

**UNIVERSIDAD DE EL SALVADOR
FACULTAD DE CIENCIAS NATURALES Y MATEMATICA
ESCUELA DE MATEMATICA**



**“SISTEMA EXPERTO PARA LA IDENTIFICACIÓN DE
ANÉLIDOS MARINOS CON ÉNFASIS EN LA FAMILIA DE
LOS ONÚFIDOS (*Polychaeta onuphidae*) DE LA COSTA DE
EL SALVADOR”**

TRABAJO DE GRADUACION PRESENTADO POR:

**JOSE LUIS NUILA MORENO
MAURICIO ERNESTO ALFARO PARADA**

**PARA OPTAR AL GRADO DE
MAESTRO EN COMPUTACION**

Ciudad Universitaria, Septiembre de 2006

San Salvador,

El Salvador,

Centro América

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE CIENCIAS NATURALES Y MATEMATICA
ESCUELA DE MATEMATICA

TRABAJO DE GRADUACION

**“SISTEMA EXPERTO PARA LA IDENTIFICACIÓN DE
ANÉLIDOS MARINOS CON ÉNFASIS EN LA FAMILIA DE
LOS ONÚFIDOS (*Polychaeta onuphidae*) DE LA COSTA DE
EL SALVADOR”**

PRESENTADO POR:

JOSE LUIS NUILA MORENO
MAURICIO ERNESTO ALFARO PARADA

ASESORES:

M.Sc HUBERTO ALFONSO SERMEÑO VILLALTA
M.Sc FRANCISCO CHICAS

CIUDAD UNIVERSITARIA, SAN SALVADOR, SEPTIEMBRE DE 2006.

UNIVERSIDAD DE EL SALVADOR
AUTORIDADES UNIVERSITARIAS

RECTOR : DRA. MARÍA ISABEL RODRIGUEZ

SECRETARIA

GENERAL : LICDA. LIDIA MARGARITA MUÑOZ VELA

FACULTAD DE CIENCIAS NATURALES Y MATEMATICA

DECANO : M.Sc. JOSÉ HÉCTOR ELÍAS DÍAZ

SECRETARIO : LIC. VICTOR MANUEL DURAN BELLOSO

ESCUELA DE MATEMATICA

DIRECTOR : LIC. MAURICIO HERNÁN LOVO CÓRDOVA

SECRETARIO : LIC. HÉCTOR DUGLAS MOLINA MELARA

CIUDAD UNIVERSITARIA, SEPTIEMBRE DE 2006.

AGRADECIMIENTOS

En el proceso de mejoramiento de las actividades y desarrollo de la Universidad de El Salvador, se requiere de un esfuerzo permanente orientado al logro de una mayor eficiencia y eficacia del personal involucrado en el manejo de los recursos disponibles para atender a la población estudiantil y el desempeño de sus labores.

La falta de atención de actividades dirigidas hacia la enseñanza y preparación de los nuevos profesionales para el desempeño de las futuras actividades, despierta un interés muy importante para atender esta deficiencia.

Es por lo que el presente trabajo denominado: “**SISTEMA EXPERTO PARA LA IDENTIFICACIÓN DE ANÉLIDOS MARINOS CON ÉNFASIS EN LA FAMILIA DE LOS ONÚFIDOS (*Polychaeta onuphidae*) DE LA COSTA DE EL SALVADOR**”, se enmarca en el desarrollo de una primera herramienta orientada a ayudar al personal de la Facultad de Ciencias Naturales y Matemáticas, para el desempeño de sus actividades cotidianas.

Este trabajo no hubiese sido posible sin el apoyo decidido de M.Sc. Francisco Chicas y el experto Lic. Carlos Rivera, a quienes dirijo un agradecimiento muy especial por compartir con el grupo sus valiosos conocimientos e intereses para llevarlo a un término feliz. Sin olvidar al Dr. Javier Leach, catedrático de la Universidad Complutense de Madrid (España), quien impulsó el desarrollo de la Maestría en Computación en la Universidad de El Salvador.

De igual manera, agradecemos al personal de la Escuela de Matemáticas de la Universidad de El Salvador, por su valiosa colaboración y apoyo incondicional para la culminación de este trabajo, al Director Lic. Mauricio Hernán Lovo Cordova. En especial a M.Sc. Humberto Alfonso Sermeño Villalta y M.Sc. Francisco Chicas, asesores que compartieron con el grupo su valiosa experiencia.

Finalmente, agradecemos a nuestros familiares por el apoyo moral y solidaridad con la que siempre contamos.

**José Luis Nuila Moreno
Mauricio Ernesto Alfaro Parada**

INDICE

RESUMEN	vii
INTRODUCCION	1
1. GENERALIDADES	4
1.1. PLANTEAMIENTO DEL PROBLEMA	4
1.1.1. OBJETIVOS	4
1.1.1.1. OBJETIVOS GENERALES	4
1.1.1.2. OBJETIVOS ESPECIFICOS	4
1.2. LIMITACIONES DEL PROYECTO	5
1.3. JUSTIFICACION	6
2. FUNDAMENTO TEORICO DE LOS SISTEMAS EXPERTOS	7
2.1. DEFINICION DE SISTEMAS EXPERTOS	7
2.2. ESTRUCTURA DE UN SISTEMA EXPERTO	9
2.2.1. BASE DE CONOCIMIENTOS	11
2.2.1.1. Reglas de Producción	11
2.2.1.2. Redes Semánticas	12
2.2.1.3. Marcos	13
2.2.1.4. Cálculo de Predicados	16
2.2.2. MEMORIA DINAMICA O MEMORIA DE HECHOS	21
2.2.3. MOTOR DE INFERENCIA	21
2.2.3.1. Estrategias de Inferencia	22
2.2.3.2. Resolución de Conflictos	29
2.2.4. COMPONENTE EXPLICATIVO	30
2.2.5. INTERFASE DE USUARIO	31

3. METODOLOGIA DE LA INVESTIGACION	33
3.1. FASE 1 - ASESORIA	34
3.2. FASE 2 – ADQUISICIÓN DEL CONOCIMIENTO	36
3.3. FASE 3 - DISEÑO	37
3.4. FASE 4 - PRUEBA	39
3.5. FASE 5 - DOCUMENTACION	39
3.6. FASE 6 - MANTENIMIENTO	40
4. DISEÑO DEL SEFTWARE NIVEL9	41
4.1. ESTRUCTURAS DE DATOS	41
4.1.1. MAQUINA P O MAQUINA PILA	43
4.1.2. TABLA DE SIMBOLOS	44
4.1.3. TABLA DE DATOS	47
4.1.4. TABLA DE CADENAS	49
4.1.5. MEMORIA DE TRABAJO O BASE DE HECHOS	50
4.1.6. BASE DE CONOCIMIENTOS	53
4.1.7. TABLA DE PREMISAS	58
4.2. MOTOR DE INFERENCIA	62
4.3. LENGUAJE PARA CODIFICAR EL CONOCIMIENTO	68
4.3.1. FASES DEL INTERPRETE NIVEL9	69
4.3.1.1. Fase de Análisis	70
4.3.1.2. Fase de Síntesis	72
4.3.2. DEFINICION DEL LENGUAJE DE PROGRAMACION	78
4.3.2.1. AUTOMATAS	80
4.3.3. DESCRIPCION FORMAL DE LA ESTRUCTURA SINTACTICA DEL LENGUAJE	84
4.4. MODULO DE RECONOCIMIENTO DE TEXTO Y SINTETIZADOR DE VOZ	88
5. DISCUSION Y ANALISIS DE RESULTADOS	89

INDICE DE FIGURAS

FIGURA 1. Proceso de Resolución de Problemas de un experto Humano	10
FIGURA 2. Proceso de Resolución de Problemas de un Sistema Experto	10
FIGURA 3. Arquitectura de un Sistema de Producción	12
FIGURA 4. Ejemplo de Red Semántica	13
FIGURA 5. Ejemplo de Formato de Marco	14
FIGURA 6. Ejemplo de Componentes de Marco	14
FIGURA 7. Ejemplo de Herencia Jerárquica en el Concepto de Marco	15
FIGURA 8. Conclusión para la Hipótesis “Hans vive en Munich”	18
FIGURA 9. Encadenamiento hacia adelante Primer Paso	24
FIGURA 10. Encadenamiento hacia adelante Segundo Paso	24
FIGURA 11. Encadenamiento hacia adelante Tercer Paso	25
FIGURA 12. Encadenamiento hacia adelante Cuarto Paso	25
FIGURA 13. Encadenamiento hacia adelante Quinto Paso	26
FIGURA 14. Encadenamiento hacia atrás Primer Paso	27
FIGURA 15. Encadenamiento hacia atrás Segundo Paso	27
FIGURA 16. Encadenamiento hacia atrás Tercer Paso	28
FIGURA 17. Encadenamiento hacia atrás Cuarto Paso	28
FIGURA 18. Encadenamiento hacia atrás Quinto Paso	29
FIGURA 19. Fases del desarrollo de los Sistemas Expertos	35
FIGURA 20. Comportamiento de la Máquina P ante una operación aritmética	45
FIGURA 21. Relación entre Memoria de Trabajo con Tabla de Datos y Tabla de Cadenas	52

FIGURA 22. Comportamiento de las estructuras de Bases de Conocimiento y Memoria de Trabajo en el momento que se dispara una regla	58
FIGURA 23. Relación entre la Base de Conocimientos con respecto a la Memoria de Trabajo y la Tabla de Premisas.	61
FIGURA 24. Estrategia de inferencia encadenamiento hacia delante	65
FIGURA 25. Interacción entre las estructuras BC y MT durante el proceso de inferencia	67
FIGURA 26. Esquema del intérprete utilizado por NIVEL9.	69
FIGURA 27. Fases de un INTERPRETE/COMPILADOR.	70
FIGURA 28. Arbol sintáctico para representar: $20 * A + (B - 30)$	73
FIGURA 29. Autómata para NUMEROS	80
FIGURA 30. Autómata para COMENTARIO SIMPLE	81
FIGURA 31. Autómata para COMENTARIO COMPUESTO	81
FIGURA 32. Autómata para ESPACIO EN BLANCO	82
FIGURA 33. Autómata para SALTO DE LINEA	82
FIGURA 34. Autómata para CADENA DE CARACTERES	82
FIGURA 35. Autómata para CHARACTER ASCII	82
FIGURA 36. Autómata para SÍMBOLO NEGATIVO	83
FIGURA 37. Autómata para OPERADORES ARITMETICOS	83
FIGURA 38. Autómata para SÍMBOLOS SEPARADORES	83
FIGURA 39. Autómata para SÍMBOLO TERMINADOR	83
FIGURA 40. Autómata para OPERADORES 1	83
FIGURA 41. Autómata para OPERADORES 2	84
FIGURA 42. Autómata para OPERADOR ASIGNACIÓN	84

INDICE DE TABLAS

Tabla 1. Comparación entre un experto humano y un sistema extpero	3
Tabla 2. Ejemplo de Cálculo de Predicado	17
Tabla 3. Estructura de datos que almacena todos los símbolos y palabras claves utilizados en la gramática correspondiente al lenguaje creado para la herramienta NIVEL9.	47
Tabla 4. Ejemplo de Tabla de Datos	49
Tabla 5. Ejemplo de Tabla de Cadenas	50
Tabla 6. Estructura de Memoria de Trabajo	52
Tabla 7. Estructura de Base de Conocimientos	57
Tabla 8. Estructura de la Tabla de Premisas	60
Tabla 9. Porcentaje de Uso de los Métodos de Sistemas de representación del conocimiento más utilizados.	68
Tabla 10. Descripción formal del lenguaje léxico SIMBOLOS UTILIZADOS POR EL LENGUAJE	78
Tabla 11. Precedencia de los operadores del lenguaje NIVEL9	87

RESUMEN

Los Sistemas Expertos como un intento de generar un nuevo estilo de software reproducen el planteamiento de personas expertas en la solución de problemas. Esto implica que un programa de computadora dispone de metodologías con las cuales se puede contar en el futuro y solucionar problemas casi inteligentes.

Los científicos han demostrado a través de diferentes proyectos de investigación cómo puede estructurarse, almacenarse y extraerse el conocimiento según el tipo de problema a tratar (si por ello se debe hablar más de procesamiento de conocimiento, que de procesamiento de datos, es algo que se deja a juicio del lector).

No cabe duda de que al concentrarse en el comportamiento humano ante la solución de un problema específico, la programación ha experimentado nuevos impulsos. Se han desarrollado en los últimos años nuevos métodos y procedimientos de programación, y el presente proyecto utiliza de entre estos métodos, específicamente, los Sistemas Expertos.

Como introducción se presenta la temática de los sistemas expertos.

La sección 1 describe las generalidades del proyecto, dando una visión general sobre el planteamiento del problema, objetivos planteados, tanto generales como específicos. Así como las limitaciones que tiene este proyecto y las justificaciones que se plantearon para la ejecución de este.

La sección 2 describe en forma detallada el campo de investigación de donde proceden los sistemas expertos, sus antecedentes y en qué forma han sido incorporados en el campo de la investigación. Dando a conocer la definición de general de lo que son los Sistemas Expertos y las estructuras de la que están compuestas.

La sección 3 ofrece información sobre las diferentes fases que se siguieron para el desarrollo del presente proyecto.

En la sección 4 se amplía la fase de diseño presentada en la sección 3, analizando detalladamente los diferentes módulos creados para la gestión de las estructuras de datos, el motor de inferencia utilizado, el lenguaje para codificar el conocimiento y el módulo de reconocimiento de texto y sintetizador de voz.

En la sección 5 se presentan las pruebas realizadas en el laboratorio de ICMARES, con el fin de comprobar los objetivos planteados en la sección 1.

La sección 6 detalla las conclusiones y recomendaciones que se obtuvieron en el desarrollo de este proyecto.

Y para finalizar el documento se presenta la bibliografía utilizada y los apendices. Entre estos se encuentra la guía de instalación de NIVEL9, la guía de usuario, el manual del lenguaje de programación, y el código de las bases de conocimientos.

INTRODUCCION

Los anélidos poliquetos son un componente importante en las comunidades bénticas de todos los mares, su abundancia numérica, biomasa y número de especies tiene repercusiones de gran magnitud en la dinámica trófica (Salazar, 1990). De este grupo, destaca la Familia Onuphidae (Kinberg, 1865) con organismos comunes en todas las profundidades oceánicas y ambientes costeros, constituyendo la segunda familia del orden Eunicida mejor representada (Fauchald, 1977).

La identificación de esta familia representa una tarea demasiado compleja para los investigadores de esta rama, por lo que una herramienta informática se hace necesaria en los procesos concernientes a las investigaciones de nuevas especies.

En el campo de la inteligencia artificial, específicamente los sistemas expertos, se busca resolver problemas de control, diseño, diagnóstico, clasificación, instrucción, interpretación, monitoreo, planeación, predicción, selección y simulación. Por lo que en la problemática presente cabe perfectamente la implementación de los mismos.

Los sistemas expertos son recursos muy valiosos para las organizaciones. Dichos sistemas pueden ofrecer ideas creativas, resolver problemas complejos, o mejorar en forma eficiente el rendimiento de tareas rutinarias. Estas contribuciones pueden cambiar en forma positiva la productividad de las empresas, mejorando consecuentemente su posicionamiento en el mercado.

La mejor forma de justificar el uso de los sistemas expertos es comparándolos con un experto humano.

Como cualquier máquina un sistema experto puede continuar trabajando después de las ocho horas laborables. Como cualquier programa de computadora, fácilmente podemos duplicar el software y enviarlo a cualquier lugar geográfico donde se disponga

de una computadora o donde haya ausencia de talento humano. También se puede aprovechar la transportabilidad del software para lugares hostiles y peligrosos donde el ser humano podría correr serios riesgos.

Los expertos humanos no son eternos. En caso de muerte, retiro, o cambio de trabajo, una organización puede perder los talentos de un experto, al capturar los talentos y experiencia de los humanos en un sistema experto una organización retiene dicho conocimiento pudiendo seguir trabajando y dando soporte o entrenamiento a nuevo personal.

Un sistema experto produce resultados más consistentes que los expertos humanos. Los talentos humanos toman decisiones que pueden estar influenciadas por muchos factores que a su vez pueden afectar el desempeño de los mismos. Por ejemplo, problemas personales pueden preocupar a la persona, obteniendo de esta forma resultados menos productivos. En una situación de emergencia el experto humano puede olvidar importantes piezas de conocimiento por presiones del tiempo o estrés. Estos problemas se obvian en un sistema experto.

Otro punto importante es la velocidad en que un talento humano puede resolver un problema, lo cual también es influido por muchos factores. Por el contrario, un sistema experto es consistente en la velocidad en que resuelve los problemas, y en muchos casos, supera en gran medida a los expertos humanos en la velocidad en que desarrolla las tareas.

Los talentos humanos tienden a ser costosos. Por lo general los talentos demandan salarios altos, por el contrario los sistemas expertos son relativamente baratos. El costo de desarrollo suele ser alto, pero en muchos casos, se puede recuperar la inversión rápidamente por los ahorros que el sistema produce a la organización.

Otras ventajas interesantes que presentan los sistemas expertos es el hecho que son capaces de explicar los procesos que han utilizado para llegar a una determinada conclusión, lo cual es bastante inusual en los humanos.

En la tabla 1. Se ilustra la comparación entre un experto humano y un sistema experto.

Tabla 1 Comparación entre un experto humano y un sistema experto

FACTOR	EXPERTO HUMANO	SISTEMA EXPERTO
Tiempo Disponible	Horas laborables	Siempre
Geográfico	Local	Disponible en cualquier lado
Seguridad	Irreemplazable	Reemplazable
Deteriorable	Sí	No
Rendimiento	Variable	Estable
Velocidad	Variable	Estable (usualmente rápido)
Costo	Alto	Viable

1. GENERALIDADES

1.1. PLANTEAMIENTO DEL PROBLEMA

La ausencia de herramientas de alta tecnología que permitan el desarrollo eficiente y eficaz de las tareas cotidianas desarrolladas por el Instituto de Ciencias del Mar y Lignología (ICMARES) de la Universidad de El Salvador, despierta el interés por el diseño e implementación de un Sistema Experto para la identificación de Anélidos marinos con énfasis en la familia de los onúfidos (*Polychaeta onuphidae*) de la costa de El Salvador.

1.1.1. OBJETIVOS

1.1.1.1. OBJETIVOS GENERALES

1. Introducir las tecnologías de Inteligencia Artificial en la Universidad de El Salvador por medio del Instituto de Ciencias del Mar y Limnología de El Salvador (ICMARES) enmarcado en el desarrollo académico y como herramienta para la investigación en la institución.
2. Optimizar el rendimiento de los procesos de investigación de los recursos marinos de El Salvador haciendo uso de tecnologías informáticas avanzadas que aumente la productividad de las metodologías empleadas hasta la fecha en El Salvador.

1.1.1.2. OBJETIVOS ESPECIFICOS

1. Desarrollar un software basado en Sistemas Expertos que permita incrementar notablemente el rendimiento y eficacia al personal de ICMARES en la tarea de identificar correctamente a los onúfidos de la costa de El Salvador.

2. Proveer a los estudiantes de la Licenciatura en Biología de la Universidad de El Salvador de una herramienta de fácil uso para todo nivel y de utilidad para llevar acabo prácticas de laboratorio.
3. Fortalecer el aprendizaje de la identificación de los onúfidos de la costa de El Salvador.
4. Crear una herramienta que funcione como una interfase amigable al usuario y como motor de inferencia de la base de conocimientos que pueda ser distribuible y alimentada fácilmente en el futuro con más y nuevas reglas de manera que se pueda identificar más especies con ella.
5. Motivar a la población universitaria a realizar investigaciones novedosas que involucren como área de desarrollo la inteligencia artificial.

1.2. LIMITACIONES DEL PROYECTO

El software NIVEL9 fue creado para trabajar, sin ningún problema, en Sistemas operativos, tales como: Windows 2000, Milenium y XP.

El proyecto fue realizado para resolver problemas que vayan de lo general a lo específico, es decir con una estrategia de inferencia de encadenamiento hacia adelante.

El proyecto no incluye ningún tipo de mantenimiento ni capacitación en el diseño y codificación de las bases de conocimiento.

El tamaño del archivo donde se codifica la base de conocimientos no puede ser mayor a 1 MB.

El formato de las imágenes que puede gestionar el software NIVEL9, en el momento de su ejecución debe ser BMP.

A pesar que entre los objetivos específicos se plantea el fortalecimiento de los estudiantes con ayuda del software, quedará a cargo del personal de ICMARES la verificación de esta tarea, lo que se logrará a través del mejor desempeño de los estudiantes.

1.3. JUSTIFICACION

La principal justificación de tipo práctica que este proyecto conlleva se enfoca en la resolución eficiente de la identificación de organismos marinos.

En cuanto a la utilidad metodológica se crea una herramienta para recolectar conocimiento perdurable para futuras investigaciones en esta área.

Siendo los sistemas expertos un área de especialización de la inteligencia artificial se pretende justificar que la utilización de esta nueva técnica se puede aplicar y explotar en nuestra sociedad.

Por otro lado, el proyecto genera conveniencias económicas (ahorro en la adquisición de licencias de este tipo de productos de alto costo en el mercado), pedagógicas (herramienta de soporte para la realización de laboratorios en el área de biología marina) y productivas (obtener mayor eficiencia en la identificación de los anélidos marinos).

No se queda atrás la relevancia social, ya que justifica la formación de Recurso Humano, a través de los módulos explicativos del software, para el desarrollo del país y esto a su vez permite un beneficio de importancia económica y comercial.

2. FUNDAMENTO TEORICO DE LOS SISTEMAS EXPERTOS

2.1. DEFINICION DE SISTEMAS EXPERTOS

Los sistemas expertos forman parte de un firme y verdadero avance en inteligencia artificial (IA). Un sistema experto o sistema basado en el conocimiento, es un software que imita el comportamiento de un experto humano en la solución de un problema. Pueden almacenar conocimientos de expertos para un campo determinado muy delimitado y solucionar problemas complejos mediante deducción lógica.

Los sistemas expertos pueden incorporar miles de reglas y utilizan árboles de búsqueda para poder trabajar esas reglas. Este busca, recorriendo las ramas del árbol de acuerdo a criterios, hasta dar con la respuesta o fin al problema. Los sistemas expertos tienen la ventaja frente a otros tipos de programas de Inteligencia Artificial, de proporcionar gran flexibilidad a la hora de incorporar nuevos conocimientos. Para ello solo se debe introducir la nueva regla en la base de conocimientos, sin necesidad de cambiar el funcionamiento propio del programa. Los sistemas expertos son "auto explicativos", al contrario que en los programas convencionales, en los que el conocimiento como tal está encriptado junto al propio programa en forma de lenguaje del computador.

Dos actores son fundamentales para el desarrollo de sistemas expertos: el Experto del Dominio (profesional X) y el Ingeniero de Conocimiento (programador), ya que estos van a enlazar sus experiencias almacenándolas en la Base de conocimientos que mediante la interfaz va a permitir al usuario llegar a comunicarse con el motor de inferencia, el cual va a tomar la decisión de aplicar toda la información almacenada en la base de conocimientos.

Los sistemas expertos son uno de los puntos que componen las investigaciones en el campo de la inteligencia artificial. Un sistema que trabaja con técnicas de inteligencia

artificial debe estar en condiciones de combinar información de forma “inteligente”, alcanzar conclusiones y justificarlas. Los sistemas expertos se aplican, por norma general, en problemas que implican un proceso basado en el conocimiento.

Un procedimiento de solución, basado en el conocimiento, tiene la capacidad de utilizar normas o estructuras que contengan conocimientos y experiencias de expertos especializados, deducción lógica de conclusiones y manipulación de conocimientos afectados por valores de probabilidad.

Las reglas deterministas constituyen la más sencilla de las metodologías utilizadas en sistemas expertos. La base de conocimientos contiene las variables y el conjunto de reglas que definen el problema, y el motor de inferencia obtiene las conclusiones aplicando la lógica clásica a estas reglas. Por regla se entiende una proposición lógica que relaciona dos o más objetos e incluye dos partes, la premisa y el consecuente. Cada una de estas partes consiste en una expresión lógica con una o más afirmaciones objeto-valor conectadas mediante operadores lógicos. Una regla se escribe normalmente como “Si premisa, entonces consecuente”.

Las personas que componen el equipo de desarrollo de los sistemas expertos son:

- El experto
- El ingeniero del conocimiento
- El programador
- El usuario

En el desarrollo del Sistema Experto, el ingeniero del conocimiento y el experto trabajan muy unidos. El primer paso consiste en elaborar los problemas que deben ser resueltos por el sistema y determinar correctamente el dominio de la aplicación. Una vez delimitado el dominio, el ingeniero del conocimiento debe diseñar la infraestructura para almacenar el conocimiento y los algoritmos de inferencia para obtener soluciones. El programador se encarga de codificar el diseño. En muchas

ocasiones el ingeniero del conocimiento y el programador son la misma persona. Finalmente, el usuario es la persona que hace uso del sistema experto, que en algunas ocasiones también es el mismo experto.

Las primeras investigaciones de sistemas expertos se llevaron a cabo en los años sesenta. En aquel entonces, se tenía como paradigma la idea que unas pocas leyes de razonamiento junto con potentes computadoras serían suficientes para producir resultados y simular el razonamiento humano. Un intento en ese sentido fue llevado a cabo por los investigadores Alan Newell y Herber Simon quienes desarrollaron un programa denominado GPS (Solucionador General de Problemas). Este programa podía trabajar con criptoaritmética, el problema de las torres de Hanoi y otros problemas similares, pero no podía resolver problemas del mundo real.

El enfoque del problema dio un giro restringiendo los problemas a un dominio específico e intentando simular el razonamiento de un experto humano. En vez de computarizar la inteligencia general, se centraron en dominios de conocimiento muy concretos.

La investigación que se lleva a cabo busca determinar un modelo de inferencia general para identificar diferentes especies de anélidos marinos, a partir del conocimiento que el experto (biólogo) pueda proporcionar a los informáticos, los cuales a su vez almacenarán en una base de conocimientos dicha información para luego, generar un motor de inferencia que permita a través de la resolución llegar a una conclusión correcta sobre la clase de especie.

2.2. ESTRUCTURA DE UN SISTEMA EXPERTO

Generalmente consideramos a una persona como experto cuando esta tiene conocimiento especializado sobre algún problema. En el campo de los sistemas expertos, este tipo de conocimiento se conoce como "dominio de conocimiento".

Usamos el término “dominio” para enfatizar que el conocimiento pertenece a un problema o área específica. Un experto almacena su dominio de conocimiento en su memoria de largo plazo (MLP).

Cuando se le da consejo a alguien, el experto primero obtiene hechos sobre el problema y luego los almacena en su memoria de corto plazo (MCP). Luego el experto entra en proceso de razonamiento combinando los hechos que tiene en su MCP y el conocimiento que almacena en su MLP. Utilizando este proceso el experto infiere información nueva del problema y eventualmente llega a la solución del mismo (Ver Figura 1.).

Los sistemas expertos resuelven problemas usando un proceso que es bastante similar por los métodos utilizados por los expertos humanos (Ver Figura 2.).

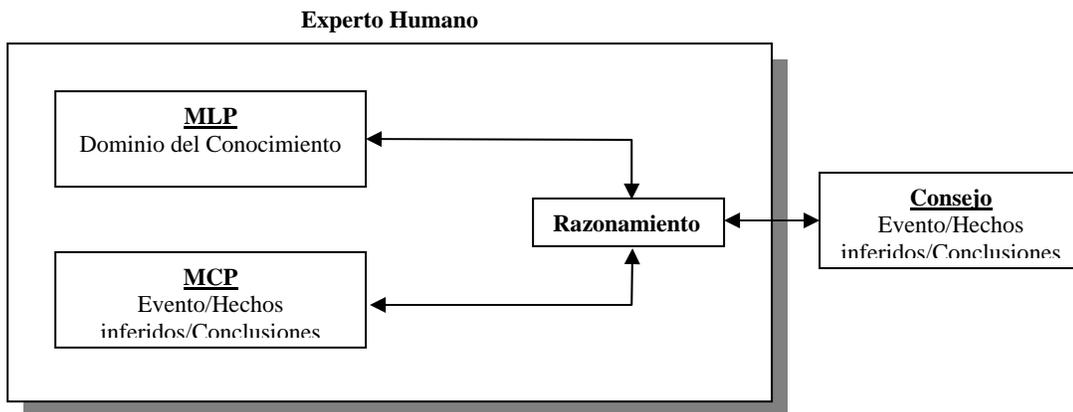


FIGURA 1. Proceso de Resolución de Problemas de un experto Humano

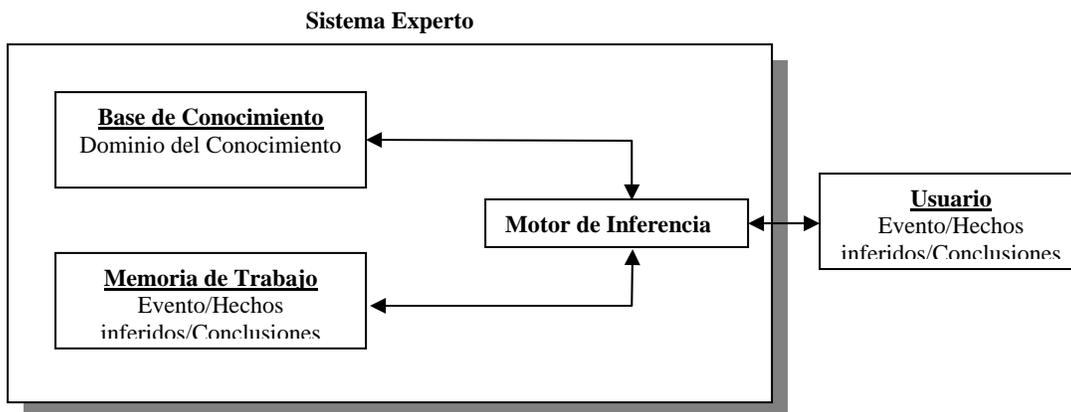


FIGURA 2. Proceso de Resolución de Problemas de un Sistema Experto

2.2.1. BASE DE CONOCIMIENTOS

El sistema experto mantiene el dominio de conocimiento del experto en un módulo conocido como base de conocimientos. La base de conocimientos es un modelo para la MLP.

El trabajo del ingeniero del conocimiento es obtener el conocimiento del experto y codificarlo en una base de conocimientos. Para ello es necesario contar con técnicas apropiadas para representar el conocimiento. Las cinco técnicas más utilizadas para desarrollar sistemas expertos son:

1. Reglas de Producción
2. Redes Semánticas
3. Marcos
4. Cálculo de Predicados

Cada técnica de representación enfatiza cierta información sobre el problema mientras ignora otra. Como es de esperar cada técnica tiene ventajas y desventajas para capturar en forma eficiente los diferentes tipos de conocimientos. Uno de los secretos para el desarrollo de sistemas expertos exitosos es saber escoger una estructura de representación adecuada para la resolución de cada problema.

2.2.1.1. Reglas de Producción

La forma más comprensible de representación del conocimiento se basa en reglas de producción. Se trata aquí de descripciones de acciones dependientes de ciertas condiciones. Una sola regla de producción puede captarse como unidad de conocimiento. Es el componente más pequeño del que consta el sistema en su totalidad.

Se ha descubierto que los expertos están mejor predispuestos a formular sus conocimientos con ayuda de reglas "SI ..., ENTONCES ...". Esta es al parecer la razón de que en la actualidad la mayoría de sistemas expertos y los de mayor éxito se basen en reglas de producción.

Los sistemas que se redactan con reglas de producción reciben el nombre de "Sistemas de Producción" (Ver Figura 3).

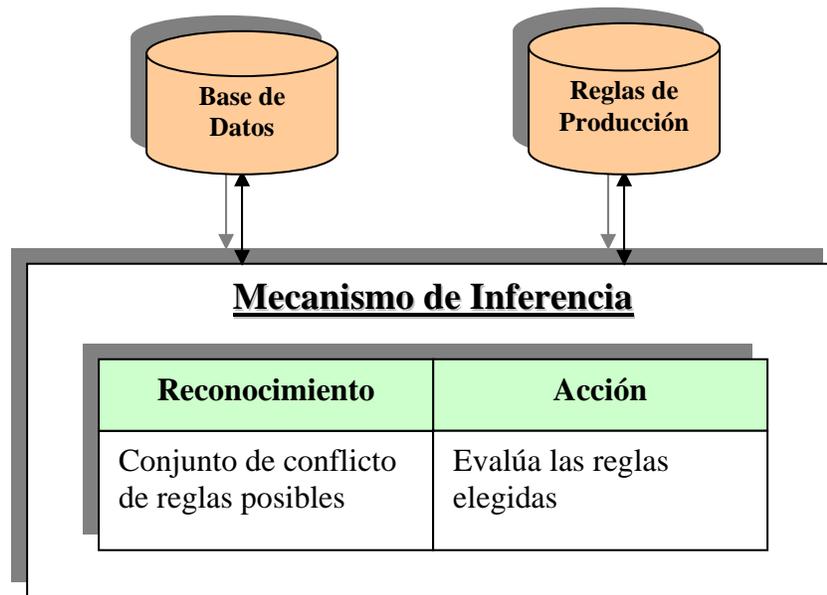


FIGURA 3. Arquitectura de un Sistema de Producción

2.2.1.2. 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 a los arcos que describen las relaciones entre los objetos. Así puede tomarse un arco con sus nodos como una sola unidad de conocimiento.

Pero con la red semántica no se da información sobre el procesamiento de la red lo cual implica que las reglas de inferencia deben estar expresadas en forma explícita.

Una red semántica ofrece una buena visión general sobre las relaciones y dependencias de un área de conocimiento y es muy apropiada para la estructuración del conocimiento y herramienta de verificación para el experto. Sin embargo, los enunciados de las relaciones mencionadas en los arcos deben ser formulados fuera de la red.

En la red semántica pueden 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 (Ver Figura 4).

