordenado. Este constructor es análogo a la definición de una estructura del lenguaje C. La sintaxis para este constructor es el siguiente:

Sintaxis: (`deftemplate` <nombre> [<comentario>] [<definición-slot>*)

donde: <definición-slot> = (slot <nombre-slot>) | (multislot <nombre-slot>)

ejemplo:

```
(deftemplate auto "Experto_Auto"
  (slot arranca)
  (slot falla)
  (slot gasolina)
  (multislot compostura)
)
```

La plantilla creada para éste ejemplo se llama "auto" y contiene cuatro slots, los primeros tres admitirán un solo valor o una sola cadena de caracteres, en cambio; el cuarto slot es de tipo múltiple y admitirá mas de una cadena de caracteres.

Los hechos no ordenados se distinguen de los ordenados mediante el primer campo. El primer campo de cualquier hecho debe ser un símbolo, pero si éste coincide con el nombre de un constructor `deftemplate`, entonces se considera no ordenado. En el constructor `deftemplate` podemos restringir el uso de los campos de los slots que forman dicha plantilla.

El constructor `defacts` permite establecer un conocimiento inicial o "a priori", mediante la especificación de una lista de hechos. Estos hechos no se pierden al ejecutar el comando `reset`. Es decir, cuando se limpia el ambiente de CLIPS, cada hecho especificado dentro de un constructor `defacts` se añade en la lista de hechos (`fact-list`).

2.5.3.3 ATRIBUTOS DE LOS SLOT (CAMPOS)

Se puede restringir su tipo, valor, rango numérico y la cardinalidad (el número mínimo y máximo de valores para un *slot*); se pueden especificar valores por defecto. Todas estas características ayudan en el desarrollo y mantenimiento de un Sistema Experto proporcionando una comprobación de restricciones.

Atributo *type* (`type` <especificación-tipo>)

Define el tipo de datos que puede tener el *slot*. Los tipos válidos son: SYMBOL, STRING, LEXEME, INTEGER, FLOAT, NUMBER. Si se especifica ?VARIABLE, significa que el *slot* puede tomar cualquier tipo de dato. Si se especifican uno o más tipos válidos, el tipo del *slot* queda restringido a uno de los tipos especificados. LEXEME equivale a especificar SYMBOL y STRING conjuntamente, y NUMBER equivale a INTEGER y FLOAT.

Atributo *allowed* (`allowed-values` <valores>)

TESIS CON
FALLA DE CUBIEN

En esta propiedad se pueden definir una lista de valores permitidos para este atributo. Existen siete atributos de esta clase: *allowed-symbols*, *allowed-strings*, *allowed-lexemes*, *allowed-integers*, *allowed-floats*, *allowed-numbers* y *allowed-values*. Cada uno de estos atributos debería ser seguido por *?VARIABLE* (que indica que cualquier valor del tipo especificado es legal) ó por una lista de valores del tipo que sigue al prefijo *allowed*. El atributo por defecto es: (*allowed-values ?VARIABLE*). Debemos de hacer notar que estos atributos *allowed* no restringen el tipo del *slot*.

Atributo *range* (*range* <límite-inferior> <límite-superior>)

Permite restringir los valores legales de un tipo numérico a un rango determinado. Tanto el límite inferior como el límite superior pueden ser un valor numérico o *?VARIABLE*.

Atributo *cardinality* (*cardinality* <límite-inferior> <límite-superior>)

Permite especificar el número mínimo y máximo que un *slot* puede contener. Ambos límites pueden ser un entero positivo ó *?VARIABLE*, que indica que no hay número mínimo o máximo de valores que el *slot* puede contener (dependiendo si se especifica como límite inferior, como superior, o en ambos). Por defecto se supone *?VARIABLE* para ambos límites.

Atributo *default* (*default* <especificación>)

Permite especificar un valor por defecto para un *slot* cuando no se añade o especifica explícitamente (por ejemplo, en un comando *assert*). La <especificación> puede ser: *?DERIVE*, *?NONE* ó una expresión simple (si se trata de un *slot* simple), ó cero o más expresiones (si se trata de un *slot* multicampo).

Si se especifica *?DERIVE*, entonces se deriva un valor para el *slot* que satisfaga todos los atributos del *slot*. Si en un *slot* no se especifica nada acerca de *default*, se supondrá (*default ?DERIVE*). En caso de un *slot* simple, esto significa que se elige un valor que satisfaga el tipo, rango y los valores permitidos para el *slot* (según indique el atributo *allowed*). En el caso de un *slot* multicampo, el valor por defecto será una lista de valores idénticos que es la cardinalidad mínima permitida para el *slot* (cero por defecto). Si aparecen uno o más valores en el atributo *default* de un *slot* multicampo, entonces cada valor satisfará el tipo, rango y los atributos *allowed* del *slot*.

Si se indica *?NONE*, entonces se debe dar un valor obligatoriamente al *slot* cuando se asevere el hecho. Es decir, no hay valor por defecto. Si no se le da un valor, se producirá un error. Si se utiliza una o más expresiones en el atributo *default*, entonces se evalúan las expresiones y el valor resultante se almacena en el *slot* siempre que no se especifique un valor para ese *slot* en un comando *assert*. En el caso de un *slot* simple, la especificación del atributo *default* debe ser exactamente una sola expresión. Si no se especifica expresión alguna cuando se trata de un *slot* multicampo, se usará un multicampo de longitud cero para el valor por defecto. En caso contrario, el *slot* multicampo contendrá cada uno de los valores devueltos por las respectivas expresiones.

```
(deftemplate auto
  (slot arranca (type SYMBOL)
    (allowed-values si no nose)
    (default nose))
  (slot falla (type SYMBOL)
    (allowed-values si no nose)
    (default nose))
  (slot gasolina (type SYMBOL)
    (allowed-values si no nose)
    (default nose))
  (multislot compostura)
)
```

Ahora en la plantilla, los primeros tres slots admitirán sólo símbolos (o cadena de caracteres) para satisfacerse y tienen definidos los valores posibles "si", "no" y "nose", esto quiere decir que CLIPS sólo admitirá únicamente estos valores para satisfacer la plantilla. El cuarto slot es de valor múltiple y puede admitir cualquier valor.

2.5.3.4 DEFINICIÓN DE HECHOS INICIALES

El constructor `deffacts` permite especificar un conjunto de hechos como conocimiento inicial.

Sintaxis: `(deffacts <nombre-colección-hechos> [<comentario>] <patrón-RHS>*)`
`(deffacts inicia (auto))`

El constructor `deffacts` no asevera los hechos dentro de la Memoria de Trabajo, simplemente los define. Los hechos de las sentencias `deffacts` son añadidas a la Memoria de Trabajo utilizando el comando `reset`. El comando `reset` elimina todos los hechos que hubiera en la lista de hechos actual, y a continuación añade los hechos correspondientes a sentencias `deffacts`.

Sintaxis: `(reset)`

Un programa CLIPS puede ser ejecutado mediante el comando `run`. Pero puesto que las reglas requieren de hechos para ejecutarse, el comando `reset` es el método clave para iniciar o reiniciar un Sistema Experto en CLIPS. Así, el comando `reset` provoca la activación de algunas reglas, y el comando `run` inicia la ejecución del programa.

2.5.3.5 VISUALIZACIÓN DE HECHOS INICIALES

Para mostrar todos los hechos ordenados definidos con el constructor `deffacts` se utiliza el comando `ppdeffacts`:

Sintaxis: `(ppdeffacts [<nombre-de-la-definición>])`

El comando `ppdeffacts` muestra exactamente la información que se introdujo con el constructor `deffacts` pero de la siguiente manera:

`(deffacts MODULO::<nombre-de-la-definición> [<comentario>]
 <hechos>*)`

donde MODULO indica el módulo en el cual el constructor se ha colocado. Los módulos son un mecanismo que permite particionar el conocimiento. Por defecto, CLIPS sólo considera un módulo, llamado MAIN, salvo que el usuario especifique la existencia de otros módulos.

CLIPS permite el control de la ejecución y desarrollo modular de Bases de Conocimiento con la construcción `defmodule`. La sintaxis de este constructor es el siguiente:

Sintaxis: `(defmodule <nombre-módulo> [<comentario>]
 <especificación-acceso>*)`

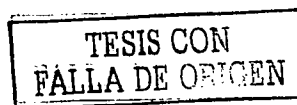
donde:

`<especificación-acceso> = (export <ítem>) | (import <nombre-módulo> <ítem>)
 <ítem> = ?ALL | ?NONE | <construcción> ?ALL | <construcción> ?NONE |
 <construcción> <nombre-construcción>+
 <construcción> = deftemplate | defclass | defglobal | deffunction | defgeneric`

Una construcción `defmodule` no puede ser ni redefinida ni borrada una vez que se define (con la excepción del módulo MAIN que puede ser redefinido una vez). La única forma de eliminar un módulo es con el comando `clear`. Al principio de una sesión CLIPS o después de un comando `clear`.

Para mostrar todos los nombres de las listas de hechos almacenados se utiliza el comando `list-deffacts` cuya sintaxis es la siguiente:

`(list-deffacts [<nombre-del-modulo>])`



- Si `<nombre-del-modulo>` no se especifica, se muestran todos los hechos del módulo de conocimiento actual. Si se especifica, se muestran los hechos que se definieron para ese módulo.
- Si se utiliza el comodín `*`, se mostraran todos los hechos definidos en todos los módulos.

Para mostrar todos los hechos no ordenados definidos con el constructor `defftemplate` se utiliza el comando **ppdefftemplate**:

Sintaxis: `(ppdefftemplate [<nombre-de-la-plantilla>])`

Para mostrar todos los nombres de las listas de hechos almacenados se utiliza el comando `list-defftemplate` cuya sintaxis es la siguiente:

`(list-defftemplate [<nombre-del-modulo>])`

2.5.3.6 BORRADO DE HECHOS INICIALES

Para borrar hechos definidos previamente se utiliza el destructor `undeffacts`.

Sintaxis: `(undeffacts <nombre-de-la-definición>)`

Utilizando el destructor no se volverán a afirmar los hechos definidos para `<nombre-de-la-definición>`. También se puede utilizar el comodín `*` para eliminar todas las definiciones existentes.

2.5.4 OBJETOS

Un objeto en CLIPS puede ser cualquier valor de un tipo primitivo de datos (un entero, un *string*, un símbolo, una dirección externa, etc.) o una instancia de una clase definida por el usuario. Los objetos son instancias de clases, y una clase es un modelo que recoge las propiedades de objetos.

Éstas dos formas de definir un objeto en CLIPS (tipo primitivo o instancia de clases) difieren en la forma en que se referencian, en cómo se crean y se borran, y en cómo se especifican sus propiedades.

La diferencia entre un hecho y un objeto es que el primero no tiene herencia y el segundo si acepta la herencia. La **herencia** permite definir las propiedades y conducta de una clase en función de otras clases.

2.5.5 LIMPIEZA DE LA MEMORIA DE TRABAJO

Existen dos comandos para limpiar la memoria de trabajo de CLIPS: `reset` y `clear`.

reset. Éste comando elimina todos los hechos que hubiera en la lista de hechos actual, y añade nuevos hechos correspondientes a sentencias `deffacts` que existieran en el código de CLIPS, incluyendo el hecho inicial `initial-fact`. Éste hecho inicial estará vacío y es útil para activar las reglas implícitamente.

Clear. Éste comando borra todas las afirmaciones y todos los modelos generados hasta ese momento (hechos, reglas, plantillas, módulos, etc.)

- Conviene usar el comando **reset** cuando se desea mantener la misma Base de Conocimiento (plantillas y reglas), y probarla con distintos hechos

- Conviene usar el comando **reset** antes de ejecutar un programa CLIPS si el programa ya ha sido ejecutado anteriormente
- Conviene usar el comando **clear** cuando se desea cambiar completamente de Base de Conocimiento

2.6 BASE DE CONOCIMIENTO (BC)

La Base de Conocimiento es el segundo elemento básico en la construcción de Sistemas Expertos y como se mencionó, contiene todos los hechos, las reglas y los procedimientos del dominio de aplicación que son importantes para la solución del problema.

2.6.1 REGLAS

Una regla consta de un antecedente y de un consecuente. El antecedente también es denominado parte "si" o parte izquierda de la regla (*Left-Hand Side*) y el consecuente también es denominado parte "entonces" o parte derecha de la regla (*Right-Hand Side*).

El antecedente está formado por un conjunto de condiciones o de Elementos Condicionales (EC) que deben satisfacerse para que la regla sea aplicable. (Existe un *and* implícito entre todas las condiciones en la parte izquierda de la regla). Para que una regla sea disparada, los elementos condicionales que forman dicha regla deben de ser cumplidos y estos se basan en la existencia o en la no existencia de los hechos específicos para tales Elementos Condicionales en la Memoria de Trabajo.

El consecuente de una regla es un conjunto de acciones a ser ejecutadas cuando la regla es aplicable. Estas acciones se ejecutan cuando el Motor de Inferencia de CLIPS es instruido para que comience la ejecución de las reglas aplicables.

Una regla en CLIPS es también considerada una entidad independiente y no es posible el paso de datos entre dos reglas. Para crear una regla se usa el constructor **defrule** y su sintaxis es la siguiente:

```
Sintaxis: (defrule <nombre-regla> [<comentario>]
          [<declaración>]<elemento-condición>* ; Parte izquierda (LHS)
          =>
          <acción>*);                               Parte derecha (RHS) de la regla
```

```
(defrule auto_arranca
  (auto (arranca si)
   =>
   (assert (compostura ninguna)))
```

La regla se llama "auto_arranca" y la parte izquierda de la regla está formada únicamente por el Elemento Condicional "(auto (arranca si))" (cuando en la plantilla "auto", el slot arranca tenga el valor "si") entonces la parte derecha de la regla tiene la aseveración de "(compostura ninguna)".

Si se introduce en la base de reglas una nueva regla con el mismo nombre que el de una existente, la nueva regla reemplazará a la antigua. El comentario, que debe ser una cadena (*string*), se usa normalmente para describir el propósito de la regla o cualquier otra información que el programador desee. Estos comentarios pueden ser visualizados junto con el resto de la regla usando el comando *ppdefrule*:

```
Sintaxis: (ppdefrule <nombre-regla>)
```

Si una regla no tiene parte izquierda, es decir, no tiene elementos condicionales, entonces el hecho (initial-fact) actuará como el elemento condicional para ese tipo de reglas, y la regla se activará cada vez que se ejecute un comando **reset**.

Elemento condicional (EC): especifica restricciones sobre elementos de las listas de hechos e instancias: sólo se satisface si existe una entidad (hecho o instancia) que cumple las restricciones expresadas.

Una regla se ejecuta cuando:

1. Todos sus elementos condición son satisfechos por la lista de hechos y/o la lista de instancias.
2. El Motor de Inferencia la selecciona.

2.6.1.1 ELEMENTOS CONDICIONALES (EC)

Entre los elementos condicionales podemos mencionar los siguientes: Pattern, test, and, not, exists, forall, logical.

Pattern

Es una colección de restricciones de campos, comodines y variables que se usan para limitar el conjunto de hechos o instancias que satisfacen el *pattern*.

test (test <llamada-a-función>)

Se usa para evaluar expresiones en la parte izquierda de una regla, interviniendo en el proceso de identidad de patrones (*pattern-matching*). El Elemento Condicional *test* se satisface si la llamada a la función que aparezca dentro de él devuelve cualquier valor distinto de FALSE; en caso contrario, éste Elemento Condicional no se satisface.

and (and <elemento-condicional>+)

CLIPS supone que todas las reglas tienen un *and* implícito que rodea todos los Elementos Condicionales de la parte izquierda de la regla. Esto significa que todos los Elementos Condicionales que aparezcan en la parte izquierda de la regla deben satisfacerse para que la regla se active. Como se verá en la siguiente sección; cuando una regla se activa no significa que se ejecute pero si es transferida a "La Agenda de CLIPS" y ahí es donde se ejecuta o no dicha regla.

Este Elemento Condicional se usa para combinar una expresión de la parte izquierda de la regla con otros elementos condicionales, es decir, una expresión de la parte izquierda de la regla será válida hasta que todos los Elementos Condicionales se satisfagan

or (or <elemento-condicional>+)

Este elemento condicional también se usa para combinar una expresión de la parte izquierda de la regla con otros elementos condicionales. El elemento condicional OR se satisface cuando al menos uno de los componentes que aparece se satisface.

not (not <elemento-condicional>)

Un elemento condicional negativo se satisface si no existe ninguna entidad que cumpla las restricciones expresadas.

exists (exists <elemento-condicional>+)

Este Elemento Condicional permite que se produzca la identidad de patrones (*pattern matching*) cuando al menos exista un hecho que satisfaga la regla, sin tener en cuenta el número total de hechos que pudiesen *interactuar entre si*. Esto permite una sola activación para una regla con la que la coincidencia (*matching*) se hace para un conjunto de hechos específicos de esa regla. El

Elemento Condicional *exists* está implementado mediante una combinación de *and*'s y *not*'s. Los Elementos Condicionales dentro de un *exists* se incluyen dentro de un *and* y luego dentro de dos *not*.

forall (forall <primer-EC> <resto-de-ECs>+)

Permite la coincidencia (*matching*) basado en un conjunto de Elementos Condicionales que son satisfechos por cada ocurrencia de otro Elemento Condicional. Su funcionamiento es el contrario que el de *exists*. Para que el Elemento Condicional *forall* se satisfaga, todo lo que se interactúe con el primer Elemento Condicional debe tener hechos que hagan el *matching* con todos los demás Elementos Condicionales que aparecen a continuación del primer Elemento Condicional. Este Elemento Condicional también puede expresarse en función de una combinación de *and*'s y *not*'s:

logical (logical <elemento-condicional>+)

Este Elemento Condicional proporciona capacidad de mantenimiento de verdad para entidades (hechos o instancias) creados por una regla que usa el Elemento Condicional *logical*. Este Elemento Condicional permite establecer una dependencia lógica entre entidades creadas en las partes derechas de las reglas (o como resultado de las acciones realizadas en la parte derecha de la regla) y las entidades que compararon los patrones contenidos en el Elemento Condicional *logical* de la parte izquierda de las reglas. Las entidades que se acoplaron con los patrones contenidos en el Elemento Condicional *logical* proporcionan un soporte lógico a los hechos e instancias que se crearon en la parte derecha de las reglas.

2.6.1.2 PROPIEDADES DE LAS REGLAS: SALIENCE Y AUTO-FOCUS

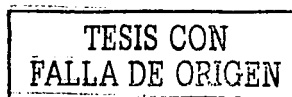
Que una regla se active no quiere decir que las acciones de su parte derecha se vayan a ejecutar. La activación consiste en el paso de la regla a una especie de Memoria de Trabajo denominada Agenda, donde queda almacenada para ser ejecutada. Sin embargo, esto no presupone que se vaya a ejecutar, ya que puede activarse otra regla con mayor prioridad que ésta y que elimine las condiciones que hacían que la primera regla estuviese activada.

De todas las reglas que hay en la agenda, CLIPS ejecuta las acciones de la parte derecha de la regla que tenga mayor prioridad. Mientras haya reglas activadas en la agenda, CLIPS irá ejecutando en cada momento la de mayor prioridad hasta que no queden reglas activas. Puede suceder también que una regla que está en la Agenda desaparezca debido a que cuando CLIPS ejecutó una regla, la parte derecha de ésta regla creó y/o cambió algunos elementos condicionales existentes y por consecuencia, las reglas que desaparecen son porque dejaron de cumplir con la coincidencia de patrones (*pattern matching*). De igual manera, al ejecutarse una regla puede suceder que se añadan nuevas reglas a la agenda, es decir; como consecuencia de los elementos condicionales generados por una regla recién ejecutada, pueden satisfacer la parte izquierda de otra regla y por consiguiente, es añadida a la Agenda.

La prioridad de una regla se puede establecer manualmente mediante la propiedad *saliencie* que se puede asociar a una regla cuando se define. El valor de la propiedad *saliencie* puede variar entre los valores de -10000 y 10000, siendo 10000 el de mayor prioridad y valiendo cero por defecto, es decir, cuando no se especifique ningún valor explícitamente, el *saliencie* de la regla será cero. Cuando una nueva regla entra a la agenda, ésta se acomoda arriba de otras reglas con *saliencie* superior o igual a la nueva regla.

Esta propiedad se declara en la parte izquierda de una regla utilizando la palabra clave *declare*. Una regla sólo puede tener una sentencia *declare* y debe aparecer antes del primer elemento condicional.

Sintaxis: (declare (*saliencie* <valor>))



```
(defrule input_arranca
  (declare (salience 9))
  (auto (arranca si))
  =>
  (assert (compostura ninguna))
)
```

Ahora la regla tiene una prioridad de nueve

La propiedad **auto-focus** ejecuta automáticamente un comando **focus** cuando la regla se activa, convirtiendo en foco actual al módulo al que pertenezca dicha regla. Este efecto se consigue con el atributo *auto-focus*, que se especifica en la sentencia *declare* junto con el atributo *salience*. Para activar el efecto del *auto-focus*, se establece a TRUE. Por defecto, este atributo tiene el valor FALSE.

Sintaxis: (auto-focus TRUE | FALSE)

La propiedad *auto-focus* es particularmente útil para reglas que se encargan de detectar violaciones de restricciones. Puesto que el módulo que contiene a dicha regla se convierte inmediatamente en el foco actual, es posible continuar trabajando en consecuencia cuando ocurre la violación, en lugar de tener un fase específica para la detección de las excepciones. Esta propiedad permite la ejecución automática de un comando **focus** cada vez que la regla se activa.

Nota: El comando focus se definirá mas adelante dentro del Motor de Inferencia.

2.6.2 COMANDOS PREDEFINIDOS POR CLIPS

Anteriormente se había dado una descripción de lo que son los comandos: Las acciones o comandos (para distinguirlas de las funciones) son operaciones que no devuelven un valor, pero tienen un efecto lateral bastante útil dentro de la programación en CLIPS. Ahora sólo se mencionarán dichos comandos pero se sugiere ir a la referencia bibliográfica para un mayor entendimiento.

- Acciones que Modifican la Memoria de Trabajo. (Assert, duplicate, retract, modify).
- Acciones Procedurales. (Ligaduras, if-then-else, while, loop-for-count, progn\$, return, break, switch).
- Comandos sobre la Agenda. (agenda, clear, halt, run).

2.6.3 FUNCIONES PREDEFINIDAS POR CLIPS

- *Nombres lógicos*. Permiten referenciar un dispositivo de E/S sin tener que entender los detalles de la implementación de la referencia. Muchas funciones en CLIPS usan nombres lógicos. Un nombre lógico puede ser un símbolo, un número ó un *string*. Entre los nombres lógicos tenemos los siguientes: Stdin, Stdout, wclips, wdialog, wdisplay, werror, wwarning y wtrace
- *Funciones de entrada y salida*. Permiten la interacción entre CLIPS y el usuario. Éstas funciones son: open, close, printout, bind, read, readline, format, rename y remove.
- *Funciones matemáticas*, de conversión. Entre las funciones aritméticas tenemos las siguientes: abs, div, flota, integer, max, min, exp, log, log10, mod, pi, round, sqrt, random, sin, cos, tan, cot, sec, tanh, sinh, entre otras. Entre las funciones de conversión podemos mencionar las siguientes: deg-grad, deg-rad, grad-deg, rad-deg, lowcase, upcase, str-cat, sym-cat, str-compare, str-index, entre otras.
- *Manipulación de valores multicampo*. Entre ellas tenemos: create\$, delete\$, explode\$, first\$, implode\$, insert\$, length\$, member\$, nth\$, replace\$, rest\$, entre otras.