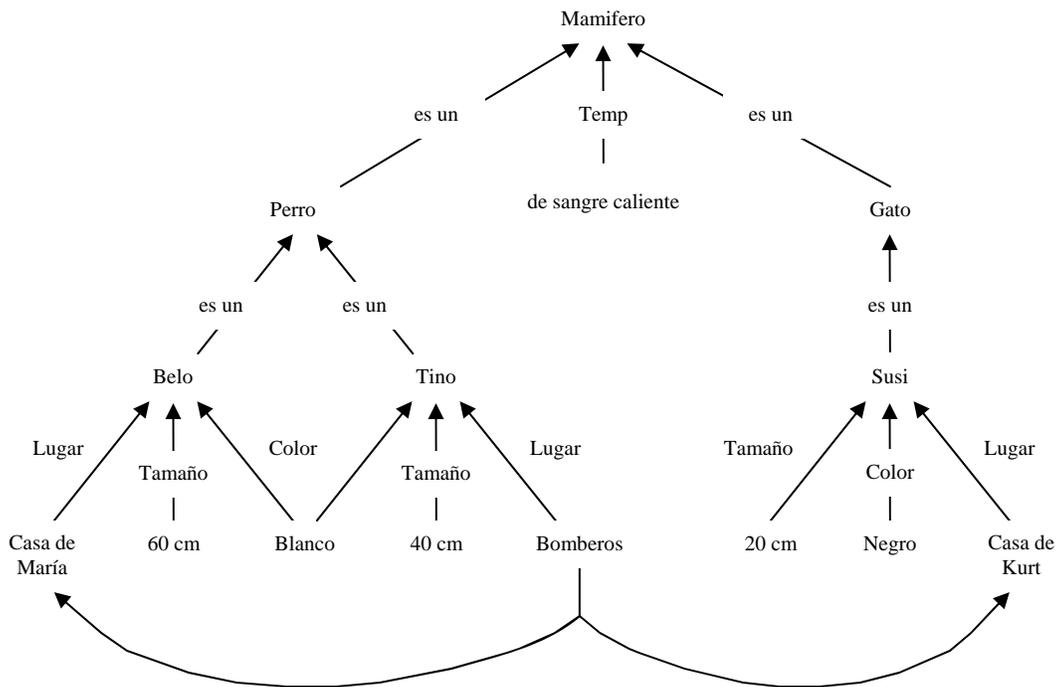


FIGURA 4. Ejemplo de Red Semántica

2.2.1.3. Marcos

“Un marco es una estructura de datos que sirve para representar una situación estereotipada, como estar en algún lugar especial de salón o ir a la fiesta de cumpleaños de un niño. Añadido a cada marco hay varios tipos de información. Parte de esta información hace referencia a cómo utilizar el marco; otra se refiere a lo que

uno puede esperar que suceda en segundo lugar. Y otra a su vez indica que hacer si tales esperanzas no son confirmadas." (Marvin Minsky, 1974).

Un Marco es, por tanto, la división de objetos, o también de situaciones, en sus componentes. Estos componentes son introducidos en las ranuras correspondientes del marco. Las ranuras pueden estar a su vez subdivididas en facetas (para una estructuración más precisa) (ver Figura 5).

Conferencia

Fecha		
Lugar		
Tema		
Participantes		

Ranura Valor de Ranura Faceta

FIGURA 5. Ejemplo de Formato de Marco

Antes de su utilización, el Marco es un armazón preestructurada de datos. La configuración del marco y las definiciones de las ranuras están ya fijadas. A lo largo del procesamiento se van rellenando las ranuras con contenidos. En este proceso puede haber varios Marcos con la misma estructura pero diferente contenido (Ver Figura 6).

Conferencia	Distribución	Conferencia	Desarrollo
Fecha	21 Mar 1986 10.00	Fecha	21 Mar 1986 10.00
Lugar		Lugar	
Tema	Distribución	Tema	Desarrollo
Participantes		Participantes	

FIGURA 6. Ejemplo de Componentes de Marco

Los valores de una ranura son heredables. De esta forma no hace falta modificar más el valor jerárquicamente superior en la ranura y todas las instancias subordinadas del marco obtienen el nuevo valor (Ver Figura 7).

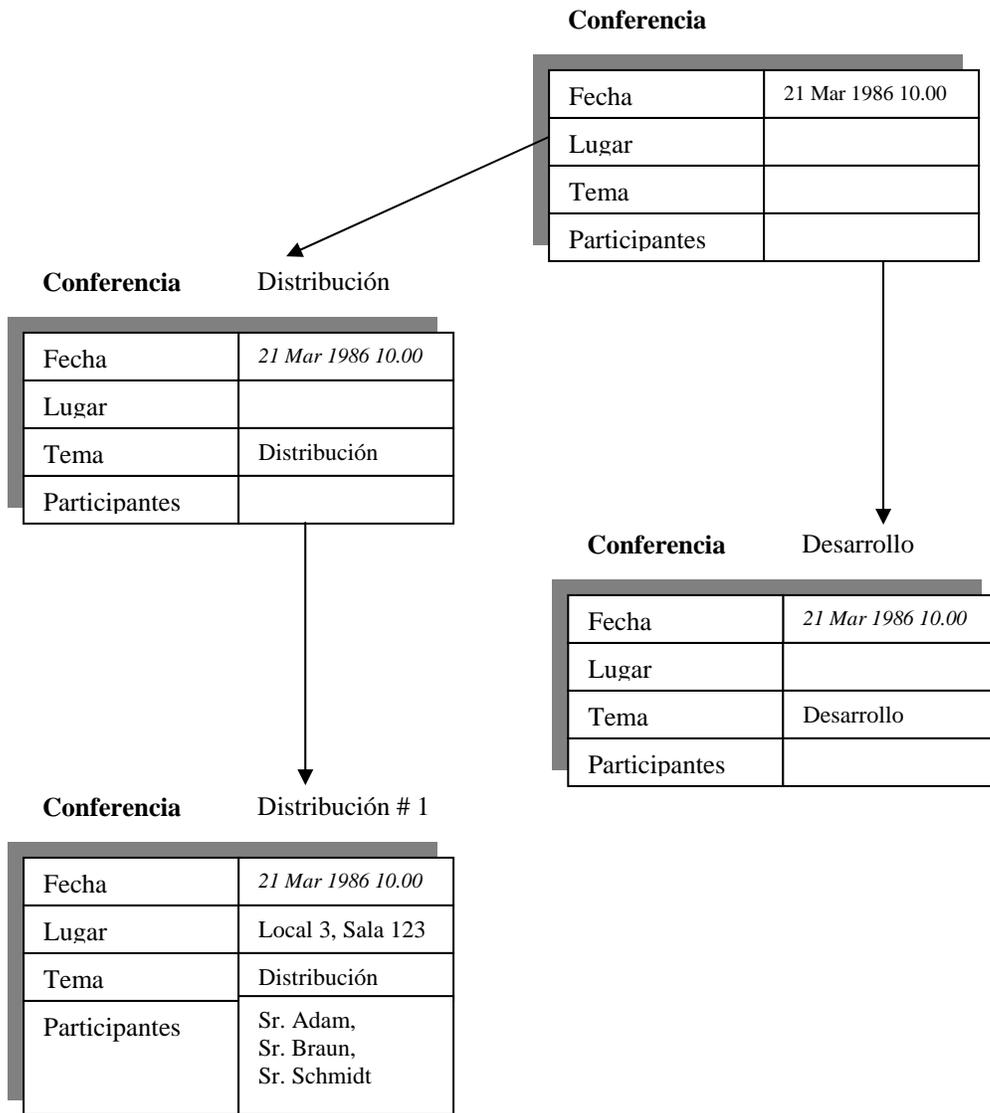


FIGURA 7. Ejemplo de Herencia Jerárquica en el Concepto de Marco

Para el procesamiento de los marcos deben existir reglas y procedimientos incorporados en el concepto, al igual que en la red semántica. Los procedimientos asociados a las ranuras son activados por determinados acontecimientos.

Si se accede a una ranura todavía vacía debe conseguirse el valor. En el procedimiento activado, el valor se calcula o se extrae de la base de conocimientos, o sino también se solicita al usuario durante el diálogo y se introduce en la ranura.

El almacenamiento de un valor en la ranura se realiza mediante un procedimiento llamado "if-added-procedure", y el borrado de un valor mediante un "if-removed-procedure". Ambos procedimientos pueden influir sobre otros marcos.

2.2.1.4. Cálculo de Predicados

El cálculo de predicados describe el conocimiento en forma de enunciados (predicados). Se trata de una notación formal para la descripción de relaciones lógicas y objetos. Contiene una gramática con la que se pueden componer enunciados lógicos válidos. El cálculo de predicados contiene reglas semánticas que relacionan los símbolos del lenguaje formal con los objetos, y reglas de procesamiento capaces de crear expresiones lógicas válidas a partir de expresiones lógicas válidas.

En resumen, el cálculo de predicados es un lenguaje formal con sintaxis y gramática propias, capaz de valorar enunciados lógicos y extraer conclusiones para la creación de nuevos enunciados.

Ejemplo:

En total hay cuatro enunciados dados (dos hechos y dos reglas). Estos enunciados se representan en lenguaje natural y en predicados según el cálculo de predicados¹.

¹ Sistemas Expertos. Dieter Nebendahl. Siemens S.A. Aktiengesellschaft, Berlín y Munich, 1988. Páginas

Tabla 2 . Ejemplo de Cálculo de Predicado

Lenguaje Natural	Cálculo de Predicados
Los dos primeros enunciados son hechos:	
1. Hans y Ute son matrimonio	matrimonio(hans, ute)
Explicación :	La relación "matrimonio" existe entre Hans y Ute
2. Ute vive en Munich	vive_en(ute, munich)
explicación:	La relación "vive en" existe entre Ute y Munich.
Los enunciados tres y cuatro son reglas generales de conclusiones.	
3. Cuando dos personas (X1y X2) son matrimonio, están casados	matrimonio(X1, X2) → casados(X1, X2)
Explicación:	La relación "matrimonio" de las personas X1 y X2 tiene como consecuencia la relación "casados" de las personas X1 y X2.
4. Cuando dos personas X3 y X4 están casadas y la persona X4 vive en el lugar X5, entonces la persona X3 vive también en X5	casados(X3, X4) ∧ vive_en(X4, X5) → vive_en(X3, X5)
Explicación:	La relación "casados" entre X3 y X4 y la relación "vive_en" entre el cónyuge X4 y una ciudad X5 tienen como consecuencia el enunciado de que la relación "vive en" también existe entre el cónyuge X3 y la ciudad X5.

Basados en estos enunciados formales puede demostrarse la suposición de que "Hans vive en Munich", aunque esta suposición hasta ahora no existía como hecho.

Forma de escritura con predicados: **vive_en(hans, munich)**

Esta suposición estará probada cuando:

- a) en la parte de la conclusión de una regla se encuentre la relación “vive_en” con dos variables, cuyo contenido pueda ser “Hans” y “Munich”, y
- b) cuando en la parte de condiciones de la misma regla se hayan cumplido todas las condiciones que llevan a esta conclusión.

Para la prueba se relacionan ahora entre sí - partiendo de la conclusión adecuada - todas las suposiciones, relaciones, reglas y hechos establecidos. Si se acoplan entre sí, sin contradicción, la suposición de que “Hans vive en Munich” es verdadera, si hay contradicciones, la suposición será falsa. Sin embargo, sería posible que los hechos y las reglas establecidos fuesen falsos o incorrectos - un problema con el que siempre nos encontraremos (consistencia de la base de conocimientos).

La hipótesis “Hans vive en Munich” encuentra en su forma de escritura de predicado vive_en (hans, munich) su conclusión adecuada en el cuarto enunciado. Las variables “X3” y “X5” reciben los significados “Hans” y “Munich” (Ver Figura 8).

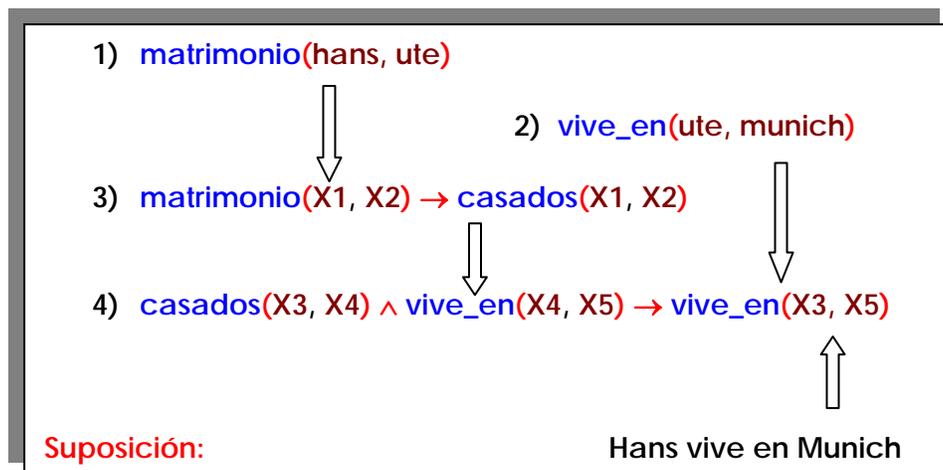


FIGURA 8. Conclusión para la Hipótesis “Hans vive en Munich”

Ahora se comprueba la validez de las dos condiciones para la permisibilidad de la conclusión en la regla 4.

Primera condición de la regla 4:

Casados(X3, X4) ¿verdadero?

Sobre esto no existe ningún hecho, sino tan sólo una conclusión en una regla - que es la regla 3. por lo tanto debe comprobarse la condición de esta regla 3.

Condición de la regla 3:

Matrimonio(X1, X2) ¿verdadero?

Esta condición se cumple por el enunciado (hecho) 1 para X1 = Hans y X2 = Ute.

Por lo tanto, la conclusión también es correcta;

Casados(hans, ute) es verdadero

La primera condición de la regla 4 queda entonces comprobada como verdadera para el caso:

Casados(hans, ute)

Es necesaria ahora la exactitud de la segunda condición

Vive_en(X4, X5)

Ya que X4 ha sido unido al símbolo "Ute", por el segundo enunciado (hecho) se asigna a X5 el símbolo "Munich". De esta manera se cumplen las dos condiciones de la regla 4 y la conclusión

Vive_en(hans, munich) es correcta.

Aunque este conocimiento no estaba explícitamente disponible, ha podido deducirse a través de los enunciados existentes en forma de hechos y reglas.

La forma más simple de inferencia es la deductiva. La inferencia deductiva usa sentencias existentes que son hechos en la base de conocimientos para descubrir nuevos hechos. Una forma de inferencia más potente es la inductiva, la cual aprende nuevas reglas de ejemplos específicos. El desarrollo de nuevas reglas a través de ejemplos es la base de la simbología de las denominadas "Máquinas de Aprendizaje". La minería de datos aplica reglas de inducción para descubrir relaciones y reglas implícitas en una lista de hechos.

La inferencia abductiva trabaja hacia atrás, tomando una hipótesis o sentencia que es un hecho observado. Al probar esta sentencia, usando sentencias lógicas en la base de conocimientos, provee una serie de asunciones o causas que llevan a este evento.

La comprobación es el centro del análisis lógico, y Modus Ponens es el centro del proceso de comprobación. La llave para la generalización de Modus Ponens es la Unificación. La Unificación es el acto de aplicar una sustitución a dos términos para hacerlos ver igual. Mientras Modus Ponens es el centro de la comprobación, la Unificación es el centro de Modus Ponens.

Mientras la generalización de Modus Ponens provee de comprobaciones correctas, esta no puede enfrentarse a todas las formas de sentencias.

Modus Ponens sólo trabaja con una subcolección específica de sentencias, conocidas como las "sentencias de Horn", denominadas de esta forma a favor a Alfred Horn quien se encargó de investigar sus usos en los comienzos de los años cincuentas. Si alguna sentencia no puede convertirse a la forma de cláusula de Horn entonces no se puede utilizar Modus Ponens para llegar a la comprobación, lo cual indica que Modus Ponens no es completa.

Sin embargo “La Resolución” es un procedimiento de inferencia completo, “La Resolución” utiliza reglas de inferencia de resolución en vez de Modus Ponens. Esto lo hace más poderoso que Modus Ponens, el cual sólo provee sentencias automáticas para sus conclusiones. Todo esto vino a generar un método completo de resolución el cual se denominó “Comprobación o Prueba por Resolución” que es el que adoptan lenguajes como PROLOG para unificar enunciados y proveer resoluciones a los enunciados.

2.2.2. MEMORIA DINAMICA O MEMORIA DE HECHOS

La memoria dinámica contiene los hechos, sobre el problema, que se van descubriendo durante la ejecución del sistema. La memoria dinámica es un modelo de la MCP.

Durante la ejecución del sistema experto, el usuario introduce información sobre el problema que se está tratando en la memoria dinámica. El sistema compara esta información con la que se encuentra almacenada en la base de conocimientos para inferir más hechos. El sistema se encarga de introducir estos nuevos hechos en la memoria dinámica para luego volver a repetir el ciclo.

La memoria dinámica contiene toda la información conocida sobre el problema que es proporcionada por el usuario y por las inferencias que realiza el sistema por si mismo.

Muchas aplicaciones de sistemas expertos se alimentan de información que se encuentra en sistemas externos como bases de datos, hojas de cálculo, sensores etc. Estos sistemas externos son capaces de cargar información en la memoria dinámica del sistema experto en el comienzo de la sesión o acceder a ella según el sistema lo necesite.

2.2.3. MOTOR DE INFERENCIA

El modelo del proceso de razonamiento del ser humano es el Motor de Inferencia.

El motor de inferencia es el que se encarga en el sistema experto de emparejar los hechos contenidos en la memoria dinámica con el dominio de conocimiento contenido en la base de conocimientos para lanzar conclusiones o nuevo conocimiento acerca del problema.

El motor busca compara las premisas de las reglas con información que se encuentra en la memoria dinámica, cuando el motor de inferencia encuentra similitudes en los dos módulos añade la conclusión de la regla en la memoria dinámica y continua explorando las reglas para encontrar nuevas similitudes.

El motor de inferencia tiene como funciones principales:

- Determinar las acciones que tendrán lugar, y el orden en que se llevará acabo 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 dialogo 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 estar adaptado al problema a solucionar.

2.2.3.1. Estrategias de Inferencia

Para el procesamiento del conocimiento es necesario formular un mecanismo, que evalúe las reglas y el conocimiento en hechos.

Para tal fin existen dos formas básicas de evaluar las reglas:

Encadenamiento hacia adelante o inferencia controlada por los datos.

Encadenamiento hacia atrás o inferencia controlada por el objetivo.

En ambos casos pueden aparecer muchas reglas para poder elegir en cuál de ellas se satisfacen las condiciones propuestas en la base de conocimientos, dándose soluciones de conflictos ante la inferencia de esas reglas.

A continuación se describe cada una de estas formas.

Encadenamiento hacia adelante:

El propósito de esta forma es buscar en la base de conocimientos las reglas cuyas premisas coincidan para los hechos conocidos y de esta forma disparar su parte de acción o conclusión. Ejecutándose este proceso las veces necesarias hasta alcanzar el objetivo o hasta que no quede más reglas donde buscar.

Este encadenamiento se puede describir de la siguiente forma:

CI = Conocimiento inicial/hechos

HASTA objetivo alcanzado o sin más reglas por disparar

HACER

1) Determina el conjunto K de todas las reglas, cuyas premisas se cumplan con CI

2) Elige de K una regla R según la estrategia de solución de conflictos

3) $CI = CI$ y el resultado de la evaluación de R aplicada sobre CI

Ejemplo:

La base de conocimientos contiene algunas reglas y hechos conocidos. El objetivo es: crear todos los hechos derivables de ello.

Primer paso

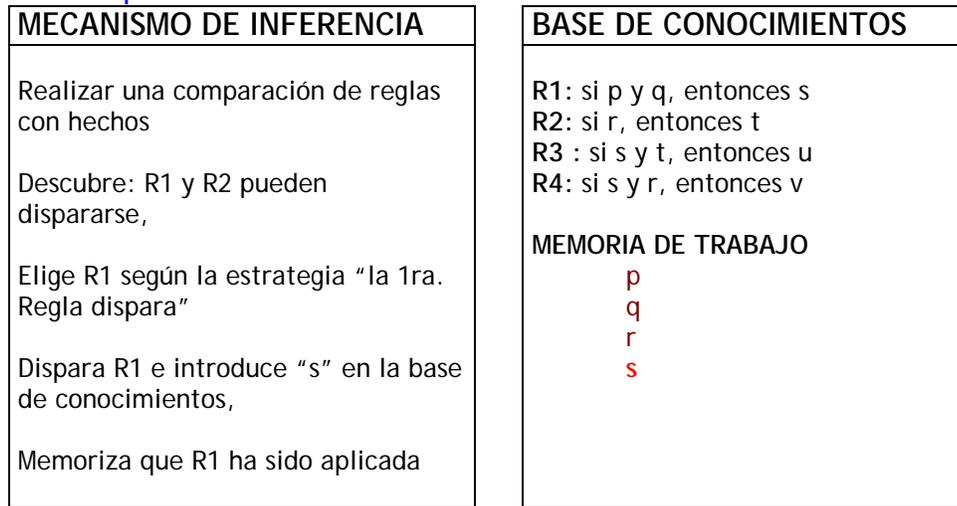


FIGURA 9. Encadenamiento hacia adelante Primer Paso

Segundo paso

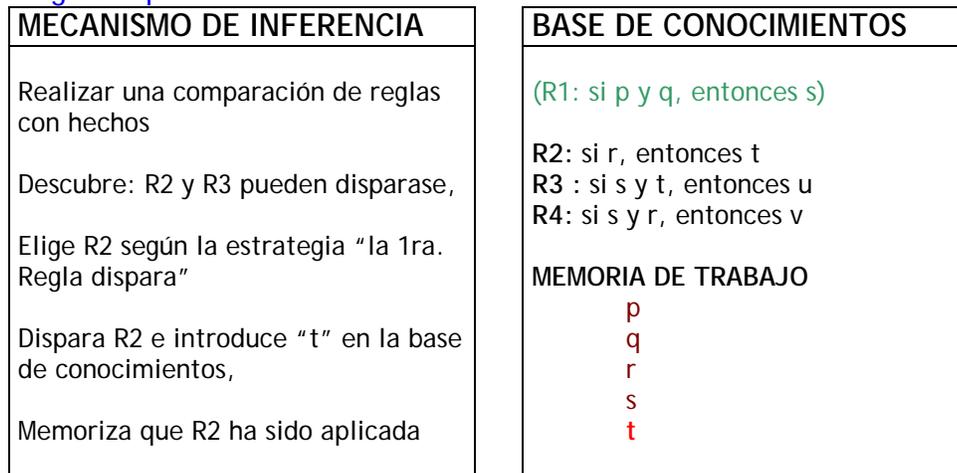


FIGURA 10. Encadenamiento hacia adelante Segundo Paso

Tercer paso

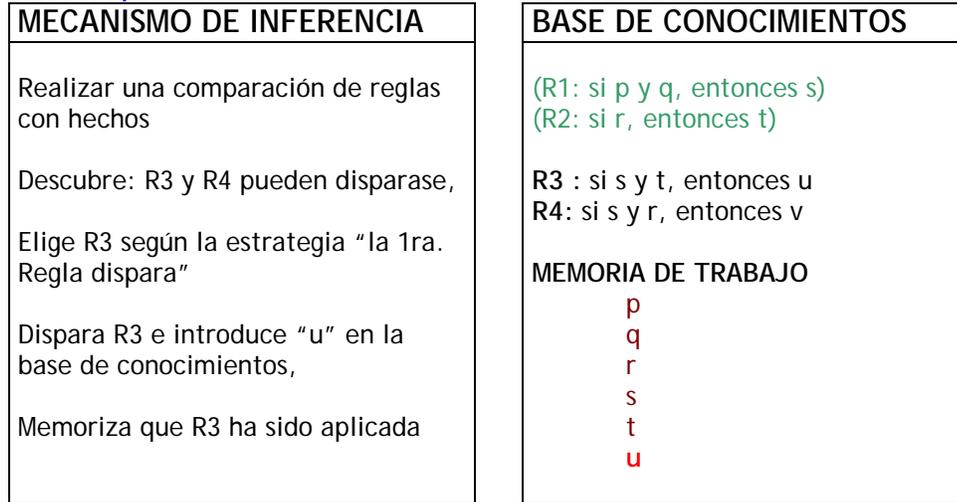


FIGURA 11. Encadenamiento hacia adelante Tercer Paso

Cuarto paso

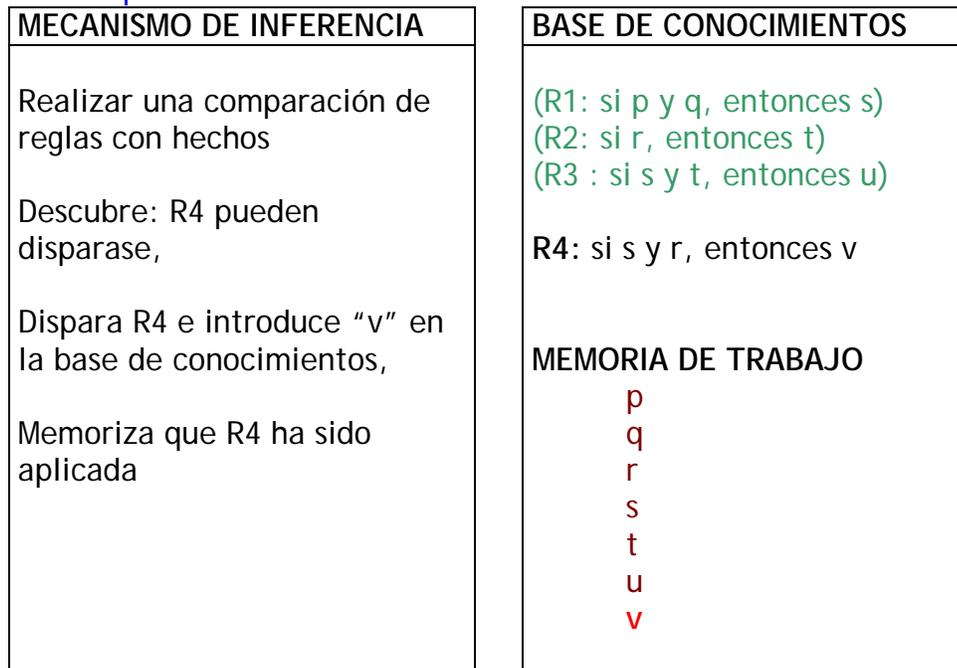


FIGURA 12. Encadenamiento hacia adelante Cuarto Paso

Quinto paso

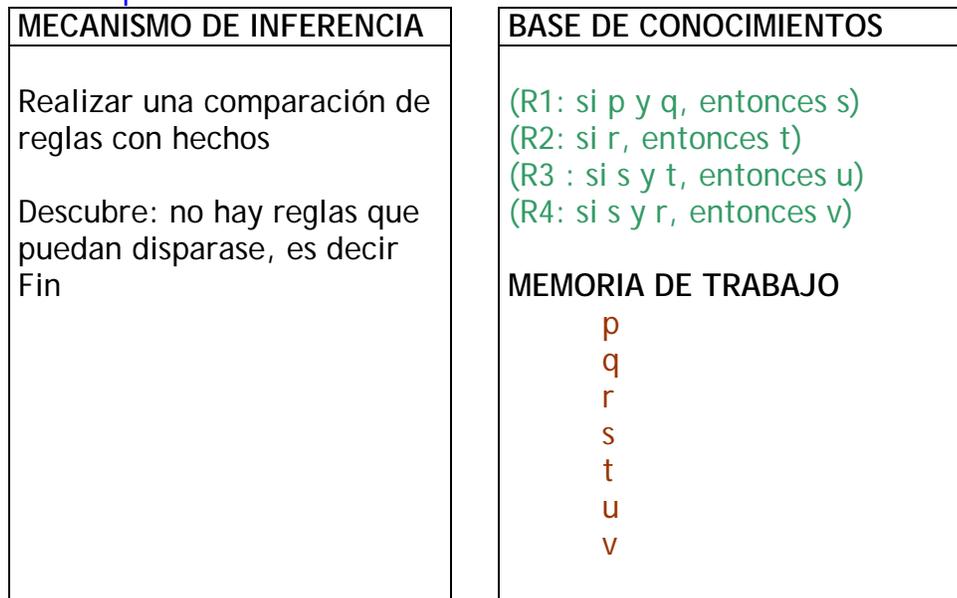


FIGURA 13. Encadenamiento hacia adelante Quinto Paso

Encadenamiento hacia atrás

Esta forma parte de un objetivo o hipótesis, de una conclusión. Todas las reglas que contienen esta conclusión son comprobadas, para saber si se cumplen las condiciones de estas reglas.

Su representación algorítmica sería:

CI = Conocimiento inicial/hechos

HASTA objetivo confirmado o ya no queda regla que pueda disparar

HACER

- 1) Determina el conjunto K de todas las reglas, cuya conclusión puede unificarse con la hipótesis
- 2) Elige de K una regla R según la estrategia de solución de conflictos

3) En caso de que la premisa de R no se encuentre en CI, realiza encadenamientos hacia atrás.

Ejemplo:

El objetivo es verificar la hipótesis "v", es decir que el mecanismo de inferencia compruebe si "v" puede ser deducido de los hechos existentes.

Primer paso

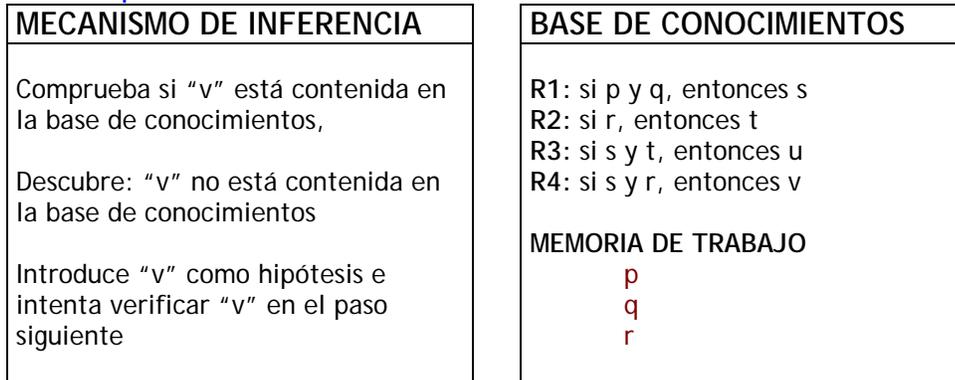


FIGURA 14. Encadenamiento hacia atrás Primer Paso

Segundo paso

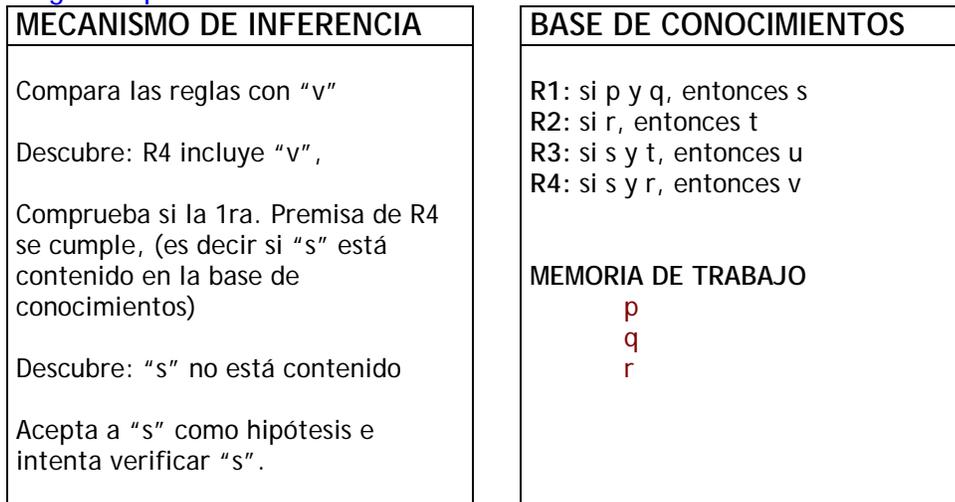


FIGURA 15. Encadenamiento hacia atrás Segundo Paso

Tercer paso

MECANISMO DE INFERENCIA	BASE DE CONOCIMIENTOS
Compara las reglas con "s"	R1: si p y q, entonces s
Descubre: R1 incluye "s",	R2: si r, entonces t
Comprueba si la 1ra. Premisa de R1 se cumple, (es decir si "p" está contenido en la base de conocimientos)	R3: si s y t, entonces u
Descubre: "p" está contenido	R4: si s y r, entonces v
Comprueba si la 2ra. Premisa de R1 se cumple, (es decir si "q" está contenido en la base de conocimientos)	MEMORIA DE TRABAJO
Descubre: "q" está contenido,	p
Descubre que con ello se cumplen todas las premisas de R1,	q
Dispara R1 e incluye "s" en la base de conocimiento.	r
	s

FIGURA 16. Encadenamiento hacia atrás Tercer Paso

Cuarto paso

MECANISMO DE INFERENCIA	BASE DE CONOCIMIENTOS
Comprueba, como a continuación del paso 2, si la 2ra. Premisa de R4 se cumple, (es decir si "r" está contenido en la base de conocimientos)	R1: si p y q, entonces s
Descubre: "r" está contenida,	R2: si r, entonces t
Descubre que con ello se han cumplen todas las premisas de R4,	R3: si s y t, entonces u
Dispara R4 e incluye "v" en la base de conocimiento.	R4: si s y r, entonces v
	MEMORIA DE TRABAJO
	p
	q
	r
	s
	v

FIGURA 17. Encadenamiento hacia atrás Cuarto Paso

Quinto paso

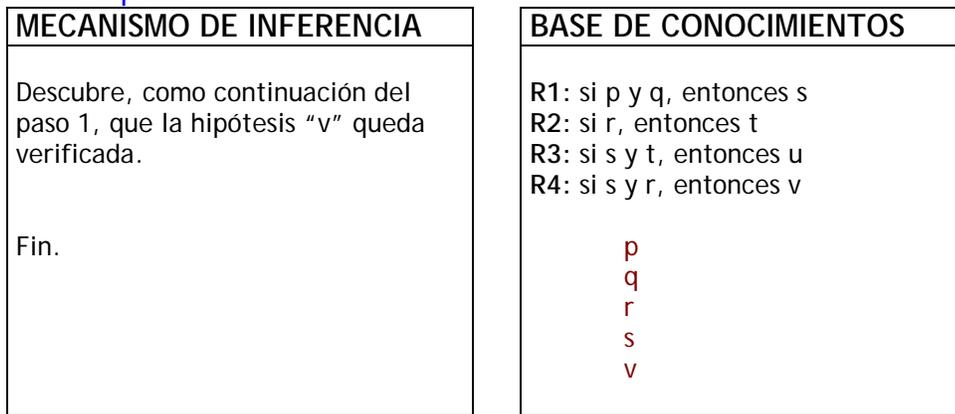


FIGURA 18. Encadenamiento hacia atrás Quinto Paso

2.2.3.2. Resolución de Conflictos

Tanto la estrategia de inferencia de encadenamiento hacia delante y encadenamiento hacia atrás chequean las reglas en el orden en que fueron introducidas a la base de conocimientos y dichas reglas se van disparando en ese orden. Sin embargo es posible que puedan existir otras reglas que se pudieran haber disparado, pero no fueron consideradas por el sistema por este ciclo de acción. Por lo tanto el orden en que están colocadas las reglas juega un papel importante en la inferencia.