- *Funciones de entorno.* Tenemos: `load`, `save`, `exit`, `batch`, `dribble-on` y `dribble-off`.

2.6.4 FUNCIONES DEFINIDAS POR EL USUARIO

Para definir funciones se utiliza el constructor `deffunction`, la sintaxis de este constructor es el siguiente:

Sintaxis: (`deffunction` <nombre> [<comentario>]
(<parámetro>* [<parámetro-comodín>])
<acción>*)

donde:

<parámetro>* es una variable unicampo y <parámetro-comodín> es una variable multicampo.

Éste constructor se compone de cinco elementos:

- 1) un nombre (que debe ser un símbolo)
- 2) un comentario opcional
- 3) una lista de cero o más parámetros requeridos (variables simples)
- 4) un parámetro comodín opcional que sirve para manejar un número variable de argumentos (variable multicampo)
- 5) una secuencia de acciones o expresiones que serán ejecutadas en orden cuando se llame a la función.

El valor devuelto por la función es la última acción o expresión evaluada dentro de la función. Si una *deffunction* no tiene acciones, devolverá el símbolo `FALSE`. Si se produce algún error mientras se ejecuta la función, cualquier otra acción de la función aún no ejecutada se abortará, y la función devolverá el símbolo `FALSE`. Si desea que una función llame a otra función, la segunda tendrá que haberse declarado y definido previamente. La única excepción es que la función sea recursiva.

Deffunction: Permite construir funciones definidas dentro del ambiente de CLIPS usando funciones predefinidas por CLIPS. Las llamadas a las funciones en CLIPS se hacen en notación prefija. Es decir, en primer lugar se escribe el nombre de la función y a continuación los argumentos de dicha función separados por uno o más espacios, todo ello encerrado entre paréntesis.

defgeneric y **defmethod** Permiten definir funciones genéricas. Éstos comandos permiten diferente tipo de código dependiendo de los argumentos pasados a la función genérica.

2.7 MOTOR DE INFERENCIA

El Motor de Inferencia es el tercer elemento que proporciona CLIPS para desarrollar un sistema experto. El ciclo básico de ésta etapa es el siguiente:

1. Si se ha alcanzado el número de ciclo de desencadenamiento expresados ó no hay foco actual, ENTONCES parar. SI NO, seleccionar para ejecutar la regla tope de la agenda del módulo actual. SI NO hay reglas en dicha agenda, ENTONCES eliminar el foco actual de la pila de focos y actualizar al siguiente de la pila. Si la pila de focos está vacía, entonces PARAR, SI NO ejecutar el paso 1 de nuevo.

2. Se ejecutan las acciones de la parte derecha de la regla seleccionada. Incrementar el número de reglas desencadenadas. Si se usa *return*, entonces eliminar el foco actual de la pila de focos.

3. Como resultado del paso anterior, se activan o desactivan nuevas reglas. Las reglas activadas se sitúan en la agenda del módulo en el que están definidas. La situación concreta depende de la prioridad de la regla (*salience*) y de la estrategia de resolución de conflictos actual. Las reglas desactivadas salen de la agenda.

4. Si se utiliza prioridad dinámica (*dynamic salience*), entonces se reevalúan los valores de prioridad de las reglas de la agenda. Ir al paso 1.

2.7.1 PATTERN MATCHING

No existe una definición oficial de lo que es el pattern matching, sin embargo es el principal motor para que el sistema experto pueda funcionar. El pattern matching es un mecanismo único de acceso a los elementos de la base de hechos y a la memoria de trabajo. Éste mecanismo toma los elementos condicionales de una regla y los compara con los que existen en la memoria de trabajo y en la base de hechos para ver con cual de ellos hace pareja y ver si dichos elementos condicionales se vuelven verdaderos.

El modelo de trabajo del pattern matching sobre la parte izquierda de la regla con los datos almacenados en la base de hechos y la memoria de trabajo es el siguiente:

- Las constantes de la parte izquierda de la regla son solo iguales a ellas dentro de la base de hechos y/o de la memoria de trabajo.
- Los comodines de la parte izquierda de la regla absorben los datos encontrados
- Las variables libres de la parte izquierda de la regla se emparejan con los datos encontrados
- Las variables emparejadas compararán su valor con los datos encontrados

2.8 ENTORNO DE CLIPS CON WINDOWS

El cuarto elemento de un sistema experto son los dispositivos de entrada y salida. La versión 6.2 utilizada para este proyecto tiene una interfase visual de ventanas, donde podemos interactuar directamente con los elementos de programación, estar en contacto directo con la Memoria de Trabajo (instancias, focos, variables globales), la Base de Hechos (ventana de hechos) y el Motor de Inferencia (Agenda).

A pesar de tener una interfaz dinámica con Windows, CLIPS carece de controles que permitan al usuario manipular de forma cómoda el código de programación del Sistema Experto así como de controles a la hora de ejecutar el programa. CLIPS se sigue basando en los controles básicos del DOS pero resulta muy útil a la hora de depurar y comprobar el programa. Para tener una mejor sincronía entre CLIPS y el usuario, fueron diseñadas librerías dinámicas (DLL's) para poder interactuar con otros lenguajes de programación (C, Visual C++, Visual Basic, HTML, entre otros) y de esta forma el manejo con el usuario sea mas eficiente. La siguiente figura muestra la interfase que tiene CLIPS con windows.

TESIS CON
FALLA DE ORIGEN

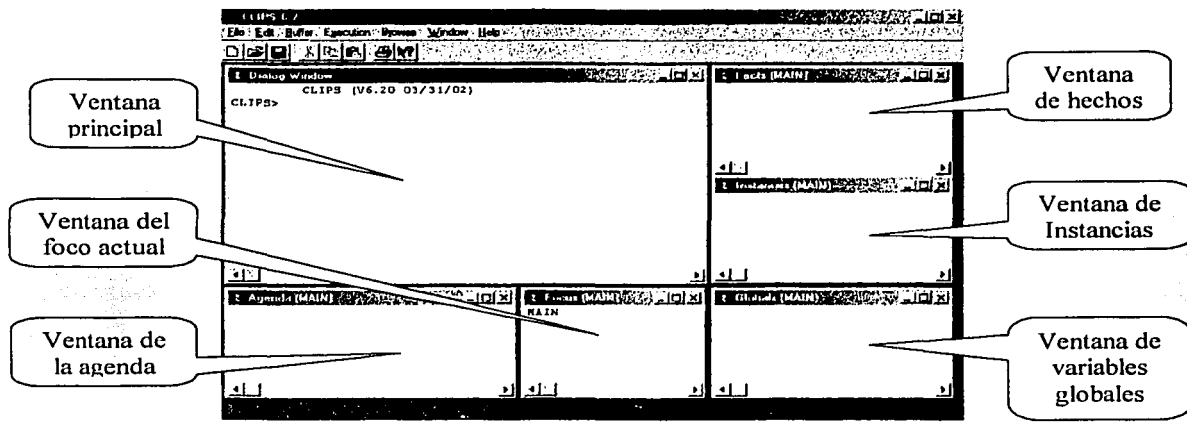


Figura 2.8.1 Interfase de CLIPS con Windows

```

CLIPS [6]
File Edit Buffer Execution Browse Window Help
CLIPS [6]
CLIPS [V6.20 03/31/02]
CLIPS> (load "D:/Teoria/VDA_Clips/pinino0.clp")
Defining defglobal: pregunta
Defining defglobal: solucion
Defining deffunction: pregunta
Defining deffunction: a1-no-p
Defining deftemplate: auto
Defining defmacro: inicio
Defining defrule: input_agenda +3
Defining defrule: input_soln +3
Defining defrule: input_gasolina +3
Defining defrule: input_botella +3
Defining defrule: input_solucion +3
Defining defrule: registra_no_solo +3
Defining defrule: registra_no_agenda_tiempo +3
Defining defrule: registra_no_llavear_al_taller +3
Defining defrule: registra_no_poner_gas +3
Defining defrule: registra_no_cargar_botella +3
Defining defrule: registra_no_llavear_al_taller +3
Defining defrule: registra_no_limpvar_al_taller +3
Defining defrule: imprime_solucion +3
TRUE
CLIPS> (reset)
CLIPS> (run 1)
El coche entra (a1/no)?
0
input_agenda: 2-1
1
2?preguntas = ""
?solucion = ""

```

Figura 2.8.2

La figura 3.8.2 muestra el ejemplo automotriz en ejecución, donde el usuario tiene que introducir las respuestas directamente desde el teclado y esto tiene la desventaja de que el usuario tiene que teclear correctamente las respuestas ya que de lo contrario, no se podrá efectuar la coincidencia de patrones (pattern matching) y CLIPS no podrá llegar a una respuesta final porque seguirá esperando dicha coincidencia para ejecutar las reglas que aún tiene en la agenda.

Nota: Recomendamos ampliamente consultar la referencia bibliográfica para entender mejor los conceptos para hacer la ejecución del programa en CLIPS.

Gracias a que CLIPS fue desarrollado para ser integrado con otros lenguajes de programación, podemos hacer uso de otro lenguaje de programación que nos facilite el diseño y la construcción de una interfase amigable para el usuario. Visual Basic proporciona dichos elementos y como se verá en las siguientes secciones, CLIPS en conjunto con Visual Basic nos permiten crear Sistemas Expertos amigables al usuario.

2.9 CLIPS ACTIVEX OCX

CLIPS ActiveX Control.ocx es una librería que permite comunicarse con otros lenguajes de programación como lo es Visual Basic y poder tener una mayor profundidad y sencillez frente al usuario final. Esto es: que si CLIPS proporciona toda la complejidad que un sistema experto necesita y si Visual Basic proporciona la flexibilidad de tener una interfase bastante agradable para el usuario, con ésta librería podemos unir los dos lenguajes y presentarle al usuario un programa amigable, funcional, flexible y de fácil uso.

Ésta librería es un archivo que contiene funciones que pueden ser llamadas desde otras aplicaciones o desde otras librerías. La librería CLIPS DLL exporta las funciones internas de CLIPS y regresa el resultado de dichas funciones al programa que lo mandó llamar. Visual Basic y la librería CLIPS OCX permite la comunicación directa con CLIPS y su programación es sencilla entre ambos. La versión utilizada para este trabajo es la 1.8.1.2. (Figura 3.8.3).

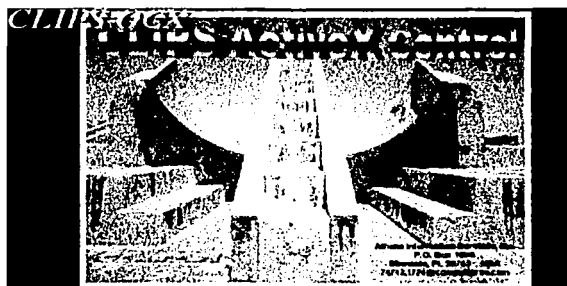


Figura 2.9.1 CLIPS OCX para Windows

2.10 CLIPS Y VISUAL BASIC

Visual Basic tiene la capacidad de agregar cualquier librería dinámica a su entorno de trabajo para interactuar con otros programas. La librería CLIPS DLL provee a Visual Basic la capacidad de programar funciones, objetos, eventos o simplemente variables para la comunicación con CLIPS. Algunos de las funciones con las cuales se interactúa directamente con CLIPS son las siguientes:

- Funciones de entrada a Visual Basic. Permiten la introducción de los datos generados por CLIPS.
- Funciones de salida de Visual Basic. Permiten introducir los datos a CLIPS generados por Visual Basic.
- Funciones combinadas. Permiten la ejecución de instrucciones en CLIPS a través de Visual Basic.

De entre las anteriores podemos destacar las siguientes:

TESIS CON
FALLA DE ORIGEN

CLIPS.Clear	Limpia y remueve todo en el entorno de CLIPS.
CLIPS.Load("nombre del archivo")	Carga el archivo a ejecutarse dentro del entorno de CLIPS.
CLIPS.Reset	Resetea el espacio de trabajo de CLIPS.
CLIPS.Run(-1)	Ejecuta el archivo cargado dentro del entorno de CLIPS .
CLIPS.GetDefglobalString("variable")	Obtiene una variable global de CLIPS al entorno de Visual Basic.
CLIPS.AssertString(hechos a insertar)	Asevera un hecho dentro del entorno de CLIPS.

Para entender mejor estas funciones, consideramos que lo mas conveniente es mostrarlas en un ejemplo y explicar su funcionamiento sobre el mismo. El código de programación que a continuación se describe es parte del ejemplo Automotriz hecho en CLIPS pero ahora con interfase en Visual Basic.

```
Dim Fact, fact2, Pregunta, solucion As String
Dim retval, retval2 As Boolean
```

```
Private Sub Command1_Click()
```

```
CLIPS.Clear
CLIPS.Load("c:\experto_automotriz.cip")
CLIPS.Reset
retval2 = CLIPS.Run(-1)
Pregunta = CLIPS.GetDefglobalString("preguntas")
solucion = CLIPS.GetDefglobalString("solucion")
Test3_Form2.Text1.Text = Pregunta
```

```
If List1.Selected(0) Then
Fact = "(auto (arranca si))"
retval = CLIPS.AssertString(Fact)
Else
fact2 = "(auto (arranca no))"
retval2 = CLIPS.AssertString(fact2)
End If
CLIPS.Run (-1)
```

```
Pregunta = CLIPS.GetDefglobalString("preguntas")
bandera = CLIPS.GetDefglobalString("bandera")
Text1.Text = Pregunta
End Sub
```

```
Private Sub Command2_Click()
If List1.Selected(0) Then
Fact = "(auto (arranca si)(falla si))"
retval = CLIPS.AssertString(Fact)
Else
fact2 = "(auto (arranca si)(falla no))"
retval2 = CLIPS.AssertString(fact2)
End If
CLIPS.Run (-1)
Pregunta = CLIPS.GetDefglobalString("preguntas")
solucion = CLIPS.GetDefglobalString("solucion")
End Sub
```

<p>TESIS CON FALLA DE ORIGEN</p>

Explicación del ejemplo

Al principio se definen las variables que se van a emplear. Enseguida se limpia el entorno de trabajo de CLIPS con la función "clear" para poder cargar el programa en CLIPS sin problemas. A continuación se carga el archivo de CLIPS con la función "load"; una vez cargado se resetea el entorno de CLIPS para inicializar los hechos con los que empezará a interactuar para activar las reglas.

Con la función "run" se procede a ejecutar el programa de CLIPS generando la activación y ejecución de algunas reglas, las cuales generarán valores a las variables globales definidas en CLIPS.

Estos valores globales que están en CLIPS tendrán que ser asignados a variables pero de Visual Basic. Para obtener esos valores se usa la función "GetDefGlobalString" y de esta manera Visual Basic puede trabajar con esos valores en su entorno de trabajo.

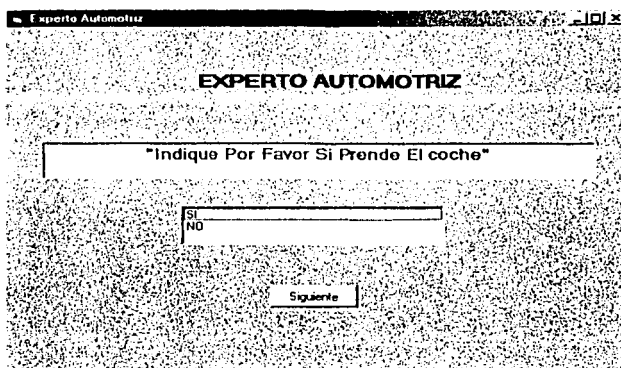
En Visual Basic se generó una interfase donde aparecen las preguntas que se le hacen al usuario y las opciones de respuesta para cada pregunta. El usuario podrá elegir una de las dos respuestas que aparecerán para cada pregunta.

Una vez que el usuario eligió la respuesta, Visual Basic tiene predefinido un hecho que será aseverado en CLIPS para que después de volver a ejecutar el entorno de CLIPS, genere nuevos valores que serán enviados a Visual Basic, y éste a su vez aseverará nuevos hechos de acuerdo a la respuesta seleccionada para la siguiente pregunta y así sucesivamente hasta que CLIPS llegue a la meta establecida.

Siguiendo con la explicación del código, después de que el usuario eligió la respuesta, se crea el hecho "(auto(arranca si))", o "(auto (arranca no))" según sea el caso; luego el hecho creado es aseverado con la función AssertString sobre la plantilla definida en CLIPS

De la misma forma, el resto del código trabajará de la misma forma hasta que CLIPS llegue a la meta.

Las siguientes figuras muestran el ejemplo Automotriz generado en CLIPS pero con interfase Visual Basic.



TESIS CON
FALLA DE ORIGEN

Figura 2.10.1 Representación del ejemplo "Experto Automotriz" con interfase Visual Basic

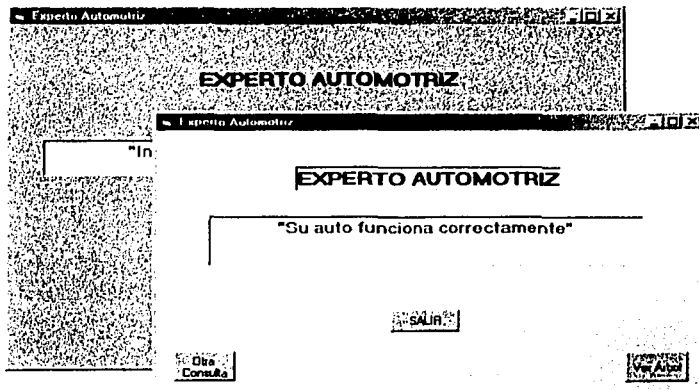


Figura 2.10.2 Representación del ejemplo "Experto Automotriz" con interfase Visual Basic

TESIS CON
FALLA DE ORIGEN

3. ENLACES MICROONDA

3.1 INTRODUCCIÓN

En este capítulo se verá un panorama general de lo que son las señales microonda, sus características y las consideraciones básicas y necesarias para realizar un enlace de microonda punto a punto.

3.2 RADIOCOMUNICACIÓN

La Radiocomunicación puede definirse como la telecomunicación realizada a través de las ondas radioeléctricas. La Unión Internacional de Telecomunicaciones (UIT), define las ondas radioeléctricas como las ondas electromagnéticas que se propagan por el espacio sin guía artificial y cuyo límite superior de frecuencia se fija convencionalmente en 300GHz.

La radiocomunicación que hace uso de elementos situados en el espacio, se denomina radiocomunicación espacial. Toda radiocomunicación distinta de la espacial y de la radioastronomía, se llama radiocomunicación terrenal o terrestre. La técnica de la radiocomunicación consiste en la superposición de la información que se desea transmitir en una onda electromagnética soporte, llamada portadora. La inserción de esa información constituye el proceso denominado modulación. La onda modulada se envía al medio de propagación a través de un dispositivo de acoplamiento con el medio denominado antena. El conjunto de equipos para el tratamiento de la información: moduladores, filtros, antenas, etc. constituye la estación transmisora (o abreviadamente, el transmisor). Cuando la onda transmitida alcanza el punto o puntos de destino, accede al sistema receptor por medio de una antena de recepción, que capta una fracción de la energía. El alcance útil o cobertura de una emisión radioeléctrica depende del tipo e intensidad de las perturbaciones.

Existen dos tipos fundamentales de transmisión inalámbrica:

Omnidireccionales: La antena transmisora emite en todas las direcciones espaciales o terrestres la información, mientras que la antena receptora recibe igualmente de toda dirección.

Direccionales: La energía emitida se concentra en un haz, para lo cual se requiere que la antena receptora y transmisora estén alineadas. Cuanto mayor sea la frecuencia de transmisión, es más factible confinar la energía en una dirección.

El espectro de frecuencias está dividido en bandas de la siguiente manera:

Símbolo	Frecuencia
VLF	3 - 30 KHz
LF	30 - 300 KHz
MF	300 - 3000 KHz
HF	3 - 30 MHz
VHF	30 - 300 MHz
UHF	300 - 3000 MHz
SHF	3 - 30 GHz
EHF	30 - 300 GHz

LEGIS CON
FALLA DE ORIGEN

Básicamente se emplean tres tipos de ondas del espectro electromagnético para comunicaciones:

Microondas: 3 MHz - 300 GHz. Muy direccionales. Pueden ser terrestres o por satélite.

Ondas de radio: 30 MHz - 1 GHz. Omnidireccionales.

Infrarrojos: *Direccionales*

La zona del espectro de las microondas está dividido de la siguiente manera:

Banda	Frecuencia
L	1 - 2 GHz
S	2 - 4 GHz
C	4 - 8 GHz
X	8 - 12 GHz
Ku	12 - 18 GHz
K	18 - 27 GHz
Ka	27 - 40 GHz

3.3 ¿QUÉ SON LAS MICROONDAS?

Se denomina así la porción del espectro electromagnético que cubre las frecuencias entre aproximadamente 3 Mhz y 300 Ghz, que corresponde a la longitud de onda en vacío entre 10 cm. y 1mm.

La propiedad fundamental que caracteriza a este rango de frecuencia es que el rango de ondas correspondientes es comparable con la dimensión física de los sistemas de laboratorio; debido a esta peculiaridad, las microondas exigen un tratamiento particular que no es extrapolable de ninguno de los métodos de trabajo utilizados en los márgenes de frecuencias con que limita. Estos dos límites lo constituyen la radiofrecuencia y el infrarrojo lejano. En radiofrecuencia son útiles los conceptos de circuitos con parámetros localizados, debido a que, en general, las longitudes de onda son mucho mayores que las longitudes de los dispositivos, pudiendo así, hablarse de autoinducciones, capacidades, resistencias, etc., debido que no es preciso tener en cuenta la propagación efectiva de la onda en dicho elemento; por el contrario, en las frecuencias superiores a las de microondas son aplicables los métodos de tipo ÓPTICO, debido a que las longitudes de onda comienzan a ser despreciables frente a las dimensiones de los dispositivos.

3.4 MICROONDAS TERRESTRES

Como se mencionaba anteriormente, se denominan microondas terrestres por no ocupar elementos espaciales para su transmisión o recepción. La antena típica de este tipo de microondas es parabólica y tiene unos tres metros de diámetro; el haz es muy estrecho por lo que las antenas receptora y emisora deben estar muy bien alineadas. A cuanto mayor altura se sitúen las antenas mayor es la facilidad para esquivar obstáculos

Una ventaja principal de las microondas terrestres es la **transmisión a larga distancia**, ya que requiere menos repetidores que otro medio de propagación (por ejemplo el cable coaxial), aunque por otra parte, las microondas terrestres necesitan que las antenas tanto del transmisor como del receptor estén alineadas. El uso de microondas es frecuente en aplicaciones de TV y voz, en **enlaces punto a punto sobre distancias cortas**, en circuitos cerrados de televisión y en interconexiones de redes locales y transmisiones entre edificios.

Las microondas cubren una parte importante del espectro radioeléctrico, de los 2 a los 40 GHz; el ancho de banda potencial y la velocidad de transmisión aumentan con la frecuencia, por lo que sus prestaciones son muy buenas y tienen múltiples aplicaciones como la transmisión de vídeo y de voz.

El problema fundamental de este tipo de comunicación es la atenuación, que dependerá de la longitud de onda que estemos utilizando, así como de las condiciones meteorológicas: por ejemplo a partir de los 10 MHz aumenta mucho la atenuación a causa de la lluvia. Además se dan problemas de interferencia entre unas y otras emisiones, por lo que es necesario regular las bandas.

3.5 ESTRUCTURA GENERAL DE UN ENLACE DE MICROONDAS

Todo sistema de comunicación parte de un modelo básico, el cual consiste en la transferencia de información (voz, datos, etc.) desde un lugar a otro. Para llevar esto a cabo, es necesario un *transmisor* que envíe la información a través de un *canal de transmisión*, y la cual es recibida por un *receptor* que sea capaz de detectar y entender el mensaje recibido.

Para las *microondas terrestres* el proceso de comunicación esta constituido por los mismos elementos. A continuación se muestra la estructura y los componentes que son necesarios para que se realice un enlace de microondas entre dos puntos considerando las características que éstas presentan y que se especificaron en temas anteriores. Sabemos que para dichas ondas, el enlace debe ser *punto a punto* concentrando la señal a transmitir en un haz unidireccional, como se observa en la *Figura 3.5.1*



Figura 3.5.1

Bajo esta perspectiva se contemplan los siguientes componentes:

* Terminales del enlace

- Equipo transmisor/receptor del lado A y B.
- Fuentes de poder o alimentación.
- Baterías, torres y accesorios menores.

* Antenas

- De alto desempeño
- Parabólicas.

* Alimentadores o Líneas de Transmisión

- Cable coaxial o guía de onda (elíptica o redonda: cobre o aluminio)
- Selección determinada por las pérdidas permitidas y por el rango de frecuencias utilizadas.

- * Alto de la antena
 - Dependiendo de la distancia a ser cubierta y el terreno intermedio.
 - Si se poseen varias líneas, se puede usar una para voz, otra para datos, video, etc.
- * Torres
 - Cableadas
 - Autosoportadas
 - Monopolares
- * Refugio (Shelters).
- * Equipo de presurización.

El enlace se lleva a cabo de la siguiente manera:

La antena utilizada generalmente en estos enlaces es la de tipo parabólico o de alto desempeño. La antena es fijada rígidamente, y transmite un haz estrecho que debe estar perfectamente enfocado hacia la antena receptora. Estas antenas de microondas se deben ubicar a una altura considerable sobre el nivel del suelo, con el fin de conseguir mayores separaciones posibles entre ellas y poder superar posibles obstáculos. Sin obstáculos intermedios la distancia máxima entre antenas es de aproximadamente 7.14 Km. Claro está que esta distancia se puede extender, si se aprovecha la característica de curvatura de la tierra, por medio de la cual las microondas se desvían o refractan en la atmósfera terrestre.

El haz lleva la información o el mensaje, el cual no es transmitido directamente, sino que se utilizan *códigos* entendibles por el equipo transmisor y el receptor. Éstos son el lenguaje utilizado para enviar el haz que es propiamente la onda portadora de esos códigos, en la cual intervienen los parámetros de frecuencia, periodo, fase y longitud de onda. Este proceso exige que la información sea codificada en la transmisión y decodificada en la recepción, con el objetivo de que la información que se quiere transmitir sea idéntica a la que se recibe. Si falla cualquiera de los elementos que intervienen (*transmisor, canal de transmisión o receptor*), se producen pérdidas de información.

Cuando la información es decodificada por el equipo receptor y se sabe que llegó a su destino, ésta se descarga de la antena receptora que puede o no estar conectada a una torre, mediante una línea de transmisión (*que puede ser un cable coaxial*); ya en pie de la torre se monta una ductería con el fin de proteger dicha línea hasta un refugio (*shelter o site*), en el cual se encuentra un dispositivo de construcción modular diseñado para alojar dos configuraciones principalmente, la de transmisor y la de receptor. Dependiendo del volumen de información que se maneje se determinará el modelo y el número de dispositivos a instalar. Un modelo general de estos equipos se muestra en la *Figura 3.5.2*.

SCB 8514-30 Enlace de microondas 1W 10 GHz

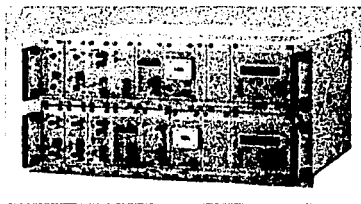


Figura 3.5.2

TESIS CON
FALLA DE ORIGEN

Una representación de lo explicado en el párrafo anterior se presenta en la Figura 3.5.3:

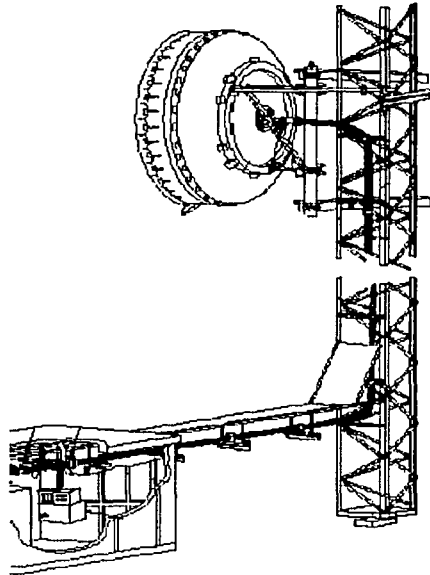


Figura 3.5.3

Por último un detalle importante es que las transmisiones a larga distancia se llevan a cabo, mediante la concatenación de enlaces punto a punto entre torres adyacentes (*repetidoras*), hasta cubrir la distancia deseada.

3.6 COMPONENTES DE UN ENLACE VÍA MICROONDAS

A continuación se hace una descripción general de lo que son los principales componentes de un enlace vía microondas del tipo punto a punto.

Antenas. Una antena es básicamente la interfase entre el espacio y la línea de transmisión, tanto para transmitir como para recibir información. Generalmente esta formada por un conjunto de conductores que, unidos a un generador, permiten la emisión de ondas, o que, conectados a una impedancia, sirven para captar las ondas emitidas por una fuente lejana. Existen diferentes tipos, y de éstos depende su operación. Sin embargo, todas ellas tienen un objetivo común, generar ondas electromagnéticas a partir de la asociación de dos tipos de ondas: unas de tipo eléctrico y otras de tipo magnético. Ambas provienen de los campos eléctrico y magnético dispuestos perpendicularmente entre sí y con la dirección de propagación de la onda.

En la práctica, las ondas electromagnéticas pueden seguir cualquier dirección en el espacio a partir de una antena, o del origen de la radiación, pero siempre seguirán manteniendo las dos componentes, eléctrica y magnética, con un desfase de 90° entre ellas, es decir, los dos campos seguirán siendo perpendiculares.

Una antena puede tener una polarización vertical debido a la disposición adoptada por el campo eléctrico generado por la misma. En caso de estar invertidos los campos eléctrico y magnético se dice que la polarización de la antena es horizontal.

El factor polarización es muy importante. Éste puede mantenerse o variar de forma continua, lo que supone conservar siempre en el mismo plano o en planos cambiantes los campos eléctrico y magnético. Si la polarización de la señal de antena es de tipo horizontal también debe adecuarse la antena receptora para recoger la máxima señal según sea el tipo de polarización, de no hacerlo así, tiene lugar una pérdida importante de energía en la recepción de la señal.

Las características que son primordiales para identificar el tipo de antena con la que se está trabajando son: su ganancia, el patrón de radiación, el ancho del haz y por último sus características mecánicas.

Los tipos de antenas mas comunes son:

- De radiodifusión (para bajas frecuencias)
- Yagui – Uda (para bajas frecuencias)
- De rejilla (para bajas frecuencias)
- Parabólicas (para bajas y altas frecuencias)
- De alto desempeño (manejan frecuencias altas y por lo tanto son direccionales)

Líneas de transmisión. Generalmente se trata de un medio de transmisión guiado, es decir, es un medio por el cual las ondas electromagnéticas se desplazan por un medio material sólido. Los más utilizados para enlaces de microondas son el cable coaxial y la guía de onda.

El cable coaxial está compuesto por un par de conductores de cobre o aluminio dispuestos de forma concéntrica. Podemos distinguir tres partes: malla, núcleo y dieléctrico.

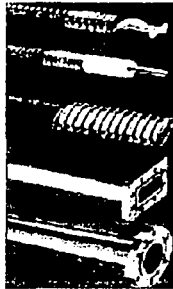
El núcleo es el que transmite la señal y esta protegido por el dieléctrico (*que es un aislante*), la malla hace de punto de conexión y atenúa las interferencias. Suele ser de cobre porque cuando un conductor se encierra sobre otro conductor, hace que se anule el campo magnético del primer conductor, con lo cual se atenúa la diafonía. Se entiende por diafonía que la señal transmitida por un par de cables logra pasar a los demás pares adyacentes del cable, produciendo de esta forma interferencias entre las líneas del cable.

La guía de onda esta generalmente formada por un tubo de material conductor que puede transportar solamente ondas electromagnéticas. Las paredes conductoras del tubo confinan la onda al interior por reflexión hacia un lado y otro de la superficie interior de la guía en un patrón de zig zag, este tubo puede estar vacío o relleno con un dieléctrico.

La onda será casi totalmente reflejada, siempre y cuando las paredes de la guía están hechas de un buen conductor. Por lo general, las guías tienen sección transversal rectangular, pero también pueden tenerla circular ó elíptica.

Una guía de onda no conduce corriente, sino que sirve como un límite que confina a la onda en su interior. Las paredes de la guía están compuestas de un material conductor, por lo tanto reflejan la energía electromagnética que choca con la superficie. Si la pared de la guía es un buen conductor y muy delgado, en las paredes del conductor fluiría poca corriente y como consecuencia se disipará poca potencia. En realidad la conducción de la energía no ocurre en las paredes, sino en el dieléctrico que se encuentra dentro de la guía de onda.

Ejemplos de cables coaxiales y guías de onda



Cable coaxial con dieléctrico de aire

Cable coaxial con dieléctrico de esponja

Guía de onda elíptica

Guía de onda rectangular

Guía de onda cilíndrica

Torres. Se entiende por torre toda estructura elevada donde va fijada una antena, que permite liberar de obstáculos que puedan interferir o atenuar los efectos de las ondas electromagnéticas recibidas o emitidas.

Dentro de las más utilizadas actualmente se tienen:

- *Torres Cables.* Se mantienen estables mediante tensores, son ligeras pero requieren un área grande para instalarlas.
- *Torres Autosoportadas.* Su estructura es piramidal, son pesadas pero requieren un área reducida para instalarlas.
- *Monopolar.* Su estructura es cilíndrica y las líneas de transmisión van por dentro del tubo.

Fuente de Alimentación. Es el dispositivo que proporciona al componente que lo disponga la corriente eléctrica precisa, tanto en intensidad como en voltaje. La fuente de alimentación se encuentra en el interior de una carcasa, generalmente es una caja metálica rectangular de gran tamaño dotada de un ventilador. De ella sale un manojo de cables que van a parar a los distintos dispositivos dentro de la carcasa.

Refugio. Es el lugar propuesto para la instalación del equipo donde se descargará la información aterrizada de la antena. Debe de contar con un sistema de ventilación con el fin de mantener el equipo en óptimas condiciones, así como fuentes de alimentación con sus respectivos contactos.

Equipo de Presurización. La presurización de cable fue instituida por primera vez en los 40s, para proteger los cables telefónicos de la entrada de humedad y los problemas subsecuentes ocasionados a la cubierta o fallas en la integridad de un empalme entre cables. Un gas puro seco tal como el aire, fue forzado bajo presión en los cables. Este gas puro bajo presión dentro del cable evita que entre humedad y purga el cable a cualquier humedad inherente. A lo largo de los años, se ha difundido el uso de la presurización para abarcar otros tipos de transportadores de comunicación tales como cable coaxial y guía de ondas.

Por último es importante considerar que aunque se tengan todos los componentes antes descritos, funcionando adecuadamente, la confiabilidad de un enlace de este tipo, no solo depende de ellos. El medio a través del cual se lleva a cabo la propagación de las ondas electromagnéticas es un factor fundamental para que la información no sufra modificaciones, por esto se consideran los siguientes fenómenos para determinar que tan factible es el enlace entre un punto emisor y un receptor:

TESIS CON
FALLA DE ORIGEN

- *Reflexión.* La reflexión se presenta al incidir una onda entre dos medios distintos. Si las características de los medios y el ángulo de incidencia son los adecuados la onda rebota en la frontera.
- *Refracción.* La refracción se presenta de manera similar a la reflexión, pero en este caso no existe rebote, solamente una desviación en la dirección de propagación.
- *Difracción.* Cada frente de onda actúa como un nuevo emisor de ondas electromagnéticas de tal manera que si la onda incide con un borde, se produce una región de sombra donde se recibe la onda pero con menor intensidad.
- *Atenuación.* Toda onda que se propaga sufre una disminución de intensidad en su campos eléctrico y magnético. Normalmente se le conoce como pérdida en el espacio libre. Este fenómeno es independiente del medio que pudiera introducir pérdidas adicionales.
- *Ruido.* Cualquier señal no deseada que se agregue a una onda en el proceso de propagación (o en los equipos).

3.6.1 TRANSMISOR Y RECEPTOR EN LAS MICROONDAS

La definición clásica de transmisor nos dice que es un instrumento que capta una variable en proceso y la transmite a distancia a un instrumento indicador o controlador; pero en realidad es eso y mucho más. La función primordial de este dispositivo es tomar cualquier señal para convertirla en una señal estándar adecuada para el instrumento receptor.

Para establecer un enlace de microondas punto a punto hace falta una antena emisora, una antena receptora y un canal para transportar la información, que vendrá especificado por su ubicación geográfica y por la frecuencia de funcionamiento.

Un enlace queda establecido cuando la potencia que llega a la antena receptora es suficiente. Siendo lo que más interesa a un diseñador de sistemas de telecomunicaciones el conocer cuánta potencia necesita la antena emisora para que la antena receptora disponga de esa potencia mínima. Para esto hace falta conocer todas las ganancias y atenuaciones que va a sufrir la señal en el enlace. Al cálculo de la potencia en cada etapa de la transmisión se le denomina cálculo del balance de enlace.

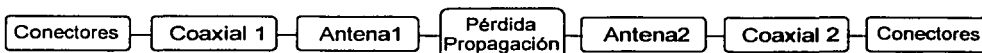
Podemos calcular el nivel de recepción de señal en función de todos los factores condicionantes:

$$\text{Nivel_recepción_señal} = \text{Potencia_transmisión_antena1} - \text{Pérdida_conectores_antena1} - \text{Pérdida_cables_antena1} + \text{Ganancia_antena1} - \text{Pérdida_propagación} + \text{Ganancia_antena2} - \text{Pérdida_cables_antena2} - \text{Pérdida_conectores_antena2}$$

O abreviando:

$$\text{Nrs} = \text{Pt}_{a1} - \text{Pco}_{a1} - \text{Pca}_{a1} + \text{Ga}_{a1} - \text{Pp} + \text{Ga}_{a2} - \text{Pca}_{a2} - \text{Pco}_{a2}$$

Un esquema más gráfico de la fórmula, el cual muestra los componentes involucrados para establecer el enlace de microondas, se presenta a continuación siguiendo el mismo orden que se mostró en la ecuación:



La descripción de estos elementos se presentó en el apartado 3.5 de este capítulo.

3.6.2 REPETIDORES

Un enlace está constituido por equipos terminales y repetidores intermedios. La función de los repetidores es salvar la falta de visibilidad impuesta por la curvatura terrestre o algún obstáculo físico (árboles, edificios, etc.) y conseguir así enlaces superiores al horizonte óptico.

Los repetidores pueden ser:

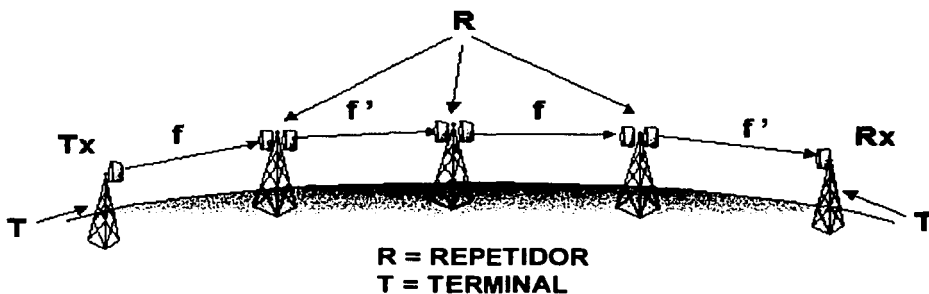
- Activos. Son aquellos que además de cambiar la dirección del haz, le dan ganancia a la señal que pasa por ellos.
- Pasivos. También llamados reflectores. En ellos no hay ganancia y se limitan a cambiar la dirección del haz.

En una estación repetidora que tiene como mínimo una antena por cada dirección, es absolutamente necesario que las frecuencias de emisión y recepción estén suficientemente separadas, debido a:

- La gran diferencia entre los niveles de las señales emitidas y recibidas, que puede ser de 60 a 90 dB.
- La necesidad de evitar los acoples entre ambos sentidos de transmisión.
- La directividad insuficiente de las antenas sobre todas las ondas métricas.

Los repetidores pasivos más comúnmente utilizados por las empresas de telecomunicaciones en la Cd. De México, son los edificios más altos:

- Torre Mayor (Altura: 225 metros)
- Torre Pemex (Altura: 214 metros)
- World Trade Center (Altura: 207 metros)
- Torre Altus (Altura: 195 metros)
- Torre Latinoamericana (Altura: 182 metros en su antena 139 en su techo)
- Los Arcos Bosques 1 (Altura: 161 metros)



TESIS CON
FALLA DE ORIGEN

3.6.3 ZONA DE FRESNEL

Las zonas de Fresnel son elipsoides que rodean la trayectoria directa entre el transmisor y el receptor. La primera zona de Fresnel, es decir, la región que encierra la primera elipsoide, contiene la mayor cantidad de potencia destinada al receptor. De existir un obstáculo en los límites de la primera zona de Fresnel, la onda reflejada tiende a cancelar la onda directa, dependiendo de las amplitudes relativas de cada onda. Para efectos prácticos se considera propagación pura por línea de vista (sin atenuación por difracción) si no existen obstáculos dentro de la primera zona de Fresnel.

Las demás zonas de Fresnel (segunda, tercera, etc.) tienen mucho menor efecto en cuanto a pérdidas por difracción debido a su contenido de potencia de la señal. El radio del conjunto de elipsoides varía a lo largo de la trayectoria y está dado por:

$$F_n = [n \cdot \lambda \cdot d_1 \cdot d_2 / (d_1 + d_2)]^{1/4} = [n \cdot \lambda \cdot d_1 \cdot d_2 / d]^{1/4}$$

Donde:

n : Número entero que caracteriza a la elipsoide correspondiente.

λ : Longitud de onda de la portadora.

d_1 : Distancia desde una terminal al punto donde se desea calcular el radio de la zona de Fresnel.

d_2 : Distancia desde la otra terminal al punto donde se calcula el radio de la zona de Fresnel.

d : $d_1 + d_2$

El radio de la primera zona de Fresnel F_1 está dado por:

$$F_1 = 17.32 [d_1 \cdot d_2 / (f \cdot d)]^{1/4} \text{ metros}$$

Donde d_1 , d_2 y d están en kilómetros y f en Ghz.

En términos generales, la zona de Fresnel es el volumen vacío que tiene que haber entre el emisor y el receptor. La altura mínima a la que se tendrán que colocar las antenas será r , es decir, la distancia del objeto más alto a la línea que forman las dos antenas. Esta distancia nunca debe ser menor del 70 % de r más la curvatura de la tierra.

Ejemplo: Para una distancia de tres kilómetros la zona de Fresnel (teniendo en cuenta la curvatura terrestre) será de 7 metros. Para una distancia de 30 Km. la zona de Fresnel será de 58 m, es decir, si formamos una línea con las dos antenas cualquier objeto tiene que estar a una distancia de 58 m de esa línea, de lo contrario producirá una pérdida.

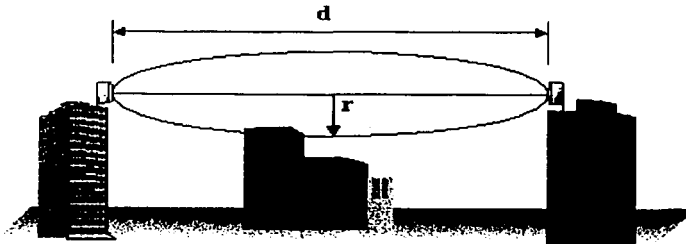


Figura 3.6.3.1

3.7 TECNOLOGÍAS DE TRANSPORTE

Un sistema de telecomunicaciones consiste en una infraestructura física a través de la cual se transporta la información desde la fuente hasta el destino, y con base a esta infraestructura se ofrece a los usuarios los diversos servicios de telecomunicaciones, para nuestro caso un enlace de microondas punto a punto. A continuación se explica de forma general como se transporta y transmite la información.

3.7.1 MULTIPLEXACIÓN

De forma general la multiplexación surge de la siguiente manera: En los sistemas de transmisión de audio, una frecuencia de audio es transportada en forma continua a lo largo de una señal portadora. Sin embargo, por mucho tiempo se cuestionó si esto es realmente necesario para transmitir una señal completa, o si la transmisión del valor de la señal en intervalos regulares pudiese ser suficiente. Los científicos Nyquist y Shannon, examinaron el problema y probaron que muestras tomadas en intervalos regulares pueden ser usadas para transmitir una señal de audio.

La ventaja de enviar información con pulsos cortos, es que los tiempos entre los dos pulsos sucesivos pueden ser usados para enviar información de pulsos de otras señales por el mismo canal de transmisión, es decir, se colocan varias comunicaciones sobre un mismo medio de transporte, a este proceso realizado en la transmisión se le conoce como multiplexación

3.7.2 SISTEMAS DE TRANSMISIÓN

Las facilidades de transmisión son caras y, a menudo, los equipos terminales de datos que se comunican por enlaces de microondas, satélite o cables coaxiales, no utilizan la capacidad total del canal, desperdiciando parte del ancho de banda disponible. Este problema se soluciona mediante unos equipos denominados *multiplexores*, que reparten el uso del medio de transmisión en varios canales independientes que permiten accesos simultáneos a los usuarios, siendo este proceso totalmente transparente a los datos transmitidos.

En un extremo, los multiplexores son equipos que reciben varias secuencias de datos de baja velocidad y las transforman en una única secuencia de datos de alta velocidad, que se transmiten hacia un lugar remoto. En dicho lugar, otro multiplexor realiza la operación inversa obteniendo de nuevo los flujos de datos de baja velocidad originales, a esta función se la denomina demultiplexar.

Dado que se efectuarán varias transmisiones distintas por la misma línea, la tasa de eficiencia del canal se ve notablemente mejorada.

Existen dos técnicas fundamentales para llevar a cabo la multiplexación:

- División de Frecuencia (FDM)
- División en el Tiempo (TDM)

3.7.3 ELEMENTOS DE MULTICANALIZACIÓN

FDM (Multiplexación por División en Frecuencia) es una técnica que consiste en dividir mediante filtros el espectro de frecuencias del canal de transmisión y desplazar la señal a transmitir dentro del margen del espectro correspondiente mediante modulaciones, de tal forma que cada usuario tiene posesión exclusiva de su banda de frecuencias (llamadas subcanales).

En el extremo de la línea, el multiplexor encargado de recibir los datos realiza la demodulación la señal, obteniendo separadamente cada uno de los subcanales. Esta operación se realiza de

manera transparente a los usuarios de la línea. Se emplea este tipo de multiplexación para usuarios telefónicos, radio o TV que requieren el uso continuo del canal.

Este proceso es posible cuando el ancho de banda del medio de transmisión excede la capacidad del mismo en cada una de las señales a transmitir. Se pueden transmitir varias señales simultáneamente si cada una se modula con una señal portadora de frecuencia diferente, y las frecuencias de las portadoras están lo suficientemente separadas como para que no se produzcan interferencias. Cada subcanal se separa por unas bandas de guarda para prevenir posibles interferencias por solapamiento.