Los mecanismos clásicos utilizados para la resolución de conflictos se pueden enumerar como sigue:

1. Orden en la base de reglas
2. Arbitrariedad
3. Prioridades asociadas a reglas
4. Especificidad
5. Edad
6. Refracción

A continuación se detalla cada uno de los mecanismos.

Orden en la base de reglas. Este mecanismo selecciona la primera regla que encuentra, retomando la técnica de FIFO (first in, first out).

Arbitrariedad. El sistema toma cualquier regla utilizando algoritmo de selección en forma aleatoria.

Prioridades. La tercera estrategia selecciona la regla que tenga mayor prioridad, la prioridad es generada por el creador de la base de conocimientos y se manifiesta por algún valor numérico en la codificación de la regla.

Especificidad. Se selecciona la regla que es más específica, es decir la regla que posee una mayor cantidad de premisas, tomando como criterio el hecho de que las reglas con más condiciones son más difíciles de cumplir.

Edad. Las instancias de reglas que utilizan hechos más recientes son preferidas a las que utilizan hechos más viejos. Esto obliga al sistema a razonar con información fresca durante la sesión.

Refracción. Una regla no debe poderse disparar más de una vez con los mismos hechos. Esta técnica previene la creación de ciclos infinitos.

2.2.5. COMPONENTE EXPLICATIVO

Las soluciones descubiertas por los expertos deben poder ser repetibles tanto por el ingeniero del conocimiento en la fase de comprobación así como por el usuario. La exactitud de los resultados sólo podrá ser controlada, naturalmente, por los expertos.

Es deseable que durante el trabajo del sistema se conozca siempre el grado de progreso en el procesamiento del problema.

¿Qué preguntas se plantean y por qué?

¿Cómo ha llegado el sistema a soluciones intermedias?

¿Qué cualidades tienen los distintos objetos?

A pesar de insistir sobre la importancia del componente explicativo es muy difícil y hasta ahora no se han conseguido cumplir todos los requisitos de un buen componente explicativo. Muchos representan el progreso de la consulta de forma gráfica. Además, los componentes explicativos intentan justificar su función, rastreando hacia atrás el camino de la solución. Aunque encontrar la forma de representar finalmente en un texto lo suficientemente inteligible las relaciones encontradas, depara las mayores dificultades. Los componentes explicativos existentes pueden ser suficientes para el ingeniero del conocimiento, ya que está muy familiarizado con el entorno del procesamiento de datos, y a veces bastan también para el experto; pero para el usuario, que a menudo desconoce las sutilezas del procesamiento de datos, los componentes explicativos existentes son todavía demasiado insatisfactorios.

2.2.6. INTERFASE DE USUARIO

En este componente se establece la forma en que el sistema experto se presentará ante el usuario.

¿Cómo debe responder el usuario a las preguntas planteadas?

¿Cómo saldrán las respuestas del sistema a las preguntas que se le plantean?

¿Qué información se representarán de forma gráfica?

Generalmente la interacción entre el usuario y el sistema experto en estilo de lenguaje natural. Esta interacción es altamente interactiva y sigue de cerca los patrones de conversación entre los humanos.

Un requerimiento básico en el diseño de las interfaces es el de preguntar al usuario. Para obtener respuestas confiables del usuario es necesario poner atención particular al diseño de las preguntas del sistema. Esto generalmente requiere diseñar menús, gráficos, etc. El usuario también requerirá de tener la habilidad de poder cambiar y explorar la información contenida en la memoria dinámica.

La interfase de usuario debe cumplir los requisitos siguientes:

- El aprendizaje del manejo debe ser rápido.
- Debe evitarse la entrada de datos errónea
- Los resultados deben presentarse en una forma clara para el usuario.
- Las preguntas y explicaciones deben ser comprensibles.

3. METODOLOGIA DE LA INVESTIGACION

Para realizar el proceso de investigación de este proyecto se trato de seguir la metodología propuesta en el anteproyecto presentado con anterioridad, esta metodología está basada en la investigación realizada por los diseñadores de sistemas expertos, los cuales definen este proceso de construcción del sistema experto como "Ingeniería del Conocimiento".

El principal interés del proyecto, como en cualquier diseño de sistemas expertos, fue "el conocimiento". La adquisición, organización y estudio del conocimiento para ganar entendimiento del problema fue clave para obtener los resultados deseados en la ejecución de este proyecto. Sin embargo en la etapa de diseño y ejecución del proyecto surgió la inquietud de generar una herramienta que fuese capaz de procesar cualquier tipo de conocimiento y no limitar el proyecto a crear un software que resolviera únicamente la problemática de identificación de especies marinas de ICMARES. El producto descrito en el anteproyecto consistía en la creación de un programa que fuese capaz de identificar a través de las metodologías de los sistemas expertos a los anélidos poliquetos de la costa de El Salvador. Esto implicaba la creación de una base de conocimientos interna en la codificación del software que sólo podría resolver el problema específico de identificación y que obviamente no podría ser gestionada por usuarios particulares que no fueran los creadores del programa, lo cual es una tremenda limitación y no representa un diseño óptimo de software. Por esta razón se propuso la creación de una herramienta que fuese capaz de analizar conocimiento en forma separada, es decir, un software que pudiese ser alimentado por conocimiento generado a través de ficheros de un editor de texto cualquiera creado por un usuario particular que tenga conocimiento del lenguaje especial generado para la herramienta en donde se pueda codificar información en forma de reglas para luego estas ser gestionadas por el programa.

En este caso el producto final del proyecto propuesto sería la creación de un código que contenga la información necesaria para que al cargarla al software pueda identificar en forma eficiente e inteligente a las familias de los anélidos poliquetos.

La justificación de este cambio tan dramático estriba en que el software será capaz de ahorrar el 50% del trabajo requerido para la elaboración de cualquier proyecto que implique la creación de un sistema experto. La herramienta permitirá al ingeniero del conocimiento concentrarse únicamente en la obtención de conocimiento de calidad y en la elaboración de códigos que sean comprendidos por el software, dejando al software encargarse de toda la parte lógica y de análisis de la información.

A diferencia de la programación convencional, desarrollar un sistema experto es un proceso altamente interactivo. El diseñador construye parcialmente un sistema, lo prueba, luego modifica el conocimiento del sistema. Este proceso se repite durante todo el proyecto a medida que en cada prueba el conocimiento del sistema aumenta así como el entendimiento del diseñador.

Para el presente proyecto se siguieron las fases clásicas para el desarrollo de sistemas expertos, las cuales se presentan en Figura 19.

3.1. FASE 1 - ASESORIA

Durante la fase de asesoría, el estudio fue conducido para determinar la viabilidad y justificación del problema de la identificación de la familia de los Onúfidios.

Las fuentes de conocimiento fueron suficientes para la construcción de la base de conocimientos ya que se contaba con un experto disponible en horarios accesibles, así como una amplia bibliografía sobre el tema.

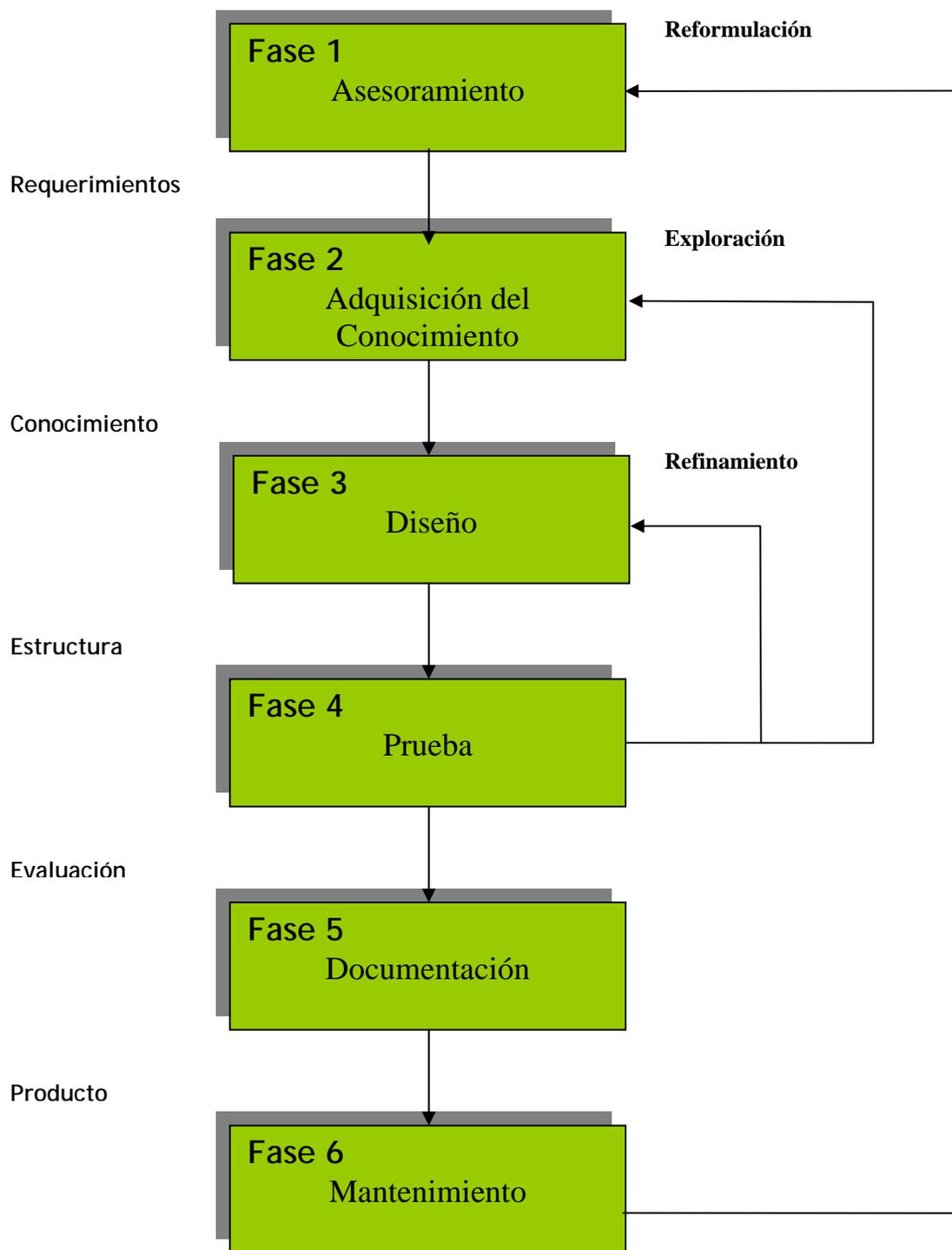


FIGURA 19. Fases del desarrollo de los Sistemas Expertos

En cuanto a los recursos técnicos se contó con una herramienta para la creación de

software sumamente sofisticada la cual permitiría crear el programa con la mayor rapidez posible, la herramienta fue suministrada por cortesía de Microsoft, lo cual de alguna forma solventa el problema económico de comprar una licencia profesional.

ICMARES por otro lado proporcionó sus instalaciones para trabajar la parte de ordenar y codificar la información referente al proyecto. También se sacó un permiso especial para no pagar la cuota monetaria correspondiente por utilizar el parqueo suministrado por las instalaciones de la UES durante la ejecución del proyecto.

Se contó con la colaboración de estudiantes de la carrera de Licenciatura en Biología en la parte de la generación de imágenes e información sobre las especies que se estaban analizando para el sistema experto.

3.2. FASE 2 - ADQUISICION DEL CONOCIMIENTO

El objetivo de la fase de adquisición del conocimiento es recolectar el conocimiento disponible del problema que es utilizado para guiar el desarrollo de diseño. Este conocimiento es usado para profundizar en el problema y como material para diseñar el sistema experto. El proceso de adquisición de conocimiento para este caso se basó en la experiencia del investigador de ICMARES Carlos Giovanni Rivera, quien se especializa en la identificación de especies marinas a nivel microscópico de las costas de El Salvador.

En esta fase del estudio se realizaron numerosas reuniones con el experto, donde algunos aspectos importantes del problema de identificación fueron discutidos, tales como: las relaciones, características físicas, número de miembros entre las familias y especies de los Anélidos. En esta etapa inicial del proyecto el enfoque de las

discusiones fueron los procesos llevados a cabo en el procedimiento básico para realizar identificaciones a nivel profesional y a nivel de prácticas de laboratorio para los estudiantes, de manera que el sistema fuera lo suficientemente amigable para poder ser utilizado por estudiantes no expertos en el área de identificación. El objetivo del proceso al inicio fue descubrir conceptos claves y métodos generales de resolución de problemas utilizados por el experto. Las sesiones posteriores, tomando ventaja de las sesiones anteriores, se enfocaron a explorar información más detallada y específica.

La "Adquisición de Conocimiento" fue como el cuello de botella en el desarrollo del sistema experto por su gran importancia. Si la información obtenida es mala el sistema no generaría resultados consistentes y todo el tiempo invertido habría sido en vano.

3.3. FASE 3 - DISEÑO

Al completar la fase de adquisición de conocimiento se profundizó en el tema, lo cual permitió dar una mejor perspectiva para abordar la solución.

El primer paso fue definir la mejor forma de representar el conocimiento y la mejor técnica de inferir dicho conocimiento para resolverlo. Para el presente problema se definió en primer lugar la inquietud de construir un emulador de sistemas expertos que fuese capaz de gestionar cualquier base de conocimientos, saliendo de la limitación de resolver un solo problema en particular. Para llevar a cabo esta modificación en el diseño se reevaluó la viabilidad sobretodo de tiempo de los diseñadores, ya que obviamente esto implicaba la creación de un lenguaje de programación apropiado para que a través de este se pudiera codificar el conocimiento del experto, y esto traería como conclusión la inversión de más tiempo de análisis y diseño por parte del programador. Como resultado de este análisis se decidió construir un software

utilizando como herramienta de desarrollo VISUAL C++, que consta de dos capas con funciones bien diferenciadas, la primera capa consistía en la creación de un lenguaje de programación que pueda almacenar conocimiento y la otra capa consistía en la creación de un programa que fuese capaz de interpretar ese código para a través de las metodologías de inteligencia artificial pudiese generar resultados y soluciones que pudiesen ser comparados a los resultados con los resultados obtenidos con un experto humano.

Tal como se manifestó en el anteproyecto, como método de inferencia se utilizó el encadenamiento hacia adelante por resolver problemas de clasificación y por consiguiente de identificación a través de la información que pide el sistema que van desde datos que van de lo general a lo específico. Otra razón de peso fue que con este tipo de inferencia el software es capaz de resolver problemas de diseño lo cual no es posible con el encadenamiento hacia atrás. Durante la etapa de diseño, las metas, estructura del sistema y organización fueron definidas. En esta etapa se elaboró una herramienta de software adaptable a las necesidades de los usuarios que fuese capaz de inferir todo el conocimiento necesario para llegar a la solución.

Durante la etapa de diseño se construyó un prototipo inicial del sistema. El propósito del prototipo fue de proveer un vehículo para obtener un mejor entendimiento del problema. Al construir inicialmente un pequeño sistema, y revisar los resultados con el dominio del experto, se revelaron nuevos requerimientos del sistema como el hecho de que era necesario que el programa fuese capaz de mostrar imágenes pertinentes a las preguntas que realizaba al usuario durante la ejecución de la sesión.

El prototipo también funcionó como punto focal para futuras entrevistas con el experto.

El diseño del sistema experto fue un proceso inherentemente interactivo en donde se definió la base de conocimiento a través de prueba y errores para ser refinada.

3.4. FASE 4 - PRUEBA

La fase de prueba no fue una tarea separada de las demás, sino que al contrario se fue desarrollando a través de todo el proyecto incluso en las etapas finales del mismo. En cada entrevista realizada con el experto del tema en estudio, nuevo conocimiento fue descubierto y agregado al sistema. Esto fue seguido por pruebas adicionales donde el conocimiento obtenido anteriormente fue modificado. El principal objetivo de la fase de prueba fue la validación de los objetivos propuestos al principio del proyecto. Además, esta etapa fue dando una buena idea del grado de complacencia y aceptabilidad del usuario final con respecto a las interfaces de usuario y la complejidad de uso del mismo. Esta fase se caracterizó por una gran cercanía que se creó entre el ingeniero del conocimiento y el experto del dominio en estudio.

3.5. FASE 5 - DOCUMENTACION

La fase de documentación concierne a la necesidad de compilar toda la información del proyecto en un documento que pueda ser un punto de encuentro entre el usuario final y el diseñador del sistema. La documentación es una práctica bastante cotidiana y necesaria para todo proyecto de ingeniería del software, y el desarrollo de sistemas expertos no es la excepción. La documentación en este caso explica cómo opera el sistema y contiene un tutorial que explica en forma sencilla las operaciones básicas del sistema. La documentación también provee información técnica acerca de los requisitos de hardware necesarios para poder ejecutar el sistema sin problemas. También en los anexos se incluye cierta información referente al dominio del tema, ya

que el sistema puede ser utilizado, no sólo por expertos sino, por personal que no conozcan demasiado del tema.

3.6. FASE 6 - MANTENIMIENTO

Una vez el sistema ha sido desarrollado e implementado generalmente es necesario darle mantenimiento periódicamente. Así como un niño, un sistema experto continúa creciendo y aprendiendo. El conocimiento no es estático, sino que crece y se madura. La base de conocimientos necesita ser refinada y actualizada conforme pase el tiempo y los requerimientos del sistema cambien. Desgraciadamente no se cuenta con un programa de mantenimiento establecido dentro de los objetivos de este proyecto.

El mantenimiento del sistema experto para este proyecto, correrá por cuenta del personal de ICMARES, basándose en la documentación del software o contratando a personal idóneo que lo realice.

4. DISEÑO DEL SOFTWARE NIVEL9

NIVEL9 se puede descomponer en cuatro grandes módulos, según la función que realizan dentro del software, estos son:

1. ESTRUCTURAS DE DATOS
2. MOTOR DE INFERENCIA
3. LENGUAJE PARA CODIFICAR EL CONOCIMIENTO
4. MODULO DE RECONOCIMIENTO DE TEXTO Y SINTETIZADOR DE VOZ

4.1. ESTRUCTURAS DE DATOS

Las estructuras de datos dentro del software tienen la función de almacenar información importante en forma ordenada en la memoria de la computadora. Toda la información está clasificada de forma conveniente dentro de las estructuras con el fin de que el software pueda diferenciar y utilizar dichos datos de manera eficiente según los propósitos de cada módulo.

NIVEL9 utiliza siete diferentes estructuras de datos para suplir a los módulos restantes del sistema estas se pueden enumerar como sigue:

- Máquina P (MP)
- Tabla de Símbolos (TS)
- Tabla de Datos (TD)
- Tabla de Cadenas (TC)
- Memoria de Trabajo (MT)
- Base de Conocimientos (BC)
- Tabla de Premisas (TP)

La Base de Conocimientos (BC) junto con la Memoria de Trabajo (MT) son las encargadas de dar apoyo al Motor de Inferencia para que este pueda realizar las búsquedas y resoluciones de la forma más rápida y eficiente.

La Tabla de Premisas (TP) junto con las dos estructuras anteriores (BC y MT) tiene la función de hacer posibles los módulos explicativos del software. Se puede decir que estas tres estructuras son las responsables de toda la parte relacionada con la inteligencia artificial, específicamente la metodología de Sistemas Expertos.

Por otro lado las cuatro estructuras restantes están más relacionadas con la gestión del lenguaje de codificación del conocimiento creado para estos fines.

La Máquina P (MP) es una pila que se encarga de gestionar diferentes operaciones de tipo aritmético, relacional y lógico. Esta funciona en base a técnicas clásicas de compiladores e intérpretes.

La Tabla de Datos (TD) proporciona a la MP los datos o valores con los cuales esta va a llevar a cabo resultados preliminares sobre operaciones de varios tipos.

La Tabla de Cadenas (TC), aunque también almacena valores de variables como lo hace la TD, no se relaciona directamente con la MP ya que en las variables de tipo cadena no se lleva a cabo ningún tipo de operación.

Por último la Tabla de Símbolos (TS), tal como su nombre lo dice, lleva el registro de todos los símbolos empleados por la codificación del conocimiento en el lenguaje propio de NIVEL9.

A continuación se detalla cada una de las estructuras de datos mencionadas anteriormente.

4.1.1. MAQUINA P O MAQUINA PILA

La Máquina P (MP) es una estructura en forma de pila la cual se encarga de gestionar todas las operaciones aritméticas, relacionales y lógicas del lenguaje NIVEL9. Esta estructura funciona basándose en las técnicas tradicionales de los compiladores y actúa en forma previa al motor de inferencia. La Máquina de NIVEL9 se basa en una gramática que genera un código intermedio de Notación Postfija el cual le permite a la pila almacenar los operandos de forma anticipada a los operadores, para de esta forma poder operar dichos valores y almacenarlos en la cima de la pila, de esta forma una vez concluida la operación se elimina los operandos relacionados con la operación que se acaba de efectuar y así sucesivamente continuar operando las nuevas expresiones que se van reconociendo.

Los valores que se introducen a la MP generalmente son extraídos de la Tabla de Datos (TD) la cual se encarga de almacenar las variables utilizadas en el código propio de la herramienta y además va actualizando dichos valores (La Tabla de Datos (TD) se explica con mayor detalle más adelante).

El método de operación de la MP para las expresiones relacionales se basa en las operaciones tradicionales de la matemática discreta. De igual manera, para resolver expresiones lógicas numéricas o entre proposiciones, la MP se apoya en las tablas de verdad y en el teorema de "modus ponens".

Una vez el código del lenguaje propio de la herramienta ha sido convertido a código intermedio (Notación Postfija), la pila va tomando los valores numéricos que son identificados por el analizador léxico. Estos números se van guardando en la pila hasta que se identifica un operador, el cual puede ser un operador aritmético, relacional o lógico; inmediatamente después la MP se encarga de realizar la operación correspondiente a las dos posiciones anteriores de donde se detectó el operador, borra estas dos posiciones e inserta el resultado en la cima de la pila y así sucesivamente.

La Máquina P (MP) se encarga también de operar lógicamente las proposiciones dentro de las reglas, de manera que en esta se van acumulando y eliminando los resultados de la veracidad o falsedad de dichas proposiciones.

En la Figura 20 se muestra como procede la MP para realizar una simple operación aritmética.

4.1.2. TABLA DE SIMBOLOS

La Tabla de Símbolos (TS) se encarga de almacenar en memoria todos los símbolos utilizados por el lenguaje NIVEL9. Estos símbolos son creados y destruidos cada vez que se inicializa y se sale de la aplicación.

La tabla de símbolos se compone de 4 campos o elementos:

Etiqueta: Se encarga de crear un índice único por cada símbolo almacenado en la estructura.

Lexema: Aquí se almacenan los símbolos utilizados por el lenguaje propio de NIVEL9.

Componente Léxico: En esta zona de la estructura se depositan los nombres correspondientes a cada uno de los símbolos utilizados por la TS.

Tipo: En esta zona se va clasificando cada símbolo según su tipo.

Para aclarar los conceptos se pone el ejemplo del signo de suma. Supongamos que este es el primer símbolo almacenado en la estructura, entonces dicho elemento quedaría identificado de la siguiente manera:

Etiqueta: 0

Lexema: +

Componente Léxico: SUMA

Tipo: Aritmético

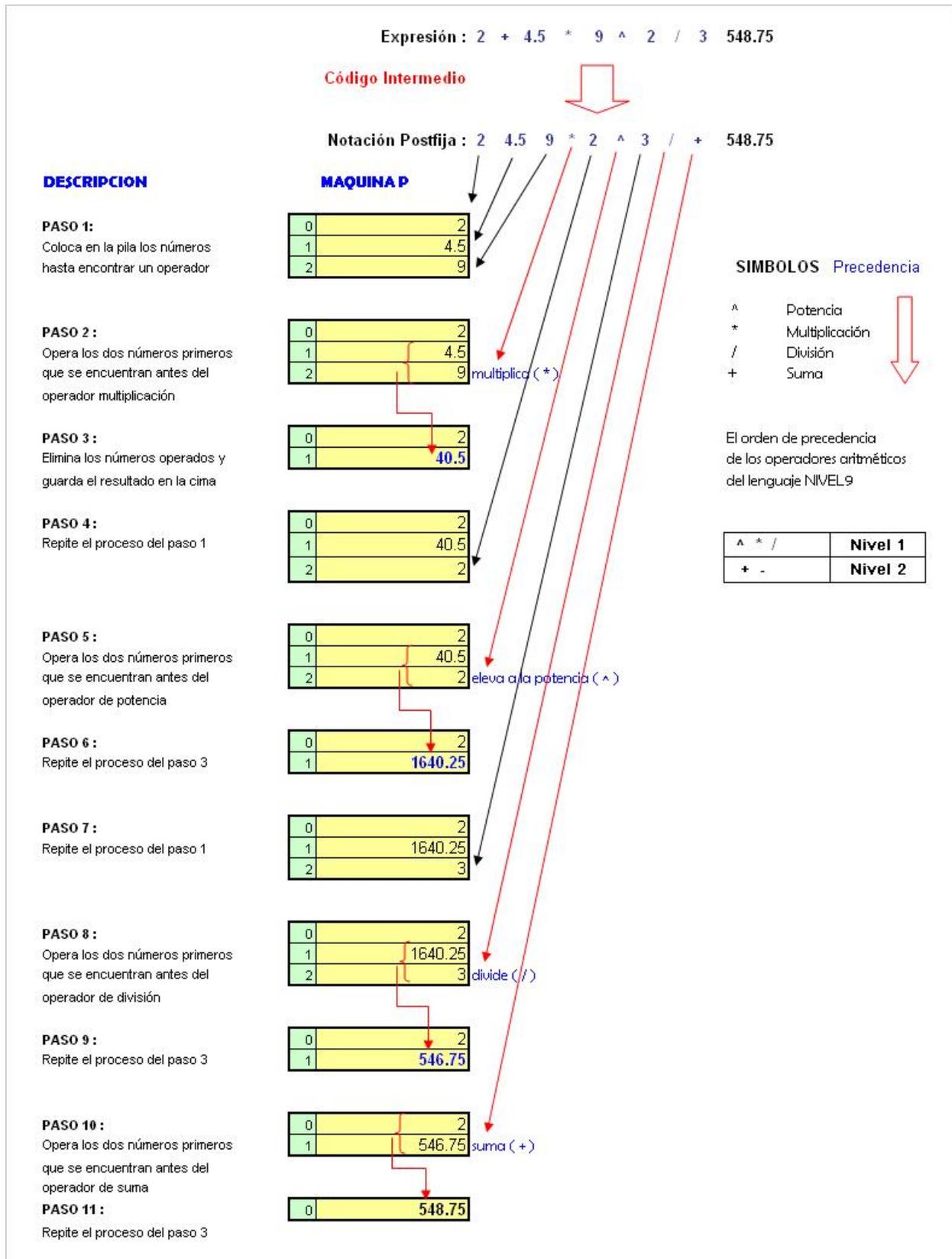


FIGURA 20. Comportamiento de la Máquina P ante una operación aritmética

En el caso de las palabras claves se sigue el mismo procedimiento. Por ejemplo supongamos que la palabra clave *num* sigue después del símbolo de suma, entonces esta quedaría identificada de la siguiente manera:

Etiqueta: 1
Lexema: sino
Componente Léxico: SINO
Tipo: Reservada

Para este caso el Tipo esta indicando que se trata de una palabra reservada propia del lenguaje NIVEL9.

Este módulo también tiene otra función adicional y es la de almacenar las nuevas variables que se van generando dentro del programa que se crea. Por ejemplo, si en el software se declara la variable *Var_1* de tipo numérico a través del siguiente código:

NUM \$Var_1\$ = 23;

La TS detectará a través del analizador semántico del lenguaje que se trata de una declaración, entonces se procederá a pasar la información a la tabla de símbolos y esta la ordenará según los campos de la estructura.

Suponiendo que la tabla cuente en ese momento con 86 elementos, la etiqueta será el próximo elemento nuevo:

Etiqueta: 87
Lexema: Var_1
Componente Léxico: NUM
Tipo: Numérico

La TS está indicando claramente que el elemento almacenado es una variable de tipo numérico. Esto se realiza con el fin de que sea posible la detección de nombres de variables repetidas y así hacer posible una depuración en el código propio de la herramienta.

En la Tabla 3, se muestra una sección de la estructura que funge como Tabla de Símbolos (TS) en el lenguaje NIVEL9.

ETIQUETA	LEXEMA	COMPONENTE LÉXICO	TIPO
0	""	NINIGUNO	Nulo
1	+	SUMA	Aritmético
2	-	RESTA	Aritmético
3	*	MULTIPLICACION	Aritmético
4	/	DIVISION	Aritmético
5	^	POTENCIA	Aritmético
6	>>	ENTRADA	Entrada/Salida
7	<<	SALIDA	Entrada/Salida
8	==	IGUAL	Relacional
9...	=...	ASIGNACIÓN...	Asignación...
...89	...fin	...FIN	...Reservada

Tabla 3. Estructura de datos que almacena todos los símbolos y palabras claves utilizados en la gramática correspondiente al lenguaje creado para la herramienta NIVEL9.

4.1.3. TABLA DE DATOS

La Tabla de Datos (TD) es la estructura encargada de almacenar todas las variables de tipo numérico (NUM) que se van declarando en el lenguaje propio de NIVEL9. La TD permite al analizador sintáctico buscar a través del nombre de la variable el valor asignado a la misma y llevarla a la MP para que pueda ser gestionada por esta (La función de la Máquina P (MP) se verá más adelante). Esta estructura consta de cuatro campos:

ID_TD: Es el identificador único asignado a cada elemento que compone la estructura.

Nombre Variable: En este campo se almacena el nombre designado a la variable. Los nombres de variables pueden tomar cualquier símbolo o carácter (incluyendo el espacio en blanco) que se encuentren entre los símbolos de dólar.

Ejemplos:

\$El hombre es negro\$;

\$El 4 + el 5 ===== 9\$;

La única excepción a esta regla es el mismo signo de dólar. El signo de dólar no puede utilizarse por ser el símbolo que identifica el comienzo y la finalización de una variable.

Ejemplo:

\$El valor del dólar es \$22.50\$;

Esto al compilar generaría un error.

Valor Numérico de Variable: Este espacio de memoria se aparta para almacenar los diferentes valores numéricos que va tomando la variable.

ID_MT: Corresponde al identificador de la memoria de trabajo donde se ha almacenado la misma variable, este identificador representa la posición de la estructura a la que hace referencia, fungiendo entonces el campo como un apuntador a la memoria de trabajo.

Esta referencia a la Memoria de Trabajo (MT) hace posible al motor de inferencia diferenciar entre hechos que son variables declaradas y hechos que son proposiciones.

Toda variable numérica declarada generalmente debe ser un hecho (una información que se da ya por conocida). Esta variable puede tomar un valor numérico como asignación desde la declaración de la misma o bien pedirá al usuario después asignarle manualmente un valor o por otro lado, podría tomar algún valor resultado de alguna operación realizada con otras variables. En cambio las variables de tipo cadena (almacenadas en la Tabla de Cadenas (TC)) generalmente se utilizarán como simples mensajes largos almacenados en variables con nombres cortos que se darán en forma repetida en diferentes partes del código del lenguaje propio de NIVEL9.

La Tabla 4, muestra cómo está estructurada la TD. Puede observarse cómo los nombres de las variables pueden tomar valores tanto flotantes como enteros y además se ve claramente cómo el último campo indica la posición en donde estas mismas variables se encuentran almacenadas en la MT.

ID_TD	Nombre Variable	Valor Numérico de Variable	ID_MT
0	Costo_hierro	48.67	3
1	Costo madera	54.12	5
2	Costo_cemento	10.45	6
3	Num_casas	12	7
4	Tamaño	6.8	9

Tabla 4. Ejemplo de Tabla de Datos

4.1.4. TABLA DE CADENAS

La Tabla de Cadenas (TC) tiene la misma función que la TD con la diferencia de que esta se encarga de almacenar las variables de tipo cadena (CAD) en vez de las de tipo numérico (NUM).

Al igual que la TD, esta estructura tiene un campo designado a servir como puntero a la Memoria de Trabajo (MT) de la cual se hablará más adelante.

Los elementos que forman la estructura son los siguientes:

ID_TC: Es el identificador único asignado a cada elemento que compone la estructura.

Nombre Variable: En este campo se almacena el nombre designado a la variable. Al igual que las variables de tipo numérico estas van delimitadas por el signo de dólar (\$).

Contenido de la Cadena: Este espacio de memoria se aparta para almacenar un arreglo de caracteres que conforma una cadena. Este arreglo corresponde con el nombre de la variable de tipo cadena y por supuesto puede estar compuesto por cualquier símbolo que sea parte del código ANSI², incluso símbolos numéricos los cuales actuarán como cadenas y no podrán ser operados aritméticamente ni lógicamente.

ID_MT: Corresponde al identificador de la Memoria de Trabajo (MT) donde se ha almacenado la misma variable, sirviendo este campo entonces, como un apuntador a la memoria de trabajo.

La Tabla 5 muestra como está estructurada la tabla de cadenas.