La señal que se transmite a través del medio es analógica, aunque las señales de entrada pueden ser analógicas o digitales. En el primer caso se utilizan las modulaciones AM, FM y PM para producir una señal analógica centrada en la frecuencia deseada. En el caso de señales digitales se utilizan modulaciones ASK (modulación por corrimiento de amplitud), FSK (modulación por corrimiento de frecuencia), PSK (modulación por corrimiento de fase) y DPSK (variación de PSK donde se toma el ángulo de fase del intervalo anterior como referencia para medir la fase de cualquier intervalo de señal), en las cuales los símbolos digitales se transforman en ondas compatibles con el canal de transmisión.

En el extremo receptor, la señal compuesta se pasa a través de filtros, cada uno centrado en una de las diferentes señales portadoras. De este modo la señal se divide otra vez y cada componente se demodula para recuperar la señal.

TDM (Multiplexación por División de Tiempo) es una técnica para compartir un canal de transmisión entre varios usuarios. Consiste en asignar a cada usuario, durante unas determinadas "ranuras de tiempo", la totalidad del ancho de banda disponible. Esto se logra organizando el mensaje de salida en unidades de información llamadas *tramas*, y asignando intervalos de tiempo fijos dentro de la trama a cada canal de entrada. De esta forma, el primer canal de la trama corresponde a la primera comunicación, el segundo a la segunda, y así sucesivamente, hasta que el n-ésimo más uno vuelva a corresponder a la primera.

El uso de esta técnica es posible cuando la tasa de los datos del medio de transmisión excede de la tasa de las señales digitales a transmitir. El multiplexor por división en el tiempo muestrea, o explora, cíclicamente las señales de entrada (datos de entrada) de los diferentes usuarios, y transmite las tramas a través de una única línea de comunicación de alta velocidad. Este tipo de multiplexores son dispositivos de señal discreta y no pueden aceptar datos analógicos directamente, sino demodulados mediante un módem.

Funcionan a nivel de bit o a nivel de carácter, los cuales se originan tras un ciclo completo de muestreo y codificación de n canales de comunicación.

Usando ésta técnica, un número de comunicaciones puede ser combinado en una señal portadora. Cada comunicación está representada por una serie de muestras, cada una de las cuales se representa en forma de un código digital.

3.7.4 JERARQUÍAS DE MULTIPLEXACIÓN

Existen tres normas de multiplexación a partir de una señal de 64 Kbps. Estas son reguladas por la UIT-T (Unión Internacional de Telecomunicaciones sección Transmisión) en la recomendación G702.

- Norma Americana (ANSI).
- Norma Europea (ETSI).
- Norma Japonesa.

TESIS CON
FALLA DE ORIGEN

Todas las jerarquías parten de una velocidad a nivel de canal de 64 Kbps sobre la que se estructuran los niveles jerárquicos en cualquier sistema.

3.7.5 ESTRUCTURA DE TDM PARA 32 CANALES

El Sistema Experto aquí propuesto, se basa en los lineamientos establecidos por la norma europea ETSI y la técnica de Multiplexación por División de Tiempo (TDM).

En Europa ha sido estandarizado y aceptado por la UIT-T un sistema TDM de 32 canales (ETSI). Cada canal tiene 8 bits, esta estructura se llama trama (Frame) y tiene 256 bits. Una llamada es asignada a un canal en una trama semejante. Esto significa que se pueden enviar 8 bits en cada trama.

Un acceso de usuario (llamada) es muestreada cada 125 microsegundos ($f_s=8\text{kHz}$), en la cual se envían 8 bits cada 125 microsegundos.

Por lo tanto la duración del canal es de: $125 \text{ microsegundos} / 32 = 3.906 \text{ microsegundos}$. La velocidad de transmisión (Bit rate) de la trama es de 256 bits en 125 microsegundos, lo cual corresponden a 2.048 Mbits/seg, que es lo que conocemos como un E1. Éste se considera básicamente como una medida de capacidad de canales.

En la estructura de la trama para un E1, la asignación de los canales es de la siguiente manera:

- Canal 0: Sincronización de la trama (alineación)
- Canal 16: Señalización
- Canal 1-15 y 17-31: Voz/Datos

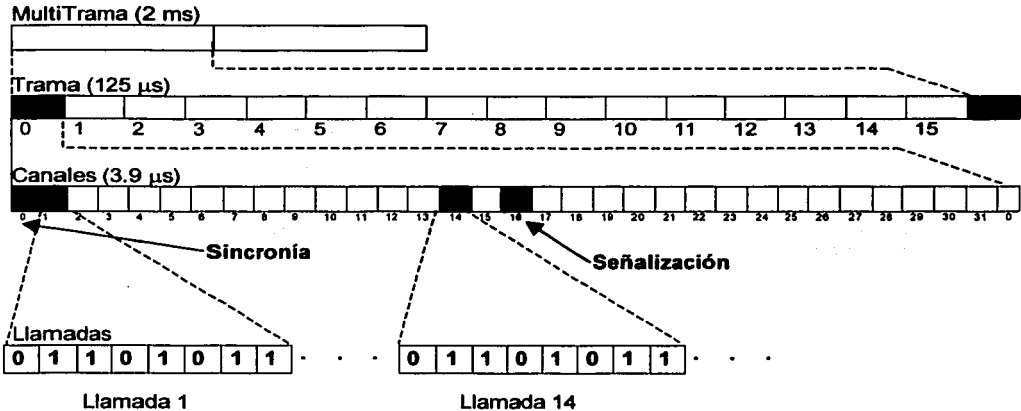
Del total de 32 canales del E1, únicamente 30 pueden ser utilizados para señales de voz/datos. Esta es la razón por la que esta estructura es algunas veces llamada, estructura de la trama de 30 canales. Cada canal usado para señales de voz contiene 8 bits, de los cuales el primero se usa como bit de signo y los otros siete son bits de magnitud codificados. En cada trama el mismo número de canal será dado al mismo usuario.

Cada canal proporciona 64 Kbps. Cuando no se requiere de un E1 completo, es posible proporcionar servicios de $N \times 64$, es decir N canales de 64 Kbps. A cada canal se le denomina E0.

3.7.6 OBJETIVO DE TDM

El objetivo de TDM es multiplexar n canales para el estándar elegido (ETSI), y así lograr lo que se denomina un E1, para esto se genera un conjunto de 16 tramas numeradas de la 0 a la 15, que es el ciclo completo de TDM en donde además de la información de las muestras de voz, se inserta información de alarmas, señalización y palabras de alineamiento tanto de trama como de lo que se denomina multitrama.

3.7.7 TRAMA DE UNA SEÑAL E1 DE 2.048 MBPS



Las características básicas de un E1 de primer orden están dadas por los siguientes parámetros:

- Frecuencia de muestreo es de 8 KHz.
- Duración del espacio de tiempo (TS) es 3.9 μ s.
- Anchura de bit de 0.488 μ s.
- Velocidad de transferencia binaria de 2.048 Mbps.
- Período de la trama de 125 μ s.
- Número de bit por palabra es 8 bit.
- Número de y tramas por multitrama de 16.
- Período de multitrama de 2 ms.
- Señal de alineamiento de trama en tramas pares.
- Señal de alineamiento de multitrama en TS16.
- Palabra de alineamiento de trama fija a 10011011
- Palabra de alineamiento de multitrama es 00001011

3.7.8 ORDENES JERÁRQUICOS DE MULTIPLEXACIÓN EUROPEA

En la jerarquía de multiplexaje de norma europea, en cada nivel jerárquico se agrupan respectivamente 4 señales digitales de orden jerárquico inferior en un tren de pulsos de orden superior.

Por ejemplo, para pasar del primer orden al segundo se agrupan 4 sistemas de 30 canales (llamados tributarias) a 2.048 Mbps. Para pasar del segundo al tercero se agrupan 4 sistemas de 120 canales (tributarias a 8.448 Mbps) y para pasar del tercero al cuarto se agrupan 4 sistemas a 480 canales (tributarias a 34.368 Mbps).

Los niveles jerárquicos de velocidades que se utilizan en México son:

<i>Velocidades de Jerarquía Europea</i>		
Orden	Velocidad Binaria	Capacidad de Canales
Primer Orden	2048 Mbps	30
Segundo Orden	8448 Mbps	120
Tercer Orden	34368 Mbps	480
Cuarto Orden	139264 Mbps	1920
Quinto Orden	564992 Mbps	7680

Así en cada paso, el multiplexor debe tomar en cuenta el hecho de que las velocidades a las que llegan en cada tributaria son distintas.

3.8 ENLACE PUNTO A PUNTO

3.8.1 ESTUDIO DE LÍNEA DE VISTA (LINE OF SIGHT- LOS)

Es el estudio realizado entre dos puntos generalmente para analizar las condiciones de operación de un enlace de microonda, dependiendo de las condiciones geográficas de los puntos y las características de operación de los equipos. Este estudio tiene como propósito general la determinación de la banda de frecuencia a utilizar en el enlace microonda, el equipo necesario para dicho enlace, la altura de las antenas que se necesitan para efectuar el enlace y todas las sugerencias de implementación a la infraestructura existente para que se garantice una comunicación con una alta calidad y disponibilidad en la aplicación final. Para lograr estas características, en el estudio de línea de vista también se debe determinar la topografía del terreno entre los puntos que se quieren enlazar, así como identificar los obstáculos existentes y las condiciones de propagación para la frecuencia a utilizar.

El estudio de línea de vista se divide en dos partes:

- Estudio del sitio a enlazar (Site Survey)
- Estudio de la trayectoria (Path Survey)

En el estudio del sitio a enlazar (Site Survey) se realiza únicamente el estudio de la infraestructura disponible en los sitios a conectar y se dan las sugerencias de implementación. El estudio de la trayectoria (Path Survey) es el análisis de las características y de los problemas de la trayectoria del enlace.

Para estudios de línea de vista urbanos, es necesario resaltar estas características. Se debe de tomar en cuenta todas las construcciones existentes en la trayectoria del enlace, así como la evaluación directa de las alturas de los obstáculos en dicha trayectoria. De acuerdo a la información topográfica, los obstáculos y la banda de frecuencia propuesta, se realizan simulaciones computarizadas para la obtención de las alturas y las dimensiones óptimas de las antenas. Esta simulación arrojará recomendaciones referentes a la aplicación, mejora o implementación de estructuras de soporte, tipos de superficie, recorridos de cableado, montaje de equipos, etc. En la figura 3.8.1 se muestra un ejemplo de que los dos puntos tienen que tener un enlace directo para poder llevar a cabo la comunicación entre los equipos.

TESIS CON
FALLA DE ORIGEN

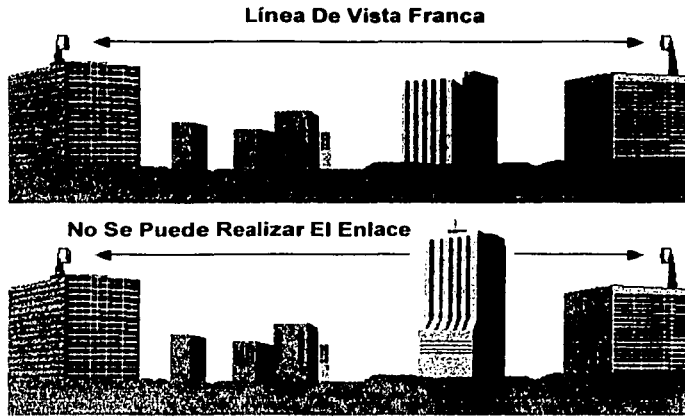


Figura 3.8.1.1 Línea de vista

3.8.2 ANÁLISIS PRELIMINAR

La metodología empleada para hacer el estudio de línea de vista considera, en su primer etapa, un análisis preliminar en el que se consideran los puntos a enlazar, su ubicación geográfica y las necesidades de disponibilidad planteadas por el cliente, de tal forma que con la ayuda de cálculos por computadora se obtenga la mejor combinación de equipos a fin de cubrir las expectativas. En éste análisis preliminar se recurre a la toma de coordenadas geodésicas auxiliándose por un DGPS o bien de un GPS para realizar posteriormente un proceso en conjunto con la estación base. Los cálculos preliminares del trayecto son usados para preparar un diseño preliminar del enlace.

Estos cálculos no son los definitivos, pero si sirven para preparar un primer presupuesto de un sistema definido. Los requerimientos técnicos y económicos son evaluados a partir de la información contenida en este diseño y será la base para una cotización detallada de los equipos necesarios para realizar en enlace.

3.8.3 PERFIL DEL ENLACE

La siguiente etapa consiste básicamente de la determinación del perfil del enlace, tomando en cuenta para ello la ayuda de soluciones cartográficas. Teniendo acceso a los mapas se localizan los dos puntos a enlazar y se toman datos correspondientes a las alturas absolutas sobre el nivel del mar con su correspondiente distancia hasta el punto tomado como inicio del enlace.

Dichas alturas son mostradas en las cartas como curvas de nivel que tienen separaciones dependientes de la escala empleada. Las escalas más comunes para el análisis topográfico son 1:50000 y 1:10000, permitiendo estas últimas una mayor precisión del relieve del terreno. Entonces se toman la mayor cantidad de puntos dependiente de las curvas de nivel presentes entre los dos puntos. Con la unión de estos puntos se obtiene un despliegue en dos dimensiones del terreno comprendido entre los dos puntos de interés definiendo así el perfil topográfico o del terreno.

Una vez obtenido el perfil topográfico es necesario anexar a este la altura y tipo de las estructuras existentes que obstaculizan la trayectoria para lo que se recurre a una inspección y medición directas e inclusive ante puntos críticos se realiza una verificación de su ubicación geográfica. (Figura 3.8.3.1)

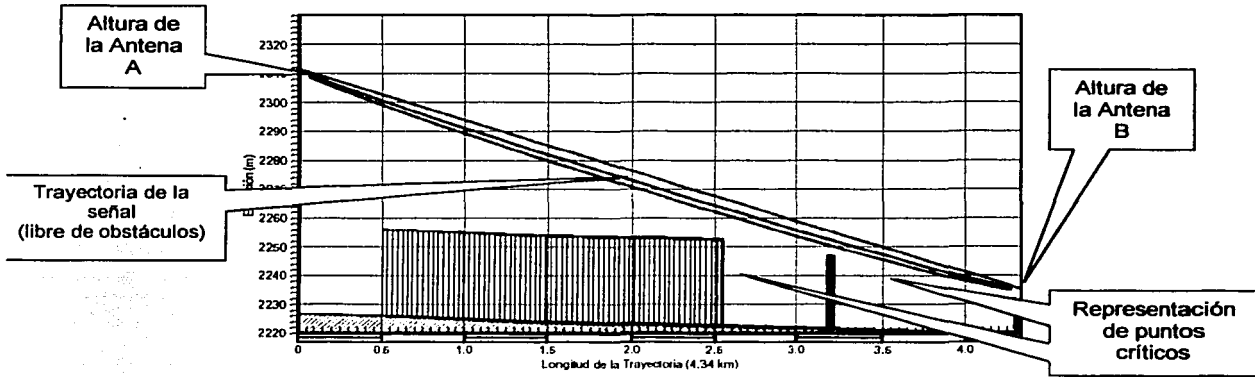


Figura 3.8.3.1 Perfil del enlace

3.8.4 CÁLCULO DE ALTURA DE ANTENAS

Los datos obtenidos en el perfil son capturados en paquetes especiales para computadora (PathLoss por ejemplo) y considerando datos como: criterios de libramiento, curvatura de la tierra, región de precipitación, parámetros del equipo propuesto, entre otras, sirven para determinar las pérdidas en la trayectoria con lo que se definen las alturas finales de las antenas en ambos sitios a fin de minimizar dichas pérdidas. El análisis así realizado podría arrojar inclusive la necesidad del empleo de otras modalidades como la diversidad en frecuencia, en polaridad, en espacio, etc.

3.8.5 ANÁLISIS DEL SITIO

El análisis del sitio incluirá la ubicación aproximada del sitio en mapas o croquis de la región a fin de determinar las vías de acceso a los sitios así como la consideración del tipo de transporte más adecuado para el traslado (Figura 4.8.5.1). El equipo propuesto requerirá de una infraestructura para su montaje y puesta en marcha para lo que se hace necesaria una supervisión del sitio asignado o propuesto para la instalación del equipo. En dicha supervisión se considera el tipo de alimentación de que se dispone, la trayectoria que describiría el cableado de banda base o la línea de transmisión según dependa, el tipo de torre existente, el tipo de construcción y las posibles necesidades para la adecuada instalación del equipo, tanto de interior como de exterior de acuerdo a las características del equipo a emplear.

Ante la inexistencia de una infraestructura se harían las observaciones y sugerencias pertinentes al tipo de equipo propuesto, así como en un primer plano se proporcionarían datos a las compañías que tomarían a su cargo el análisis de la obra civil.

TESIS CON
FALLA DE ORIGEN

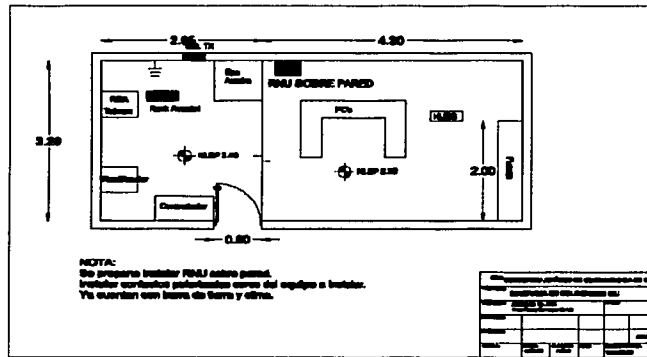


Figura 3.8.5.1 Croquis de acceso al Sitio del Cliente

3.8.6 INFORME

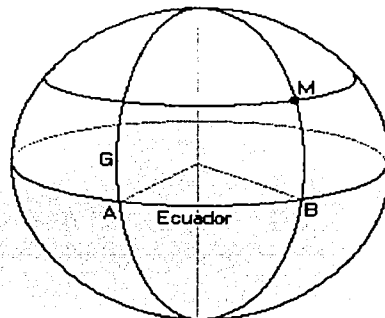
Los resultados obtenidos en el análisis preliminar arrojan una idea general de la factibilidad del enlace, estos de inicio indicarían la conveniencia de continuar con el análisis del enlace.

El informe final contendrá toda la información obtenida en la supervisión de los sitios así como el análisis de ingeniería para la o las propuestas resultantes.

3.9 CÁLCULO DE DISTANCIA

Como la Tierra es, aproximadamente una esfera podemos aplicar los conceptos de la trigonometría esférica para calcular la distancia entre dos puntos. Cada punto M de la Tierra está localizado por sus coordenadas geográficas: *longitud* y *latitud*.

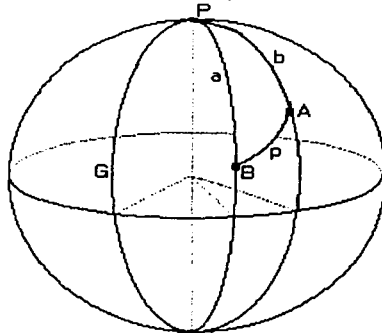
Se denomina *longitud del punto M* a la graduación del arco AB (medido sobre el ecuador, siendo G el meridiano de Greenwich o meridiano origen). Se determina la posición indicando si está al Este (+) o al Oeste (-). El arco BM determina la latitud del punto M . Se determina la posición si está al Norte (+) o al Sur (-) del Ecuador.



Ejemplo. Cálculo de distancia geográfica entre dos puntos A y B cuyas coordenadas geográficas son:

A(longitud; latitud) = A(55° 45' 13" E; 55° 48' 10" N)

B(longitud; latitud) = B(48° 50' 2" E; 20° 30' 40" N).



Considerando el triángulo esférico PBA, se tiene:

$$a = PB = 90^\circ - 48^\circ 50' 2'' = 41^\circ 9' 58''$$

$$b = PA = 90^\circ - 55^\circ 45' 13'' = 34^\circ 14' 47''$$

$$P = 55^\circ 48' 10'' - 20^\circ 30' 40'' = 35^\circ 17' 30''$$

Aplicando el teorema del coseno de la trigonometría esférica (fórmulas de Bessell) resulta:

$$\cos(p) = \cos(a) \cos(b) + \sin(a) \sin(b) \cos(P)$$

Ocupando la fórmula antes presentado y sustituyendo valores:

$$\cos(p) = 0,925 \text{ de donde } p = 22,386^\circ = 22^\circ 23' 9,6''$$

Mediante una proporción (a 360° corresponden 40000 km. de longitud de un círculo máximo) resulta entonces que ambos puntos están separados por 2487,333 km.

Estos conceptos se utilizan en el sistema a tratar en el siguiente capítulo para calcular la distancia que existe entre la ubicación geográfica de un cliente que solicite el enlace de microondas y los repetidores con los que éste tenga línea de vista franca.

3.10 CÁLCULO DE DISPONIBILIDAD Y CONFIABILIDAD

Las normas de seguridad de funcionamiento de los sistemas de microondas han alcanzado gran rigidez. Por ejemplo, se utiliza un 99.98% de confiabilidad general en un sistema patrón de 6000 Km. de longitud, lo que equivale a permitir solo un máximo de 25 segundos de interrupción del año por cada enlace.

Las empresas industriales que emplean sistemas de telecomunicaciones también hablan de una confiabilidad media del orden de 99.9999%, o sea un máximo de 30 segundos de interrupciones por año, en los sistemas de microondas de largo alcance.

Los cálculos estimados de interrupciones del servicio por fallas de propagación, emplean procedimientos parcial o totalmente empíricos. Los resultados de dichos cálculos generalmente se dan como tiempo fuera de servicio (TFS) anual por enlace o porcentaje de confiabilidad por enlace.

La confiabilidad de los enlaces de microondas puede darse según fallas de equipo, aplicando cálculos de probabilidad.

3.11 CÁLCULO DE ENLACE

Los factores más significativos que van a condicionar y determinar el funcionamiento y el rendimiento del enlace de microondas son los siguientes:

- Línea de vista.
- Cálculo de la distancia entre antenas.
- Cálculo de la altura de las antenas.
- Cálculo de las ganancias según el tipo de antena.
- Cálculo de la Zona de Fresnel.
- Cálculo de atenuación por lluvia.
- Cálculo de despolarización.
- Cálculo de las pérdidas por desvanecimiento.
- Cálculo de las pérdidas por atenuación debida a gases atmosféricos.
- Condiciones del terreno y meteorológicas.

Los cálculos que hacen referencia a las características de las antenas, condiciones climáticas y Zona de Fresnel son realizados por softwares especializados como lo son PathLoss ó ATDI. De los resultados que estos presenten dependerá la conclusión final del sistema propuesto en este trabajo.

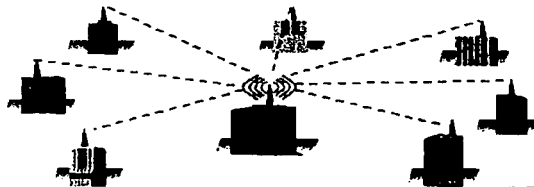
3.12 ENLACE MULTIPUNTO

Se define como enlace multipunto aquel, en donde dos antenas tienen que tener línea visual entre si para poder intercambiar señales (que transportan información). Si bien la antena de un cliente solo puede enfocar hacia la antena central de la empresa de telecomunicaciones a la que le contrató el servicio, esta última no es exclusiva para dicho cliente, siendo posible que una misma antena central atienda señales de varios usuarios.