ID_TC	Nombre Variable	Contenido de Cadena	ID_MT
0	Nombres	Mauricio Ernesto	1
1	Apellido	Alfaro	2
2	Segundo Apellido	Parada	4
3	Sexo	M	8
4	Tipo	RH2+	10

Tabla 5. Ejemplo de Tabla de Cadenas

4.1.5. MEMORIA DE TRABAJO O BASE DE HECHOS

La Memoria de Trabajo (MT) es una de las estructuras principales del sistema, es a través de la interacción de esta con la Base de Conocimientos (BC) que se puede llevar a cabo la inferencia del conocimiento. Esta estructura va almacenando todo el conocimiento útil que se va obteniendo de la BC de acuerdo a las respuestas que se van respondiendo a lo largo de la sesión. El motor de inferencia se encarga de disparar los hechos o conclusiones que pueden ser generados a través de análisis lógico matemático entre proposiciones y variables, las cuales a su vez, son gestionadas por la

² Código ANSI : Código de símbolos utilizados por Windows para poder representar diferentes lenguajes en diferentes idiomas en las aplicaciones Windows.

MP, la cual actúa conjuntamente con el motor de inferencia para traducir reglas en conocimiento.

Los elementos que componen esta estructura son los siguientes:

ID_MT: Identificador único que va controlando la posición de los hechos almacenados.

ID_BC: Este campo funge como un apuntador a la posición de donde se obtuvo el hecho en la Base de conocimientos. Este apuntador es indispensable para que el sistema sea capaz de explicar de donde obtuvo las conclusiones finales o intermedias.

Número de Regla: El número de regla permite al sistema ubicar la regla de la cual se pudo obtener conocimiento importante para resolución del problema. El número de regla puede obtenerse una vez se ha ubicado el ID_BC.

Tipo de Instrucción: Este campo es un indicador que le sirve al motor de inferencia para incrementar la velocidad de búsqueda, ya que el motor reconoce los tipos de instrucciones que le pueden ser útiles para inferir conocimiento.

Tipo de Producción: Al igual que el campo anterior este elemento es otro indicador que le sirve al motor de inferencia para acelerar la búsqueda de información importante. Existen tipos de producciones que simplemente son inútiles para inferir conocimiento.

Contenido de Hecho: En este espacio de memoria se almacenan los hechos que se van disparando a través del motor de inferencia, los cuales provienen de las reglas almacenadas en la Base de Conocimientos (BC). Los hechos almacenados luego servirán para obtener más conocimiento.

En la Tabla 6 se muestra con mayor detalle cómo está compuesta la estructura de la Memoria de Trabajo (MT).

ID_MT	ID_BC	Número de Regla	Tipo de Instrucción	Tipo de Producción	Contenido de Hecho
0	5	2	INST_HECHO	PROP	La viga cumple la normas ASTM R2006
1	8	2	INST_HECHO	EXP	El área de acero de refuerzo > 20.4
2	12	3	INST_HECHO	EXP	El peso del concreto es >= 12.78

Tabla 6. Estructura de Memoria de Trabajo

La figura 21, muestra cómo se interrelacionan la Memoria de Trabajo (MT) con la Tabla de Datos (TD) y la Tabla de Cadenas (TC) las cuales mantienen un apuntador hacia la primera, con el fin de poder identificar de dónde provienen los hechos que va almacenando la Memoria de Trabajo.

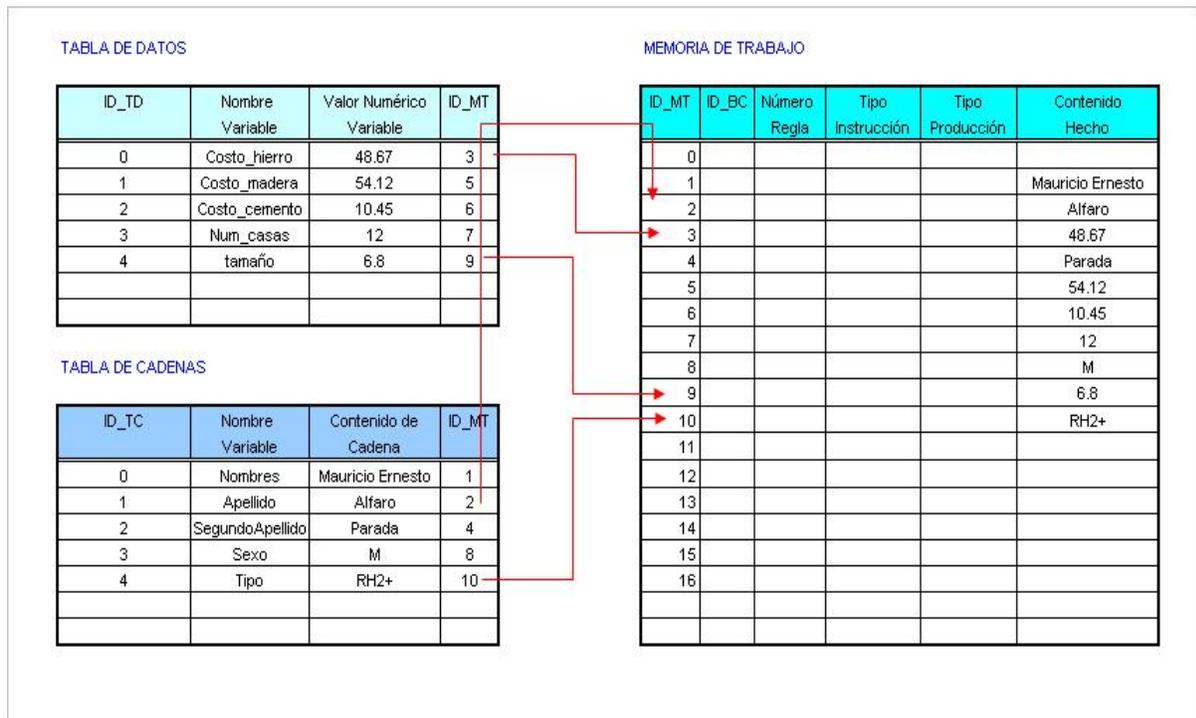


FIGURA 21. Relación entre Memoria de Trabajo con Tabla de Datos y Tabla de Cadenas

4.1.6. BASE DE CONOCIMIENTOS

La Base de Conocimientos (BC) es la principal estructura del sistema. Aquí se almacenan todo el conocimiento obtenido en forma de reglas por medio del lenguaje propio de la herramienta.

Esta estructura tiene la función de identificar y ubicar parámetros claves recogidos del código de donde se obtuvo el conocimiento. Estos parámetros serán reconocidos por el motor de inferencia para realizar en forma eficaz y eficiente la resolución de las proposiciones y así poder dar una conclusión apropiada a los hechos y premisas gestionadas.

No es difícil adivinar que la BC es la estructura más grande que maneja el sistema por contener en ella toda la información del experto en forma de parámetros apropiados que vienen a formar los campos de la estructura. Esta estructura se compone de los siguientes campos:

ID_BC: Este es el identificador único que identifica cada elemento de la estructura. Estos elementos pueden ser de diferentes tipos (sentencias, proposiciones, variables, delimitadores etc.) y se separan convenientemente según los símbolos utilizados en el lenguaje NIVEL9.

Instrucción: Aquí se van almacenando las instrucciones propias del lenguaje como por ejemplo: *SI, ENTONCES, SALIDA, RESP...* de esta forma el sistema puede reconocer que hacer en cada caso.

Contenido de Proposición: En este espacio se guardan las proposiciones las cuales representan las afirmaciones y hechos sacadas de las reglas con los cuales el motor de inferencia jugará para evaluar y sacar conclusiones efectivas. Para

aclarar el concepto de proposición se muestra el siguiente ejemplo, si tenemos la siguiente regla:

*SI Todos los hombres son mortales,
Y Mauricio es hombre,
ENTONCES Mauricio es mortal.*

Las palabras escritas en mayúscula representan instrucciones en el lenguaje de NIVEL9, por lo que las proposiciones para este ejemplo son:

*Todos los hombres son mortales,
Mauricio es hombre,
Mauricio es mortal.*

Número de Regla: Este campo, tal como su nombre lo indica, almacena el número de regla al que pertenece cada elemento de la Base de Conocimientos. Este es tomado de la parte superior de cada conjunto de proposiciones que están encerradas por llaves. Retomando el ejemplo anterior el código adecuado para representar esa regla en el lenguaje NIVEL9 sería:

```
REGLA: 20
{
  SI ¿Todos los hombres son mortales?;
  Y ¿Mauricio es hombre?;
  ENTONCES ¿Mauricio es mortal?;
  SI_FIN
}
```

De manera que la Base de Conocimientos puede reconocer que la instrucción *SI* y la proposición *Todos los hombres son mortales* pertenecen a la regla número 20. Nótese que se ha agregado a la regla la instrucción *SI_FIN*, Esto se hace con el fin de que el sistema pueda identificar el fin de la instrucción *SI*, y las llaves denotan los límites de la regla, definiendo el inicio de la regla y el final de la regla.

Tipo de Instrucción: En este campo se almacena un entero definido por cada tipo de instrucción que identifica a las proposiciones.

Cada proposición puede ser o bien una premisa (INST_PREMISA), o un hecho (INST_HECHO) o una declaración (INST_DECLARACION), por lo tanto en este campo se almacena las constantes que identifican estos tipos como tales. La identificación de los tipos de instrucción, tal como su nombre lo dice, viene dado por las instrucciones utilizadas. Por ejemplo: después de una instrucción *SI* tiene que venir una proposición de tipo premisa (INST_PREMISA), después de un *ENTONCES* tiene que venir un hecho (INST_HECHO), y de esta forma es posible que el sistema pueda diferenciar y llenar este campo según convenga.

Todo esto ayuda al motor de inferencia a optimizar las búsquedas ya que este sólo tomará las premisas como factores importantes para disparar un hecho.

Tipo de Producción: El tipo de producción clasifica todas las producciones o afirmaciones contenidas en una regla de acuerdo al propósito que tiene dicha afirmación. En este caso también se identifican tres tipos diferentes: proposiciones (PROP), expresiones (EXP) y sentencias (SENT).

Las proposiciones ya se definieron en el campo Contenido de Proposición visto anteriormente. Sin embargo, las producciones pueden estar formadas por expresiones, relaciones y por sentencias, las cuales deben tratarse de forma diferente por el sistema. Una expresión relacional será enviada a la máquina P para comparar lógicamente su veracidad y en base a esto mandar un veredicto sobre si debe dispararse esa regla o no. Una sentencia en cambio es una instrucción que se maneja específicamente a nivel de programación para hacer más interesante y amigable el uso del software.

Analizar la siguiente regla codificada en el lenguaje propio de NIVEL9:

```

REGLA: 12
{
  Si $El peso de la viga$ > 235.78;
  Y $La carga uniformemente distribuida$ <= 42.5;
  O $La deflexión de la viga$ >= 0.0023;
  Y ¿La separación de la capa de mortero no cumple la norma?;
  (SALIDA<<"¿Cumple la separación de la capa de mortero la norma
  estructural RT2006?"");
  ENTONCES ¿El diseño está defectuoso?;
  SALIDA<<"El diseño de la viga no cumple las normas."";
}

```

En el ejemplo se puede observar que el lenguaje utiliza los símbolos de dólar (\$) como delimitadores de variables que fueron declaradas con anterioridad. Este delimitador permite poder utilizar casi cualquier símbolo (menos el dólar) como nombre de la variable. Estas variables son comparadas con valores numéricos a través de operadores relacionales para ver si son producciones verdaderas o falsas. Por ejemplo, la primera producción: *\$El peso de la viga\$ > 235.78;*

Por otro lado, las proposiciones están delimitadas por los signos de interrogación y se valen de sentencias para interactuar con los usuarios permitiendo a estos definir la veracidad de la producción como ocurre con la afirmación:

¿La separación de la capa de mortero no cumple la norma?;

Luego el sistema utiliza la sentencia *SALIDA* para permitir al usuario responder afirmativamente o negativamente.

(SALIDA<<"¿Cumple la separación de la capa de mortero la norma estructural RT2006?"");

El uso de los paréntesis en la instrucción *SALIDA* le indican al sistema que debe hacer una interrupción para permitir al usuario interactuar con el mismo, también tiene el propósito de definir a cuál de todas las proposiciones de tipo premisa se la ha asignado el conjunto de sentencias encerradas por dichos paréntesis.

Indicador de Disparo de Regla: Este campo es de gran importancia para el motor de inferencia, ya que aquí se va almacenando las premisas, hechos y sentencias que se van disparando a lo largo de la sesión. Esto permite al motor de inferencia reconocer reglas que ya han sido gestionadas y así no volver a buscar en estas que ya están almacenadas en la Memoria de Trabajo, incrementando notablemente el rendimiento de las búsquedas. Este campo almacena los valores lógicos 1 y 0 para indicar si se la producción fue disparada o no.

Peso de Regla: Este campo permite darle un peso a cada regla de manera que el sistema pueda distinguir algún tipo de orden o importancia en la forma en cómo debe de realizar la resolución de las reglas.

NOTA: Esta instrucción no se ha incluido en la programación del software, mas se ha tomado en cuenta en el diseño del lenguaje NIVEL9 con el fin de incorporar esta funcionalidad en futuras versiones del sistema.

La forma de la estructura de la Base de conocimientos podría representarse gráficamente de la siguiente forma (Ver Tabla 7):

ID_BC	Instrucción	Contenido de Proposición	Número de Regla	Tipo de Instrucción	Tipo de Producción	Indicador disparo de Regla	Peso de Regla
0	SI	Todos lo hombres son mortales	3	INST_PREMISA	PROP	0	0
1	Y	Mauricio es hombre	3	INST_PREMISA	PROP	0	0
2	PARENTESIS_IZQ SALIDA	(¿Es Mauricio hombre?	3 3		SENT	1	0
3	PARENTESIS_DER ENTONCES) Mauricio es mortal	3 3	INST_HECHO	PROP	0	0
4	SI_FIN		3				

Tabla 7. Estructura de Base de Conocimientos

En la Figura 22, puede observarse la relación que hay entre las estructuras de Base de Conocimientos (BC) con la Memoria de Trabajo (MT).

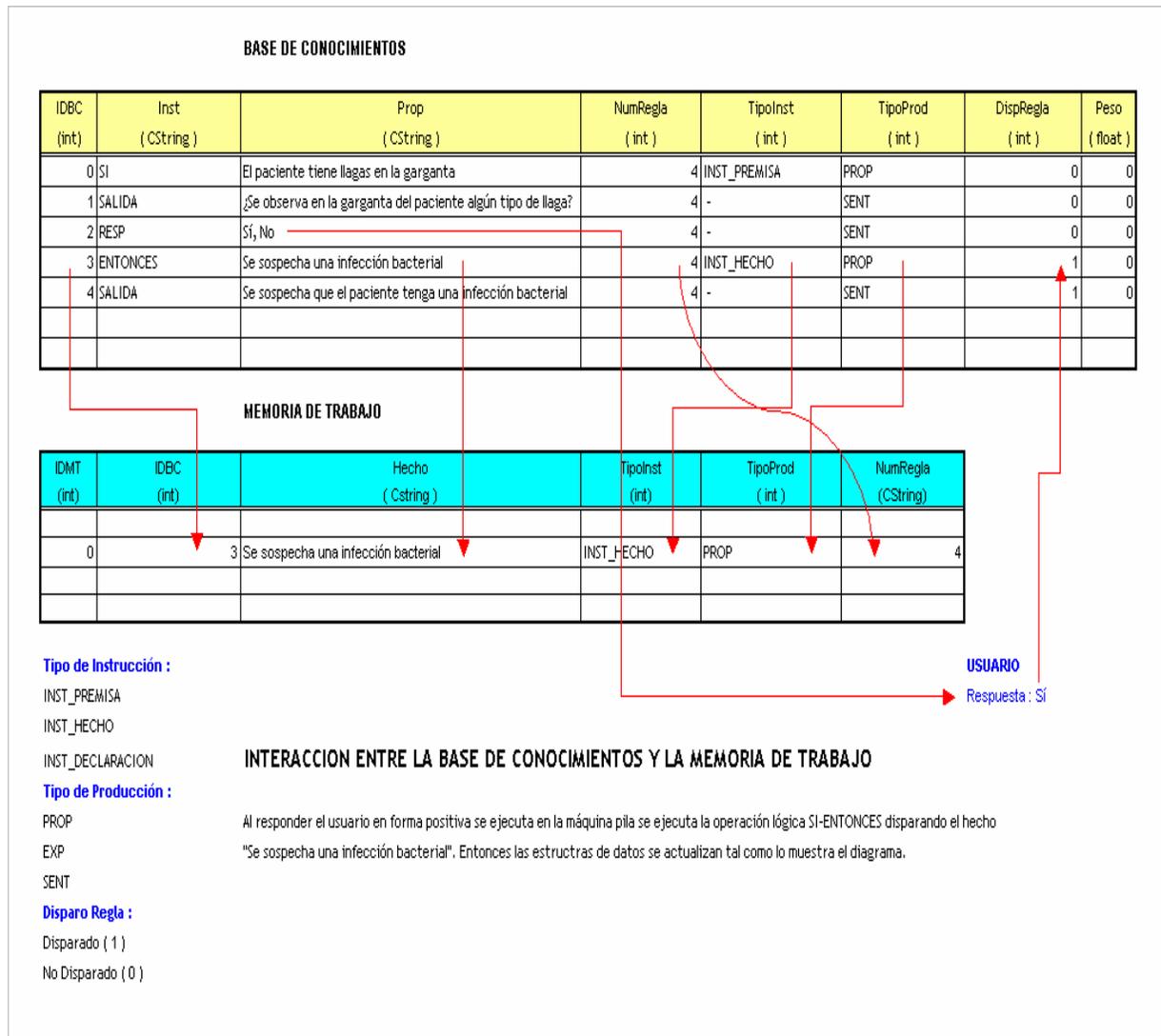


FIGURA 22. Comportamiento de las estructuras de Bases de Conocimiento y Memoria de Trabajo en el momento que se dispara una regla.

En el ejemplo el sistema muestra en la interfaz de usuario a través de la instrucción SALIDA la pregunta *¿Se observa en la garganta del paciente algún tipo de llaga?*, el usuario se asume que responde afirmativamente y entonces se dispara la regla pasando el hecho *Se sospecha una infección bacterial* a la Memoria de Trabajo junto con todos los parámetros que dicha estructura solicita.

4.1.7. TABLA DE PREMISAS

La Tabla de Premisas (TP) es la encargada de ir almacenando las proposiciones que necesitan algún tipo de interacción por parte del usuario. Esta estructura va

almacenando la producción junto con su equivalente respuesta para que el sistema pueda conocer el historial del usuario en cuanto a las aseveraciones realizadas durante una sesión determinada. Este registro es de gran valor para los módulos explicativos del sistema ya que permiten tener control de las proposiciones ciertas y falsas, y así poder proporcionar esta información a la MP para que esta a su vez pueda realizar un análisis lógico apropiado de acuerdo al teorema de Modus Ponens y el uso de tablas de verdad.

Los componentes de esta estructura son los siguientes:

ID_TP: Es el identificador único que distingue una proposición de otra

ID_BC: Representa un apuntador a la Base de Conocimientos (BC) para tener registrada la posición en donde se encuentra la proposición almacenada.

Contenido de Premisa: En este campo se guarda la proposición de interés. Todas las proposiciones van delimitadas por los símbolos de interrogación. El signo de interrogación de apertura indica el comienzo de la proposición y el signo de interrogación de cierre indica el cierre de la proposición. Al igual que las variables esto permite utilizar casi cualquier carácter para la construcción de las proposiciones.

Ejemplo:

¿El cielo se ve nublado?;
¿El factor es mayor que 3.5?;

Como es lógico la excepción es el uso de símbolos de interrogación en las proposiciones para evitar al compilador errores de interpretación. Las siguientes proposiciones generarían un error en el compilador.

Ejemplo:

¿El día es Lunes o ¿Martes??;
¿La calle es ¿¿¿¿ roja?;

Contenido de Respuesta: Este campo representa la respuesta del usuario ante la proposición de interés. El usuario debió mandar una respuesta positiva o negativa y esta respuesta es la que se almacena en este campo.

Número de Regla: Esta estructura lleva también un registro del número de regla para poder indicar en los módulos explicativos en que regla es que se dieron las respuestas almacenadas.

Tipo de instrucción: El tipo de instrucción debe estar bien definida para saber si la proposición es efectivamente una premisa (INST_PREMISA) ya que todos los demás tipos de instrucción en esta estructura serán descartadas. Luego este campo sirve como control de errores a lo largo de la sesión.

Tipo de producción: Al igual que el elemento anterior el tipo de producción debe dar señales para ver si es una proposición (PROP) ya que los demás tipos de producción no serán tomados en cuenta en esta estructura.

La representación gráfica de esta estructura se muestra a continuación en la Tabla 8:

ID_TP	ID_BC	Contenido de Premisa	Contenido de Respuesta	Número de Regla	Tipo de Instrucción	Tipo de Producción
0	4	El paciente tiene una temperatura mayor que 40 grados	1	2	INST_PREMISA	PROP
1	8	El paciente ha estado mas de un mes en cama	0	4	INST_PREMISA	PROP
2	22	El paciente presenta llagas en la garganta	0	9	INST_PREMISA	PROP
3	23	El paciente padece de la faringe	1	9	INST_PREMISA	PROP

Tabla 8. Estructura de la Tabla de Premisas.

La Figura 23 muestra como se relacionan las estructuras responsables de que se lleve a cabo la inferencia y los módulos explicativos dentro del sistema. La Base de Conocimientos (BC) se comunica con la Memoria de Trabajo (MT) a través del apuntador ID_BC que se encuentra como segundo elemento dentro de la MT, esta comunicación permite al motor de inferencia distinguir y ordenar las búsquedas de forma eficiente. Por otro lado la TP se comunica con la BC a través del apuntador ID_BC que se encuentra como segundo elemento dentro de la TP. Esta comunicación hace posible el funcionamiento de los módulos explicativos que ofrece NIVEL9.

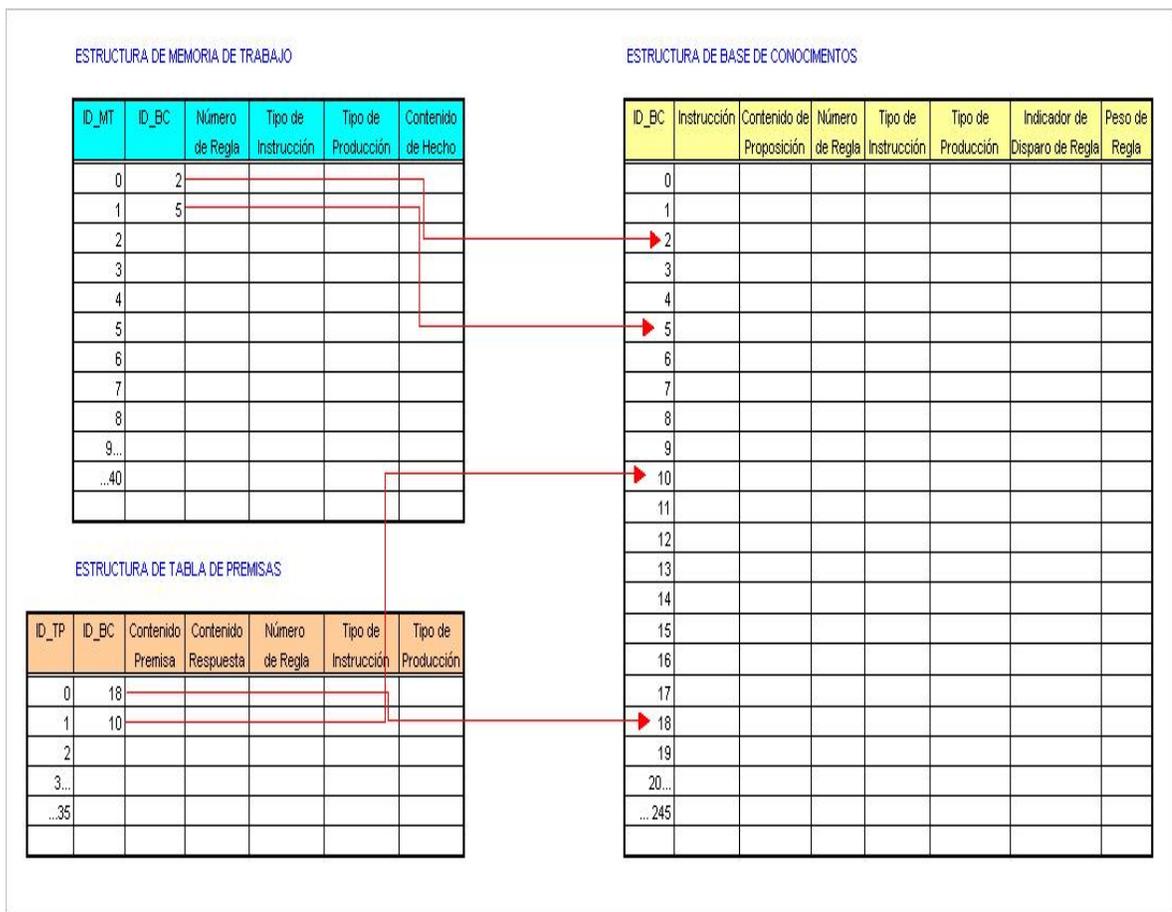


FIGURA 23. Relación entre la Base de Conocimientos con respecto a la Memoria de Trabajo y la Tabla de Premisas.

4.2. MOTOR DE INFERENCIA

El motor de inferencia es el módulo encargado de construir en forma controlada y sistemática las soluciones de los diferentes problemas que se intentan resolver por medio de información almacenada en la BC.

El motor de inferencia tiene como función principal el seleccionar, decidir, interpretar y aplicar el conocimiento almacenado en la BC sobre la MT con la finalidad de recoger la solución buscada y luego arrojarla a la interfaz de usuario por los mecanismos de salida del sistema.

El funcionamiento del motor de inferencia de NIVEL9 puede clasificarse, según el orden de ejecución de los algoritmos que se generan, en:

- **EVALUACION:** El motor de inferencia selecciona la información útil a emplear en la inferencia. Debe distinguir cuándo se trata de una premisa y cuándo de un hecho, además debe identificar en las producciones cuándo se trata de proposiciones y cuándo de expresiones de tipo aritmético, relacional o lógico.
- **COMPROBACION:** El motor debe realizar diferentes comprobaciones para cerciorarse de que las producciones que le van llegando son aplicables para la inferencia.
- **INFERENCIA:** En esta fase el motor de inferencia debe realizar búsquedas dentro del la MT o Base de Hechos para comparar hechos anteriormente inferidos con reglas nuevas y así inferir nuevos hechos.
- **OPTIMIZACION:** El motor es responsable también por optimizar de la mejor forma posible las búsquedas de el mismo va generando. Para esto se vale de parámetros obtenidos de la BC con los cuales puede definir anticipadamente rangos limitados de búsqueda.

- **CONTROL:** Este enunciado se refiere al control de posiciones en las estructuras de datos y al control del orden en que se van gestionando las producciones obtenidas de las reglas codificadas.
- **CHEQUEO DE ERRORES:** El motor también es capaz de gestionar la mayoría de errores debidos a codificaciones deficientes del conocimiento en el lenguaje que puedan ocurrir durante la inferencia del conocimiento. El motor manda mensajes de error a los módulos de salida del sistema que permitan al usuario descifrar la causa del error.

El mecanismo de búsqueda que utiliza NIVEL9 es la búsqueda ordenada de soluciones. Esta se basa en el encadenamiento del conocimiento representado en forma de reglas de producción. Este encadenamiento se consigue haciendo que el consecuente (hecho) o acción de una regla se convierta en el antecedente (premisa) de la siguiente regla a considerar.

El encadenamiento genera estructuras en forma de árbol y este puede darse por terminado cuando se encuentra una solución o el conjunto total de las mismas.

La búsqueda ordenada de soluciones del motor de inferencia de NIVEL9 utiliza el uso de heurísticos para la poda del árbol de soluciones. Un heurístico es un proceso creador, el cual permite a partir de un conjunto de operaciones elementales suficientes, resolver todos los problemas de una clase específica. Los heurísticos utilizados consideran el conocimiento representado en las reglas para dirigir la búsqueda de la solución, para lo que el lenguaje cuenta con instrucciones especiales para este propósito. Esto significa que el sentido de la búsqueda puede hacerse tanto por profundidad o por amplitud, ya que esto depende de cómo se ordenen las reglas en el código del lenguaje.

Como se mencionó anteriormente NIVEL9 utiliza el encadenamiento hacia adelante o deductivo como método de inferencia. El motor de inferencia enlaza los conocimientos

a partir de unas producciones con el fin de obtener una solución de un problema. El encadenamiento hacia delante fue escogido como método de inferencia por ser el modo de trabajo al que estamos acostumbrados y el más práctico para cálculos de tipo numérico. Además éste tipo de inferencia permite la resolución de problemas de diseño, lo cual no es posible a través del encadenamiento hacia atrás.

La operatividad del motor de inferencia con encadenamiento hacia adelante de NIVEL9 se basa en el "modus ponens" de la lógica formal el cual enuncia lo siguiente:

Si conocemos la regla: <<Si A entonces B>> y es cierto <<A>>, entonces podemos deducir que <> es también cierto.

En el caso de NIVEL9 las premisas son obtenidas de las reglas almacenadas en la Base de Conocimientos (BC) según el orden en que van siendo codificadas por el lenguaje, mientras que los hechos son obtenidos de la Memoria de Trabajo (MT). La MT se va llenando a partir de los hechos obtenidos de las reglas que se van disparando. Cuando es aplicable, el motor de inferencia compara premisas con hechos para separarlos en una sección de conflicto en donde se lleva a cabo la resolución del conflicto entre proposiciones, de tal manera que si el conflicto se resuelve positivamente la regla se dispara y el hecho extraído de esa regla se añade a la MT y así vuelve a comenzar el ciclo.

La Figura 24 muestra el algoritmo utilizado por el motor de inferencia de NIVEL9 para inferir conocimiento a través del método de encadenamiento hacia adelante.

La Figura 25 muestra la forma en que interrelacionan la BC y la MT durante el proceso de inferencia de NIVEL9.

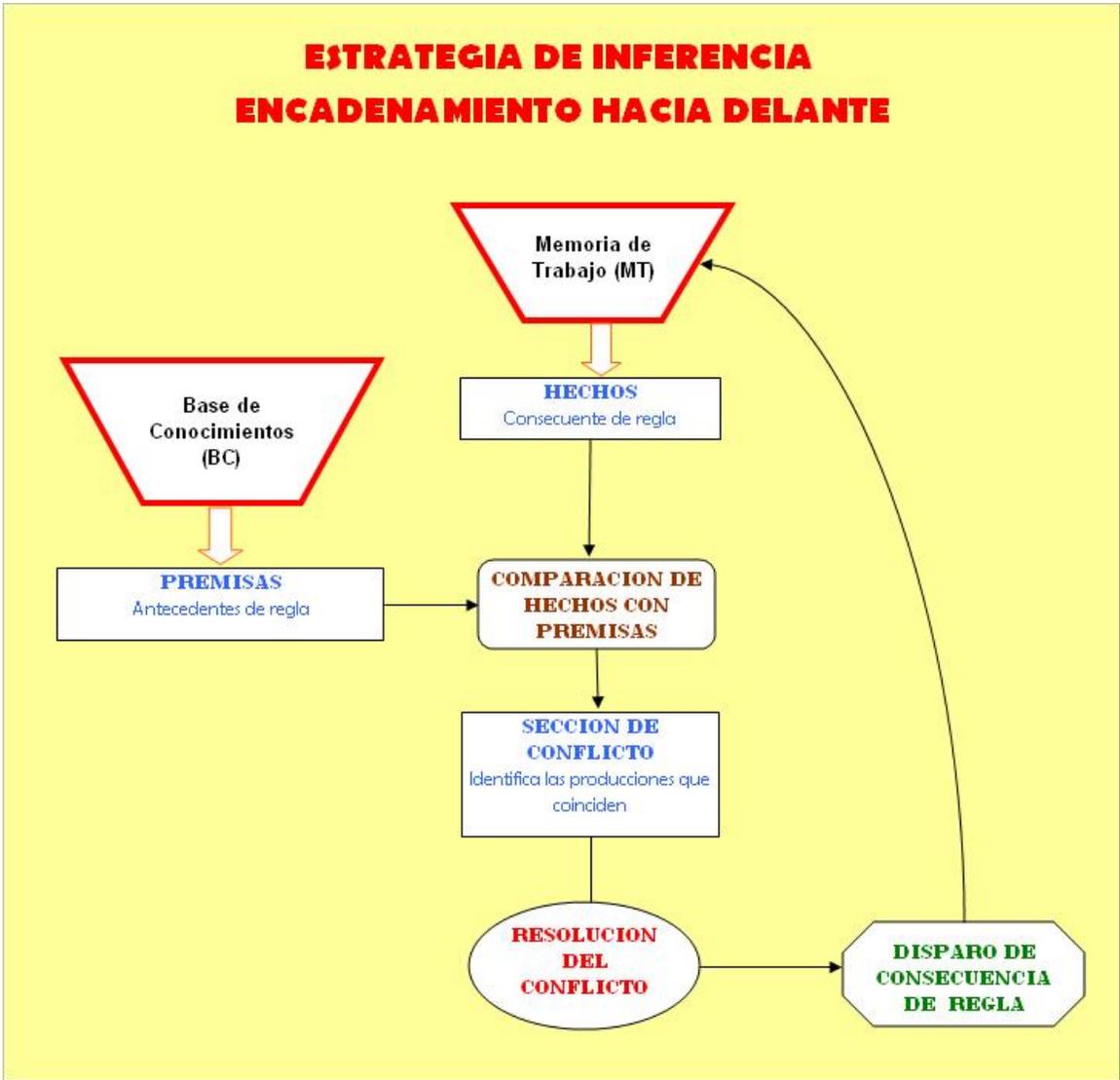


FIGURA 24. Estrategia de inferencia encadenamiento hacia delante.

ESQUEMA DISEÑO DE INFERENCIA ENCADENAMIENTO HACIA DELANTE

ESTADO INICIAL

BASE DE CONOCIMIENTOS						
IdBC	Tipo Inst	Regla	Num Regla	Salida	Resp	Disparo
1	premisa	El paciente tiene llagas en la garganta	1	¿Tiene el paciente llagas en la garganta?		0
2	premisa	Se sospecha una infección bacterial	1	¿Se sospecha una infección bacterial?		0
3	hecho	El paciente tiene un estreptococal	1			0
4	premisa	La temperatura del paciente es mayor que 100	2	¿Tiene el paciente una temperatura mayor que 100?		0
5	hecho	El paciente tiene fiebre	2			0
6	premisa	El paciente ha estado enfermo por más de 1 mes	3	¿Ha estado enfermo por más de 1 mes?		0
7	hecho	Se sospecha una infección bacterial	3			0

MEMORIA DE TRABAJO			
IdMT	IdBC	Hecho	Num Regla

1er Paso

BASE DE CONOCIMIENTOS						
IdBC	Tipo Inst	Regla	Num Regla	Salida	Resp	Disparo
1	premisa	El paciente tiene llagas en la garganta	1	¿Tiene el paciente llagas en la garganta?	si	1
2	premisa	Se sospecha una infección bacterial	1	¿Se sospecha una infección bacterial?		0
3	hecho	El paciente tiene un estreptococal	1			0
4	premisa	La temperatura del paciente es mayor que 100	2	¿Tiene el paciente una temperatura mayor que 100?		0
5	hecho	El paciente tiene fiebre	2			0
6	premisa	El paciente ha estado enfermo por más de 1 mes	3	¿Ha estado enfermo por más de 1 mes?		0
7	hecho	Se sospecha una infección bacterial	3			0

MEMORIA DE TRABAJO			
IdMT	IdBC	Hecho	Num Regla

2º Paso

BASE DE CONOCIMIENTOS						
IdBC	Tipo Inst	Regla	Num Regla	Salida	Resp	Disparo
1	premisa	El paciente tiene llagas en la garganta	1	¿Tiene el paciente llagas en la garganta?	si	1
2	premisa	Se sospecha una infección bacterial	1	¿Se sospecha una infección bacterial?	no	0
3	hecho	El paciente tiene un estreptococal	1			0
4	premisa	La temperatura del paciente es mayor que 100	2	¿Tiene el paciente una temperatura mayor que 100?		0
5	hecho	El paciente tiene fiebre	2			0
6	premisa	El paciente ha estado enfermo por más de 1 mes	3	¿Ha estado enfermo por más de 1 mes?		0
7	hecho	Se sospecha una infección bacterial	3			0

MEMORIA DE TRABAJO			
IdMT	IdBC	Hecho	Num Regla

3º Paso

BASE DE CONOCIMIENTOS						
IdBC	Tipo Inst	Regla	Num Regla	Salida	Resp	Disparo
1	premisa	El paciente tiene llagas en la garganta	1	¿Tiene el paciente llagas en la garganta?	si	1
2	premisa	Se sospecha una infección bacterial	1	¿Se sospecha una infección bacterial?	no	0
3	hecho	El paciente tiene un estreptococal	1			0
4	premisa	La temperatura del paciente es mayor que 100	2	¿Tiene el paciente una temperatura mayor que 100?	si	0
5	hecho	El paciente tiene fiebre	2			0
6	premisa	El paciente ha estado enfermo por más de 1 mes	3	¿Ha estado enfermo por más de 1 mes?		0
7	hecho	Se sospecha una infección bacterial	3			0

MEMORIA DE TRABAJO			
IdMT	IdBC	Hecho	Num Regla

4º Paso

BASE DE CONOCIMIENTOS						
IdBC	Tipo Inst	Regla	Num Regla	Salida	Resp	Disparo
1	premisa	El paciente tiene llagas en la garganta	1	¿Tiene el paciente llagas en la garganta?	si	1
2	premisa	Se sospecha una infección bacterial	1	¿Se sospecha una infección bacterial?	no	0
3	hecho	El paciente tiene un estreptococal	1			0
4	premisa	La temperatura del paciente es mayor que 100	2	¿Tiene el paciente una temperatura mayor que 100?	si	1
5	hecho	El paciente tiene fiebre	2			1
6	premisa	El paciente ha estado enfermo por más de 1 mes	3	¿Ha estado enfermo por más de 1 mes?		0
7	hecho	Se sospecha una infección bacterial	3			0

MEMORIA DE TRABAJO			
IdMT	IdBC	Hecho	Num Regla
1	5	El paciente tiene fiebre	2

El motor de inferencia verifica si en la Base de Conocimientos existe alguna premisa que coincida con el hecho recién disparado. Para el caso no



FIGURA 25. Interacción entre las estructuras BC y MT durante el proceso de inferencia.

4.3. LENGUAJE PARA CODIFICAR EL CONOCIMIENTO

Este módulo se encarga de crear los mecanismos necesarios de análisis gramatical (análisis léxico, análisis sintáctico, análisis semántico) para que el conocimiento pueda ser codificado en forma de reglas de producción. La representación del conocimiento en forma de reglas de producción fue propuesta por Post en 1943 y es la forma más común de representar el conocimiento, debido a su gran sencillez por ser la formulación más inmediata del principio de causalidad (causa-efecto).

Según una encuesta realizada en 1986 en Japón por el JIPDEC³ sobre la creación de empresas que realizaban sistemas expertos durante esa época, los sistemas de representación del conocimiento más utilizados eran los siguientes (Ver tabla 9):

METODOS	PORCENTAJE DE USO
Reglas de producción	65.3%
Marcos	42.9%
Redes semánticas	7.1%
Otros	11.2%

Tabla 9. Porcentaje de Uso de los Métodos de Sistemas de representación del conocimiento más utilizados.

La generación de código para representar el conocimiento está relacionada con las teorías de creación de compiladores. Esto incluye la creación de autómatas finitos para el analizador léxico, la invención de una gramática libre de contexto para el analizador sintáctico que se pueda codificar de la forma más natural posible (es decir que se parezca lo más posible al lenguaje natural utilizado por las personas diariamente), la generación de una gramática de atributos que pueda gestionar los errores semánticos y por fin la codificación de una máquina virtual que pueda convertir el lenguaje fuente (reglas de producción expresadas en lenguaje natural) a un lenguaje objeto (código intermedio que pueda ser gestionado convenientemente por la Base de Conocimiento).

³ JIPDEC = Japan Information Processing Development Center.

La figura 26 muestra en forma general el funcionamiento del intérprete creado para la herramienta NIVEL9.

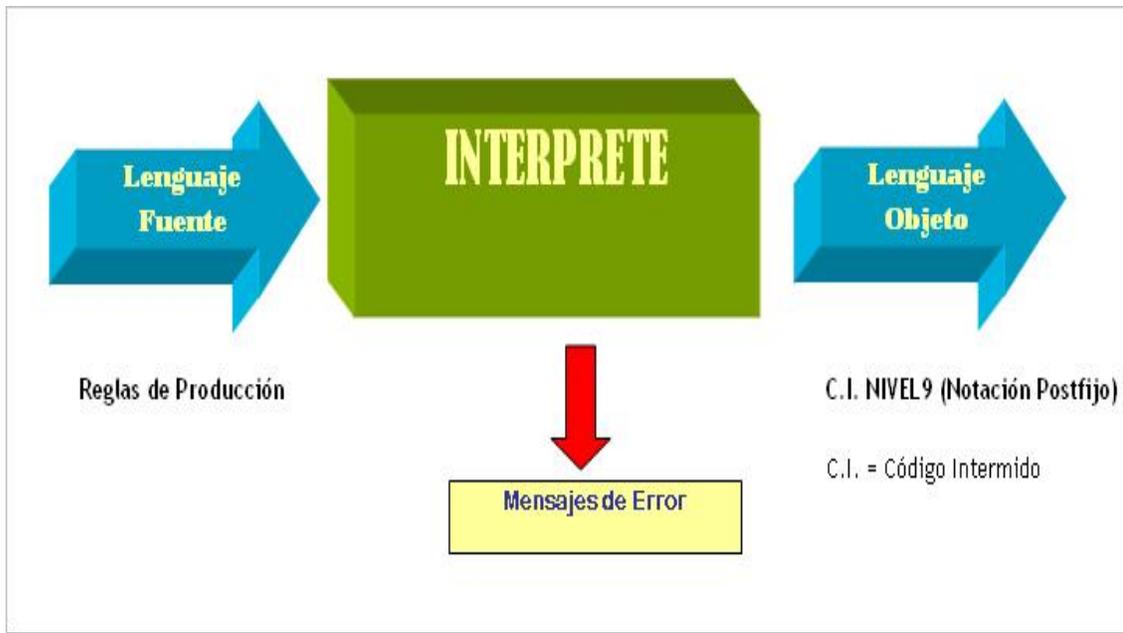


FIGURA 26. Esquema del intérprete utilizado por NIVEL9.

4.3.1. FASES DEL INTERPRETE NIVEL9

El intérprete del lenguaje puede descomponerse en dos fases, las cuales transformarán al programa fuente de una representación a otra, teniendo en cada fase una serie de tareas fundamentales. La fase que analiza la entrada y genera estructuras intermedias se llama FASE DE ANÁLISIS, y la fase que sintetiza la salida se denomina FASE DE SÍNTESIS. La Figura 27, muestra un diagrama de las fases de un intérprete/compilador.

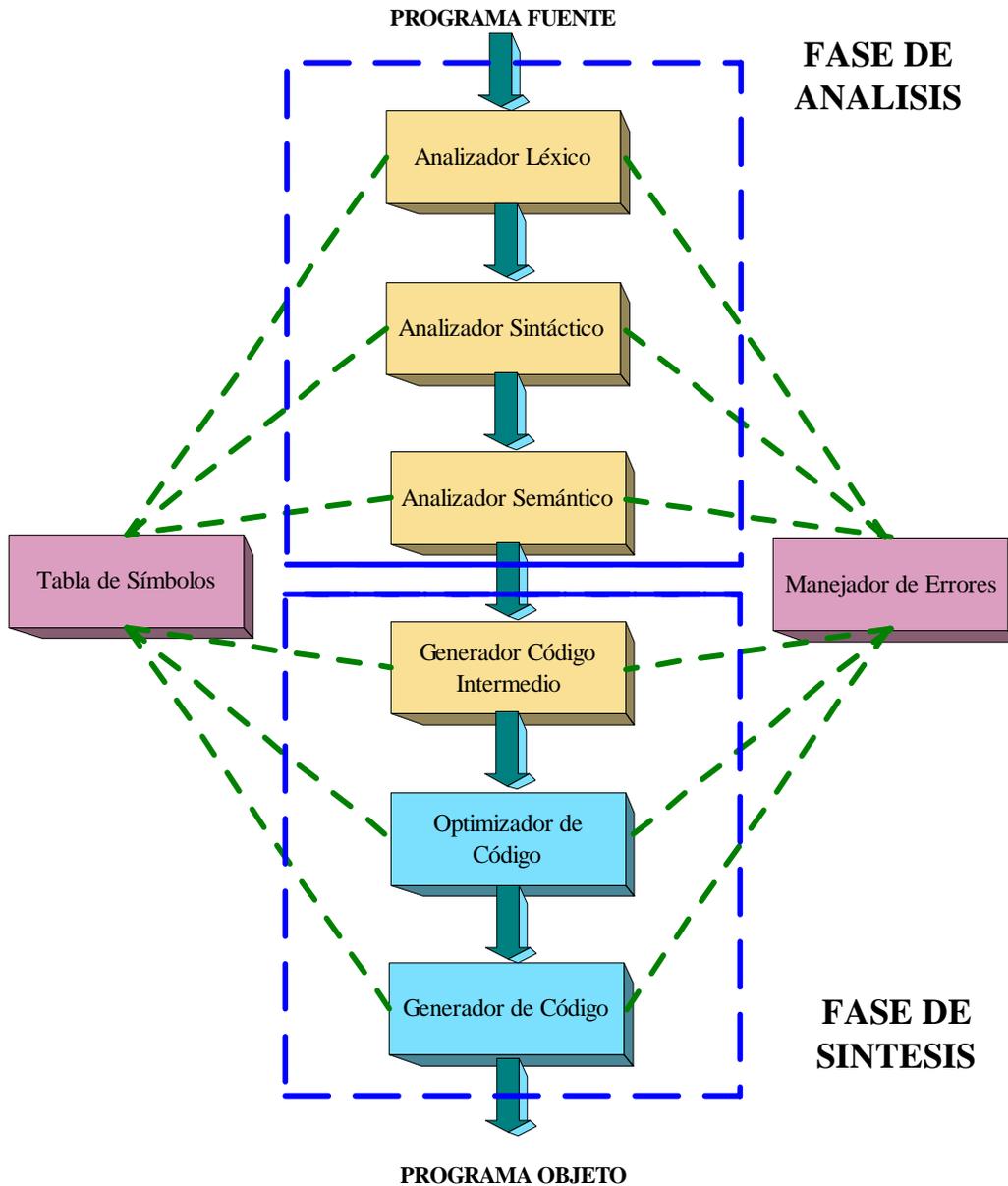


FIGURA 27. Fases de un INTERPRETE/COMPILADOR.

4.3.1.1. Fase de Análisis

La Fase de Análisis tiene como objetivos el controlar la corrección del programa fuente y generar las estructuras necesarias para comenzar la síntesis, los que se llevan a cabo realizando las tareas siguientes:

- **Análisis Léxico:**

Este divide el programa fuente en los componentes básicos: números, identificadores de usuario (variables, constantes, tipos, nombres de procedimientos, ...), palabras reservadas, signos de puntuación, y otros símbolos, asociándole a cada uno de estos la categoría a la que pertenece.

Este reconoce los componentes denominados "Tokens", agrupa secuencias de caracteres desde el programa fuente de entrada en unidades llamadas componentes léxicos. Los tokens (componentes léxicos) son las unidades básicas del mismo nodo en que las palabras y los signos de puntuación son las unidades básicas de una oración. La elección de los tokens es una tarea del diseñador del compilador o intérprete.

Además de reconocer las palabras claves, identificadores, operadores, etc., el analizador léxico realiza las tareas de eliminar espacios en blanco y comentarios, así como también identificar y detectar errores sencillos como los de ortografía.

- **Análisis Sintáctico:**

Con ayuda de reglas gramaticales, comprueba que la estructura de los componentes básicos sea la correcta.

Este analizador es más complicado que el analizador léxico. Este agrupa los tokens en estructuras sintácticas en forma muy similar a las estructuras gramaticales de las oraciones de un lenguaje estructurado, identificando las diferentes partes de la oración que lo componen. De esta forma el analizador léxico reconoce los tokens que se encuentran agrupados en una expresión, lo que puede llegar a determinar una

sentencia de asignación; determinando a su vez si la secuencia de tokens es sintácticamente correcta, de acuerdo a la definición del lenguaje.

- **Análisis Semántico:**

Comprueba todo lo relacionado con el significado, chequeo de tipos, rangos de valores, existencia de variables, números de regla repetidos, etc.

Este analizador determina el significado del programa mediante la creación de tablas de símbolos, verificando cuáles de las variables utilizadas han sido definidas y una infinidad de tareas. Dos de las tareas principales son:

- a) la verificación estática del programa y
- b) la generación de una representación intermedia.

4.3.1.2. Fase de Síntesis

La Fase de Síntesis tiene como objetivo el construir el programa objeto a partir de las estructuras generadas por la fase de análisis, el que se lleva a cabo realizando las siguientes tareas:

- **Generación de código intermedio:**

Genera un código independiente de la máquina que facilita la optimización de código.

El código intermedio consiste en una estructura de datos que representa el programa fuente durante la traducción a lenguaje objeto, generando una linealización del árbol sintáctico, es decir, una representación del

árbol sintáctico en forma lineal. NIVEL9 utiliza como forma de representar el código intermedio el Código P (también conocido como Máquina P o Máquina Pila). El diseño del Código P para el software involucra principalmente a la Máquina P (MP) la cual ya se explico anteriormente, y esta involucra a las estructuras siguientes:

- Tabla de Datos (TD)
- Tabla de Cadenas (TC)

Las cuales juegan el papel de pequeñas memorias que almacenan información importante que la MP utiliza para gestionar su traducción en forma directa.

Ejemplo:

Como un ejemplo sencillo de código P, se analiza la siguiente expresión:

$$20 * A + (B - 30)$$

El árbol sintáctico para esta expresión se representaría de la siguiente forma:

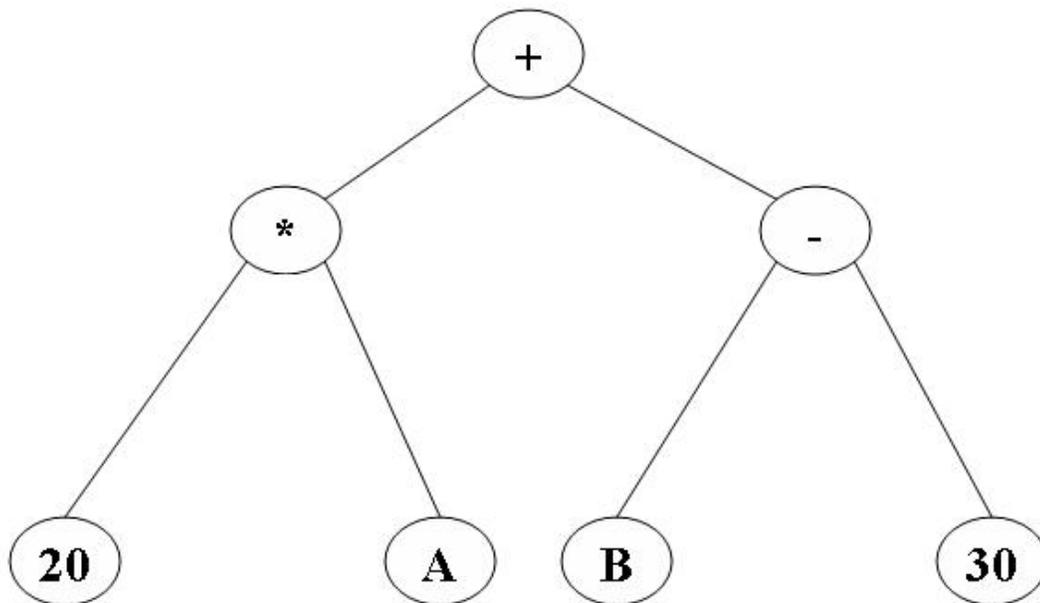


FIGURA 28. Árbol sintáctico para representar: $20 * A + (B - 30)$

La versión de Código P de NIVEL9 para la expresión anterior se muestra a continuación:

MP_Apila(x) → MP_Apila(20) → Carga el número 20
MP_Apilar(x) → MP_Apila(A) → Carga el valor de la variable A
*MP_Multiplica(x, y) → MP_Multiplica(20, A) → Multiplicación de números (20*A)*
MP_Apila(x) → MP_Apila(B) → Carga el valor de la variable B
MP_Apilar(x) → MP_Apila(30) → Carga el número 30
MP_Resta(x, y) → MP_Resta(B, 30) → Sustracción de números (B-30)
*MP_Suma(x, y) → MP_Suma((20*A), (B-30)) → Suma de números (20*A) + (B-30)*

Estas instrucciones representan las operaciones en la Máquina P (MP). En primer lugar se inserta el valor 20 en la pila temporal. Luego se inserta el valor de la variable A en la pila. La instrucción MP_Multiplica(x, y) toma como parámetros los dos valores de la pila, los multiplica (en orden inverso) e inserta el resultado en la pila. Las siguientes dos instrucciones (MP_Apila(B) y MP_Apila(30)) insertan el valor de B y la constante 30 en la pila (Ahora se tienen tres valores en la pila). Posteriormente, la instrucción MP_Resta(x, y) toma de parámetros los dos valores superiores de la pila, resta el primero del segundo, e inserta el resultado. Finalmente, la instrucción MP_Suma(x, y) extrae los dos valores restantes de la pila, los suma e inserta el resultado. El código finaliza con un solo valor en la pila, que representa el resultado del cálculo.

Los valores de las variables A y B son extraídos de la Tabla de Datos (TD) la cual se encarga de buscarlos a través de sus nombres (A y B).

En la figura 20 y en la sección de las estructuras de datos, mostrada en párrafos anteriores, se muestra otro ejemplo en forma gráfica y clara sobre el funcionamiento de la Máquina P.

Para las expresiones relacionales y lógicas la Máquina P (MP) va almacenando los enteros 1 ó 0 dependiendo si el resultado de las expresiones es verdadero o falso, respectivamente.

Cada expresión tiene su instrucción en el Código P la cual se encarga de gestionar el resultado de las expresiones.

Ejemplo:

$(20 < B) \text{ Y } (B == 50)$

El código P generado sería el siguiente:

MP_Apila(x) → MP_Apila(20) → Carga el número 20

MP_Apila(x) → MP_Apila(B) → Carga el valor de la variable B

MP_MenorQue(x, y) → MP_MenorQue(20, B) → Gestiona si 20 es menor que B

MP_Apila(x) → MP_Apila(20 < B) → Carga el resultado de la instrucción anterior

MP_Apila(x) → MP_Apila(B) → Carga el valor de la variable B

MP_Apila(x) → MP_Apila(50) → Carga el número 50

MP_IgualQue(x, y) → MP_IgualQue(B, 50) → Gestiona si B es igual a 50

MP_Y(x, y) → MP_Y((20 < B), (B == 50)) → Gestiona el operador AND para (20 < B) y (B == 50)

Este caso es bastante similar que el anterior en cuanto al funcionamiento de la Máquina P (MP), sin embargo los resultados almacenados por las instrucciones de tipo relacional y las instrucciones de tipo lógico son enteros que simbolizan un resultado de tipo booleano, es decir, cierto o falso, los cuales se representa por los enteros 1 ó 0 en la MP, respectivamente. La instrucción *MP_MenorQue(x, y)* que toma como parámetros los valores almacenados en la pila (20 y B) generan como resultado un 1 o un 0 dependiendo del valor de la variable B. Si el valor de B es 50 entonces el resultado devuelto será 1, este uno se almacenará en la cima de la pila. La instrucción *MP_IgualQue(x, y)* funciona de la misma forma, esta se encarga de comparar los dos últimos valores almacenados en la pila. Si el valor de la variable B es 50 entonces el resultado de la instrucción será 1. Finalmente la instrucción lógica (*MP_Y(x, y)*) se encarga de gestionar los resultados devueltos por las dos instrucciones anteriores operando de esta forma los dos enteros restantes de la pila (1 y 1) para almacenar en la pila el resultado que sería 1 (1 AND 1 → 1) → (true AND true → true).

Las instrucciones lógicas también son capaces de operar producciones, de manera que los parámetros de la instrucción $MP_Y(x, y)$ pueden ser proposiciones, las cuales son gestionadas en forma dinámica por el usuario en tiempo de ejecución.

Ejemplo:

¿Todos los hombres son mortales? Y ¿Mauricio es hombre?

La forma en que NIVEL9 gestiona la producción anterior es la siguiente:

*¿Todos los hombres son mortales? → Proposición (producción)
Respuesta del usuario: Sí → Gestión dinámica de la producción
MP_Apila(x) → MP_Aplia(1) → Carga el resultado booleano 1
¿Mauricio es hombre? → Proposición (producción)
Respuesta del usuario: Sí → Gestión dinámica de la producción
MP_Apila(x) → MP_Aplia(1) → Carga el resultado booleano 1
MP_Y(x, y) → MP_(1, 1) → Gestiona el operador AND para 1 y 1*

Para las instrucciones de control como el SI/ENTONCES/SI_FIN o el MIENTRAS/HACER/MIENTRAS_FIN el código intermedio tiene funciones especiales que gestionan estas operaciones, dependiendo de los resultados generados por las operaciones lógicas dentro de las sentencias almacenadas en la Máquina P (MP). Para el caso de la instrucción SI, las expresiones o producciones que se evalúan dentro de las instrucciones propias de la sentencia produciendo unos o ceros. Si los resultados son verdaderos las conclusiones posteriores a la palabra clave ENTONCES son almacenadas en la Memoria de Trabajo (MT) y en el caso opuesto simplemente las conclusiones no se almacenan.

- **Optimización:**

Esta se realiza durante la generación de código intermedio y durante la generación de código para la BC, pero ambos procesos son independientes entre sí. La optimización del código intermedio se produce al generarse una gramática que elimina la recursión de las producciones y que facilita

las operaciones de la pila en forma directa y de una sola pasada. Una vez que las variables o constantes de la pila han sido operadas estas son eliminadas de la pila, de esta forma no se utiliza memoria adicional y las operaciones aritméticas relacionales y lógicas se realizan de forma más eficiente. Las instrucciones para la gestión de las expresiones lógicas utilizan los algoritmos denominados de 'mínimo esfuerzo', es decir, que en vez de evaluar toda la tabla de verdad para obtener el resultado sólo evalúan si los dos parámetros tomados por la función son iguales para saber si el resultado es cierto o falso.

Ejemplo:

MP_Y(x, y) → Gestiona el operador AND lógico

Si los parámetros de MP_Y(x, y) son ambos unos, entonces devuelve 1 de lo contrario devuelve 0.

MP_Y(x, y) → MP_Y(1, 1) → 1

MP_Y(x, y) → MP_Y(1, 0) → 0

MP_Y(x, y) → MP_Y(0, 1) → 0

MP_Y(x, y) → MP_Y(0, 0) → 0

MP_O(x, y) → Gestiona el operador OR lógico

Si los parámetros de MP_Y(x, y) son ambos ceros, entonces devuelve 0 de lo contrario devuelve 1.

MP_Y(x, y) → MP_Y(0, 0) → 0

MP_Y(x, y) → MP_Y(1, 0) → 1

MP_Y(x, y) → MP_Y(0, 1) → 1

MP_Y(x, y) → MP_Y(1, 1) → 1

La optimización al generarse código para la BC consiste en la creación de etiquetas que se relacionan mutuamente entre las demás estructuras de datos del software. Al existir identificadores que apuntan a las posiciones de las estructuras relacionadas con la BC es posible generar punteros que busquen directamente en memoria dicha información en vez de realizar búsquedas que entorpecerían el rendimiento del intérprete.

En la sección de las estructuras de datos se explica las relaciones entre estructuras con mayor detalle.

- **Generación de código para la BC:**

En los compiladores tradicionales esta etapa se encarga de generar el ejecutable “.exe” del programa directamente o la generación de un archivo “.obj”. Para el caso de NIVEL9 esta etapa se encarga de generar un código adecuado que pueda ser gestionado por la Base de Conocimientos (BC). Para el caso de NIVEL9 el código intermedio es interpretado por otras funciones las cuales se encargan de ir colocando los valores en forma ordenada en la Base de Conocimientos para que esta ha su vez pueda ser gestionada por el motor de inferencia.

4.3.2. DEFINICIÓN DEL LENGUAJE DE PROGRAMACIÓN

Tabla 10. Descripción formal del lenguaje léxico
SIMBOLOS UTILIZADOS POR EL LENGUAJE

LEXEMA	COMPONENTE LEXICO	TIPO
" "	NINGUNO	Nulo
"+"	SUMA	Aritmético
"-"	RESTA	Aritmético
"*"	MULTIPLICACION	Aritmético
"/"	DIVISIÓN	Aritmético
"^"	POTENCIA	Aritmético
"!"	NEGATIVO	Lógico
"<<"	SALIDA	Entrada/Salida
">>"	ENTRADA	Entrada/Salida
"=="	IGUAL	Relacional
"<="	MENOR_IGUAL	Relacional
">="	MAYOR_IGUAL	Relacional
"<"	MENOR	Relacional
">"	MAYOR	Relacional
"<>"	DIFERENTE	Relacional
"="	ASIGNACION	Asignación
"Y"	Y	Lógico/Reservada
"y"	Y	Lógico/Reservada
"O"	O	Lógico/Reservada
"o"	O	Lógico/Reservada

"("	PARENTESIS_IZQ	Separador
")"	PARENTESIS_DER	Separador
" ,"	COMA	Separador
"."	PUNTO	Separador
"{"	LLAVE_IZQ	Delimitador
"}"	LLAVE_DER	Delimitador
"//"	COMENTARIO	Delimitador
"/*"	COMENTARIO_IZQ	Delimitador
"*/"	COMENTARIO_DER	Delimitador
"¿"	INTERROGACION_IZQ	Delimitador
"?"	INTERROGACION_DER	Delimitador
"\$"	DÓLAR	Delimitador
""	COMILLA_DOBLE	Cadena
""	COMILLA_SIMPLE	Cadena
". ,"	PUNTO_COMA	Terminador
"regla"	REGLA	Reservada
"REGLA"	REGLA	Reservada
"salida"	SALIDA	Reservada
"SALIDA"	SALIDA	Reservada
"entrada"	ENTRADA	Reservada
"ENTRADA"	ENTRADA	Reservada
"resp"	RESP	Reservada
"RESP"	RESP	Reservada
"imagen"	IMAGEN	Reservada
"IMAGEN"	IMAGEN	Reservada
"mensaje"	MENSAJE	Reservada
"MENSAJE"	MENSAJE	Reservada
"ir_a_regla"	IR_A_REGLA	Reservada
"IR_A_REGLA"	IR_A_REGLA	Reservada
"eliminar_reglas"	ELIMINAR_REGLAS	Reservada
"ELIMINAR_REGLAS"	ELIMINAR_REGLAS	Reservada
"num"	NUM	Reservada
"NUM"	NUM	Reservada
"cad"	CAD	Reservada
"CAD"	CAD	Reservada
"si"	SI	Reservada
"SI"	SI	Reservada
"sino"	SINO	Reservada
"SINO"	SINO	Reservada
"si_fin"	SI_FIN	Reservada
"SI_FIN"	SI_FIN	Reservada
"entonces"	ENTONCES	Reservada
"ENTONCES"	ENTONCES	Reservada
"mientras"	MIENTRAS	Reservada
"MIENTRAS"	MIENTRAS	Reservada
"mientras_fin"	MIENTRAS_FIN	Reservada
"MIENTRAS_FIN"	MIENTRAS_FIN	Reservada
"hacer"	HACER	Reservada
"HACER"	HACER	Reservada

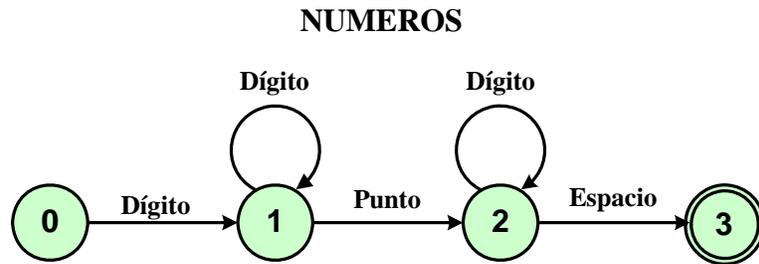
"inicio"	INICIO	Reservada
"INICIO"	INICIO	Reservada
"fin"	FIN	Reservada
"FIN"	FIN	Reservada
"terminar"	TERMINAR	Reservada
"TERMINAR"	TERMINAR	Reservada

4.3.2.1. AUTOMATAS

Un autómata describe las acciones necesarias ha realizar para reconocer un token. El autómata es una gráfica dirigida cuyos nodos representan los estados y las líneas adyacentes representan los caracteres de entrada o salida de cada estado.

Para la realización del compilador, se han utilizado tres autómatas para el reconocimiento de NUMEROS, COMENTARIOS SIMPLES y COMENTARIOS COMPUESTOS.

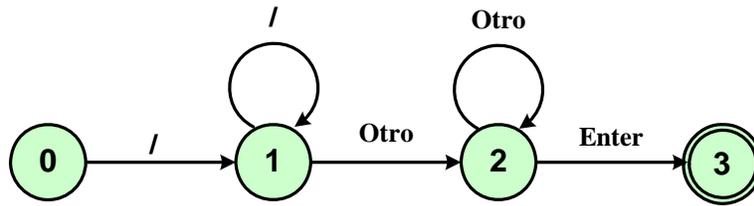
A continuación se representas cada uno de los autómatas utilizados, acompañados de sus respectivas tablas de comportamiento.



ESTADOS	0 DIGITO	1 PUNTO	2 ESPACIO
0	1	-1	-1
1	1	2	-1
2	2	-1	3
3	-1	-1	-1

FIGURA 29. Autómata para NUMEROS

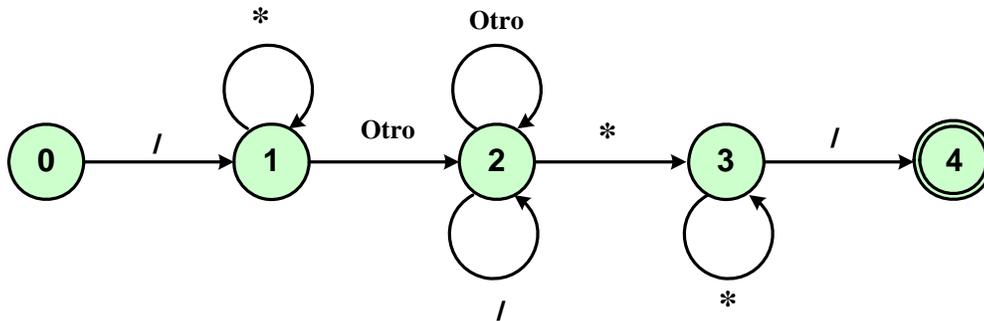
COMENTARIO SIMPLE



ESTADOS	0 /	1 OTRO	2 ENTER
0	1	-1	-1
1	1	2	-1
2	1	2	3

FIGURA 30. Autómata para COMENTARIO SIMPLE

COMENTARIO COMPUESTO



Para el comentario simple, el "otro", representa cualquier carácter diferente a la "/" y a la tecla "ENTER". Para el comentario compuesto, el "otro", representa cualquier carácter diferente a la "/" y al "*".