El los primeros enlaces de microondas los enlaces eran punto a punto, donde el cliente tenía una antena que enfocaba directamente hacia su par distante la cual estaba exclusivamente colocada para ese enlace.

Los punto-multipunto no son tan rápidos como los punto a punto pero la velocidad es adecuada para las prestaciones requeridas por el Internet inalámbrico.

Gracias a esta tecnología es posible el acceso inalámbrico a Internet a usuarios en general y a costos razonables.



TESIS CON
FALLA DE ORIGEN

4. PLANTEAMIENTO DEL SISTEMA EXPERTO

4.1 INTRODUCCIÓN

En este capítulo se describen los aspectos mas relevantes para el desarrollo e implementación de un sistema que permita evaluar la factibilidad de un enlace de microondas punto a punto. Se presenta de forma general la definición del sistema, el alcance del mismo, los recursos, el análisis y la especificación estructurada, el diseño y por último, parte de la codificación utilizada.

4.2 PROBLEMA (DEFINICIÓN DEL SISTEMA)

Se requiere construir un sistema que tenga la capacidad de determinar la factibilidad de un enlace microonda punto a punto a partir de los datos básicos propuestos por parte del cliente, así como determinar los datos necesarios para dicho enlace.

El sistema será capaz de elegir de entre las opciones para realizar el enlace, la solución óptima de dichas opciones de acuerdo a tres factores importantes: distancia del enlace, disponibilidad del repetidor y validación del cálculo del enlace. En caso de no contar con los requerimientos para la mejor solución, el sistema deberá proponer la segunda mejor solución de entre las opciones que tiene y así sucesivamente hasta elegir la adecuada.

El sistema dará a conocer los datos óptimos tanto del punto A (cliente) como del punto B (Empresa de Telecomunicaciones) pero para éste último propondrá también un control sobre la asignación de los E1's en las antenas del repetidor asignado.

Finalmente, el sistema tendrá una interfase cómoda y amigable para que cualquier usuario nuevo pueda hacer uso de éste fácilmente.

4.2.1 ANÁLISIS DEL PROBLEMA

El problema pareciera tener solución a simple vista con cualquier lenguaje de programación, ya que correspondería a un tipo causa-efecto, es decir; ante un conjunto único de datos de entrada, se devuelve un único valor de salida; sin embargo, un lenguaje de programación estructurada no podría inferir ante una segunda causa sin tener que reinicializar su proceso, es decir; si queremos realizar una segunda consulta, se tendrá que iniciar nuevamente el proceso, perdiendo los datos almacenados e introducir los nuevos datos para la consulta. Por los datos naturales del problema, se requiere que los valores únicos de entrada se guarden para ser utilizados con otros datos que darán la solución óptima. De esta manera, el enfoque que se le dará a la solución del problema se orienta hacia el desarrollo de un sistema experto. Por lo tanto se plantea dar solución al problema, desarrollando un Sistema Experto.

Con un sistema experto podemos utilizar ampliamente el motor de inferencia que este tipo de herramienta posee para darle solución al corazón del problema, mientras que para la interfase amigable podemos utilizar cualquier lenguaje de programación estructurada compatible con el sistema experto utilizado.

4.2.2 RECURSOS

4.2.2.1 RECURSOS DE HARDWARE

Dentro de los recursos de hardware se consideran tres categorías: "El sistema de desarrollo, la máquina objetivo y otros elementos de hardware del nuevo sistema".

Para el sistema de desarrollo contamos con una computadora Pentium II y con otra computadora similar a Pentium IV. La primera es una Pentium II con Procesador Intel Celeron con una velocidad de 500 MHz, memoria RAM de 256 MB, Disco duro de 40 MB de capacidad, CD-Rom 52x Fax Modem 56.6 Kb y un monitor a color con resolución máxima de 800 x 600 pixeles.. La segunda computadora tiene procesador Athlon a una velocidad de 1.8 GHz, 515 MB en memoria RAM, disco duro de 40 GB, CD-Rom 52x, fax Modem y monitor con resolución máxima de 1280x1024.

Dentro de las máquinas objetivo se cuentan con computadoras Pentium II, Pentium III y Pentium IV.

4.2.2.2 RECURSOS DE SOFTWARE

También podríamos considerar tres categorías: el software de la máquina de desarrollo, el software de la máquina objetivo y el software adicional para la realización del nuevo sistema.

Las computadoras de desarrollo cuentan con el sistema operativo Windows Millennium, Shell de sistemas expertos: I2, Clips, WxClips; Base de datos Access 97 y 2000

Las computadoras objetivo cuentan con sistemas operativos Windows 98 y Windows 2000, Access 97 y Access 2000, no cuentan con cascarones de sistemas expertos.

Sistemas Operativos

Windows 95,98, Millennium: Son sistemas operativos que utilizan formato de 16 o de 32 bits (FAT16 o FAT32), perdiendo seguridad local y en red.

Windows NT, 2000, XP: Son sistemas operativos que utilizan formato NTFS, permitiendo la seguridad local entre usuarios y en red.

Sistemas Expertos

I2: Cascarón para crear sistemas expertos que utiliza el encadenamiento hacia atrás.

Jess: Cascarón para crear sistemas expertos basado en CLIPS y trabaja con programación Java

Clips: Cascarón para crear sistemas expertos basado en encadenamiento hacia delante.

Wxclips: Cascarón para crear sistemas expertos basado en CLIPS con una pequeña interfase gráfica.

Bases de Datos

Access 97: Es una herramienta para la gestión de base de datos relacionales, que permite crear y construir aplicaciones de uso práctico en base de 32 bits.

TESIS CON
FALLA DE ORIGEN

Access 2000: Es una Base de Datos de tipo Relacional que además permite importar la información de otras fuentes, navegar por la Web, interaccionar con otros productos de Microsoft tales como Excel o Word, desarrollar aplicaciones gráficas.

SQL: Es un lenguaje de consulta relacional.

Dbase: Es un lenguaje de consulta de bases de datos no relacionales.

Oracle: Es un sistema robusto, potente y completo de base de datos relacional SQL.

Lenguajes de Programación

Java: Es un lenguaje de programación orientado a objetos que es independiente de la plataforma utilizada.

C++: Es un lenguaje de programación orientado a objetos que se utiliza generalmente para la creación de programas de aplicación a gran escala

Visual Basic: Es un lenguaje de programación de propósito general basado en métodos, propiedades y eventos. Basado también en objetos pero no orientado a ellos.

4.3 ANÁLISIS DEL SISTEMA

El desarrollo de un sistema de información, independientemente de su tamaño y complejidad, requiere muchas actividades coordinadas y el empleo de una diversidad de herramientas y modelos. La metodología del desarrollo de sistemas es una forma estándar de organizar y coordinar estas actividades.

El análisis del sistema llega a la raíz del problema o a la necesidad y define los requerimientos de los usuarios. Con frecuencia, lo que los usuarios creen que necesitan o lo que parece ser el problema al principio, resulta ser algo totalmente diferente después de realizar un análisis profundo. Cuando el programador se reúne con los usuarios surgen nuevos y en ocasiones diferentes requerimientos que al principio no eran evidentes necesariamente.

Después de iniciar un proyecto de sistemas, se debe definir su alcance y desarrollar un enfoque profundo en los requerimientos de los usuarios.

4.3.1 LIMITACIONES Y DELIMITACIONES

Para el análisis de un enlace microonda punto a punto deben de considerarse tres estudios importantes: El Cálculo del Enlace, El Cálculo de Interferencia y El Cálculo de Disponibilidad. Para la realización de estos cálculos deben de considerarse factores importantes como son: Coordenadas de ambos nodos, número de E1's a instalar, altura de las antenas, frecuencia a utilizar, polaridad a utilizar, tipo y características de operación de las antenas, condiciones atmosféricas entre nodos, verificación visual de que exista una franca línea de vista entre ambos nodos, ente otras cosas.

El cálculo del enlace y el cálculo de interferencia involucran bastantes cálculos matemáticos que no serán tratados en éste proyecto, debido a que para realizarlos se requieren de programas especializados no disponibles para uso público. Tampoco se publica información de carácter confidencial para la empresa de telecomunicaciones.

Ayudándose de una herramienta para la realización de dichos enlaces (Pathloss por ejemplo), los usuarios u operarios finales podrán realizar con mayor seguridad y confiabilidad el cálculo adecuado para el repetidor adecuado.

Para la asignación de la frecuencia final que se empleará para realizar el enlace utilizaremos únicamente la frecuencia de 15 GHz y 23 GHz. Por convención de la empresa de telecomunicaciones en estudio, para los enlaces menores o iguales a 4E1's se utilizará la frecuencia de 15 GHz y para enlaces mayores de 4E1's se utilizará la frecuencia de 23 GHz a menos que los cálculos antes mencionados salgan negativos para dicha frecuencia.

De igual manera, para enlaces mayores de 12 Kilómetros se manejará la frecuencia de 15 GHz y para enlaces menores a 12 Kilómetros se podrá usar la de 23 o 15 GHz según el número de E1's solicitados.

El cálculo de enlace y el cálculo de interferencia que no son tratados por éste sistema, tendrán que realizarse con un software adicional especial para dichos cálculos. Dichos cálculos tomarán en cuenta los resultados que el sistema propondrá a partir de los siguientes datos básicos:

- Coordenadas geográficas del cliente
- Verificación de línea de vista
- Número de E1's

El sistema realizará el cálculo de la distancia geográfica entre los puntos dados por las coordenadas geográficas del cliente y las coordenadas geográficas del o de los repetidores. La respuesta de nuestro sistema optimizará la instalación del enlace para el cliente de acuerdo a los criterios antes mencionados, logrando reducir costos en la realización de los cálculos de enlace, debido a que los operarios o usuarios de la empresa de Telecomunicaciones pueden realizar el cálculo del enlace a partir de los datos preliminares que el sistema arroja.

Gracias a que el sistema puede elegir la mejor ruta de entre varios posibles caminos, el usuario solo se enfocará a realizar los cálculos necesarios para esa ruta elegida, de resultar negativo el cálculo realizado, el usuario le informará al sistema que esa ruta no es factible y, éste le devolverá la segunda mejor opción. En caso de que tampoco esta opción sea factible, el sistema dará la siguiente ruta de las asignadas. En caso de no contar con ninguna solución el sistema será capaz de detectar que no existe una solución óptima para ese enlace y dará una recomendación al usuario de lo que debe hacerse.

Con la ruta más óptima, el sistema podrá hacer un análisis sobre los E1's existentes para el repetidor seleccionado y optimizará la conexión de dichos E1's sobre las antenas instaladas en ese repetidor.

4.3.2 FACTIBILIDAD Y VIABILIDAD DEL SISTEMA EXPERTO

Como ya se ha mencionado anteriormente, un sistema experto puede inferir ciertos valores a partir de unos cuantos datos de entrada y almacenar y/o generar nuevos datos que serán necesarios para llegar a la meta final propuesta para el sistema experto.

Para la realización de este proyecto se reúnen las condiciones necesarias para poder diseñar, construir y tener un nivel óptimo de utilidad en operación del sistema experto para Enlace de Microondas Punto a Punto (EMPP).

4.3.3 ALCANCE DEL SISTEMA EXPERTO

El alcance del sistema Experto EMPP se enfoca al veredicto de la factibilidad del enlace microonda punto a punto, dando un mejor control sobre la asignación de cada enlace.

4.3.4 SELECCIÓN DE RECURSOS A UTILIZAR

Una vez estudiado, probado y analizado los diferentes cascarones para la construcción de sistemas expertos, los recursos de software utilizados para la realización de ésta tesis son los siguientes:

Sistema Operativo: Windows Millennium
 Sistema Experto: CLIPS V.6.2
 Base de Datos: Access 2000
 Lenguaje de Programación: Visual Basic
 ActiveX Control DLL : ClipsActiveX Control.OCX V.1.8.1.1

4.4 DIAGRAMA DE FLUJO DE DATOS (DFD)

El diagrama de flujo de datos es un modelo lógico que se utiliza en el análisis y diseño de sistemas de información, es además una herramienta gráfica de modelado, éste, describe los flujos de información y los procesos o actividades que cambian o transforman los datos en un sistema. El diagrama de flujo de datos puede mostrar los procedimientos que debe seguir el personal de una organización para llevar a cabo ciertas tareas y así convertir los datos en información.

El diagrama de flujo de datos es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por líneas y círculos o rectángulos; se le conoce también como carta de burbujas, DFD, diagrama de burbujas, modelo de proceso, diagrama de flujo de trabajo o modelo de función. Los diagramas de flujo de datos son gráficas dirigidas en donde los nodos especifican las actividades de proceso y los arcos la transferencia de datos entre nodos de proceso.

El DFD es un modelo lógico, y por lo tanto, no identifica discos, cintas, impresoras, computadoras o algún otro dispositivo físico. Los DFD se construyen en forma descendente. Los detalles de más alto nivel de los DFD son utilizados por los programadores para desarrollar el software que soporta las aplicaciones.

Los DFD's que nos permiten visualizar al sistema EMPP como una red de procesos funcionales son:

DFD DE NIVEL 0

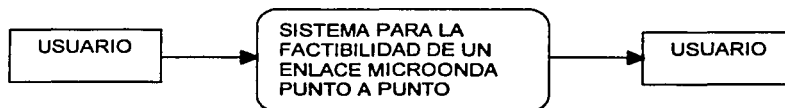


Figura 4.4.1

TESIS CON
FALLA DE ORIGEN

En el *DFD de Nivel 0* (Figura 4.4.1), se muestra el sistema como un diagrama general el cuál llevará a cabo todos los procesos que realizan las operaciones planteadas en el estudio de la definición del sistema. Los datos de entrada los genera el usuario de acuerdo a la actividad que desee realizar y la información que resulte del proceso solo será validada por el Ingeniero a cargo del estudio de enlace.

DFD DEL NIVEL 1

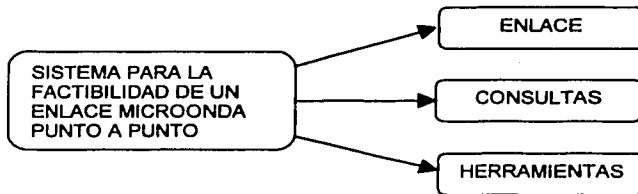


Figura 4.4.2

Para el *DFD de Nivel 1* (Figura 4.4.2), se presenta el desglose de los procesos que se propusieron para dividir las funciones que llevará a cabo el sistema EMPP. Básicamente se exponen tres: el primero (ENLACE), pretende cumplir la mayor parte de los objetivos señalados en el análisis del problema, ya que es en él donde se verifican los parámetros elegidos para evaluar la factibilidad de un enlace y así dar como resultado la conclusión de un diagnóstico preliminar y final para reportar las características que presentará el mismo; el segundo (CONSULTAS), permite al usuario realizar la consulta de los datos generales de los clientes, antenas y repetidores que son los principales actores del sistema ya que son ellos los que realizan las acciones fundamentales para que el sistema tenga sentido, tanto el cliente solicita un enlace, como los repetidores y antenas son evaluados para darle atención a dicha solicitud; el tercero (HERRAMIENTAS), da mantenimiento a los principales actores en el sentido de que son éstos a los que es posible dar de alta, de baja y modificar sus atributos con el fin de llevar un control en cuanto a la disponibilidad del sistema EMPP.

DFD DEL NIVEL 2 PARA EL PROCESO "ENLACE"

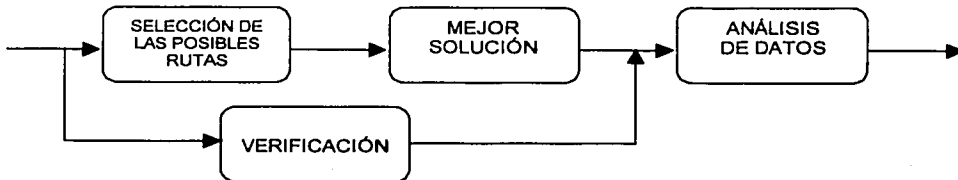


Figura 4.4.3

TESIS CON
FALLA DE ORIGEN

En el DFD DEL NIVEL 2 PARA EL PROCESO ENLACE (Figura 4.4.3), se establecen los procesos que realizarán el análisis de factibilidad del enlace: primero, la evaluación de las posibles rutas a seguir en el enlace solicitado (proceso de "Selección de las posibles rutas"), más adelante, la propuesta preliminar del mejor diagnóstico encontrado por el sistema EMPP que satisfaga a los parámetros del enlace y cumplan con las prioridades establecidas en el planteamiento del sistema (proceso de "Mejor solución"), y por último la organización de las líneas solicitadas con respecto a la disponibilidad de antenas del repetidor reportado en la propuesta preliminar, finalizando con un reporte general de la solicitud evaluada y aprobada como conclusión final al estudio realizado (proceso de "Análisis de Antenas"). Si el análisis de factibilidad de un enlace ya existe, no es necesario realizar el proceso de "Selección de las posibles rutas" y el de "Mejor solución", únicamente se tendrá un proceso de "Verificación" en donde se compruebe la existencia de dicho análisis y la disponibilidad de antenas para el repetidor que resultó en el estudio verificado, y así concluir con el proceso de "Análisis de Antenas", comentado en las líneas anteriores.

ESPECIFICACIONES DEL PROCESO "ENLACE" :

Selección de las posibles rutas. El proceso requiere como entrada los datos generales del cliente que solicita el enlace (Nom_Cliente, Tel_Cliente, Clave_Cliente, Estado, Ciudad, Dirección, Latitud, Longitud, Num_E1, Elevación.) y la lista de los repetidores visibles a partir de la localización geográfica del cliente; con el fin de realizar una evaluación de las rutas posibles y determinar algunos de los parámetros (polaridad, distancia y frecuencia). La función principal de este proceso es que el sistema especifique la causa del por qué son o no factibles dichas rutas.

Escoger mejor solución. Una vez que se tiene la lista de las rutas y su evaluación, el sistema debe de ser capaz de escoger y diagnosticar la ruta mas viable, de acuerdo a las dos principales prioridades: distancia entre el cliente y el repetidor seleccionado como mejor candidato, y la disponibilidad con la que el repetidor cuente. La primer prioridad a considerar es la distancia, ya que se tomará en cuenta la ruta con distancia mas corta, en segundo término aparece la disponibilidad, y aquí el sistema EMPP, será capaz de ofrecer tres alternativas: *aprobado*, *no tiene disponibilidad* y *saturado*. Entendiéndose por *"aprobado"*, que el repetidor seleccionado cuenta con las líneas suficientes para ofrecer el servicio, por *"no tiene disponibilidad"*, que el repetidor no cuenta con las líneas suficientes para satisfacer la solicitud del cliente, y por *"saturado"*, que el repetidor puede dar servicio a la solicitud pero se quedará en el rango de disponibilidad de líneas de reserva y por tanto tendrá que dar mantenimiento urgente al repetidor si decide instalar las líneas solicitadas.

Este proceso, presentará una propuesta preliminar, puesto que aún debe considerar la respuesta de un estudio externo (cálculo de enlace) el cual es necesario para asegurar si es factible o no, un enlace de microondas. Éste no es realizado por el sistema EMPP, sin embargo, si la respuesta a dicho estudio es aprobatoria se procede al siguiente proceso propuesto. De no ser aprobado, el sistema solicitará la modificación de algunos parámetros que intervienen en el cálculo de enlace para así hacer aprobatorio el estudio y continuar con los procesos. Si no se aprueba el estudio para esa ruta, aún modificando los parámetros, entonces el sistema propondrá la segunda mejor solución y así sucesivamente hasta que de la lista de rutas viables se apruebe para un repetidor el estudio del cálculo de enlace y se de fin a este proceso o que la lista quede sin repetidores y por consecuencia no se pueda realizar el enlace. Siempre que se cumpla la existencia de al menos una ruta con disponibilidad de "aprobado" para la lista de los repetidores seleccionados, se tendrá el análisis antes planteado, pero si se da el caso de que no se existe ninguna ruta "aprobada" debido a que los repetidores seleccionados presentan como disponibilidad "no hay disponibilidad" ó "saturado", el sistema EMPP mandará una advertencia al usuario comentando dicha situación y dará opción para continuar con los siguientes procesos indicando que será necesario dar mantenimiento a el repetidor elegido por mínima distancia entre cliente-repetidor.

Análisis de antenas. Este proceso tiene la tarea de organizar las líneas solicitadas por el cliente en las antenas disponibles del repetidor propuesto como mejor solución. Prácticamente la tarea del proceso "Realizar Análisis de Factibilidad" esta por terminar aquí, ya que solo hace falta presentarle al usuario un reporte de cómo fue establecida la asignación de líneas de acuerdo a la disponibilidad en las antenas del repetidor propuesto para pasar al proceso de conclusiones finales.

Verificación. Solo se podrá tener acceso a este proceso si ya se cuenta con un cliente dado de alta en el sistema EMPP, ya que básicamente con este proceso el sistema pretende dar la opción al ingeniero a cargo de verificar las características generales del cliente y su estudio de factibilidad realizado previamente, y así tener la oportunidad que si en un futuro se solicitan más líneas por parte del cliente, no sea necesario realizar nuevamente dicho estudio, y solo se indique el número de líneas que se sumarán a las ya instaladas. El proceso, además tendrá que verificar si existe disponibilidad en el repetidor indicado en la solicitud anterior y mandar la información necesaria para que se lleve a cabo el proceso de "Análisis de Antenas" y así presentar nuevamente una conclusión.

DFD DEL NIVEL 3 PARA EL PROCESO "SELECCIÓN DE LAS POSIBLES RUTAS"

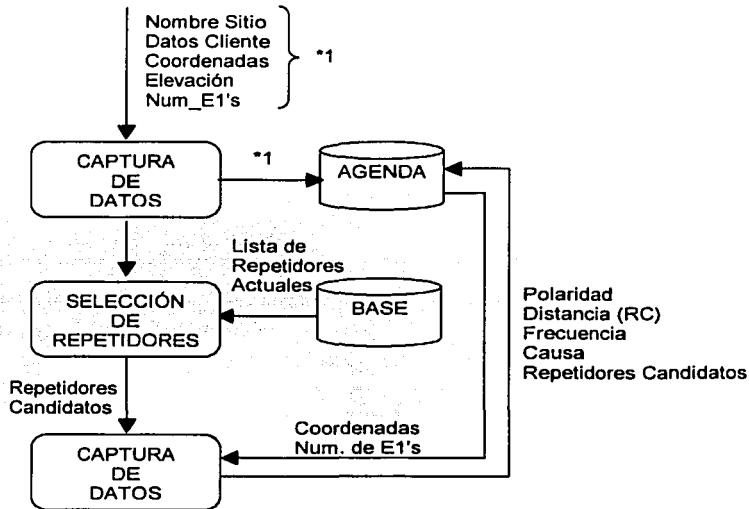


Figura 4.4.4

ESPECIFICACIONES DEL PROCESO "SELECCIÓN DE LAS POSIBLES RUTAS" :

Captura de Datos. Proceso para recopilar los siguientes datos de entrada: *Nom_Cliente*, *Tel_Cliente*, *Clave_Cliente*, *Estado*, *Ciudad*, *Dirección*, *Num_E1*, *Longitud*, *Latitud* y *Elevación*.

TESIS CON FALLA DE ORIGEN

Selección de repetidores. Se pretende presentar la lista de todos los repetidores disponibles por parte de la empresa de telecomunicaciones, con el fin de seleccionar aquellos con los que se tenga un contacto visual sin interferencia aparente debido a obstáculos como pueden ser árboles, edificios, postes, etc. De existir alguno de éstos se planteará la opción de librarlos mediante el uso del aditamento "torre" para que se pueda establecer una línea de vista directa con los repetidores existentes y formen parte de los candidatos para establecer una ruta de enlace entre cliente-repetidor.