ESTADOS	0 /	1 *	2 OTRO
0	-1	1	-1
1	1	1	2
2	1	3	2
3	4	3	-1
4	-1	-1	-1

FIGURA 31. Autómata para COMENTARIO COMPUESTO

ESPACIO EN BLANCO

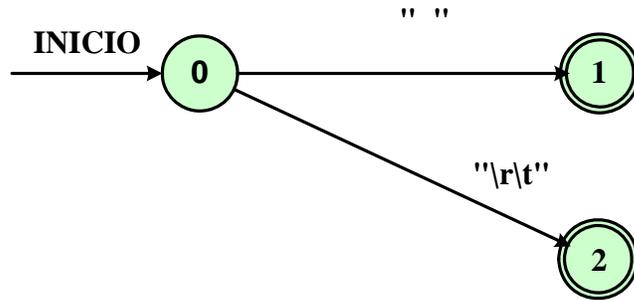


FIGURA 32. Autómata para ESPACIO EN BLANCO

SALTO DE LINEA

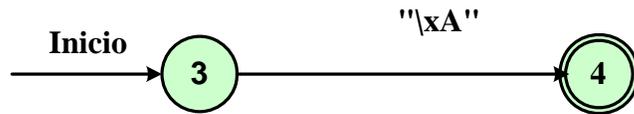


FIGURA 33. Autómata para SALTO DE LINEA

CADENA DE CARACTERES

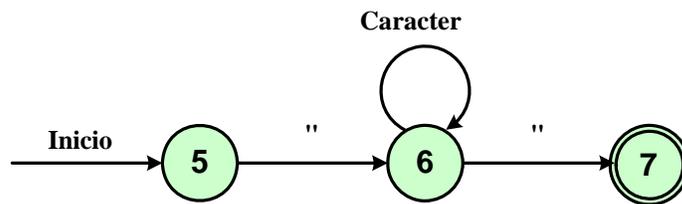


FIGURA 34. Autómata para CADENA DE CARACTERES

Símbolos Separadores

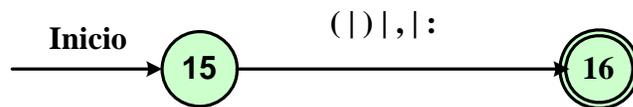


FIGURA 35. Autómata para CARACTER ASCII

Símbolo Negativo

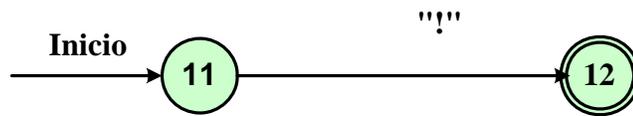


FIGURA 36. Autómata para SÍMBOLO NEGATIVO

Operadores Aritméticos

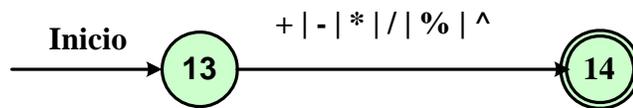


FIGURA 37. Autómata para OPERADORES ARITMETICOS

Símbolos Separadores

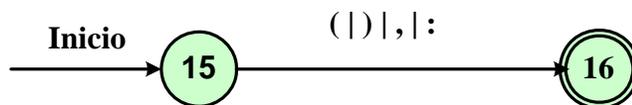


FIGURA 38. Autómata para SÍMBOLOS SEPARADORES

Símbolo Terminador

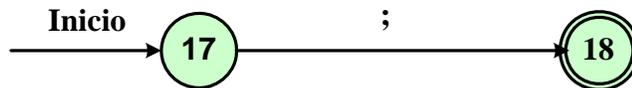


FIGURA 39. Autómata para SÍMBOLO TERMINADOR

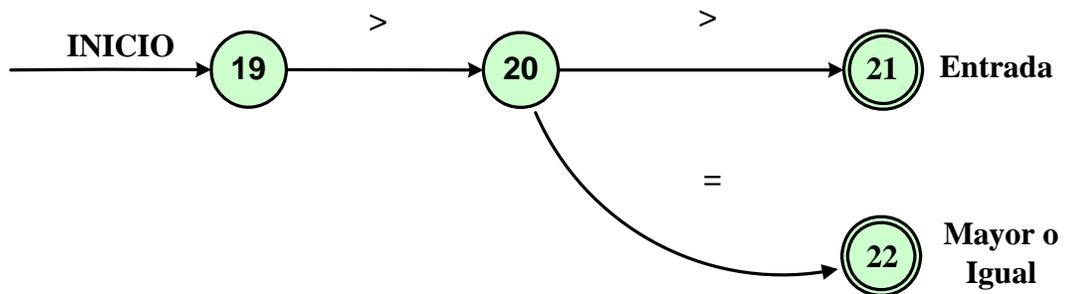


FIGURA 40. Autómata para OPERADORES 1

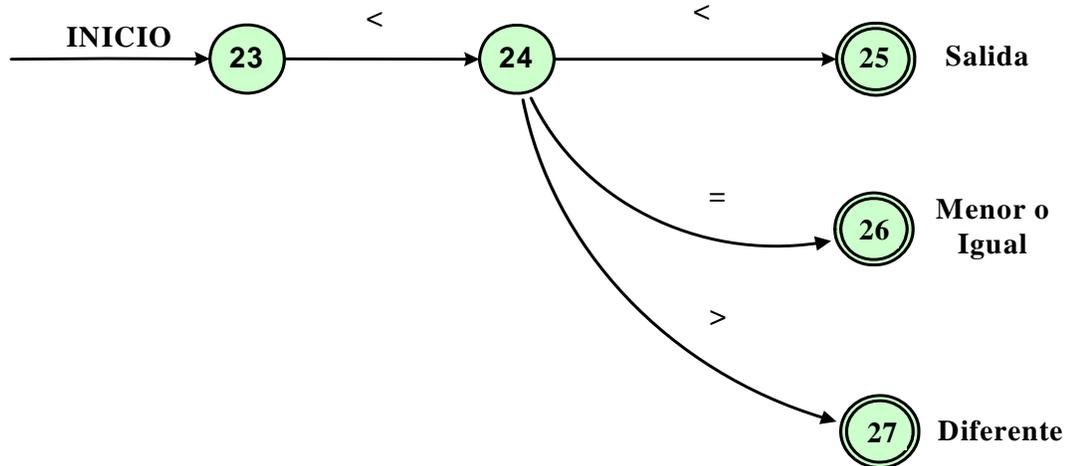


FIGURA 41. Autómata para OPERADORES 2

Asignación

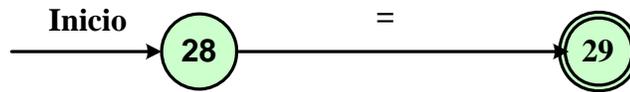


FIGURA 42. Autómata para OPERADOR ASIGNACION

4.3.3. DESCRIPCIÓN FORMAL DE LA ESTRUCTURA SINTÁCTICA DEL LENGUAJE

GRAMATICA LIBRE DE CONTEXTO LENGUAJE "NIVEL9"

(ELIMINANDO RECURSION)

$P \rightarrow \text{INICIO} \{ \text{bloque_sentencias } \text{bloque_reglas} \} \text{FIN}$
 $\text{bloque_sent_prop} \rightarrow \text{bloque_sentencias } \text{bloque_sent_prop} \mid \epsilon$
 $\text{bloque_sent_prop} \rightarrow \text{bloque_proposiciones } \text{bloque_sent_prop} \mid \epsilon$
 $\text{bloque_sentencias} \rightarrow \text{sentencia } \text{bloque_sentencias} \mid \epsilon$
 $\text{bloque_proposiciones} \rightarrow \text{proposición}; \text{bloque_proposiciones} \mid \epsilon$
 $\text{bloque_reglas} \rightarrow \text{regla } \text{bloque_reglas} \mid \epsilon$
 $\text{proposición} \rightarrow \text{¿cadena?}$

$\text{sentencia} \rightarrow \text{sentencia_asignación}$
 $\text{sentencia} \rightarrow \text{sentencia_declaración}$
 $\text{sentencia} \rightarrow \text{sentencia_entrada}$
 $\text{sentencia} \rightarrow \text{sentencia_salida}$
 $\text{sentencia} \rightarrow \text{sentencia_ir_a_regla}$
 $\text{sentencia} \rightarrow \text{sentencia_parada}$
 $\text{sentencia} \rightarrow \text{sentencia_imagen}$
 $\text{sentencia} \rightarrow \text{sentencia_rango_respuestas}$

sentencia_asignación → **\$ID\$** = expresión_aritmética;
sentencia_declaración → tipo **\$ID\$** lista_variables

lista_variables → lista_variables_numéricas | lista_variables_cadenas
lista_variables_numéricas → ;
lista_variables_numéricas → , **\$ID\$** lista_variables_numéricas
lista_variables_numéricas → = expresión_aritmética;
lista_variables_cadena → ;
lista_variables_cadena → , **\$ID\$** lista_variables_cadena
lista_variables_cadena → = "cadena";
sentencia_salida → **ENTRADA** >> **\$ID\$**;
sentencia_salida → **SALIDA** << lista_salida
lista_salida → "cadena" final_salida
lista_salida → **\$ID\$** final_salida
final_salida → ;
final_salida → << lista_salida

sentencia_ir_a_regla → **IR_A_REGLA** : número;
sentencia_mensaje → **MENSAJE** : CADENA;
sentencia_eliminar_reglas → **ELIMINAR_REGLAS** [número_entero : número_entero];
sentencia_parada → **TERMINAR**;
sentencia_imagen → **IMAGEN** : "cadena";
sentencia_rango_respuestas → **RESP** : (lista_cadena);
lista_cadena → "cadena", lista_cadena
lista_cadena → "cadena"

regla → **REGLA** : número_entero peso { enunciado_si } bloque_reglas
peso → [número_real]
peso → ε
enunciado_si → **SI** expresión **ENTONCES** bloque_sent_prop enunciado_sino **SI_FIN**
enunciado_mientras → **MIENTRAS** expresión **HACER** bloque_sent_prop **MIENTRAS_FIN**
enunciado_sino → **SINO** bloque_sent_prop
enunciado_sino → ε
lista_sentencias → (bloque_sentencias)
lista_sentencias → ε

expresión → expresión_lógica | expresión_relacional | expresión_aritmética
expresión_lógica → expresión_relacional más_exp_lógica
expresión_lógica → proposición más_exp_lógica
más_exp_lógica → **Y** expresión_relacional; lista_sentencias más_exp_lógica
más_exp_lógica → **O** expresión_relacional; lista_sentencias más_exp_lógica
más_exp_lógica → **Y** proposición lista_sentencias más_exp_lógica
más_exp_lógica → **O** proposición lista_sentencias más_exp_lógica
más_exp_lógica → ε
expresión_relacional → expresión_aritmética más_exp_relacional
más_exp_relacional → < expresión_aritmética
más_exp_relacional → > expresión_aritmética
más_exp_relacional → <= expresión_aritmética
más_exp_relacional → >= expresión_aritmética
más_exp_relacional → <= expresión_aritmética
más_exp_relacional → == expresión_aritmética
más_exp_relacional → <> expresión_aritmética
más_exp_relacional → ε
expresión_aritmética → término más_term
más_term → + término más_term

más_term → - término más_term
 más_term → ε
 término → factor más_fact
 más_fact → * factor más_fact
 más_fact → / factor más_fact
 más_fact → ^ factor más_fact
 más_fact → ε
 factor → (expresión_aritmética)
 factor → **negativo** factor
 factor → **\$ID\$**
 factor → **número**
 proposición → ¿cadena?
 tipo → **NUM** | **CAD**

ID → **carácter**

número → **número_entero** | **número_real**

número_entero → **dígito** (**dígito**)*

número_real → **número_entero.número_entero** | **.número_entero** | ε

cadena → **carácter** (**carácter**)*

carácter → **letra** | **dígito** | **+** | **-** | ***** | **/** | **^** | **%** | **,** | **;** | **.** | **:** | **=** | **(** | **)** | **{** | **}** | **<** | **>** | **!** | **|** | **¿** | **?** | **_**

| **"** | **[** | **]** | **#** | **\$**

letra → [**a - z** , **A - Z**]

dígito → [**0 - 9**]

guión → **_**

negativo → **!**

NOMENCLATURA :

Símbolos Terminales – Color Rojo

Símbolos No Terminales – Color Negro

Componentes Léxicos – Color Azul

Al analizar la gramática libre de contexto del lenguaje NIVEL9 es claro que se ha creado un lenguaje mucho más potente que la representación de simple reglas de producción (*SI/ENTONCES/SI_FIM*). NIVEL9 posee instrucciones especiales que permiten al usuario mostrar imágenes pertinentes en cada regla (*IMAGEN*), así como instrucciones que permiten mostrar opciones de las posibles respuestas que el usuario puede enviar al sistema en los casos en que haya interacción dinámica con el este (*RESP*), o hasta instrucciones especiales para crear ciclos repetitivos como *MIENTRAS/HACER/MIENTRAS_FIN* ó *IR_A_REGLA* (lo cual no es posible en la mayoría de sistemas expertos ya que esta instrucción se sale de la definición de lo que tradicionalmente se define como regla).

El lenguaje NIVEL9 puede declarar los nombres de las variables en forma de oraciones, es decir, con espacios vacíos entre el texto del nombre de la variable ($\$ = \text{símbolo delimitador de variables}$) de manera que puedan declararse variables que denoten proposiciones o frases que tengan sentido en el lenguaje natural humano. Esto permite a los módulos explicativos expresar análisis y conclusiones de forma clara y sencilla.

Algo parecido ocurre con las proposiciones las cuales van delimitadas por el signo de interrogación ($\{ = \text{para iniciar}, ? = \text{para finalizar}$), el propósito de esto es igual que el de las variables. Sin embargo, debe usarse símbolos diferentes para que el analizador sintáctico pueda diferenciar entre variables y proposiciones.

La gramática también pone de manifiesto la precedencia de los operadores en el lenguaje. En la tabla 11, se muestra la precedencia de los operadores del lenguaje NIVEL9 la cual es un poco diferente a las que utilizan los lenguajes tradicionales. La tabla esta ordenada de forma descendente de mayor a menor precedencia. El nivel de precedencia indica en que cual operador tiene mayor precedencia, entre menor es el número del nivel, mayor precedencia tiene el operador.

NIVEL DE PRECEDENCIA	OPERADORES	NOMBRES
1	!	Símbolo negativo
1	()	Paréntesis
2	^	Potencia
2	*	Multiplicación
2	/	División
3	+	Suma
3	-	Resta
4	<	Menor
4	<=	Menor o Igual
4	>	Mayor
4	>=	Mayor o igual
4	==	Igual
4	<>	Diferente
5	Y	Y (AND)
5	O	O (OR)

Tabla 11. Precedencia de los operadores del lenguaje NIVEL9

La gramática también pone remanifiesto que el lenguaje sólo posee dos tipos de datos que son el tipo NUM y el tipo CAD. Los datos de tipo NUM representan aquellos con los cuales pueden realizarse operaciones de tipo aritmético, sean estos de valores constantes, enteros o flotantes. Los datos de tipo CAD son aquellos que pueden almacenar cualquier tipo de símbolos o caracteres los cuales se representan a través de cadenas.

4.4. MODULO DE RECONOCIMIENTO DE TEXTO Y SINTETIZADOR DE VOZ

El módulo de reconocimiento de voz y sintetizador de voz se encarga, tal como su nombre lo dice, de reconocer textos, números, fechas pertenecientes a algún lenguaje natural humano conocido que se presente en la salida de la interfaz del software y a la vez emular dichos símbolos a través de un sintetizador de voz el cual se encarga de generar e imitar a través de tecnología multimedia la voz humana.

Para generar este módulo se utilizó la Speech Application Programming Interface (SAPI) de Microsoft. SAPI es parte del modelo WOSA (Windows Open Services Architecture) de Microsoft el cual consta de reconocimiento de voz "speech recognition (SR)" y el texto para voz "text-to-speech (TTS)".

NIVEL9 utiliza el servicio TTS el cual provee la habilidad de convertir texto escrito en palabras habladas. El TTS se vale de sintetizadores de voz creados por otras empresas para generar voces en diferentes idiomas según sea el gusto del usuario.

Si se desea conocer más información sobre esta tecnología se recomienda visitar el sitio de Microsoft <http://www.microsoft.com>

5. DISCUSION Y ANALISIS DE RESULTADOS

Durante la elaboración de este proyecto se realizaron varias pruebas de manera informal, esto con el fin de poder demostrar (no en forma exacta) uno de los objetivos principales de este proyecto el cual se remite a un significativo incremento en la tarea de identificación de las familias de los anélidos con énfasis en la especie de los onúfidos.

Se pudo observar que en la identificación de especies que son consideradas sencillas de identificar por el experto (tiempos entre 5 y 8 minutos), NIVEL9 es capaz de identificarlos en un tiempo entre 1 y 1.5 minutos, garantizando a ICMARES una reducción de tiempo del 80.77% y un incremento del rendimiento en aproximadamente 520%.

En lo que se refiere a las especies consideradas como altamente complejas de identificar, el software presenta un aumento aproximado del rendimiento del 200% al 250%. Estas pruebas fueron realizadas por estudiantes con cierta experiencia en el conocimiento de las familias en estudio. En cuanto al desenvolvimiento de los expertos se espera un rendimiento aproximado del 75%, ya que estos sin necesidad del software realizan identificaciones con mayor velocidad que los estudiantes.

Es importante hacer notar que la utilización de la base de conocimientos generada funciona en forma, óptima con personal con cierto conocimiento del tema, si el estudiante conoce poco o nada sobre las estructuras de los anélidos es posible que se cometan errores. Sin embargo la probabilidad de cometer errores utilizando el software se reduce notablemente.

En cuanto a la percepción de los estudiantes acerca del software se denota gran entusiasmo por lo novedoso, dinámico y amigable, y por supuesto, por el obvio incremento en el rendimiento en el proceso de identificación. Se percibe en una

primera impresión como un software difícil de usar por la considerable cantidad de iconos y menús que este posee. No obstante al utilizar el sistema dos o más veces, la adaptación es inmediata sintiendo un dominio del mismo.

En lo que a la generación de las bases de conocimiento se refiere, debido al diseño de un lenguaje amigable el personal de ICMARES estima que con el código generado para este proyecto les será fácil implementar otras identificaciones parecidas sin ninguna capacitación referente al software.

También se han dado buenos comentarios acerca del potencial pedagógico del software, tanto por parte del experto como por parte de los estudiantes, poniendo de manifiesto el módulo explicativo que presenta NIVEL9, en cuanto a la capacidad de resolver inquietudes referentes a la forma de cómo se realizó la identificación. En este sentido también se comentó que el proceso de identificación se puede mejorar notablemente, así como la disminución de errores, si se utilizaran imágenes más explícitas referente a las estructuras de las especies en estudio, detallando zonas claves y modificando las imágenes para señalar las zonas de interés sobre las que hace referencia el software en las sesiones de identificación.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

Los resultados obtenidos en el desempeño del software ante la problemática de la identificación de especies, confirma y valida el uso de la metodología de Sistemas Expertos como un recurso efectivo para resolver problemas específicos.

El método de encadenamiento hacia delante como técnica para inferir conocimiento se adapta perfectamente para la resolución de problemas de identificación, ya que estas se basan en claves de identificación que van de lo general a lo específico.

La codificación de conocimiento a través de reglas de producción facilita la generación de bases de conocimientos ya que su sintaxis es mucho más amigable para personas que no tienen experiencia en el área de informática y además es la forma habitual que se utiliza en nuestro medio para realizar o establecer el razonamiento.

La interfaz gráfica de la aplicación se adapta de manera conveniente a la resolución del problema en particular, debido a que se pueden mostrar imágenes referentes a la estructura biológica de la especie que está siendo identificada.

El diseño de un lenguaje de programación que permita gestionar variables y proposiciones de forma similar al lenguaje natural, permite una sensación de dinamismo y mayor eficiencia en las sesiones realizadas en el software.

Debido a la flexibilidad del diseño implementado resultaría relativamente sencilla la creación de versiones del software en otros idiomas (además del español).

Finalmente se puede concluir que los sistemas expertos pueden ser iguales o más eficientes que los expertos humanos en la tarea de resolver problemas y además son

una excelente alternativa como medio de aprendizaje sobre cualquier dominio de conocimiento.

6.2. RECOMENDACIONES

Se recomienda codificar el resto de las familias pendientes de la base de especies marinas de El Salvador en el código de la base de conocimiento, para almacenar de esta manera en forma ordenada un registro importante sobre la metodología seguida en la identificación, y que además sea de utilidad para las futuras generaciones.

Se recomienda el uso de imágenes más explícitas que detallen y muestren al usuario el proceso de identificación de las especies en forma más clara e intuitiva, y de esta forma optimizar el rendimiento y disminuir la probabilidad de error.

Crear los medios que permitan darle seguimiento al desarrollo del software generado en el presente proyecto, de manera que en el futuro se puedan implementar otros tipos de análisis complementarios dentro de las metodologías de sistemas expertos, entre estos se podrían mencionar: el método de encadenamiento hacia atrás, análisis apoyados en lógica difusa, el manejo de probabilidades a través del teorema de Bayes, la generación de auto aprendizaje a través de máquinas de aprendizaje, etc. Sería interesante llevar el prototipo creado en esta tesis a una fase de mejoramiento y depuración de tal forma que pudiera convertirse en una herramienta con alto potencial comercial internacionalmente.

Generar proyectos tanto a nivel universitario como gubernamental para codificar conocimiento en áreas especializadas con el que no se cuenta en El Salvador, también sería recomendable crear bases de conocimiento en dominios en el que se vea la necesidad de mejorar la calidad de dicho conocimiento, tales como la medicina y la ingeniería, de manera que todo este conocimiento pueda ser gestionado por medio del software creado.

Crear grupos de excelencia en el área de informática e inteligencia artificial como medios importantes para la resolución de problemas nacionales complejos y desarrollo tecnológico de nuestra sociedad. Esto por supuesto conlleva a impulsar proyectos de investigación que sean rentables y autosostenibles.

Impulsar la creación de diplomados, maestrías y hasta doctorados en las áreas de la informática y de la inteligencia artificial que vayan orientadas a la investigación científica.

Fomentar investigaciones referentes a otras metodologías diferentes a los sistemas expertos, tales como la programación genética, redes neuronales, y lógica difusa, y todo esto apoyado por la creación de un centro de investigación orientado a la inteligencia artificial, tal como se maneja en otros países más desarrollados.

Apoyar la creación de grupos multidisciplinarios de investigación en donde se agrupen expertos que dominen áreas básicas de conocimiento como la estadística, la matemática y la informática, y de esta manera crear una plataforma científica que permita resolver problemas en áreas como el modelado, la optimización, la identificación de patrones complejos, y análisis avanzado de datos e información, los cuales sean auxiliados por software científico especializado hecho en El Salvador.

7. REFERENCIAS BIBLIOGRAFICAS

AVRON BARR, PAUL R. COCHEN, "ARITIFICIAL INTELLIGENCE, VOLUMEN IV"

Addison-Wesley Publishing Company, Inc, 1989. ISBN: 0-201-1731-0. Pág. 149-191

Barraza, J. E. (1994). "*Guía Ilustrada de algunos poliquetos (Annelida: Polychaeta) de Solymar, La Libertad, El Salvador*". Boletín Técnico No 2. Asociación Amigos del Árbol (A. M. A. R.). 17 pp.

Berkeley, E. & C. Berkeley. (1939). "*On a collection of Polychaeta Chiefly from the west coast of Mexico*". Annals and Magazine of Natural History London II: 321-346.

Calles, A. D.; N. A. Fuentes & N. A. Serrano. (1994). "*Estudio preliminar de la Comunidad Macrobentónica de la Bahía de Jiquilisco, Departamento de Usulután, Marzo y Septiembre de 1993*". Escuela de Biología, Facultad de Ciencias Naturales y Matemática. Universidad de El Salvador (Tesis de Licenciatura). 40 pp.

Canjura, J. N. (1996). "*Composición de anélidos poliquetos durante la estación transicional lluviosa-seca en el manglar del Estero de San Diego, La Libertad*". Escuela de Biología, Facultad de Ciencias Naturales y Matemática, Universidad de El Salvador (Tesis de Licenciatura). 41 pp.

Daniel Grevier, "AI. The Tumultuous History Of The Search For Artificial Intelligence"
Basic Books, 1993. ISBN: 0-465-02997-3. Pág.: 145-162.

De León, J. A.; C. G. Rivera & M. Y. Romero. (2004). "*Sublittoral Eunicidae and Onuphidae (Polychaeta) from soft bottoms off El Salvador, Eastern Pacific*". J. Mar. Biol. Ass. U.K. 84: 93-101.

DIETER NEBENDAHL, "SISTEMAS EXPERTOS"

Siemens S.A., Aktiengesellschaft, Berlín y Munich, 1988. ISBN : 84-267-0678-9

Elaine Rich, Kevin Knight, "Inteligencia Artificial"

McGraw - Hill, Segunda edición, 1994. ISBN: 84-481-1858-8. Pág.: 607-618

Fauchald, K. (1968). "*Onuphidae (Polychaeta) from western México*". Allan Hancock Monographs in Marine Biology. University of Southern California, Los Angeles, California. 82 pp.

----- (1973). "*Polychaeta from Central American sandy beaches*". Bulletin of Southern California Academy of Sciences. 72: 19-31.

----- (1977). "*The Polychaete Worms. Definitions and Keys to the Orders, Families and Genera*". Science Series 28. Los Angeles, California. 188 pp.

----- (1982). "*Revision of Onuphis, Nothria and Paradiopatra (Polychaeta: Onuphidae) based upon type material*". Smithsonian Contr. Zool. 356: 6-109.

Hartman, O. (1944). "*Polychaetous annelids. 5. Eunicea*". Allan Hancock Pacific Expeditions. 10: 1-237.

Hartmann-Shröder, G. (1959). "*Zur ö Kologie der Polychaeten de Mangrove-Estero Gebietes von El Salvador*". Beitr. neutr. Fauna. 1: 69-183 pp.

J.P. SÁNCHEZ Y BELTRÁN, "SISTEMAS EXPERTOS, UNA METODOLOGÍA DE PROGRAMACIÓN"

Macrobit Editores, S.A. de C.V. ISBN : 970-604-037-4

JOHN DURKIN, "EXPERT SYSTEMS, DESIGN AND DEVELOPMENT"

Prentice-Hall, Englewood Cliffs, New Jersey, 1994. ISBN : 0-02-330970-9

Kinberg, J. (1865). "*Annulata nova*". Öfv. Vetesnk, Akad, Stockholm, Förh. 21: 559-574.

Molina, O. A. (1992). "*Estructura del macrobentos en el manglar del Estero de Jaltepeque, El Salvador*". Escuela de Biología. Facultad de Ciencias, Universidad de Costa Rica (Tesis de Maestría). 89 pp.

Moore, J. (1911). "*The Polychaetous annelids dredged by the U. S. S. Albatross off the coast of the Southern California in 1904. 3. Euphrosinidae to Goniadidae*". Proceedings of the Academy of Natural Sciences of Philadelphia. 63: 243-318.

PATRICK HENRY WINSTON, "INTELIGENCIA ARTIFICIAL"

Addison-Wesley Iberoamericana, Tercera edición. ISBN: 0-201-51876-7. Pág.: 131-177

Rivera, C. G. & M. Y. Romero. (2002). "*Distribución de poliquetos (Annelida: Polychaeta) en la zona costera de El Salvador. Resultado del crucero de investigación R/V URRACÁ del Instituto Smithsonian de Investigaciones Tropicales*". Escuela de Biología. Facultad de Ciencias Naturales y Matemática, Universidad de El Salvador (Tesis de Licenciatura). 64 pp.

Rivera, J. R. & R. E. Ibarra. (1995). "*Estudio preliminar de los poliquetos (Annelida: Polychaeta) del Estero de Barra de Santiago, Ahuachapán, durante la*

estación lluviosa". Escuela de Biología, Facultad de Ciencias Naturales y Matemática, Universidad de El Salvador (Tesis de Licenciatura). 49 pp.

Roberto Hernández Sampieri, Carlos Fernández Collado, Pilar Baptista Lucio, "Metodología de la Investigación". McGraw Hill, Segunda Edición, 1998. ISBN: 970-10-1899-0. Pág.: 1 - 54

Ruppert, E. & R. Barnes. (1996). "*Zoología de los Invertebrados*". 6ª Ed. Mc. Graw-Hill. México. 504-599 pp.

Salazar, S. (1990). "*Poliquetos (Annelida: Polychaeta) de la Bahía de Manzanillo, con una clave ilustrada para las especies de Colima, México*". Rev. Biol. Trop. 38 (2A): 211-229.

Stuart Russell, Petter Norvig, "Artificial Intelligence, A Modern Approach"
Prentice-Hall, Inc., 1995. ISBN: 0-13-103805-2. Pág.: 149-332

Treadwell, A. (1941). "*Polychaetous annelids from the west coast of Mexico and Central American*". Zoológica. 26: 17-24.

Uebelacker, J. M. & P. G. Johnson (Editors). (1984). "*Taxonomic Guide to the Polychaetes of the Northern Gulf of Mexico*". Final Report to the Minerals Management Service, contract 14-12 -001-29091. Barry A. Vittor & Associates, Inc., Mobile, Alabama. 7 vols.

Vasconcelos, H. A. & L. J. Fuentes. (1997). "*Comunidad Macrobentónica del Golfo de Fonseca (El Salvador)*". Escuela de Biología, Facultad de Ciencias Naturales y Matemática, Universidad de El Salvador (Tesis de Licenciatura). 61 pp.

Woodin, S. A. (1974). "*Polychaete abundance patterns in a marine soft-sediments environment the importance of biological interaction*". Ecol. Monogr. 44: 171-187.

SITIOS WEB CONSULTADOS

- a. http://www.cee.hw.ac.uk/~alison/ai3notes/tableofcontents2_1.html
Extracto del libro: "Essence of Computing, Essence of Artificial Intelligence" Publicado por Alison Cawsey
- b. <http://www.emclab.umn.edu/consortium/Whatis/node7.html>
Extracto de la tesis de Navin Kashyap, "An Expert Systems Application in Electromagnetic Compatibility," Universidad de Missouri-Rolla, 1997 . Publicado por Universidad de Missouri-Rolla ECM Consortium

- c. <http://xiniom.com/se/index.php>
Demostración de la aplicación de los sistemas expertos, sobre todo de aquellos basados en el Web. Publicado por Lic. Javier Rosas Hernández
- d. <http://www.monografias.com/trabajos10/exper/exper.shtml>
Artículo publicado en www.monografias.com. Autor desconocido.
- e. <http://elies.rediris.es/elies9/index.htm>
"Diseño e Implementación de un Lexicón Computacional para Lexicografía y Traducción Automática". Autor, Antonio Moreno Ortiz. Facultad de Filosofía y Letras, Universidad de Málaga
- f. http://greco.dit.upm.es/~tomas/cursos/isi/trabajos/2002/jpalacios_t.pdf
"RDF & RDF Schema". Autor, Juan Pablo Palacios E. Universidad Politécnica de Madrid. Facultad de Telecomunicaciones.
- g. <http://www.w3c.org/RDF/>
Documento con la especificación para la construcción de redes semánticas utilizando la Infraestructura de descripción de recursos (Resource Description Framework). Autores, Eric Miller, Semantic Web Activity Lead, Ralph Swick, Development Lead, Dan Brickley, RDF Interest Group Chair and RDF Core Working Group co-chair
Version: 1.170

APENDICES

A. ¿QUE ES NIVEL9?

En la actualidad se están analizando estrategias de solución y conocimientos de expertos para la solución de problemas. Es por ello, que se están utilizando los Sistemas Expertos, como una nueva generación de desarrollo de software y dar respuesta a una serie de situaciones cotidianas, en la que de alguna manera se aprovecha el conocimiento de los expertos para obtener los resultados deseados de una forma inteligente y de esta forma optimizar el tiempo del experto en tareas de mucha mayor envergadura.

Nivel9 es un software basado en los Sistemas Expertos que contiene una interfase gráfica de usuario, creado con el objeto de ejecutar código de programas englobado en una base de conocimientos que se apoyan en los conceptos de inteligencia artificial, con el fin de dar interpretación a esta base de conocimientos para obtener un resultado. Este software utiliza un programa basado en la experiencia y el conocimiento del experto, con el fin de llegar a un resultado apegado a la realidad.

El software contiene una interfase al usuario y un editor. La interfase presenta el proceso que utiliza el usuario para obtener la respuesta. El editor ayuda y facilita la creación de programas que generan las bases de conocimientos.

El lenguaje que se utiliza en la programación es semejante al lenguaje natural, con el fin de facilitar al usuario en la creación de las bases de conocimientos y la creación de programas para dar solución a la problemática. Una de las características, muy importante, es que se utiliza el lenguaje español para su programación, haciendo a NIVEL 9 un sistema de fácil comprensión y programación para los usuarios. Es basado en REGLAS, las que se encuentran relacionadas, así dependiendo de las condiciones que están contenidas en cada regla se pasa de una regla a otra o, se obtiene una conclusión.

NIVEL9 es un lenguaje fácil de utilizar y su portabilidad hace posible adaptar el software escrito para una computadora con una versión de sistema operativo Windows a otra versión.

B. GUIA DE INSTALACION DE NIVEL9

INSTALACION DE NIVEL9

Antes de instalar NIVEL9, es necesario estar seguro que el equipo donde se instalará cumpla con los requisitos mínimos necesario para su ejecución e instalación.

Requisitos Mínimos de equipo:

- Microprocesador Pentium a 133 MHz o superior (o su equivalente)
- Computadora con 64 MB en RAM (128 MB recomendado)
- Sistema operativo cualquier versión de Windows
- Unidad de CD-ROM
- Monitor a colores con adaptador de alta resolución gráfica
- 12 MB de espacio en disco duro
- Ratón

A continuación, se describe el procedimiento que se debe seguir para instala en un computador el software NIVEL9.

- Colocar el CD con el programa de instalación de NIVEL9.
- Abrir el explorador de Windows y ubicarse en la carpeta denominada "Nivel9_Install", abrir esta carpeta y ubicar el cursor del ratón sobre el archivo llamado SETUP.EXE, hacer doble clic sobre el archivo y luego seguir las instrucciones (ver figura 1).

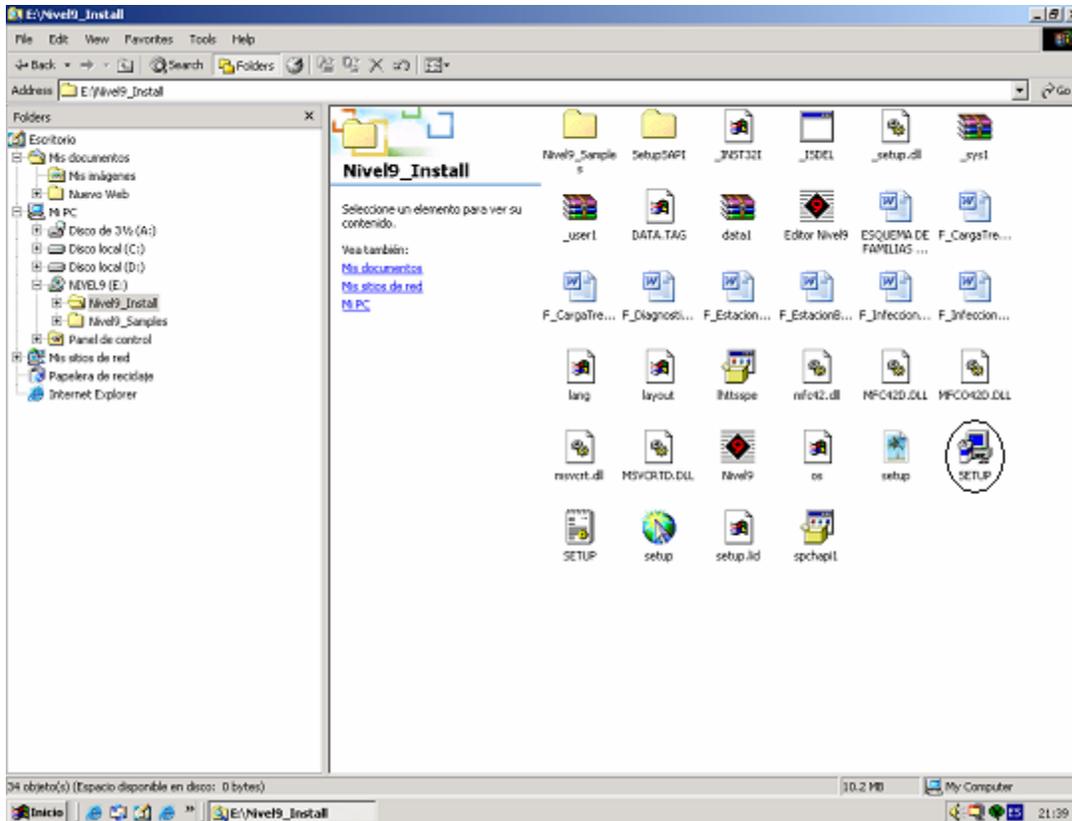


Fig. 1 Listado de archivos en el CD

Esta acción presentará la ventana semejante a la de la figura 2, que recomienda cerrar todos los programas que se estén ejecutando en el momento de la instalación del software.

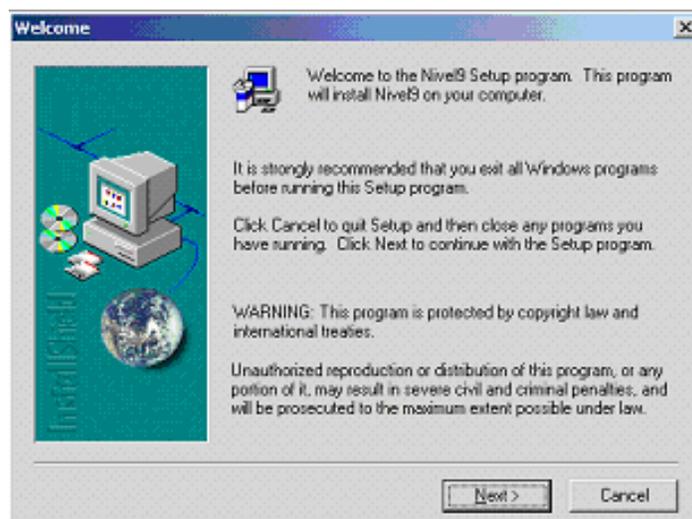


Fig. 2. Inicio de la instalación

A continuación, oprimir el botón con el mensaje "Next", y se obtendrá la ventana de la figura 3.

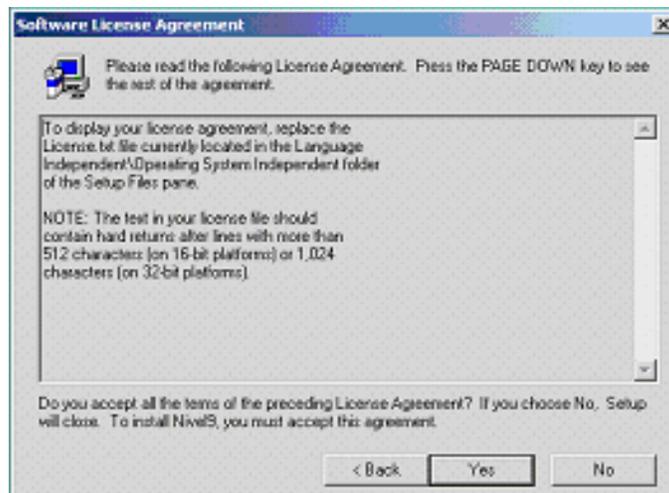


Fig. 3. Términos de Licenciamiento

La figura 3, presenta el licenciamiento del software. Si se selecciona "No", no se aceptan todos los términos del licenciamiento y el proceso terminará. El seleccionar "Yes", se aceptan los términos del licenciamiento y el proceso de instalación continúa presentando la ventana que se presenta en la figura 4.

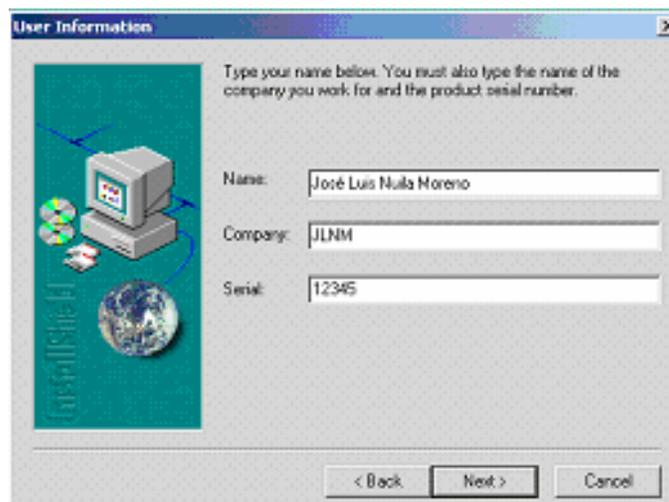


Fig. 4. Identificación del propietario

Por omisión, en el proceso de la instalación, el nombre y la compañía, este programa de instalación toma como parámetros iniciales los valores introducidos al momento de instalar el sistema operativo. Estos valores pueden ser modificados por el usuario a su

conveniencia. Para activar el botón "Next", deberá introducir un número de serie, valor que está a decisión del usuario. En futuras versiones se pretenderá realizar una mayor protección al software activando la verificación del número de serie. El oprimir este botón se presenta la ventana de la figura 5.

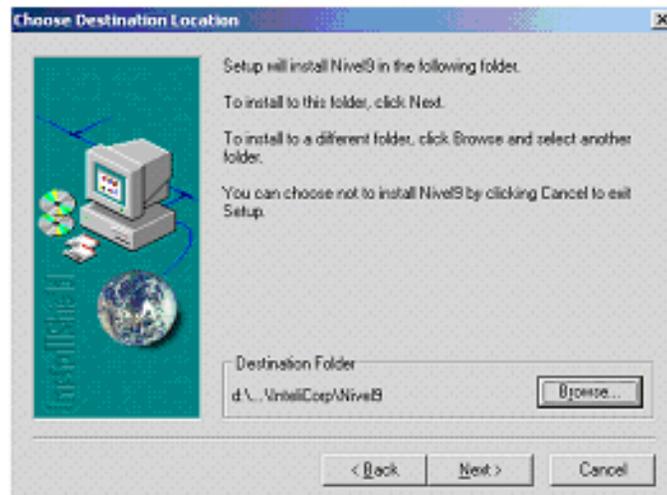


Fig. 5. Ubicación del programa

En la figura 5 se muestra la ventana donde se decide la ubicación dentro del disco duro donde se instalará el programa.

Si es necesario cambiar de nombre y/o de posición, simplemente se oprime el botón con el mensaje "Browse" y se sigue las instrucciones. Una vez decidida la posición se oprime el botón con el texto "Next", cuya acción le presenta al usuario la pantalla que se muestra en la figura 6.

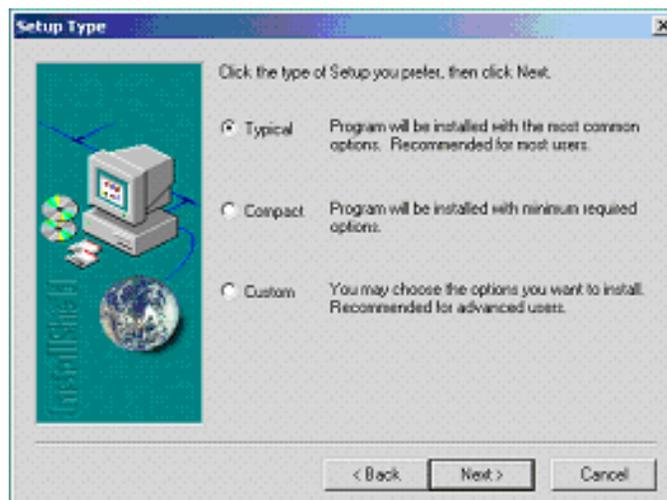


Fig. 6. Tipo de configuración

En esta ventana se selecciona el tipo de configuración del programa que se desea instalar. Las opciones de configuración que se presentan son:

- Typical: Instala los programas ejecutables, la ayuda y los ejemplos
- Compact: Instala los programas ejecutables y la ayuda.
- Custom: Da las opciones que el usuario desea instalar. Por defecto el programa tiene seleccionada todas las opciones que son: los programas ejecutables, la ayuda, los ejemplos y, el módulo SAPI y los sintetizadores de voz en español.

Luego, es necesario continuar, presionando el botón con el texto "Next", cuya acción presenta la pantalla que se muestra en la figura 7.

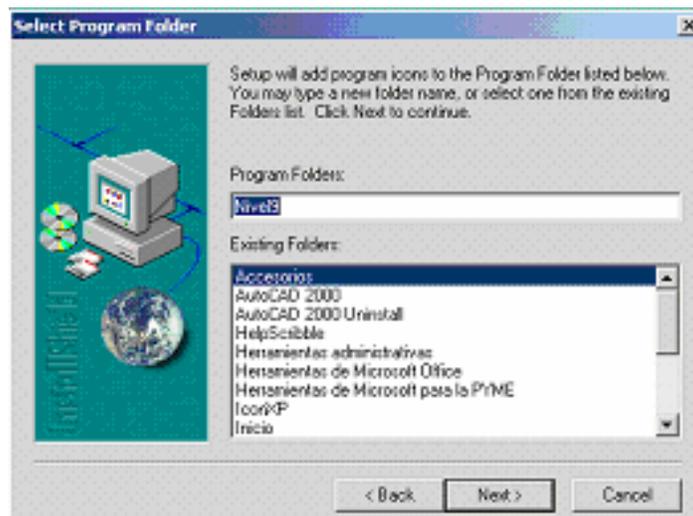


Fig. 7. Carpeta de instalación

La figura 7, le presenta al usuario la opción de poder crear un fólдер nuevo o un fólдер existente donde se ubicarán los archivos de acceso directo de la aplicación.

La figura 8, es el resultado de accionar el botón "Next" de la ventana de la figura 7. En ella, se observan las barras de progreso de la instalación del programa en el disco

duro. Una vez terminada la instalación se presenta la ventana de la figura 9, indicando que la instalación del programa Nivel9 ha finalizado correctamente y está listo para ser utilizada.

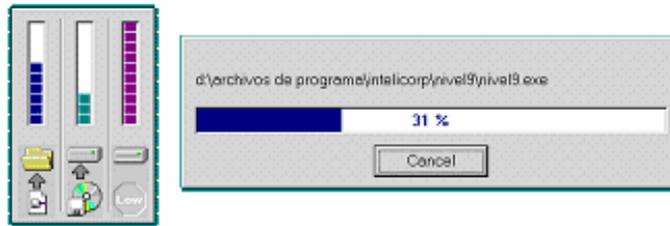


Fig. 8 Barra de progreso del proceso de Instalación

La figura 9, representa la ventana resultado de la instalación satisfactoria del programa. Oprimir el botón "Finish" para terminar la instalación.



Fig. 9. Finalización de la instalación

C. ENTORNO DE TRABAJO

Nivel9 para plataformas de Windows, es un entorno integrado de desarrollo de bases de conocimientos y ejecución de las mismas. Este ofrece desarrollar bases de conocimientos para dar solución a cualquier tipo de problemas. Así como comprobar la sintaxis de las bases de conocimientos una vez terminadas.

La interfase de usuario se muestra en la figura 10.

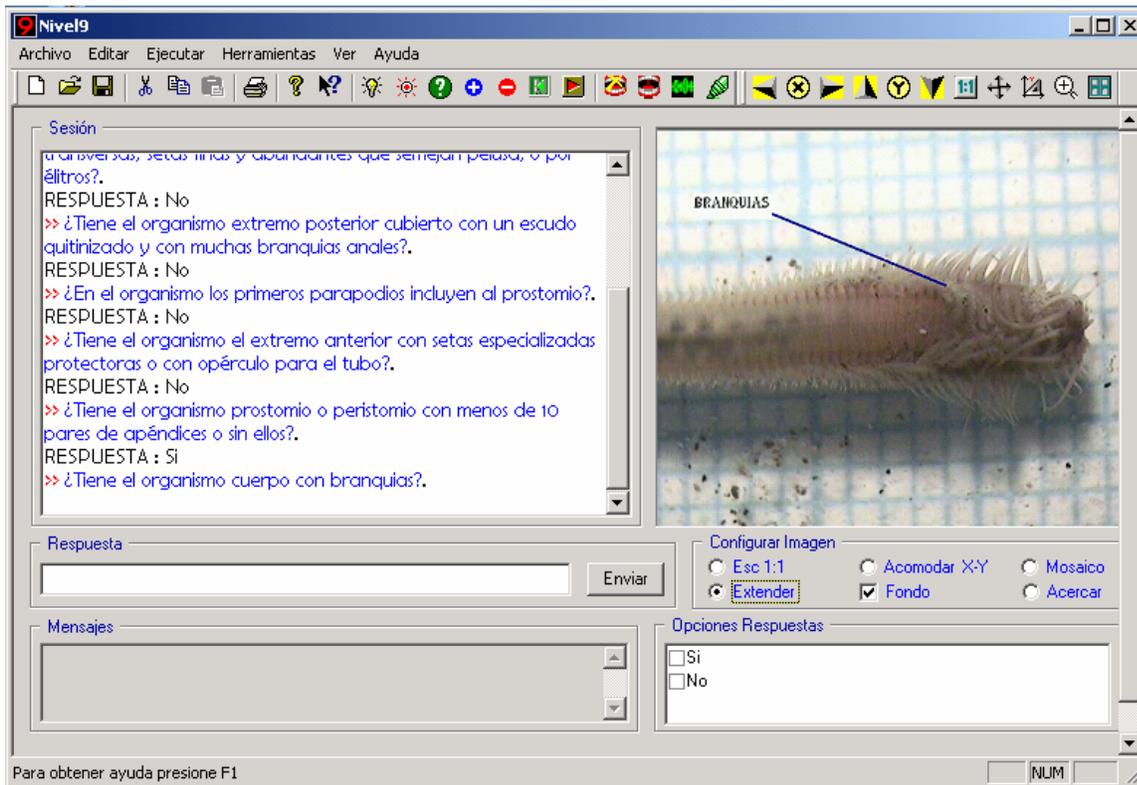


Fig. 10 Interfase de usuario de NIVEL9.

En la parte superior de la interfase de usuario, se encuentra la barra de título (Ver Figura 11), donde se presenta el icono de la aplicación acompañado de su nombre. En el extremo derecho de esta barra, se encuentran los botones de control, con los cuales se puede minimizar o maximizar la ventana o cerrar la aplicación.



Fig. 11. Barra de Título.

Debajo de la barra de título, se encuentra la barra de menú (Ver Figura 12), en la cual se visualiza el conjunto de opciones disponibles para la aplicación. Al activar alguna de las opciones con el ratón, sobre su nombre se visualiza el conjunto de órdenes que la forman.



Fig. 12. Barra de Menú.

A continuación, se puede observar que se encuentran: la barra de herramientas (figura 13) y la barra de imagen (figura 14), ambas barras presentan los íconos de acceso directo de las ordenes con que cuenta el programa. Y en la parte inferior de la ventana se encuentra la barra de estado (Figura 15).

Barra de herramientas



Fig. 13. Barra de Herramientas.

La barra de herramientas, aparece en la parte superior de la ventana de la aplicación, debajo de la barra del menú. Mediante el ratón, la barra de herramientas brinda acceso rápido a muchas herramientas de la aplicación Nivel9.

Para ocultar o mostrar la barra de herramientas, elija la opción Ver del menú principal y seleccione la etiqueta Barra de herramientas.

Barra de imagen



Fig. 14. Barra de Imagen.

La barra de imagen, aparece en la parte superior de la ventana de la aplicación, debajo de la barra del menú y al lado derecho de la barra de

herramientas. Mediante el ratón, la barra de imagen, brinda un acceso rápido para posicionar y variar las dimensiones de la imagen de la aplicación Nivel9.

Para ocultar o mostrar la barra de imagen, elija la opción Ver del menú y seleccione la etiqueta Barra de imagen.

Barra de estado



Fig. 15. Barra de Estado.

La barra de estado, aparece en la parte inferior de la ventana principal de la aplicación. Use el comando Barra de estado del menú Ver para mostrar u ocultar la barra de estado.

El área izquierda de la barra de estado, describe las acciones de los elementos de menú. Esta descripción aparece cuando se posiciona el cursor del ratón sobre cada una de los elementos que presenta el menú o cuando, utilizando las teclas de movimiento o cursor. Si después de ver la descripción de uno de los botones de la barra de herramientas y no se desea ejecutar el comando, suelte el botón del ratón mientras el puntero se halla fuera de la barra de herramientas.

Las áreas de la derecha de la barra de estado indican cuál de las siguientes teclas han sido activadas o presionadas:

<u>Indicador</u>	<u>Descripción</u>
MAY	Se ha presionado la tecla Bloq Mayús.
NUM	Se ha presionado la tecla Bloq Num.
DESP	Se ha presionado la tecla Bloq Despl.

A continuación se detalla cada uno de los elementos que componen la Barra de Menú:

Archivo

El accionar esta opción de la barra de menú se despliega el contenido de esta opción. La figura 16 muestra su detalle.



Fig. 16. Opciones de menú Archivo.

A continuación se detalla el contenido de la opción de la barra de menú "Archivo", que pueden ser seleccionadas al activarse.

Nueva sesión:

Esta opción, permite limpiar la sección de la interfase de usuario denominada "Sesión". Al optar con esta opción es necesario cargar nuevamente la base de conocimientos.

Abrir sesión...:

Esta opción, le permite al usuario llamar una sesión guardada previamente, con el fin de estudiar o darle seguimiento al proceso realizado.

Guardar sesión:

Una vez obtenida una conclusión, esta opción permite al usuario almacenar en disco el resultado obtenido para que después pueda ser estudiada.

Guardar sesión como...:

Opción que permite al usuario guardar una sesión con otro nombre.

Cargar base conocimientos...:

Permite al usuario, traer a memoria la base de conocimientos que se desea procesar. Esta opción es necesaria para procesar la base de conocimientos.

Imprimir...:

Opción que permite al usuario, imprimir el resultado obtenido en el proceso de la base de conocimientos.

Al seleccionar esta opción, se presenta la ventana de diálogo de selección de la impresora, donde se selecciona el impresor y es posible configurar los parámetros deseados para la impresión de los resultados.

Configurar impresora...

Esta opción presenta la ventana de diálogo de configuración del impresor. Al seleccionar esta opción permite seleccionar la impresora y configurar sus parámetros.

Salir

Opción que permite salirse del sistema

Editar

El accionar esta opción de menú, se despliega el siguiente detalle:



Fig. 17. Opción de menú Editar.

Copiar

Comando que se utiliza para copiar los datos seleccionados en el portapapeles. Este comando no está disponible si no hay datos actualmente seleccionados.

Al usar el comando Copiar para colocar datos en el portapapeles, se reemplazan los datos que previamente estaban almacenados en el mismo.

Seleccionar todo

Sombrea el contenido total del texto que en ese momento se encuentra en la ventana denominada Sesión.

Ejecutar

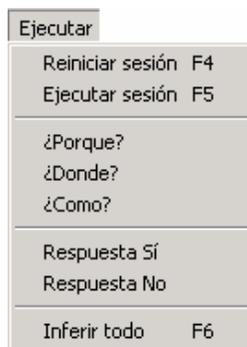


Fig. 18. Opción de menú Ejecutar.

Reiniciar sesión:

Opción que le permite al usuario reiniciar la sesión. Esto implica que si el usuario ha iniciado una sesión y se encuentra en un proceso intermedio o ha finalizado, se inicia la sesión ejecutándose el programa con la primera regla de la base de conocimientos.

Ejecutar sesión:

Inicia el proceso del sistema experto para procesar la base de conocimientos.

¿Por qué?

Con esta instrucción, el software intenta indicarle al usuario la razón por la cual el sistema está formulando la pregunta en curso. Esta instrucción está inspirada en el hecho de que muchas veces se quiere conocer cuál es la razón por la cual un experto está preguntando cosas que se cree no tienen relación con la secuencia del proceso o con el problema que se está tratando de resolver.

¿Dónde?

Con esta opción, se obtiene el razonamiento que sigue el sistema experto, en forma descendente, para inferir una conclusión que se encuentran en la base de conocimientos. Esta instrucción muestra información referente a las reglas procesadas que han sido consideradas hasta el momento en que se ejecuta esta instrucción. Sin embargo, este comando se ocupa, generalmente, cuando el sistema ya ha terminado la sesión en curso.

¿Cómo?

Se refiere a cómo se infiere y se llega a la conclusión obtenida, basada en las reglas de la base de conocimientos. Esta inferencia se obtiene en forma ascendente, considerando las reglas y respuestas que el usuario ha ingresado al sistema y basado en los conocimientos en forma de hechos encontrados. Muestra información adicional referente a: a) los hechos que han sido declarados previamente en el código de la base de conocimientos y b) los hechos confirmados por el usuario.

Respuesta Si

Le permite al usuario contestar la pregunta formulada en el área de sesión cuando la respuesta es "Si".

Respuesta No

Le permite al usuario contestar la pregunta formulada en el área de sesión cuando la respuesta es "No".

Inferir todo

Opción que se utiliza cuando se han codificado reglas en forma desordenada, es decir, que se ha creado una base de conocimientos que no sigue un orden secuencial lógico.

Al seleccionar esta opción se ejecuta el motor de inferencia buscando nuevos conocimientos a través de las reglas procesadas hasta ese momento.

Herramientas

Opción del menú principal que tiene las siguientes alternativas:



Fig. 19. Opción menú Herramientas.

Editor Nivel9...

Invoca el editor del programa Nivel9. Editor especial para crear las bases de conocimientos en el lenguaje propio de Nivel9. Más adelante se explicará el funcionamiento del editor Nivel9.

Habilitar sintetizador de voz

Al accionar esta opción, permite activar el sintetizador de voz, con ello se escuchará a través de los parlantes los mensajes de salida que presenta el sistema experto.

Deshabilitar sintetizador de voz

El accionar esta opción, desactiva el sintetizador de voz, con ello se evitará escuchar a través de los parlantes los mensajes de salida que presenta el sistema experto.

Leer sesión completa

Al accionar esta opción, el sistema lee todo el contenido que se encuentra desplegado en el área de la Sesión de inicio a fin.

Configurar Voz...

Pretende seleccionar el sintetizador de voz deseado configurando los parámetros de Tono, velocidad y volumen de la voz que se escuchará.

La siguiente figura muestra la ventana de configuración.

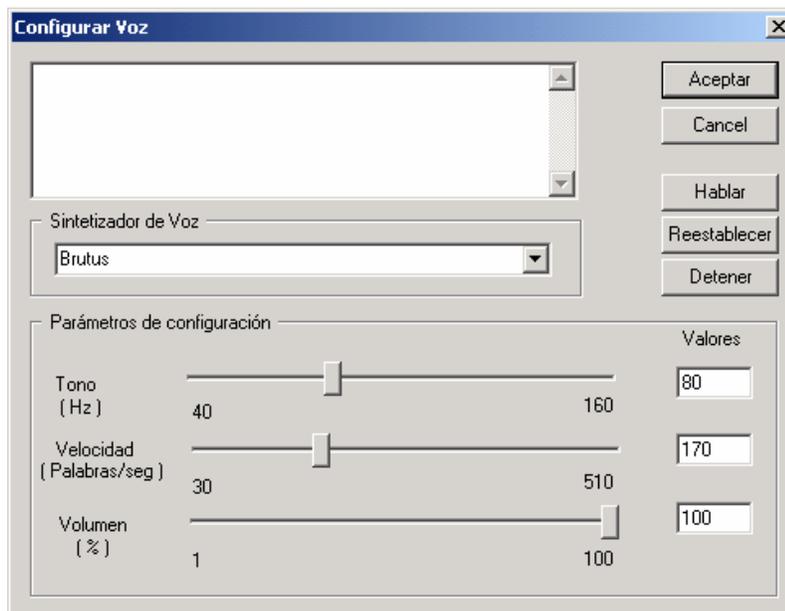


Fig. 20. Opción Configurar Voz.

Configurar fuentes sesión...

Con esta opción, el usuario puede establecer la fuente, el estilo de la fuente, el tamaño, su color y algunos efectos. La ventana de configuración se muestra a continuación.

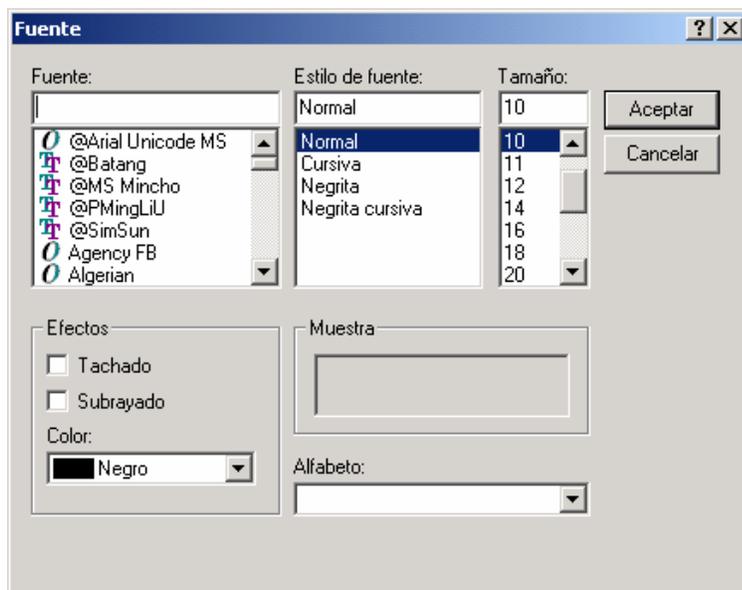


Fig. 21. Opción Configurar Fuente.

Seleccionar fondo sesión...

Opción que permite cambiar el color del fondo de la sesión, acorde al deseo del usuario, según las opciones que se presentan en la figura 22.



Fig. 22. Opción Color.

Autoformato sesión

Esta opción permite colorear el texto que se encuentra en el área de sesión, diferenciando números (rojo), preguntas (azul) y respuestas (negro) con colores diferentes. Esto es útil cuando el usuario ha decidido cambiar la fuente del área de sesión, ya que al hacerlo, el texto se vuelve todo del mismo color.

Acomodar imagen

Opción que permite ubicar la imagen según el deseo del usuario.

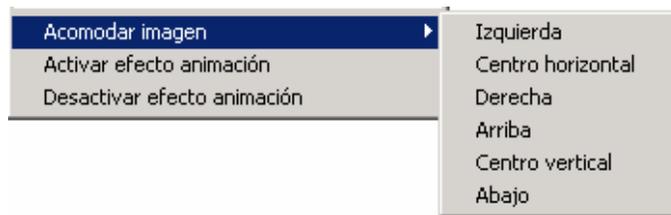


Fig. 23. Opción Acomodar imagen.

Acomodar imagen, ubica la imagen presentada en la interfase de usuario en la posición seleccionada, de acuerdo a las opciones mostradas en el submenú (ver figura 23).

Activar efecto animación

Hace que la figura que se muestra en la interfase de usuaria se anime.

Desactivar efecto animación

Hace que la figura que se muestra en la interfase de usuaria se detenga.

Ver

Opción del menú principal que permite visualizar u ocultar las barras de instrucciones. Barras que presenta la interfase del usuario en la parte inferior a la barra de menú principal.

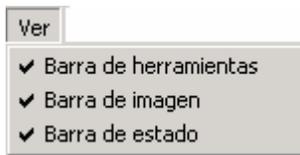


Fig. 24. Opción de menú Ver.

Barra de herramientas

Accionando esta opción, permite visualizar la barra de herramientas en la interfase del usuario, accionándola nuevamente oculta esta barra. Cuando se presenta un cheque al lado izquierdo del literal la barra se encuentra visible al usuario.

Barra de imagen

Opción que permite visualizar la barra de imagen en la interfase del usuario, accionándola nuevamente oculta esta barra. Cuando se presenta un cheque al lado izquierdo del literal la barra se encuentra visible al usuario.

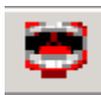
Barra de estado

Permite visualizar la barra de estado en la interfase del usuario, accionándola nuevamente oculta esta barra. Cuando se presenta un cheque al lado izquierdo del literal la barra se encuentra visible al usuario.

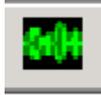
A continuación se muestran los íconos que aparecen en la barra de herramientas y en la barra de imagen como medio de acceso directo a las opciones presentadas en la barra de menú.

Barra de herramientas:

<u>Presionar</u>	<u>Para</u>
	Inicia una nueva sesión
	Abrir una sesión previamente almacenada
	Guardar una nueva sesión
	Copiar el texto seleccionado en el área de sesión
	Imprimir la sesión actual
	Saber Acerca de Nivel 9 y los créditos
	Obtener ayuda acerca del funcionamiento del software
	Activar la instrucción ¿Cómo?
	Activar la instrucción ¿Dónde?
	Activar la instrucción ¿Por qué?
	Contestar en forma afirmativa "Si" a una pregunta
	Contestar en forma negativa "No" a una pregunta
	Reiniciar la sesión de la base de conocimientos
	Ejecutar la sesión cargada en el sistema
	Apagar el sintetizador de voz



Encender el sintetizador de voz



Configurar el sintetizador de voz



Identificar símbolos con colores diferentes en la sesión

Barra de imagen

Presionar

Para



Mover a la izquierda la imagen



Centrar la imagen con respecto al eje horizontal



Mover a la derecha la imagen



Mover hacia arriba la imagen



Centrar verticalmente la imagen



Mover hacia abajo la imagen



Retornar al tamaño original de la imagen



Extender la imagen en toda el área disponible



Acomodar la imagen con respecto al eje "X" y al eje "Y".



Aumentar el tamaño de la imagen



Mostrar la figura como un mosaico

D. INTERFASE DE USUARIO

COMPONENTES DE LA INTERFASE DE USUARIO

El sistema de desarrollo NIVEL9 es un entorno integrado de desarrollo de sistemas expertos sobre plataformas Windows.

Al iniciar NIVEL9, presenta como interfase al usuario la ventana principal que se muestra en la figura 25.

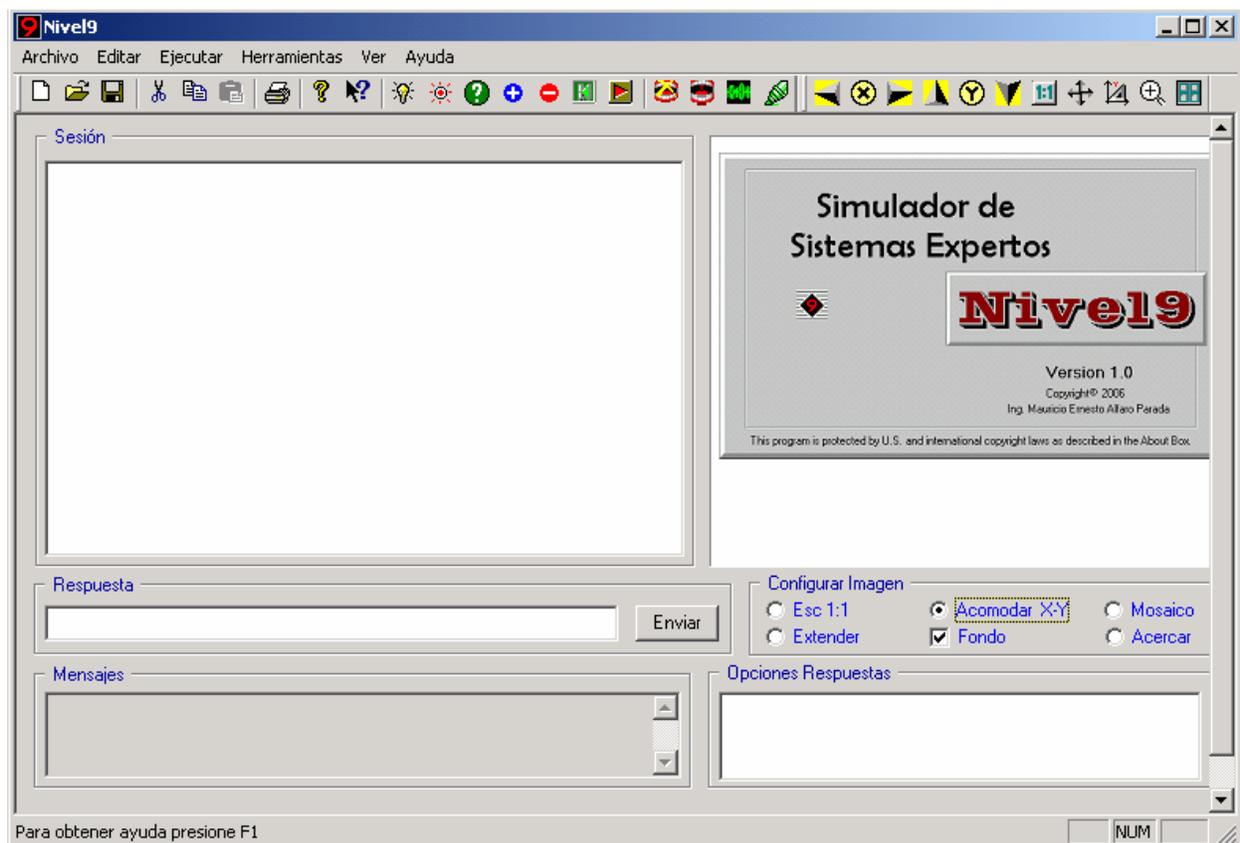


Fig. 25. Interfase de Usuario de Nivel9.