Determinación de los parámetros a evaluar para todas las rutas. La función de este proceso es dar un diagnóstico preliminar, en el cual se reporten los resultados de distancia, frecuencia, polaridad y causa de disponibilidad, los cuales nos ayudarán a tener una solución preliminar basada en la primer prioridad especificada en el sistema, que es la "distancia" entre cliente-repetidor. Estos resultados los analizará y arrojará la herramienta que se propuso para el diseño del sistema experto EMPP "CLIPS", siempre y cuando la recopilación de datos que estará a cargo de Visual Basic sea la correcta y se esté enviado de forma adecuada para que el shell los interprete en base a las estructuras que maneja y al mecanismo de inferencia que tiene para que nos devuelva los valores requeridos y éstos sean expuestos al usuario mediante la interfaz gráfica de Visual Basic.

Base. Es necesario tener un almacén de datos permanentes, con el fin de que se pueda extraer de éste la información guardada por solicitudes de clientes y equipo disponible por la empresa de telecomunicaciones para satisfacer las necesidades del cliente y poder darle el servicio adecuado en base a la disponibilidad de equipos. El almacén denominado "Base", tiene la única función de ser la base de datos, de la cual se tomará o se depositará información únicamente si el proceso la solicita explícitamente, manteniendo su condición original.

Agenda. Se propone otro almacén al cual lo denominamos "Memoria de Trabajo". Los datos que se guarden en esta memoria, solo lo harán de forma temporal, ya que al terminar cualquier operación relacionada con el análisis del enlace, no se requerirán mas para estudios posteriores. Sin embargo son de gran valor en el transcurso del mismo puesto que con ellos se realizan todas las evaluaciones para el análisis de factibilidad de un enlace.

TESIS CON
FALLA DE ORIGEN

DFD DEL NIVEL 3 PARA EL PROCESO "MEJOR SOLUCIÓN"

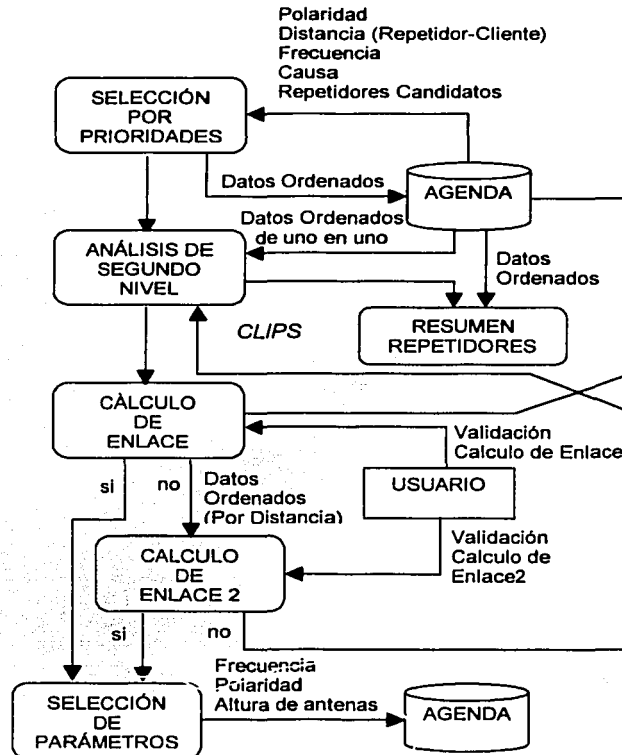


Figura 4.4.5

ESPECIFICACIONES DEL PROCESO "MEJOR SOLUCIÓN" :

Selección por prioridades. El sistema EMPP ordenará por mínima distancia la lista de repetidores candidatos y les anexará la causa de disponibilidad (*aprobado*, *no hay disponibilidad* y *saturado*). Para dar una conclusión preliminar el sistema se enfrenta a los siguientes casos: (1) verifica de la lista ordenada qué repetidor presenta una distancia más corta con respecto al cliente y analiza su causa de disponibilidad, de ser "aprobado", concluye para ese repetidor; (2) si al verificar la lista no se encuentra ningún repetidor con causa de disponibilidad "aprobado", se concluye con el primer repetidor que tenga la distancia más corta y que su causa sea "saturado"; (3) de no encontrar ningún repetidor con causa de disponibilidad "aprobado", ni tampoco "saturado", entonces el sistema concluirá con el primer repetidor de la lista, que obviamente tendrá como causa de disponibilidad "no hay disponibilidad". Hay que recordar que la causa de disponibilidad se obtiene de las conclusiones que arroja CLIPS.

TESIS CON
FALLA DE ORIGEN

"Análisis de segundo nivel". En este proceso se presenta al usuario, mediante una interfase de Visual Basic, la conclusión preeliminar a la que llegó el sistema EMPP. Se espera la aprobación de ésta según sea el caso que el sistema enfrente: si nos encontramos en el caso (1) expuesto en el proceso *"Selección por prioridades"*, las opciones son continuar con el estudio del cálculo de enlace o detener el análisis de factibilidad y realizar una nueva consulta; si estamos en el caso (2) se puede continuar con el estudio del cálculo de enlace, pero al exponer al usuario la conclusión final del análisis de factibilidad, se le recomendará dar mantenimiento urgente al repetidor seleccionado; y por último para el caso (3), el sistema tendrá que advertir al usuario que puede continuar o no con el estudio del cálculo de enlace, pero en los procesos posteriores no tendrá la opción de guardar el análisis realizado, únicamente se le presentará un resumen con las características encontradas en el análisis de factibilidad.

Cálculo de enlace 1. Este proceso espera respuesta por parte del usuario, el cual le debe indicar si el estudio externo "cálculo de enlace" ha sido aprobado o no para la ruta seleccionada entre repetidor-cliente. De ser aprobado, el proceso enviará toda la información requerida por el sistema y con ella organizar las líneas solicitadas por el cliente en el repetidor propuesto como mejor opción. Si es no aprobado, el sistema tendrá que enviar al usuario la opción del proceso *"Cálculo de enlace 2"*.

Cálculo de enlace 2. Como el cálculo de enlace no es aprobado, se presenta en este proceso la opción de modificar algunos de los parámetros que intervienen en el cálculo de enlace. Si el estudio es aprobado, con ciertas modificaciones, se mandará al usuario la lista de los parámetros que pudieron ser modificados. En el caso contrario, éste proceso tendrá que mandar la información necesaria al proceso *"Análisis de segundo nivel"*, para que éste proponga una segunda mejor opción en base a la lista de repetidores candidatos, considerando las prioridades del sistema EMPP (distancia y causa de disponibilidad).

En caso de que en la lista de repetidores candidatos, aparezcan otros con causa de disponibilidad "aprobado", se propondrán de uno en uno, en un orden por mínima distancia, como mejor opción de repetidor. De no ser así, se tomará otra vez de uno en uno a los repetidores con mínima distancia y causa *"saturado"*. Si no se tiene en la lista repetidores con estas dos causas, entonces el sistema tomará al primer repetidor con causa *"no hay disponibilidad"*, pero entonces solo dejará las opciones expuestas en el proceso de *"análisis de segundo nivel"* para este caso. Este análisis lo llevará a cabo el shell "CLIPS", gracias a la información que se mantiene almacenada temporalmente en *"Agenda"* y a la recopilación de la misma por parte de la interfaz gráfica de Visual Basic.

Selección de parámetros modificados. Este proceso únicamente presenta al usuario la lista de parámetros a modificar para el estudio del cálculo de enlace. Permite que el usuario indique cuáles fueron y manda esta información a la *"Agenda"* con el fin de concluir con estos datos en los siguientes procesos.

Resumen repetidores. Para mayor comodidad, éste proceso, le muestra al usuario la lista de los repetidores candidatos. Acompañados cada uno, de su causa de disponibilidad y de la distancia que existe entre el repetidor y el cliente en estudio.

Usuario. Para los procesos involucrados en el DFD NIVEL 3 de 2, "Usuario", es la persona que únicamente tiene la función de validar el estudio del cálculo de enlace.

DFD DEL NIVEL 3 PARA EL PROCESO "ANÁLISIS DE ANTENAS"

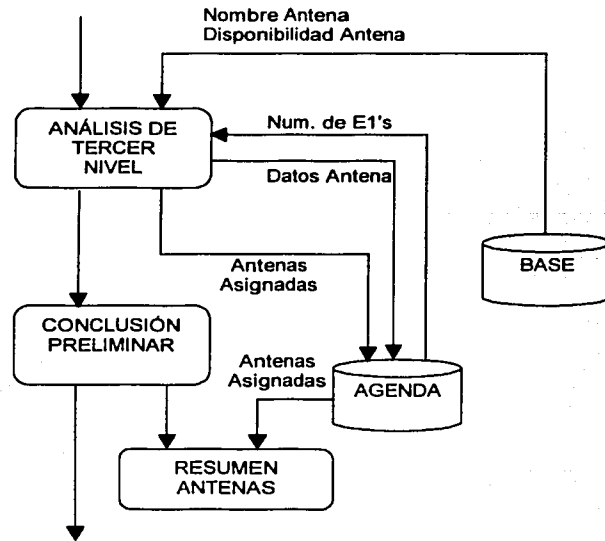


Figura 4.4.6

ESPECIFICACIONES DEL PROCESO "ANÁLISIS DE ANTENAS" :

"Análisis de tercer nivel". El sistema EMPP, hace uso del shell "CLIPS", para lograr que en este proceso se lleve a cabo la organización de las líneas solicitadas, en las antenas disponibles del repetidor propuesto como mejor opción. CLIPS arrojará el nombre de las antenas y la causa de disponibilidad para cada una de ellas, después de haber realizado el análisis correspondiente de asignación de líneas. Las causas de disponibilidad son: "Ya no hay E1's para instalar", "no hay espacio en la antena" ó "Instalar X E1's".

Conclusión preliminar. Mediante una interfaz gráfica se le presentará al usuario únicamente la lista de las antenas que tengan como causa de disponibilidad el número de líneas que se ocuparán de cada una de ellas, para instalar el nuevo enlace solicitado por el cliente (*"Instalar X e1's"*).

Resumen antenas. Se le muestra al usuario la lista de todas las antenas instaladas en el repetidor propuesto como mejor opción. Acompañadas cada una, de su causa de disponibilidad.

TESIS CON
FALLA DE ORIGEN

DFD DEL NIVEL 2 PARA EL PROCESO "CONSULTAS"

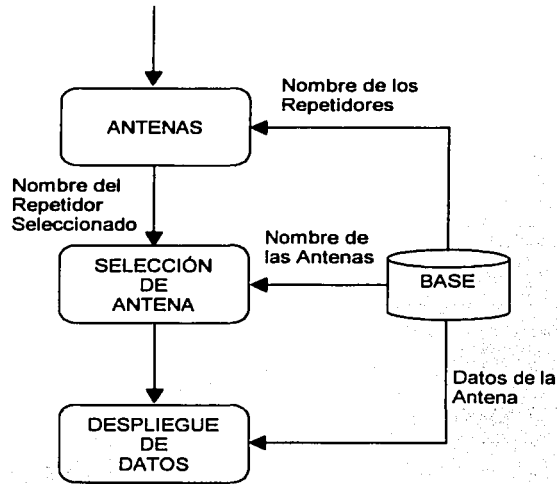


Figura 4.4.7

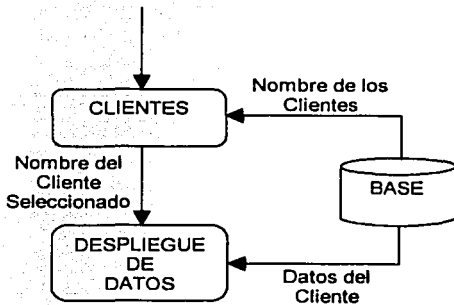


Figura 4.4.8

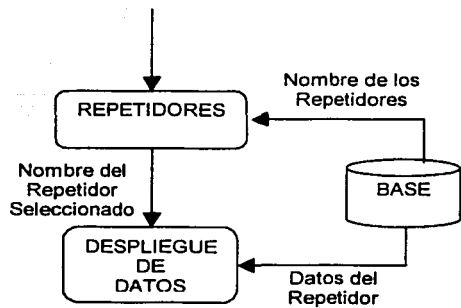


Figura 4.4.9

TESIS CON
FALLA DE ORIGEN

En el *DFD DEL NIVEL 2 PARA EL PROCESO CONSULTA* (Figuras 4.4.7, 4.4.8, 4.4.9), se establecen los procesos que permitirán consultar los datos generales de los actores principales del sistema EMPP: Antenas, Repetidores y Clientes.

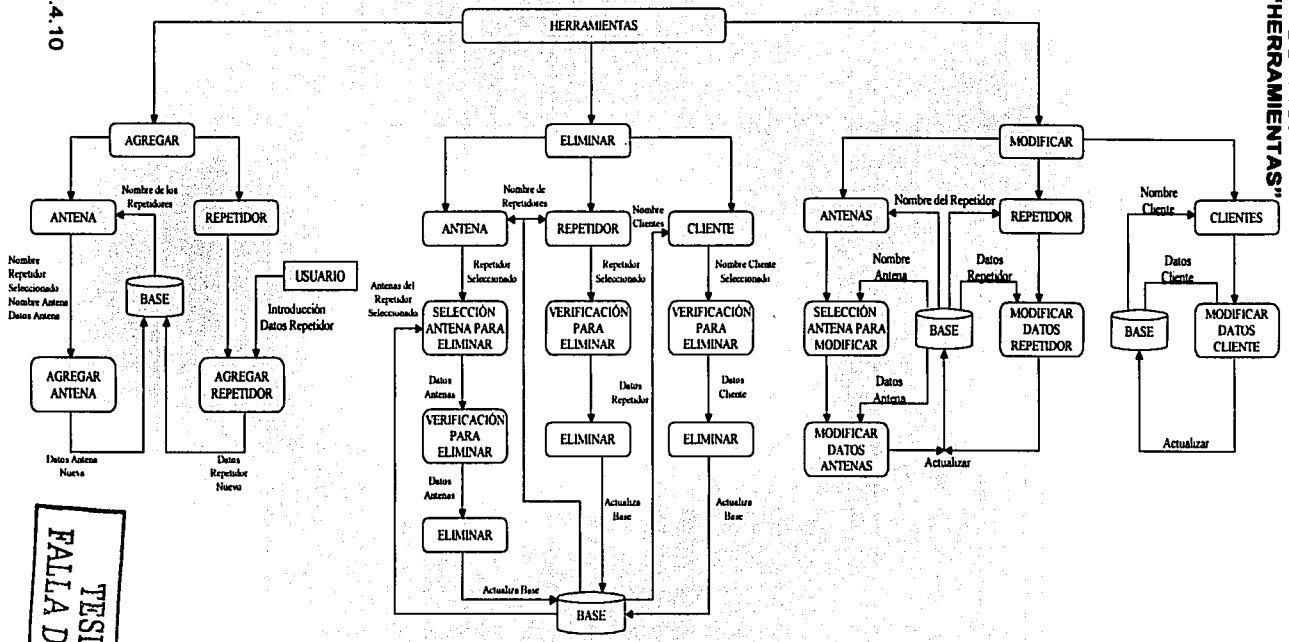
Para el caso de las Antenas: Nombre del Repetidor (Nom_Repetidor), Nombre de la Antena (Nom_Antena), E1's disponibles (E1_Disponibles), Frecuencia y Polaridad.

Para los repetidores: Nombre del Repetidor (Nom_Repetidor), Antenas Instaladas (Num_Antenas), Número de E1's instalados, Número de E1's disponibles (E1_Disponibles), Polaridad (Pol_Rep), Frecuencia (Frec_Rep), Latitud, Longitud y Elevación.

Y por último, para los Clientes: Nombre del Cliente (Nom_Cliente), Clave (Clave_Cliente), Dirección, Ciudad, Teléfono (Tel_Cliente), Estado, Nombre del Repetidor (Nom_Repetidor), Número de E1's instalados (Num_E1), Polaridad, Frecuencia, Latitud, Longitud y Elevación.

TESIS CON
FALLA DE ORIGEN

Figura 4.4.10



TESIS CON FALLA DE ORIGEN
 73

En el *DFD DEL NIVEL 2 PARA EL PROCESO HERRAMIENTAS* (Figura 4.4.10), se presentan los procesos que llevarán a cabo el mantenimiento del sistema EMPP: Agregar, Eliminar y Modificar.

Agregar. Sólo se podrán dar de alta, antenas y repetidores. Esta decisión depende del ingeniero a cargo, ya que se tendrá que tener la certeza de que el equipo a instalar esté funcionando correctamente y cumpla con las características y reglamentos establecidos por la compañía de telecomunicaciones y la COFETEL.

Eliminar. El sistema da la opción de dar de baja a Clientes, Antenas y Repetidores.

En el caso de los Clientes, es posible eliminar la solicitud por completo y así liberar todas las líneas que se mantenían ocupadas en ese enlace. O se pueden eliminar únicamente el número de líneas que el cliente desee dar de baja, sin anular su solicitud por completo.

Para las Antenas, es necesario que el sistema verifique que ningún cliente se encuentre conectado a ninguna antena, de lo contrario no será posible darla de baja.

Y por último para los Repetidores, el sistema repite la acción de verificar, pero para este caso revisa que no se cuente aún con antenas instaladas que sigan prestando servicio.

4.5 DIAGRAMA ENTIDAD-RELACIÓN DEL SISTEMA EMPP

El diagrama de entidad-relación es un modelo que describe la distribución de los datos almacenados en el sistema. También se le conoce como DER. El principal propósito del DER es representar los objetos de datos y sus relaciones.

Es importante modelar los datos de un sistema ya que las estructuras de datos y sus relaciones son complejas y resulta necesario examinarlas independientemente del proceso del sistema.

Los componentes principales para el DER del sistema EMPP son: *Tipos de objetos, Relaciones e Indicadores asociativos de tipo objeto.*

Tipos de Objetos. Los objetos que juegan un papel necesario en el sistema EMPP son:

- Clientes
- Repetidores
- Antenas
- C-A

Ya que el sistema no puede operar sin el acceso a éstos.

Relaciones. Éstas representan algo que debe ser recordado por el sistema: algo que no pudo haberse calculado ni derivado mecánicamente. Es decir, las relaciones representan la memoria del sistema EMPP.

Indicadores asociativos de tipo objeto. Un indicador asociativo de tipo objeto representa algo que funciona como objeto y como relación. Es algo acerca de lo cual se desea mantener alguna información.

Para el DER del sistema EMPP, la relación que asocia a un cliente con una o mas antenas, representa un indicador asociativo de tipo objeto debido a que es indispensable para el sistema recordar algunos de los datos acerca de la instancia C-A, como lo es el nombre de la antena y el

número de líneas que se le asignaron al cliente en esa antena. Estos datos, no son atributos de *Cientes*, ni de *Antenas*. Mas bien, se asocia con a la relación entre ambos.

Por lo tanto, en el DER que se muestra en la Figura 4.5.1 se indica que C-A funcionará como: un tipo de objeto y como una relación que conecta los dos tipos de objeto *Cientes* y *Antenas*. Lo importante en éste caso es que *Cliente* y *Antena* se mantienen solos, ya que *C-A* aparece únicamente como resultado de una relación entre los otros objetos con los cuales está conectada.

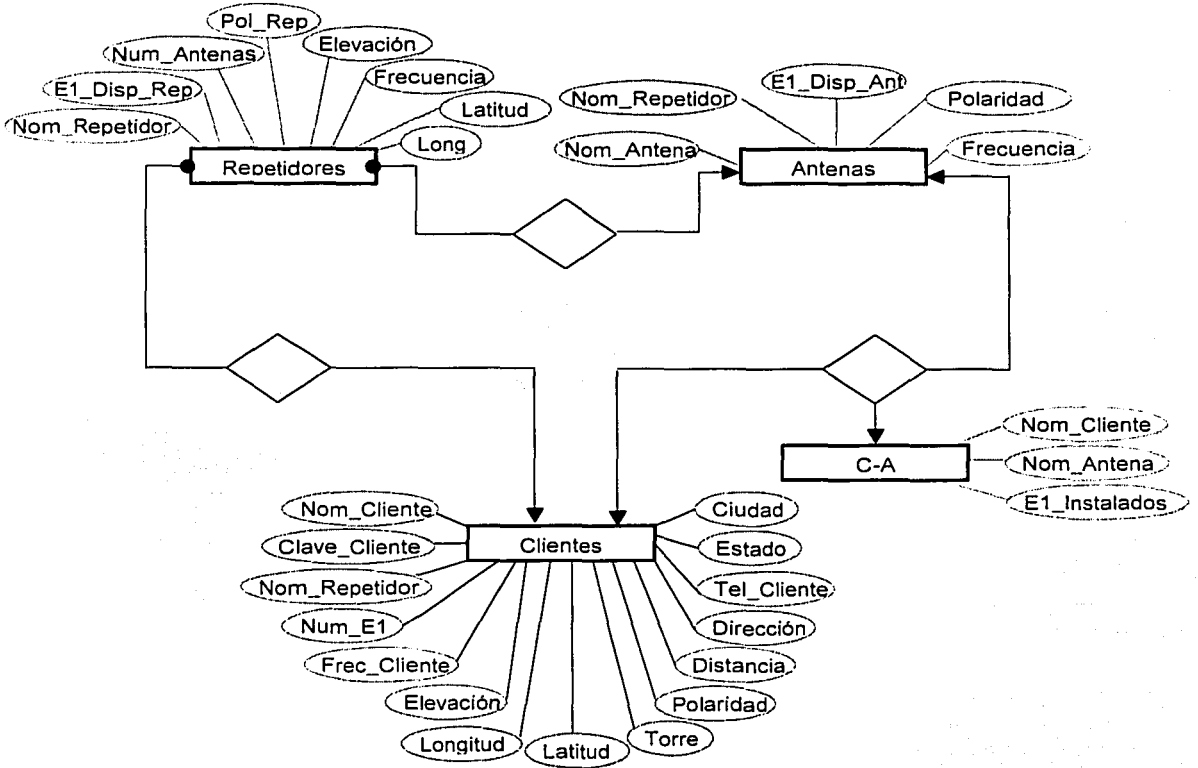


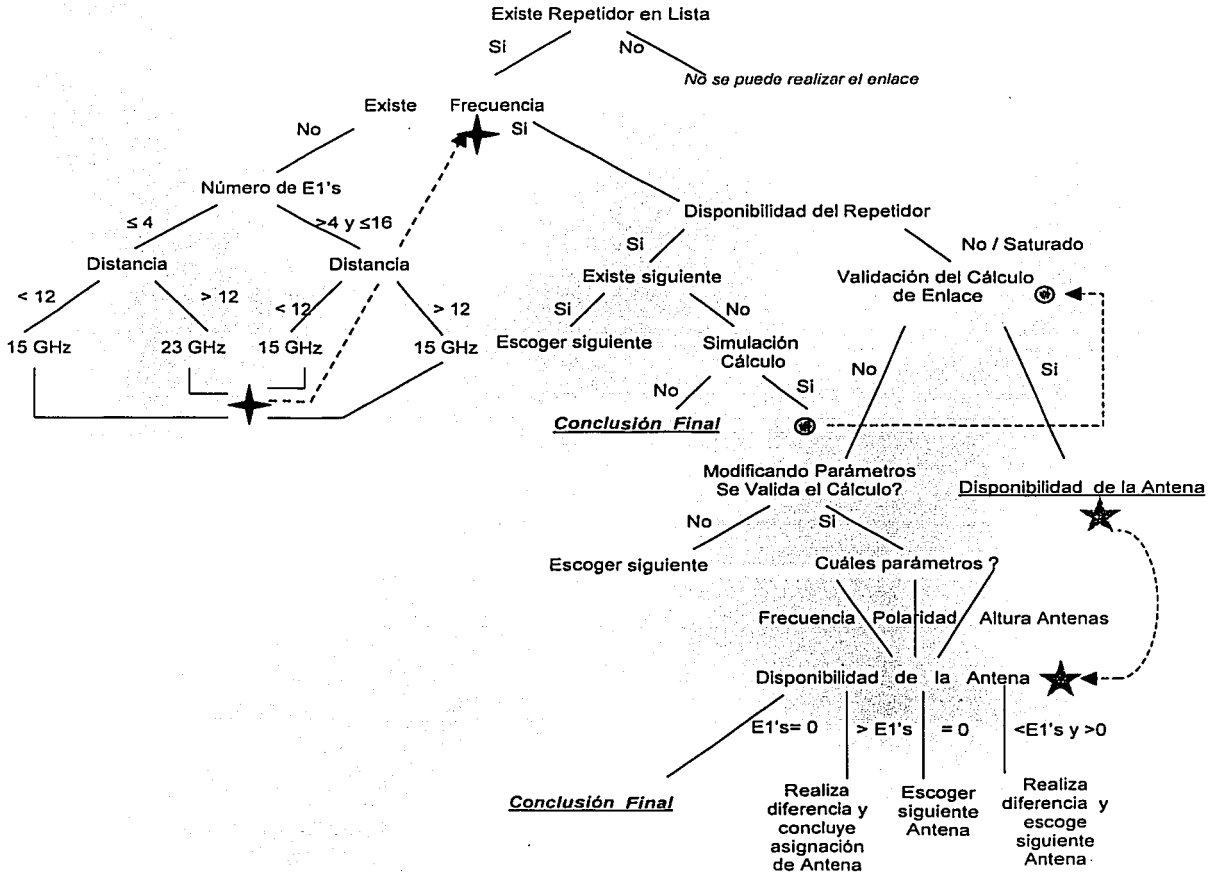
Figura 4.5.1

4.6 ÁRBOL DE DECISIÓN

El árbol proporciona una visión gráfica de las variables que se prueban, las decisiones que se toman y el orden en que se realiza esta toma de decisiones. El árbol de decisiones hace que los analistas de sistemas identifiquen las decisiones reales que deben tomarse, y de esta manera reduce la probabilidad de que se pasen por alto decisiones críticas. Asimismo, el árbol de decisiones obliga a los analistas de sistemas a considerar la secuencia de decisiones.

Los árboles de decisión se usan en los sistemas expertos por que son mas precisos que el hombre para poder desarrollar un diagnóstico con respecto a algo, ya que el hombre puede dejar pasar sin querer un detalle, en cambio mediante un sistema experto con un árbol de decisión se puede dar un resultado exacto.

Por lo anterior, el árbol de decisión que dará secuencia de las condiciones y decisiones que tomará el sistema experto para analizar la factibilidad de un enlace de microondas punto a punto, mediante el shell CLIPS, se plantea a continuación:



TESIS CON
 FALLA DE ORIGEN

Es importante señalar que CLIPS es llamado, en el desarrollo de este sistema EMPP, por el lenguaje Visual Basic. Con el objetivo de que sea éste shell el que desarrolle la función de dar la deducción lógica de conclusiones al análisis del sistema EMPP. Para después retomarla como datos de entrada al lenguaje que lo llamó, regresándole así el control a Visual Basic.

4.7 DISEÑO

El objetivo del diseño estructurado es la creación de un sistema, el cual esté formado por la interacción de módulos independientes entre sí, esto hace que se facilite el mantenimiento a dichos módulos. Además, el diseño estructurado muestra gráficamente todos los módulos a desarrollar.

4.7.1 CARTA ESTRUCTURADA

La carta de estructura generalmente presenta una organización de tipo jerárquico, la cual permite que el sistema se defina en unidades más pequeñas y manejables, además de proporcionar una definición más clara de la relación entre los diferentes elementos del sistema.

Dentro de la organización jerárquica que guarda la carta de estructura, se cuenta con procesos concurrentes que se pueden ejecutar en forma paralela y que se comunican a través de variables comunes entre ellos. Además, esta carta proporciona las interfaces de comunicación entre los módulos.

La carta estructurada para el sistema EMPP es la mostrada en la Figura 4.7.1.1 :

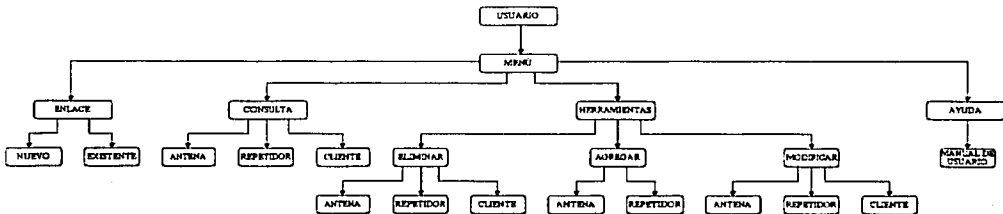


Figura 4.7.1.1

4.7.2 MÓDULOS DE LA CARTA DE ESTRUCTURA

❖ Módulo de Enlace

Este módulo se divide a su vez en dos submódulos: Nuevo y Existente

Dentro del submódulo de Nuevo se presentan al usuario las siguientes interfaces:

- *FrmNuevo*. Aquí el usuario captura los datos personales del cliente así como los datos requeridos para la verificación del enlace
- *FrmSeleccRepet*. Se presenta la lista de los repetidores existentes de los cuales el usuario podrá elegir aquellos donde la línea de vista es franca.
- *FrmMejorOpcionR*. El sistema sugiere la mejor opción de entre los repetidores seleccionados, mostrando los parámetros de frecuencia, polaridad y distancia, y presenta la opción de ver el resumen de ellos con el fin de que el usuario verifique la sugerencia hecha por el sistema EMPP.
- *FrmResumenRepet* Presenta la lista de todos los repetidores seleccionados, donde aparecen ordenados por distancia y mostrando la causa asignada en base a su disponibilidad.
- *FrmCalc_Enlace*. El sistema EMPP pide la aprobación del cálculo de enlace para la sugerencia presentada, junto con los siguientes parámetros: Nombre de los sitios a enlazar, coordenadas geográficas de ambos sitios, Elevación sobre el nivel del mar de ambos sitios y la frecuencia y polaridad recomendada.
- *FrmCalc_Enlace2*. Si no se aprobó el cálculo de enlace, aparece esta interfase en donde se le pregunta al usuario si es posible aprobar dicho cálculo modificando los parámetros de Frecuencia, polaridad y altura de antenas.
- *FramAdvertencia*. Esta interfase se despliega, cuando no se ha encontrado ningún repetidor con la disponibilidad suficiente para darte servicio a la solicitud de líneas pedidas. Presenta la opción de continuar con el estudio de cálculo de enlace pero señalando como mejor opción al primer repetidor que tenga menor distancia.
- *FrmConclusionPre*. Presenta al usuario la factibilidad aprobada para realizar el enlace y sugiere también la distribución y conexión de los E1's en las antenas del repetidor asignado.
- *FrmResumenAnt*. Presenta la lista de todas las antenas instaladas en el repetidor asignado, mostrando la causa asignada en base a su disponibilidad.
- *FrmConclusion*. Esta pantalla muestra el diagnostico final del sistema EMPP y da la opción de guardar el estudio realizado, así como la posibilidad de regresar a la pantalla principal.

Dentro del submódulo de Existente se presentan al usuario las siguientes interfaces:

- *FrmExistente*. Esta interfase pide la selección del cliente al cual se le añadirán mas E1's
- *FrmConclusionPre*, *FrmResumenAnt*, *FrmConclusion*. Estas interfaces son exactamente iguales a las descritas por el submodulo de Nuevo.

❖ Módulo de Consulta

Para este módulo se presenta solo una interfase llamada *frmConsultas*, la cual contiene tres frames: *fraAntenas*, *fraRepetidores* y *fraClientes*. Cada uno será desplegado al usuario, siempre y

cuando éste indique de que elemento desea consultar sus características generales, ya sea de Antena, Repetidor o Cliente.

❖ Módulo de Herramientas

Muestra tres interfaces diferentes, *frmAgregar*, *frmModificar*, *frmEliminar*, para Agregar, Modificar y Eliminar, tanto Clientes, Antenas y Repetidores. Con este módulo se pretende dar mantenimiento al sistema EMPP, actualizando la base de datos y conservando la información del sistema EMPP en orden.

4.8 CODIFICACIÓN

Descomponer un programa en términos de recursos abstractos consiste en dividir una determinada acción compleja en términos de un número de acciones más simples capaces de ejecutarlas o que constituyan instrucciones de computadoras disponibles.

Bajo este concepto se desarrolló la codificación del sistema EMPP, puesto que tratándose de un sistema experto basado en CLIPS, la idea principal fue representar en reglas simples y concisas las acciones necesarias y representativas que llevan a cabo la factibilidad de un enlace de microondas punto a punto.

La codificación en Visual Basic que nos permite tener la comunicación con CLIPS, para la causa de disponibilidad "aprobado", es la siguiente:

```

CLIPS.Clear
retval = CLIPS.Load(App.Path & "\tesism1.clp")
CLIPS.Reset
n = frmMemoria.lstDistancia.ListCount

For i = 0 To n - 1
    fact = "(distancia " & frmMemoria.lstDistancia.List(i) & ")"
    fact2 = "(num_e1 " & frmMemoria.txtNum_E1 & ")"
    fact3 = "(disp_rep " & frmMemoria.lstE1_Dispatch.List(i) & ")"

    retval = CLIPS.AssertString(fact)
    retval = CLIPS.AssertString(fact2)
    retval = CLIPS.AssertString(fact3)

    retval2 = CLIPS.Run(-1)

    solucion2 = CLIPS.GetDefglobalString("diferencia")
    solucion3 = CLIPS.GetDefglobalString("causa")

    frmMemoria.lstDif_E1.AddItem solucion2
    frmMemoria.lstCausa_Rep.AddItem solucion3

Next i

```

Las reglas que se programaron en CLIPS con el fin de determinar el caso en donde la causa de disponibilidad del repetidor fuese "aprobado", se muestran a continuación:

```

(defglobal
  ?*causa* = "")
(defglobal
  ?*bandera* = 0)
(defglobal
  ?*diferencia* = 0)

(deftemplate datos
  (slot num_e1 (type NUMBER))
  (slot distancia (type NUMBER))
  (slot disp_rep (type NUMBER))
  (slot diferencia (type NUMBER))
  (slot bandera (type NUMBER))
  (slot disp_ant(type NUMBER))
  )

(deffacts inicia (datos))

(defrule dif_Rep
  ?A <- (num_e1 ?lineas)
  ?B <- (disp_rep ?dispon)
  =>
  (bind ?*diferencia* (- ?dispon ?lineas))
  (assert (diferencia ?*diferencia*))
  )

(defrule causaC
  ?A <- (diferencia ?dif)
  (test (> ?dif 5))
  =>
  (bind ?*causa* "APROBADO")
  (bind ?*bandera* 3)
  (retract ?A)
  )

```

Siguiendo esta misma lógica y estructura se codifican las reglas necesarias para evaluar las prioridades y factibilidad del enlace de microondas. Estas prioridades se expusieron en la sección 4.4 del presente capítulo.

5. UTILIZACIÓN DEL SISTEMA EMPP

5.1 INTRODUCCIÓN

En este capítulo veremos un ejemplo cotidiano para la evaluación de un enlace microonda punto a punto utilizando el sistema EMPP. Se presenta la simulación del sistema EMPP y una breve explicación sobre la interacción que el sistema tiene con el usuario.

5.2 DESCRIPCIÓN DEL PROBLEMA

Se requiere realizar un enlace de microonda punto a punto con un nuevo cliente que solicita la instalación de 5E1's y cuyos datos aparecen en el siguiente reporte (Figura 5.2.1).

E M P P			
REPORTE PARA EL SISTEMA EMPP			
DETALLES DEL SITIO			
Nombre del Sitio:	<input type="text" value="FASNET"/>	Latitud:	<input type="text" value="N 19° 23' 39.2"/>
		Longitud:	<input type="text" value="O 99° 08' 34.4"/>
		Elevación:	<input type="text" value="2220 m"/>
Dirección:	<input type="text" value="Horacio No. 325-A, Col. Polanco"/>		
Ciudad:	<input type="text" value="México"/>	Estado:	<input type="text" value="D.F."/>
Teléfono:	<input type="text" value="56980225"/>		
Número de E1's requerido:	<input type="text" value="5"/>		

Figura 5.2.1 - Reporte inicial donde aparecen los datos del cliente

De este reporte podemos obtener los datos generales del cliente, la ubicación geográfica, el número de E1's que se desean instalar y la verificación visual del repetidor o de los repetidores

candidato, así como la propuesta de instalación para tener una línea de vista franca hacia dichos repetidores, es decir; la propuesta de instalar una torre para tener línea de vista franca hacia uno o varios repetidores.

5.3 INTERACCIÓN DEL SISTEMA EMPP CON EL USUARIO

La figura 5.3.1 muestra la pantalla principal del sistema EMPP en donde se visualiza el menú general con sus diferentes opciones:

- **Enlace:** Ésta opción permite la realización de un nuevo enlace, la instalación de nuevos E1's para un cliente existente y la terminación del programa.
- **Consultas:** Ésta opción permite realizar consultas sobre las distintas características de las antenas instaladas en cualquier repetidor, así como realizar consultas sobre los clientes existentes y consultas sobre los repetidores actuales.
- **Herramientas:** Ésta opción permite la adición de nuevas antenas en el repetidor seleccionado así como la adición de nuevos repetidores. También con esta opción se pueden realizar modificaciones o eliminar una antena, un repetidor o un cliente sobre la base de datos.

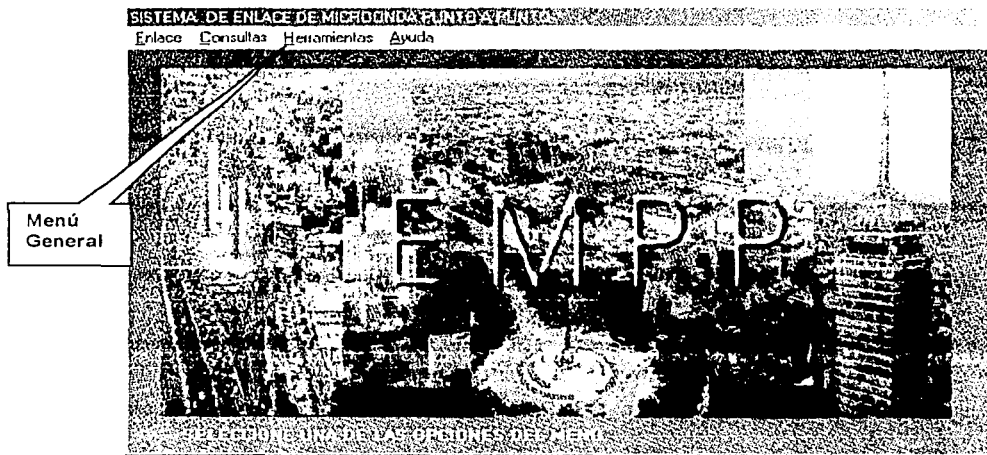


Figura 5.3.1 - Menú General

El submenú de Enlace contiene las tres opciones comentadas anteriormente. (Figura 5.3.2)



Figura 5.3.2 - Submenú de Enlace

Al ejecutar la opción de Enlace "Nuevo", el sistema EMPP presenta la plantilla para ser llenada con algunos de los datos que aparecen en el formato inicial (Figura 5.2.1). Ésta plantilla contiene la opción de regresar al menú general y la de continuar con el Análisis (Figura 5.3.3).

DATOS DEL CLIENTE

DETALLES DEL SITIO

NOMBRE DEL SITIO: FASNET

CALLE Y NÚMERO: HOFACIO No. 325-A

COLONIA: POLANCO C.P.: 15500

CIUDAD: Mexico TELEFONO: 56980225

ESTADO: D.F.

LATITUD: 19°23'39.2" Insertar

LONGITUD: 99°08'34.4" Insertar

ELEVACION: 2220

EN SU ORDENADOR: 9

Menú Siguiente

Figura 5.3.3 - Plantilla de datos del nuevo cliente

El siguiente paso para el sistema EMPP es la solicitud para seleccionar los repetidores candidato propuestos por el formato inicial. La figura 5.3.4_A muestra la manera en que el sistema EMPP solicita al Ingeniero a cargo la selección de los repetidores donde sin utilizar una torre la línea de vista es franca.

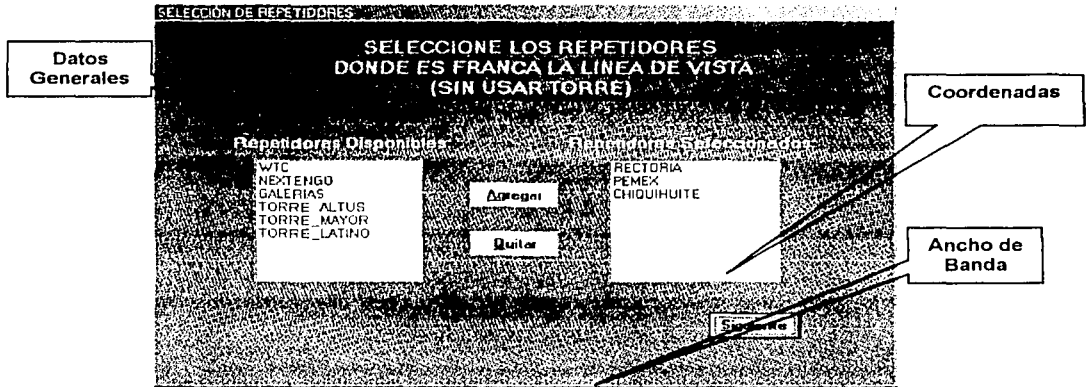


Figura 5.3.4_A

Mientras que la figura 5.3.4_B muestra la solicitud de selección de los repetidores donde SI es necesario una torre para tener línea de vista franca.

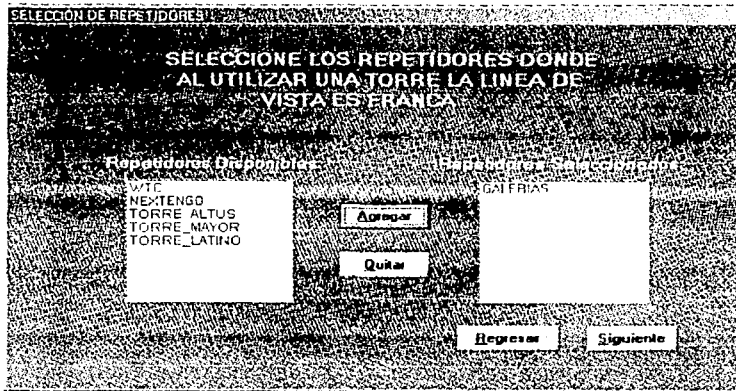


Figura 5.3.4_B

Con la introducción de todos los repetidores candidato, el sistema EMPP es capaz de realizar la evaluación en base a los criterios y prioridades expuestos en el capítulo anterior (Distancia y Disponibilidad del repetidor).

Dicha evaluación contiene el nombre del repetidor y la distancia que hay entre éste punto y el cliente. El sistema EMPP al visualizar la evaluación, permite la opción de regresar al menú principal (con su respectiva confirmación posteriormente), la de continuar con el análisis y la de visualizar un resumen de las evaluaciones que el sistema hizo con respecto a cada repetidor candidato. (Figura 5.3.5).

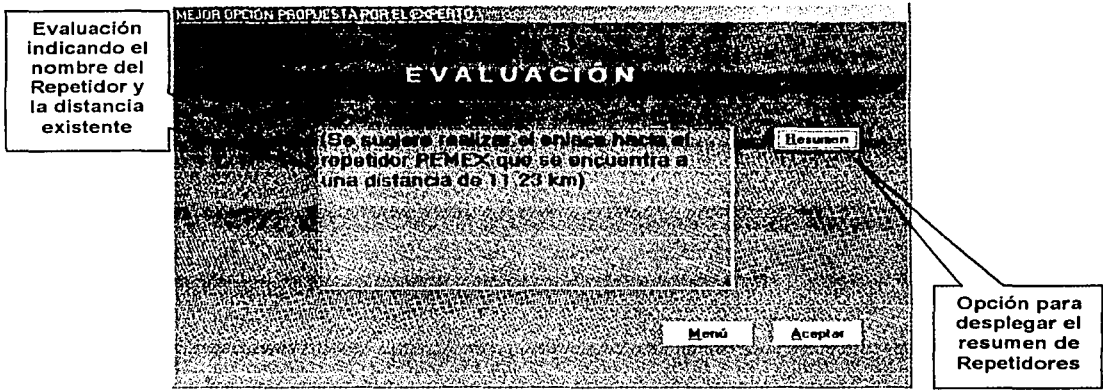


Figura 5.3.5 - Evaluación preliminar

La opción de resumen presenta una lista ordenada de los repetidores candidato respecto a la distancia que existe hacia el cliente y contiene además la evaluación que el sistema EMPP hizo de acuerdo a la disponibilidad del repetidor (Figura 5.3.6).



Figura 5.3.6 - Resumen de los repetidores candidatos

En éste resumen podemos observar que aunque el repetidor Galerias tiene la mínima distancia respecto al cliente, sus antenas quedarían saturadas si se llegaran a instalar los 5 E1's solicitados por el cliente, por lo que el sistema EMPP no aprueba esta evaluación ("Saturado") por el momento, y analiza el siguiente repetidor. El siguiente repetidor de la lista es Rectoría y se puede observar que "No hay suficientes E1's para instalar", por lo que el sistema EMPP tampoco aprueba ésta opción y analiza el siguiente repetidor candidato. El siguiente repetidor en esta ocasión es Pemex y se observa que su evaluación si es "Aprobada" por el sistema. Éste

repetidor junto con la distancia hacia el cliente, aparecen en la evaluación del sistema EMPP (Figura 5.3.5).

Al continuar con el análisis, el sistema EMPP presenta en pantalla la mejor propuesta para realizar el cálculo de enlace. *El cálculo de enlace tendrá que ser evaluado con un software adicional específico para dicho cálculo.*

Esta propuesta contiene los parámetros básicos que el Ingeniero a cargo debe de tomar en cuenta al utilizar el software adicional. Dicha propuesta solicita la validación del cálculo de enlace para continuar trabajando (Figura 5.3.7).



Figura 5.3.7 - Propuesta para realizar el cálculo de enlace

Para éste ejemplo, al utilizar el software adicional y evaluar el cálculo de enlace introduciendo los datos propuestos por el sistema EMPP, se observó que **NO** fue válido. Seleccionando ésta opción, el sistema EMPP pregunta ahora que si cambiando alguno de los datos propuestos originalmente (frecuencia, altura de antenas o polaridad), el cálculo de enlace puede validarse afirmativamente. (Figura 5.3.8)

TESIS CON
FALLA DE ORIGEN

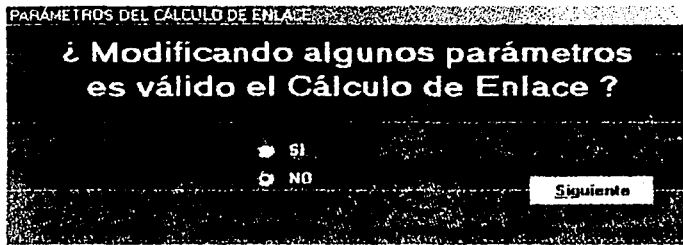


Figura 5.3.8 - Petición para validar el cálculo de enlace

Utilizando nuevamente el software adicional y cambiando los parámetros sugeridos, se concluye que el cálculo de enlace tampoco es válido, por lo que al seleccionar esta opción, el sistema EMPP vuelve a mostrar la pantalla de evaluación pero ahora con el nuevo nombre del repetidor para realizar el enlace (Figura 5.3.9).

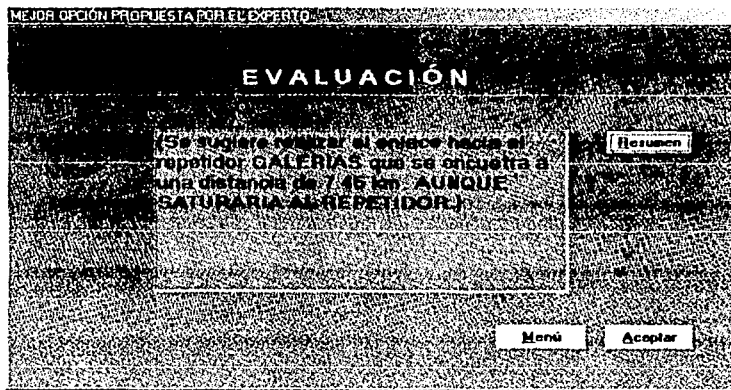


Figura 5.3.9 - Evaluación preliminar actualizada.

Continuando con el análisis, vuelve a aparecer la validación del cálculo de enlace así como los parámetros sugeridos para la siguiente mejor opción hecha por el sistema EMPP. (Figura 5.3.10)

TESIS CON
FALLA DE ORIGEN



Figura 5.3.10 - Segunda propuesta para validar el cálculo de enlace

Nuevamente el cálculo de enlace no fue valido utilizando los parámetros sugeridos y el sistema EMPP cuestiona si dicho cálculo puede validarse cambiando los parámetros (Figura 5.3.8).

En esta ocasión, el cálculo de enlace SI es valido al cambiar el parámetro de "Frecuencia". Al conocer la respuesta, el sistema EMPP solicita la información acerca de cual o cuales fueron los parámetros cambiados para validar satisfactoriamente el cálculo de enlace (Figura 5.3.11).

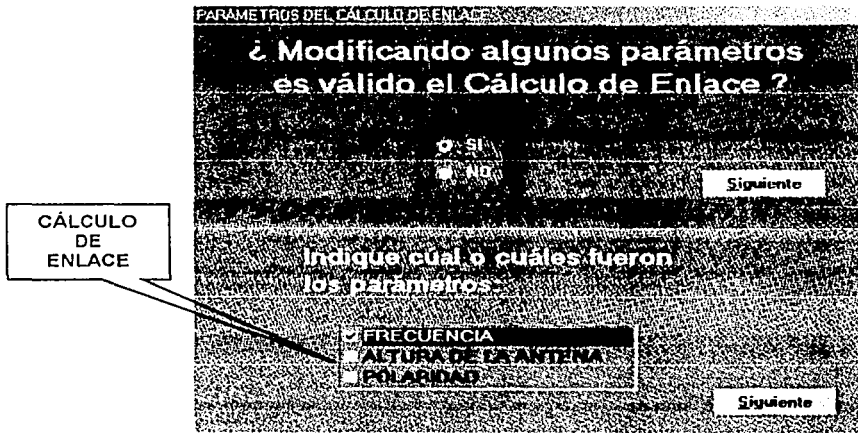


Figura 5.3.11 - Selección de parámetros modificados

TESIS CON FALLA DE ORIGEN

Una vez seleccionado el parámetro que se modificó y continuando con el análisis, el sistema EMPP visualiza la recomendación de la forma de instalación para los E1's solicitados en las diferentes antenas que se encuentran actualmente en el repetidor asignado (Figura 5.3.12)

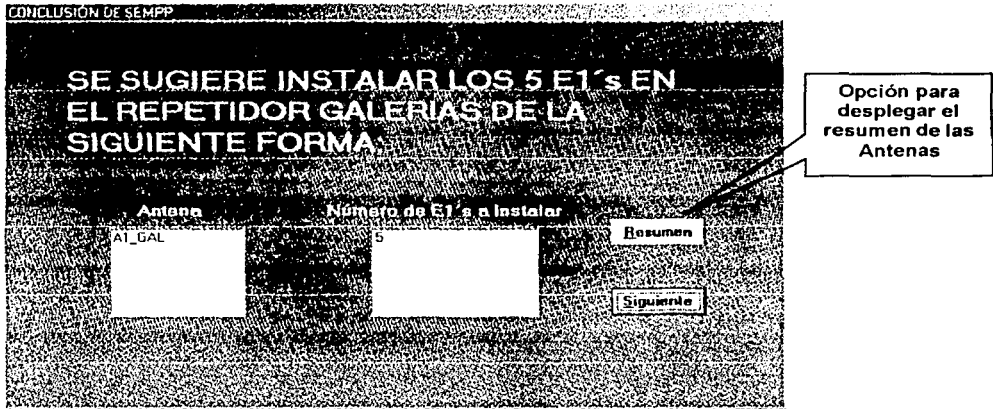


Figura 5.3.12 - Sugerencia para instalar los E1's en las antenas.

Esta recomendación presenta también una opción para desplegar un listado de las antenas instaladas en el repetidor propuesto, así como la evaluación que el sistema EMPP efectuó para cada antena de acuerdo al criterio de disponibilidad (Figura 5.3.13).



Figura 5.3.13 - Resumen de las antenas instaladas en el repetidor

Finalmente, el sistema EMPP concluye y despliega un mensaje donde aparece el nombre del repetidor seleccionado, la distancia del enlace, la frecuencia y la polaridad a utilizar para realizar el enlace y algunas características de instalación para ser tomadas en cuenta (Figura 5.3.14).

TESIS CON
FALLA DE ORIGEN

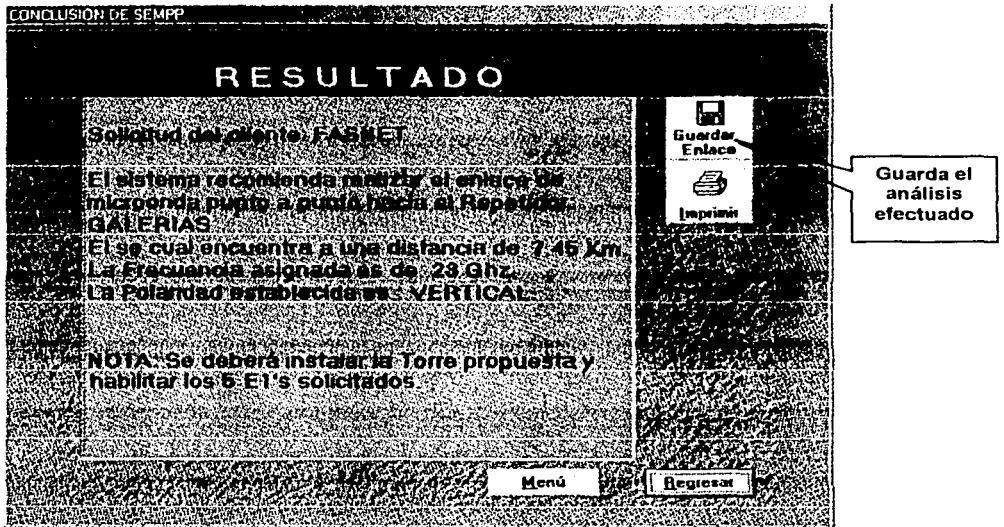


Figura 5.3.14 - Conclusión Final

Junto con el repetidor final, el sistema EMPP presenta la opción para guardar la consulta y de ésta forma, todo el análisis queda almacenado en la base de datos. Esto significa que al guardar el enlace se actualiza la capacidad disponible para un siguiente enlace; información que se toma en cuenta la próxima vez que se consulte el sistema

TESIS CON
 FALLA DE ORIGEN

5.4 COMENTARIO FINAL

Como se ve en este ejemplo, en la interacción del sistema EMPP con el usuario:

- Se presenta una interfase muy amigable y de cómodo acceso para los operadores, ya que la mayoría de las ventanas son muy claras en cuanto al paso siguiente que hay seguir para continuar evaluando el enlace.
- Se entiende en que consiste la búsqueda de la solución, puesto que el sistema EMPP presenta al operador el diagnóstico sobre la evaluación en cada repetidor candidato.
- El resultado final que se muestra contiene todos los elementos relevantes de la solución
- En el análisis del sistema se previó la posibilidad de mantener actualizada la base de datos por ser vital para su implementación.

En el ejemplo anterior, se muestran las capacidades que el sistema EMPP posee al permitir la aprobación del cálculo de enlace cuando la evaluación para cada repetidor es "Aprobado" o "Saturado". Sin embargo, el sistema EMPP tiene también la posibilidad de que en caso de no haber encontrado la solución óptima con algún repetidor candidato con diagnóstico "Aprobado o Saturado", el sistema evaluará a los repetidores que existan con diagnóstico "No hay suficientes E1's para el cliente" y permitirá realizar la simulación del cálculo de enlace, concluyendo en caso de ser aprobado dicho cálculo que la mejor solución corresponde a ese repetidor pero que por el momento no se cuenta con la disponibilidad necesaria para realizar el enlace y sugiere que se retrase la fecha de entrega hasta que se habiliten más E1's en ese repetidor.

En el caso de que el cálculo de enlace no haya sido aprobado para ningún repetidor candidato, el sistema EMPP concluirá que no se puede realizar el enlace con la confiabilidad requerida debido a posibles factores que impiden la adecuada emisión y recepción de la señal en el medio de transporte.

TESIS CON
FALLA DE ORIGEN

CONCLUSIONES

El sistema EMPP sirve como una buena herramienta de apoyo para las empresas de telecomunicaciones, debido a que puede responder ante situaciones distintas de manera similar a la del Ingeniero Experto.

El sistema EMPP cuenta con el criterio y la capacidad para tomar la mejor decisión que permita realizar el enlace de microonda punto a punto de entre varias posibles rutas existentes, gracias a que toma en cuenta parámetros combinados de la red de empresas de telecomunicaciones (disponibilidad de los repetidores) y de los datos naturales del enlace (número de E1's, distancia).

Debido a la toma de decisión que el sistema EMPP realiza, se refleja una considerable reducción en los costos del levantamiento del estudio de línea de vista, al considerar únicamente el sugerido por el sistema y no por todas las posibles opciones para realizar el enlace.

De igual forma, se refleja una reducción en tiempo para instalar y poner en operación el nuevo enlace hacia el cliente.

Con la administración y toma de decisión que el sistema EMPP efectúa en la asignación de los E1's a instalar, en la(s) antenas(s) existente(s) en cada repetidor, se logra un mejor control dentro del cuarto de control (switch) y se obtiene un mayor rendimiento para el mantenimiento del cableado y equipo dentro de la empresa de telecomunicaciones.

La estructura con la que el sistema EMPP esta diseñada, permite la fácil ampliación y adaptación de nuevos parámetros que permiten la toma de decisión de una manera mas completa. Esto es posible gracias a que se maneja una herramienta para la construcción de los sistemas expertos (CLIPS V.6.2) de uso libre y gratuito que cuenta con un mecanismo de inferencia en reglas y objetos, los cuales permiten integrarse con otros lenguajes de programación que realicen los cálculos matemáticos y retomen el control y manejo del conocimiento del sistema experto

El sistema EMPP muestra información general de una forma segura y confiable, ya que al ofrecer la opción de ser almacenados en una base de datos hace mas sencillo el medio para consultarlos, aumentando así la productividad del departamento al conservar su valiosa información y hacer que su difusión sea mas sencilla.

En general el sistema EMPP es confiable ya que muestra la capacidad para explicar el por qué de sus conclusiones. Se basa principalmente en las experiencias del Ingeniero Experto y sobre todo, reduce las decisiones erróneas y acelera los procesos de toma de decisión.

TESIS CON
FALLA DE ORIGEN

BIBLIOGRAFÍA

1. David W. Roston, "Principios de Inteligencia Artificial y Sistemas Expertos", 1990.
2. Chatain, Dussauchoy, "Sistemas Expertos. Métodos y Herramientas", 1988.
3. Sánchez, Beltrán, "Sistemas Expertos: Una Metodología de Programación", 1990.
4. "Introducción a los Sistemas Expertos: Fundamentos de la IA.", Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Alicante, España.
5. Iglesias Fernández Carlos Ángel, "Sistemas Basados en Conocimientos" e "Introducción a CLIPS", Universidad de Navarra, España, 1999.
6. Prof. Luis Daniel Hernández Molinero, "Tutorial de CLIPS", Departamento de Informática, Inteligencia Artificial y Electrónica, Fac. Informática, Universidad de Murcia, España. 2002.
7. Joseph Giarratano, "Expert Systems: Principles and Programming", 1998.
8. Alcatel University México, Diplomado en Telecomunicaciones, Módulo I (Introducción a las Telecomunicaciones), Módulo II (Medios de Transporte), Módulo III (Transporte), 2002.
9. Constantino Carlos Reyes. "Apuntes de Microondas, Satélites y Antenas", ITAM.
10. Pressman Roger, "Ingeniería de software: Un enfoque práctico", México, 1993.
11. Fairley Richard, "Ingeniería de Software", México , 1988
12. <http://www.ghgcorp.com/clips/CLIPS.html> - Página Oficial de CLIPS
13. <http://www.ghg.net/clips/download/documentation/> - Manual Oficial de CLIPS
14. <http://mail.udlap.mx/> - Tesis Electrónicas de La Universidad de Puebla

TESIS CON
FALLA DE ORIGEN

APÉNDICE A

; Programa que Permite hacer el cálculo de la
 ; Factibilidad para un Enlace Vía Microonda
 ; A partir del Estudio De línea de Vista.

;El siguiente grupo de variables globales son las que se utilizan para la comunicación ente Visual
 ;Basic y CLIPS.

```
(defglobal
  ?*frecuencia* = 0)
(defglobal
  ?*reanalisis* = 0)
(defglobal
  ?*solucion* = "")
(defglobal
  ?*causa* = "")
(defglobal
  ?*bandera* = 0)
(defglobal
  ?*diferencia* = 0)
(defglobal
  ?*antena* = "")
(defglobal
  ?*causa_ant* = "")
(defglobal
  ?*antena_filtro* = 2)
(defglobal
  ?*numeroe1* = 0)
(defglobal
  ?*salir* = 0)
```

```
(defglobal
  ?*conclusion1* = 0)
(defglobal
  ?*filtro3* = 0)
(defglobal
  ?*filtro2* = 0)
(defglobal
  ?*filtro1* = 0)
```

;Las plantillas que a continuación se construyen sirven para almacenar los datos que utilizarán
 ;las reglas para determinar el valor de nuestras variables globales.

```
(deftemplate datos
  (slot num_e1 (type NUMBER))
  (slot distancia (type NUMBER))
  (slot disp_rep (type NUMBER))
```

TESIS CON
FALLA DE ORIGEN

```
(slot diferencia (type NUMBER))
(slot bandera (type NUMBER))
(slot disp_ant(type NUMBER))
)
```

```
(deftemplate repetidores
  (slot filtro1 (type NUMBER))
  (slot filtro2 (type NUMBER))
  (slot filtro3 (type NUMBER))
  (slot nombre_rep (type SYMBOL))
  (slot distancia (type NUMBER))
  (slot filtro (type NUMBER))
)
```

;Inicializamos la plantilla de datos para poner todas las variables a cero.

```
(defacts inicia (datos))
```

;Las siguientes 4 reglas devuelven el valor de la frecuencia a utilizar en la variable global ;frecuencia.

```
(defrule frec_15AGHz
  ?A <- (num_e1 ?lineas)
  ?B <- (distancia ?dist)
  (test (<= ?lineas 4))

  (test (< ?dist 12))
  =>
  (bind ?**frecuencia* 15)
  (retract ?A)
  (retract ?B)
)
```

```
(defrule frec_23AGHz
  ?A <- (num_e1 ?lineas)
  ?B <- (distancia ?dist)
  (test (> ?lineas 4))
  (test (< ?dist 12))
  =>
  (bind ?**frecuencia* 23)
  (retract ?A)
  (retract ?B)
)
```

```
(defrule frec_15BGHz
  ?A <- (num_e1 ?lineas)
  ?B <- (distancia ?dist)
  (test (<= ?lineas 4))
  (test (> ?dist 12))
  =>
  (bind ?**frecuencia* 15)
  (retract ?A)
)
```

<p>TESIS CON FALLA DE ORIGEN</p>

```

(retract ?B)
)

(defrule frec_23BGHz
  ?A <- (num_e1 ?lineas)
  ?B <- (distancia ?dist)
  (test (> ?lineas 4))
  (test (> ?dist 12))
=>
  (bind ?*frecuencia* 23)
  (retract ?A)
  (retract ?B)
)

```

;La siguiente regla regresa el valor de la disponibilidad del repetidor que será utilizada en las siguientes reglas.

```

(defrule dif_Rep
  ?A <- (num_e1 ?lineas)
  ?B <- (disp_rep ?dispon)
=>
  (bind ?*diferencia* (- ?dispon ?lineas))
  (assert (diferencia ?*diferencia*))
)

```

;Las siguientes tres reglas se ocupan para saber si en el repetidor existen suficientes E1's para realizar el enlace o para ver si los E1's solicitados saturarán al repetidor.

```

(defrule causaA
  ?A <- (diferencia ?dif)
  (test (< ?dif 0))
=>
  (bind ?*causa* "NO HAY SUFICIENTES E1'S PARA EL CLIENTE")
  (bind ?*bandera* 1)
  (retract ?A)
)

```

```

(defrule causaB
  ?A <- (diferencia ?dif)
  (test (<= ?dif 10))
=>
  (bind ?*causa* "LOS E1'S SATURARÁN EL REPETIDOR")
  (bind ?*bandera* 2)
  (retract ?A)
)

```

```

)
(defrule causaC
  ?A <- (diferencia ?dif)
  (test (> ?dif 10))
=>
  (bind ?*causa* "APROBADO")
  (bind ?*bandera* 3)
  (retract ?A)
)

```

<p style="text-align: center;">TESIS CON FALLA DE ORIGEN</p>
--

Las siguientes 4 reglas se utilizan para asignar en las antenas del repetidor seleccionado, los números de E1's por antena a instalar y también devuelven el resultado de la consulta a cada antena.

```
(defrule antenas1
  ?A <- (num_e1 ?lineas)
  (test (= ?lineas 0))
=>
  (bind ?*salir* = 10)
  (bind ?*numeroe1* ?lineas)
  (bind ?*antena* "Ya no hay e1's para instalar")
  (bind ?*antena_filtro* 1)
  (retract ?A)
)
```

```
(defrule antenas2
  ?A <- (disp_ant ?disant)
  ?B <- (num_e1 ?lineas)
  (test (= ?disant 0))
=>
  (bind ?*salir* = 10)
  (bind ?*antena* "no hay espacio en la antena")
  (bind ?*numeroe1* ?lineas)
  (bind ?*antena_filtro* 1)
  (retract ?A)
  (retract ?B)
)
```

```
(defrule antenas3
  ?A <- (num_e1 ?lineas)
  ?B <- (disp_ant ?disant)
  (test (<> ?lineas 0))
  (test (<> ?disant 0))
  (test (<= (- ?disant ?lineas) 0))
=>
  (bind ?*antena* ?disant)
  (bind ?*antena_filtro* 0)
  (bind ?lineas (- ?lineas ?disant))
  (bind ?*causa_ant* INSTALAR)
  (retract ?A)
  (retract ?B)
  (bind ?*numeroe1* ?lineas )
)
```

```
(defrule antenas4
  ?A <- (num_e1 ?lineas)
  ?B <- (disp_ant ?disant)
  (test (<> ?lineas 0))
  (test (<> ?disant 0))
  (test (> (- ?disant ?lineas) 0))
=>
  (bind ?*antena* ?lineas)
  (bind ?*antena_filtro* 0)
```

<p>TESIS CON FALLA DE ORIGEN</p>

```
(bind ?lineas 0)
(bind ?causa_ant* INSTALAR)
(bind ?*numeroe1* ?lineas )
(retract ?A)
(retract ?B)
)
```

;Las siguientes reglas sirven para saber cuantos de los repetidores candidatos hay de cada uno y ;utilizarlos después para generar la mejor solución y en caso de que el cálculo de enlace no sea ;valido, para tener el control de cual es el siguiente repetidor como mejor opción.

```
(defrule Aprobados
  ?A <- (repetidores (filtro1 ?filtro1)(filtro2 ?filtro2)(filtro3 ?filtro3)(nombre_rep ?nombre)(distancia
?dist)(filtro ?filtro))
  (test (> ?filtro3 0))
  (test (= ?filtro 3))
=>
  (retract ?A)
  (bind ?*filtro3* (- ?filtro3 1))
  (bind ?*filtro2* ?filtro2)
  (bind ?*filtro1* ?filtro1)
  (bind ?*conclusion1* Se sugiere realizar el enlace hacia el repetidor ?nombre que se encuentra
a una distancia de ?dist Km.)
)
```

```
(defrule Saturados
  ?A <- (repetidores (filtro1 ?filtro1)(filtro2 ?filtro2)(filtro3 ?filtro3)(nombre_rep ?nombre)(distancia
?dist)(filtro ?filtro))
  (test (= ?filtro3 0))
  (test (> ?filtro2 0))
  (test (= ?filtro 2))
=>
  (retract ?A)
  (bind ?*filtro3* ?filtro3)
  (bind ?*filtro2* (- ?filtro2 1))
  (bind ?*filtro1* ?filtro1)
  (bind ?*conclusion1* Se sugiere realizar el enlace hacia el repetidor ?nombre que se encuentra
a una distancia de ?dist Km. AUNQUE SATURARIA AL REPETIDOR)
)
```

```
(defrule NoAprobados
  ?A <- (repetidores (filtro1 ?filtro1)(filtro2 ?filtro2)(filtro3 ?filtro3)(nombre_rep ?nombre)(distancia
?dist)(filtro ?filtro))
  (test (= ?filtro3 0))
  (test (= ?filtro2 0))
  (test (> ?filtro1 0))
  (test (= ?filtro 1))
=>
  (bind ?*filtro3* ?filtro3)
  (bind ?*filtro2* ?filtro2)
  (bind ?*filtro1* (- ?filtro1 1))
  (bind ?*conclusion1* NO SE PUEDE REALIZAR EL ENLACE HACIA EL REPETIDOR ?nombre
DEBIDO A QUE NO HAY LINEAS DISPONIBLES PARA DARLE SERVICIO AL CLIENTE.)
)
```

TESIS CON
FALLA DE ORIGEN

```
(defrule Salir
```

```
  ?A <- (repetidores (filtro1 ?filtro1)(filtro2 ?filtro2)(filtro3 ?filtro3))
  (test (= ?filtro3 0))
  (test (= ?filtro2 0))
  (test (= ?filtro1 0))
=>
  (bind ?*salir* 1)
)
```

```
(defrule Mantener
```

```
  ?A <- (repetidores (filtro1 ?filtro1)(filtro2 ?filtro2)(filtro3 ?filtro3)(filtro ?filtro))
=>
  (bind ?*filtro3* ?filtro3)
  (bind ?*filtro2* ?filtro2)
  (bind ?*filtro1* ?filtro1)
)
```

TELECOM
FALLA DE ORIGEN