Como puede verse, la interfase de usuario consta de diferentes áreas, que presentan la información necesaria para que el usuario comprenda el funcionamiento del sistema experto. Estas áreas son: Sesión, Imagen, Respuesta, Mensajes, Configurar Imagen y Opciones Respuestas.

Para comprender el funcionamiento del sistema, a continuación se detalla cada una de las áreas que componen la interfase de usuario:

Sesión:

Sesión

Monday 12-Jun-2006

>> Universidad de El Salvador. Instituto de Ciencias del Mar y Limnología de la Universidad de El Salvador. Departamento de Ecología Marina.

>> Clave para la identificación de las Familias de Anélidos Poliquetos de la costa de El Salvador.

>> ¿Tiene el organismo dorso cubierto por paleas en hileras transversas; setas finas y abundantes que semejan pelusa, o por élitros?.

RESPUESTA : Si

>> ¿Tiene el organismo el dorso cubierto por paleas en hileras transversas?.

RESPUESTA : Si

>> El organismo es de la Familia Chrysopetalidae Ehlers 1864

Los crisopetálicos son organismos de cuerpo corto o largo, usualmente aplanado, con tres antenas; notosetas ensanchadas (paleas) en filas transversas; neurosetas compuestas.

Época	Especie	Autores que la reportaron	Hábitat
	Chrysopetalum macrophthalmum	Hartmann-Schröder	
1959	Hartmann-Schröder 1959	Molina 1992	Estero de Jaltepeque. En Fondos arenosos; limo-arcillosos.
			Lluviosa y seca

Fig. 26. Área de Sesión.

En esta área se presentan las salidas al usuario de la ejecución del sistema experto. Las frases en color azul, representan las preguntas que el sistema experto realiza al usuario para ser contestadas. La frase "RESPUESTA" es la opción que el usuario le suministra al sistema experto.

El resto de información que se presenta en color negro, es información de salida para el conocimiento del usuario, así como la conclusión que el sistema experto entrega al usuario una vez recorrido el proceso.

Imagen:

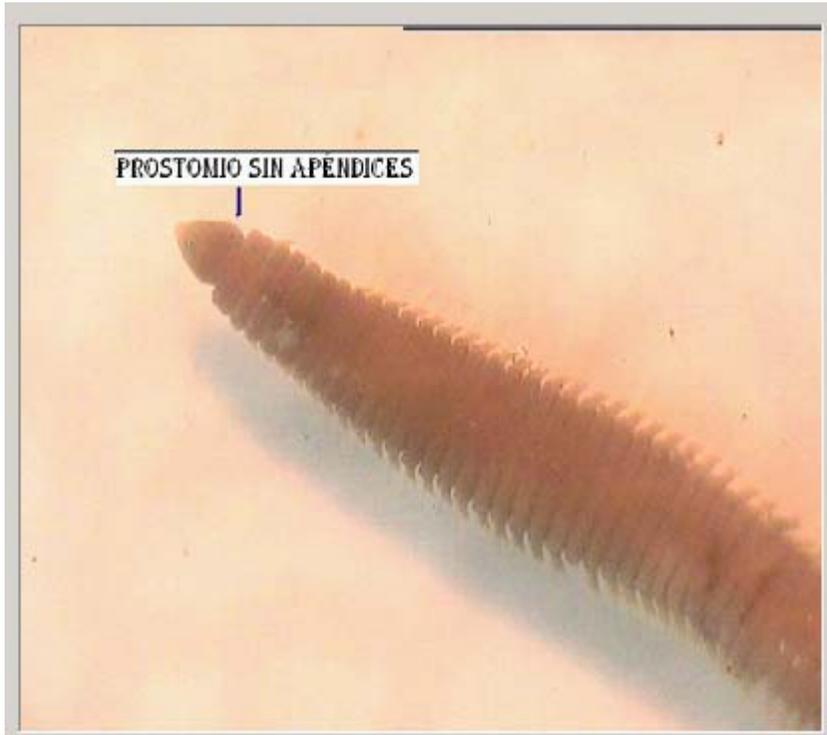


Fig. 27. Área de Gráfico.

Área donde se presenta la figura o imagen alusiva a la pregunta que el sistema de conocimiento le realiza al usuario. En ella puede verse algún detalle que el sistema de conocimiento le permite facilitar al usuario el entendimiento y aclaración de la pregunta o conclusión obtenida.

Respuesta:

A user response input area consisting of a rectangular text box with a thin border. Above the text box, the word 'Respuesta' is written in blue. To the right of the text box is a button labeled 'Enviar'.

Fig. 28. Área de Respuesta.

Esta área está compuesta por: una caja de texto, donde se presenta o escribe la respuesta que el usuario proporciona a la pregunta efectuada por el sistema experto, y un botón con la frase "Enviar", este es utilizado para enviar la respuesta del usuario al sistema experto y así poder ser procesada.

Mensajes:

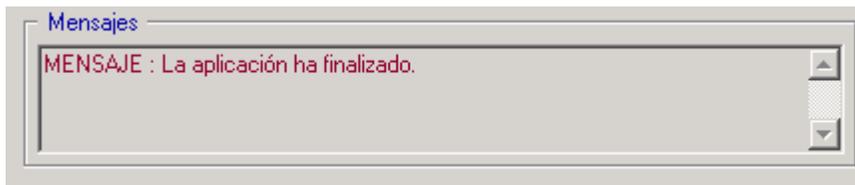


Fig. 29. Área de Mensajes.

Área que presenta mensajes de interés para el usuario, tales como la finalización de la aplicación. Así como mensajes de respuestas al usuario por la acción de oprimir las opciones de ¿Dónde?, ¿Cómo? y ¿Por qué? Y mensajes de error en la ejecución del sistema experto.

Configurar imagen:



Fig. 30. Área de Configurar Imagen.

En esta sección se encuentran las opciones para configurar las propiedades de la imagen. Estas variables de configuración son las equivalentes a las últimas 5 que se encuentran en la barra de imagen.

Como nuevo parámetro se encuentra la opción Fondo. El seleccionar esta opción, el color del fondo de la imagen se torna blanco, al deseccionarlo el color del fondo del área de la imagen se torna del mismo color que el del resto de la ventana.

Opciones Respuestas:

The image shows a rectangular dialog box with a light gray border. At the top left, the text "Opciones Respuestas" is written in a blue font. Below this text, there are two radio button options. The first option is "Si" with an unchecked radio button to its left. The second option is "No" with an unchecked radio button to its left. The rest of the dialog box is empty.

Fig. 31. Área de Opciones de Respuestas.

En esta área se presentan las posibles opciones a las respuestas que en un momento determinado presenta el programa de la base de conocimiento. Como puede verse en este caso se presentan dos opciones "Si" y "No". El usuario seleccionará la opción que le sea pertinente. Estas opciones dependerán del tipo de respuesta que ha sido programada en la base de conocimientos. Al ser marcada una opción, esta se traslada al área de respuesta.

E. EDITOR NIVEL9

El software simulador de Inteligencia Artificial Nivel9, cuenta con un editor propio, el cual permite construir los programas que conllevan la base de conocimiento.

Este editor es sencillo y fácil de utilizar, presentando características propias para la edición y la programación de las bases de conocimientos. La figura 32 muestra la interfase del editor.

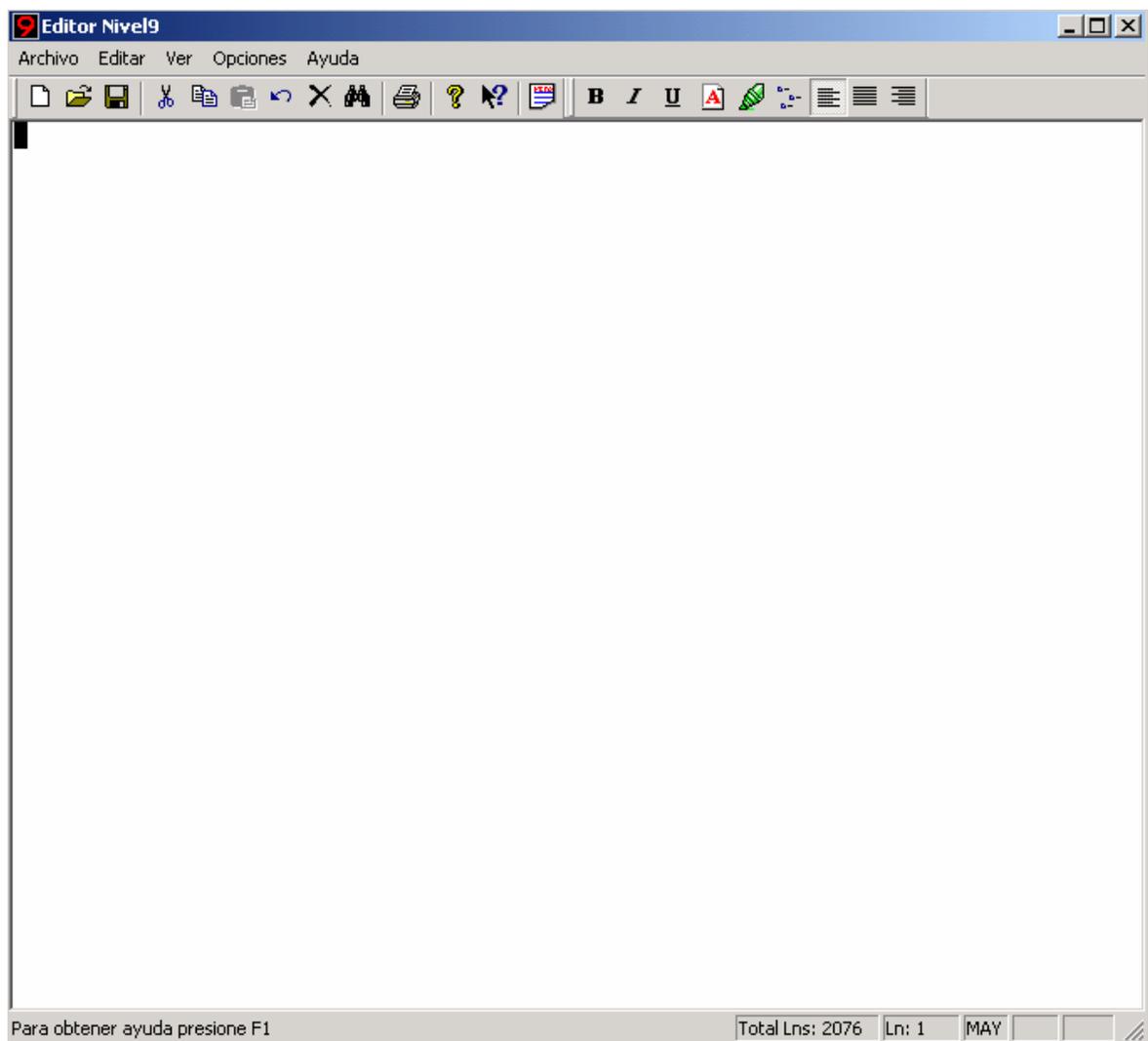


Fig. 32. Editor de Nivel9.

En la barra de menú puede observarse las diferentes opciones que contiene. En la figura 33 se presenta el contenido de esta barra.

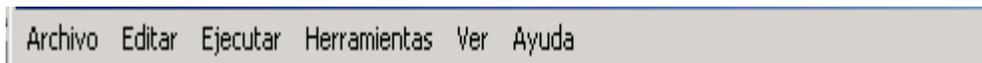


Fig. 33. Barra de Menú.

A continuación se detalla cada una de las opciones que se encuentran en la barra de menú.

Archivo:

El seleccionar esta opción, se despliegan los elementos que se observan en la figura 34.

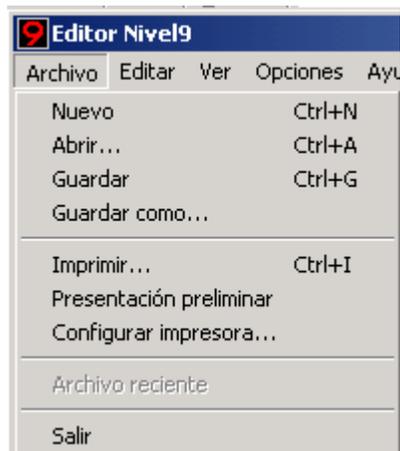


Fig. 34. Barra de menú opción Archivo.

A continuación se detalla el contenido de la opción de la barra de menú "Archivo", que pueden ser seleccionadas al activarse.

Nuevo:

Esta opción, permite limpiar la sección de la interfase de usuario del editor disponiendo el área para editar un nuevo código empleado para generar la base de conocimientos

Abrir...:

Esta opción, le permite al usuario llamar una base de conocimientos guardada previamente, con el fin de modificar y/o continuar con el código empleado para generar la base de conocimientos previamente editado.

Guardar:

Opción que permite al usuario almacenar en disco el programa del código empleado para generar la base de conocimientos editada.

Guardar como...:

Permite guardar la presente base de conocimientos con otro nombre, con el fin de poder ser modificada y mantener sin alterar el código empleado para generar la base de conocimientos anterior.

Imprimir...:

Opción que permite al usuario imprimir el contenido del código empleado para generar la base de conocimientos editada.

Presentación Preliminar:

Comando que se pretende implementar en futuras versiones del NIVEL9.

(No disponible en esta versión)

Configurar impresora...:

Esta opción presenta la ventana de diálogo de configuración del impresor. Al seleccionar esta opción permite seleccionar la impresora y configurar sus parámetros.

Salir:

Opción que permite salirse del editor.

Editar

Opción que se encuentra en la barra de menú y conlleva lo referente a la manipulación del texto que se escribe en el editor. La figura 35 muestra las opciones que pueden utilizarse.

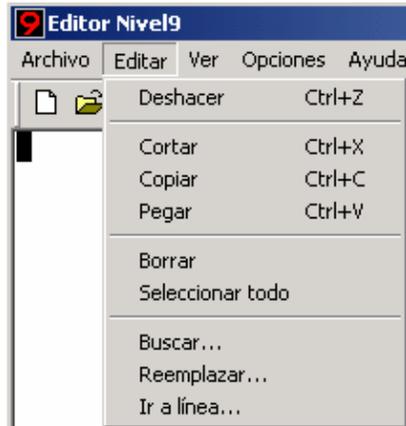


Fig. 35. Barra de menú opción Editar.

Deshacer:

Comando que restablece el cambio de la última acción de edición que se haya cambiado.

Cortar:

Comando que se utiliza para borrar los datos actualmente seleccionados del documento y colocarlos en el portapapeles. Este comando no está disponible si no hay datos seleccionados.

Al usar el comando Cortar para colocar datos en el portapapeles, se reemplazan los datos que previamente estaban almacenados en el mismo.

Copiar:

Comando que se utiliza para copiar los datos seleccionados en el portapapeles. Este comando no está disponible si no hay datos seleccionados.

Al usar el comando Copiar para colocar datos en el portapapeles, se reemplazan los datos que previamente estaban almacenados en el mismo.

Pegar:

Opción que permite al usuario insertar una copia del contenido del portapapeles en el punto de inserción. Este comando no está disponible si el portapapeles está vacío.

Borrar:

Esta opción se utiliza para borrar los datos seleccionados del documento y se difiere de la opción "Cortar" que esta instrucción NO coloca en el portapapeles lo borrado. Este comando no está disponible si no hay datos actualmente seleccionados.

Seleccionar todo:

Sombrea el contenido total del texto que en ese momento se encuentra en el editor.

Ir a línea...:

Opción que le permite al programador ubicarse en una línea determinada.

Ver

Esta opción permite al usuario mostrar u ocultar las diferentes barras de instrucciones con que cuenta la interfase.



Fig. 36. Barra de menú opción Ver.

Barra de herramientas:

Opción que permite visualizar la barra de herramientas en la interfase del editor, accionándola nuevamente oculta esta barra. Cuando se presenta un cheque al lado izquierdo del literal la barra se encuentra visible al usuario.

Barra de estado:

Permite visualizar la barra de estado en la interfase del editor, accionándola nuevamente oculta esta barra. Cuando se presenta un cheque al lado izquierdo del literal la barra se encuentra visible al usuario.

Opciones

Este ítem de la barra de menú, contiene comandos que engloban acciones referentes a: darle formato de edición al texto del código que genera la base de conocimientos, y análisis gramatical sintáctico del código que genera la base de conocimiento. La figura 37 muestra el contenido de este ítem.

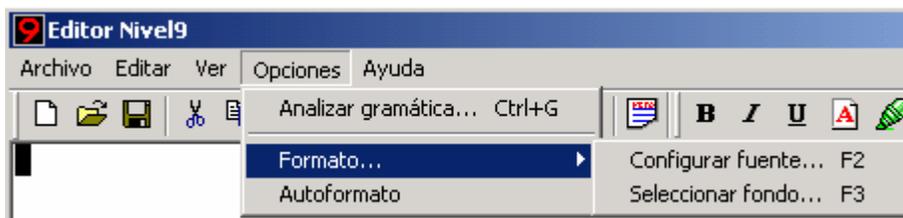


Fig. 37. Barra de menú opción Opciones.

Analizar gramática...:

El seleccionar esta opción se realiza el análisis gramatical de las instrucciones que contiene el programa de la base de conocimiento.

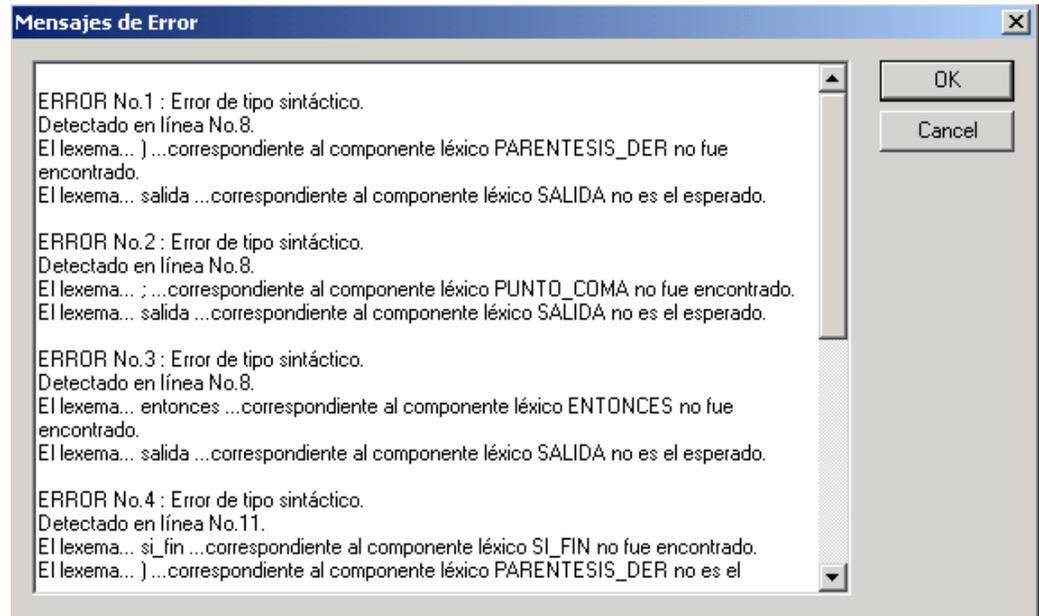


Fig. 38. Ventana de diálogo para Mensajes de errores sintácticos.

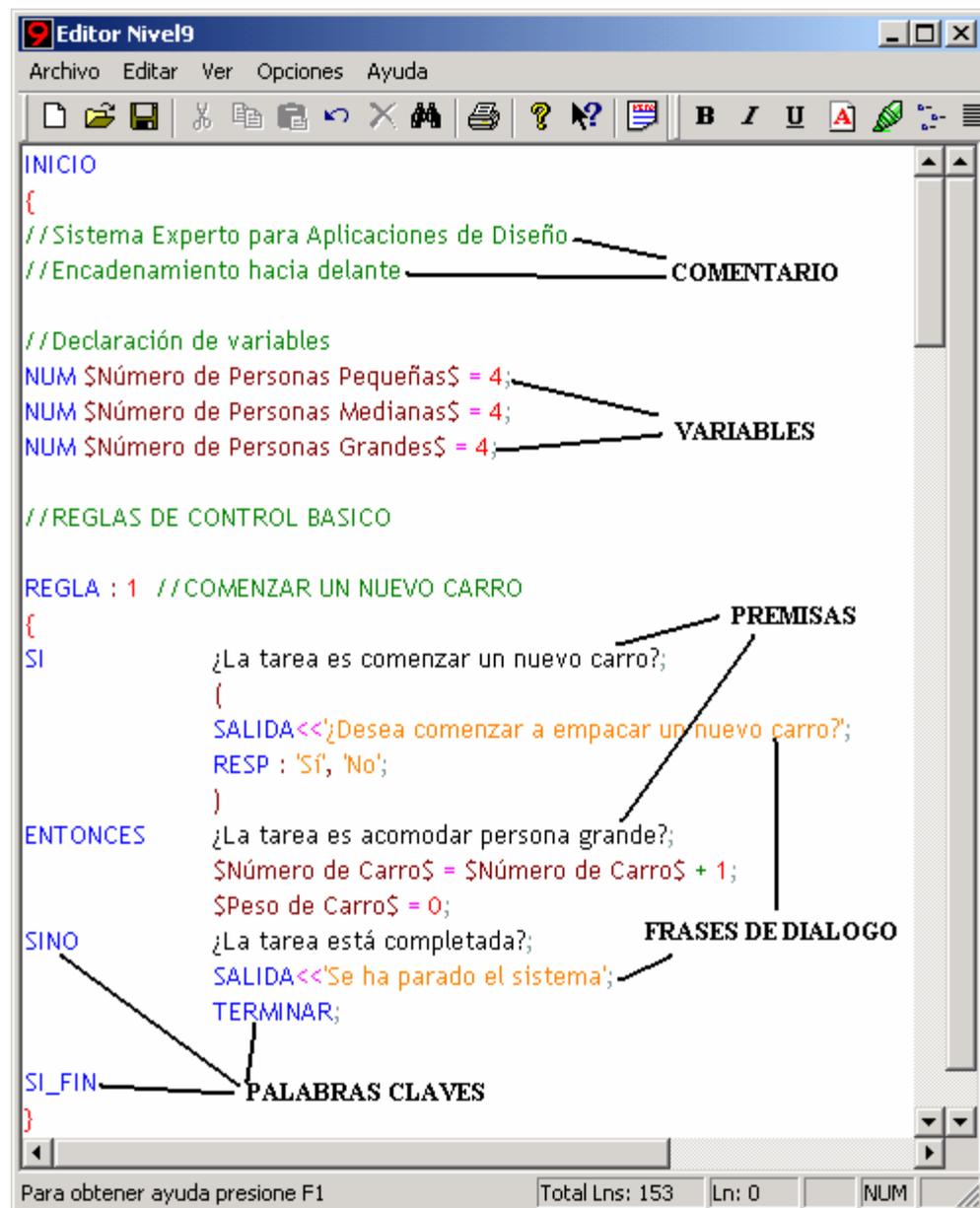
En esta ventana de diálogo se puede observar que el analizador sintáctico presenta el número del error, tipo de error que se refiere, el número de línea donde se puede localizar el error y el detalle a que se refiere el error.

Formato...:

Como puede observarse en la figura 37, Formato presenta dos opciones una es la configuración de la fuente; que se refiere a determinar el tipo de letra que se desea para la presentación del texto del programa de la base de conocimiento y a seleccionar el color del fondo que se desea para el ambiente del editor.

Autoformato:

Opción que permite colorear los símbolos utilizados y las palabras claves del lenguaje que se utiliza para la creación del código de las bases de conocimientos. Con esto puede identificarse comentarios, palabras claves del lenguaje, frases que se utilizarán como diálogo con el usuario, premisas, variables utilizadas, entre otras. Ejemplo de ello se puede ver en la figura 39.



The screenshot shows a text editor window titled "Editor Nivel9" with a menu bar (Archivo, Editar, Ver, Opciones, Ayuda) and a toolbar. The code is color-coded as follows:

- COMENTARIO:** Green text for lines starting with "//".
- VARIABLES:** Red text for variable declarations.
- PREMISAS:** Blue text for conditionals like "SI" and "SINO".
- FRASES DE DIALOGO:** Orange text for user prompts and responses.
- PALABRAS CLAVES:** Blue text for keywords like "ENTONCES", "TERMINAR", and "SI_FIN".

```
INICIO
{
//Sistema Experto para Aplicaciones de Diseño
//Encadenamiento hacia delante
//Declaración de variables
NUM $Número de Personas Pequeñas$ = 4;
NUM $Número de Personas Medianas$ = 4;
NUM $Número de Personas Grandes$ = 4;

//REGLAS DE CONTROL BASICO

REGLA : 1 //COMENZAR UN NUEVO CARRO
{
SI      ¿La tarea es comenzar un nuevo carro?
{
SALIDA<<'¿Desea comenzar a empacar un nuevo carro?';
RESP : 'Sí', 'No';
}
ENTONCES ¿La tarea es acomodar persona grande?
$Número de Carro$ = $Número de Carro$ + 1;
$Peso de Carro$ = 0;
SINO     ¿La tarea está completada?
SALIDA<<'Se ha parado el sistema';
TERMINAR;
SI_FIN
}
```

Fig. 39 Identificación de colores del autoformato

Barra de Herramientas

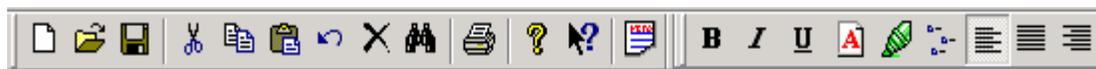


Fig. 40 Barra de Herramientas del Editor.

La barra de herramientas aparece en la parte superior de la ventana del editor de la aplicación, por debajo de la barra del menú. Mediante el ratón, la barra de herramientas brinda un acceso rápido a las herramientas de la aplicación del Editor Nivel9.

Para ocultar o mostrar la barra de herramientas, elija la opción Ver del menú y seleccione la etiqueta Barra de herramientas.

Barra de Estado



Fig. 41 Barra de Estado del Editor de NIVEL9

La barra de estado aparece en la parte inferior de la ventana del editor. Use el comando Barra de estado del menú "Ver" para mostrar u ocultar la barra de estado.

El área izquierda de la barra de estado describe las acciones de los elementos de menú mientras se usan las teclas de flecha para desplazarse por los menús. Esta área también muestra mensajes que describen la acción que realizan los botones de la barra de herramientas mientras los mantiene presionados. Si después de ver la descripción de uno de los botones de la barra de herramientas no desea ejecutar el comando, suelte el botón del ratón mientras el puntero se halla fuera de la barra de herramientas.

Las áreas de la derecha de la barra de estado indican cuál de las siguientes teclas han sido accionadas:

<u>Indicador</u>	<u>Descripción</u>
MAY	Se ha presionado la tecla Bloq Mayús.
NUM	Se ha presionado la tecla Bloq Num.
DESP	Se ha presionado la tecla Bloq Despl.

Además puede observarse dos indicadores más, ellos son: El total de líneas que presenta el programa de la base de conocimiento y la línea que en el momento se encuentra el cursor.

A continuación se presenta el detalle de los íconos que componen la barra de herramientas:

<u>Presionar</u>	<u>Para</u>
	Crear un nuevo documento
	Abrir un documento ya existente
	Guardar el documento activo
	Cortar la selección y colocarla en el portapapeles
	Copiar la selección y colocarla en el portapapeles
	Insertar el contenido del portapapeles en la posición del cursor
	Deshacer la última acción
	Borrar el texto seleccionado



Posicionar el cursor en la línea indicada



Imprimir el documento actual



Mostrar información del Editor



Mostrar ayuda de las herramientas del editor



Realizar un chequeo de la gramática del programa



Resaltar el texto seleccionado en negrita



Poner el texto en cursiva



Subrayar el texto seleccionado



Mostrar cuadro de diálogo para seleccionar color



Colorear símbolos y palabras claves del lenguaje



Colocar viñeta al texto seleccionado



Alinear el texto seleccionado a la izquierda



Centrar el texto seleccionado



Alinear el texto seleccionado a la derecha



F. EL LENGUAJE DE PROGRAMACION NIVEL9

El lenguaje de programación utilizado por software NIVEL9, para la creación de bases de conocimientos que se utilizan para dar solución a situaciones y pretenden ser consultas de expertos humanos en cualquier dominio de conocimientos, tiene cierta similitud a otros lenguajes que se utilizan comúnmente, tales como C, C++, BASIC, entre otros; a diferencia que NIVEL9 utiliza un sistema de variables, sentencias y expresiones que se asemejan al habla hispana.

Así como todo intérprete o compilador, NIVEL9 cuenta con una sintaxis de programación para la creación de bases de conocimientos. De igual forma existen declaraciones de variables, tipos de datos, utilización de sentencias y estructuras de programación, y la utilización de expresiones lógicas, aritméticas y relacionales.

NIVEL9 posee un conjunto de operadores, sentencias y expresiones sencillas que lo vuelven fácil de programar. Algunas son comunes y semejantes a otros lenguajes de programación conocidos, pero otras son propias del lenguaje.

A continuación se detalla todo el contenido del lenguaje de programación de NIVEL9.

Tipos de Datos

Los tipos de datos que se pueden utilizar para la programación en NIVEL9 son: numéricos y cadenas; utilizando las palabras claves de NUM para numéricos y CAD para cadenas.

Con la declaración NUM puede considerarse tanto variables numéricas de tipo entera como de punto flotante.

Variables

Una variable es una posición de memoria con nombre que se usa para mantener un valor que puede ser modificado por el programa. Todas las variables deben ser declaradas antes de ser utilizadas. La forma general de declaración considerando el tipo de la variable es:

Tipo \$variable\$;

Estas variables pueden ser declaradas en cualquier posición del programa, por lo que son reconocidas a lo largo de todo el programa y se pueden utilizar en

cualquier parte del código, siempre y cuando hayan sido declaradas con anterioridad.

A diferencia de otros lenguajes de programación, NIVEL9 permite que el nombre de la variable pueda tener espacios entre las palabras.

Ejemplo:

```
NUM $Variable numerica$;  
CAD $variable de cadena$;
```

Es de considerar, que al realizar este tipo de declaración los valores que por defecto toman las variables son: cero para las numéricas y vacíos para las cadenas.

También se pueden declarar variables del mismo tipo en la misma instrucción separadas por comas. En este caso los valores que toman estas variables serán los valores por defecto de cada tipo de variable.

Ejemplo:

```
NUM $Variable numérica 1$, $Variable numérica 2$;  
CAD $variable de cadena1$, $variable de cadena2$, $variable de cadena3$;
```

Es de tener en cuenta que las variables pueden ser primeramente declaradas y posteriormente, asignarles un valor.

Ejemplo:

```
NUM $Variable numerica$;  
CAD $variable de cadena$;  
  
$Variable numerica$ = 1;  
$variable de cadena$ = ' Bienvenido al mundo de NIVEL 9';
```

Además, esta acción se puede realizar en forma simultánea. Así:

```
NUM $Variable numerica$ = 1;  
CAD $variable de cadena$ = ' Bienvenido al mundo de NIVEL 9';
```

Vale la pena aclarar, que no es posible declarar varias variables en una línea y asignarles valores individuales simultáneamente. Así:

```
NUM $Variable numérica 1$ = 1 , $Variable numérica 2$ = 4;
```

Lo que sí es posible, es declarar varias variables en una sola línea y la última variable asignarle un valor. El valor que tomaran las variables que no se les asigne el valor, será el valor por defecto que toma el tipo de variable.

Ejemplo:

```
NUM $Variable numérica 1$, $Variable numérica 2$, $Variable numérica 3$ = 4;
```

En este caso, las variables \$Variable numérica 1\$ y \$Variable numérica 2\$, toman el valor por defecto del tipo de variable NUM (cero). Y la variable \$Variable numérica 3\$ toma el valor de 4.

Inicio del programa

Todo programa en NIVEL9 comienza con un delimitador de inicio utilizando la palabra clave INICIO y termina con la palabra clave FIN. Todo el contenido del programa debe estar encerrado entre estas dos palabras claves mediante las llaves de inicio y cierre ({ ...}).

La sintaxis utilizada para ello es la siguiente:

INICIO { bloque de reglas, bloque de sentencias } FIN

Todo el contenido entre las llaves que encierran estas dos palabras claves es el contenido del programa que compone el código de la base de conocimientos.

Ejemplo:

```
INICIO
{
...

REGLA :1
{
...
SALIDA<<'Bienvenido al mundo de NIVEL 9';
...
}
...
}
FIN
```

REGLA

Esta instrucción es una especie de delimitador que encierra a un conjunto de instrucciones que pueden llegar a determinar una conclusión u obtener un resultado que conlleve a otra REGLA.

La sintaxis para esta sentencia es:

```
REGLA: numero
{
.....
}
```

Donde:

REGLA	es la palabra clave que identifica la instrucción
:	Separador de los términos que componen la instrucción
numero	número entero que representa el identificador de la regla.
{ ... }	delimitadores del conjunto de instrucciones que componen
la regla	

Ejemplo:

```
REGLA :1
{
SI      ¿La temperatura del paciente es mayor de 40?;
      (
        SALIDA<<'¿Tiene el paciente ...?';
        RESP : 'Sí', 'No';
        IMAGEN : 'C:\\Nivel9_Samples\Figura.bmp';
      )
ENTONCES ¿El paciente ...?;
        SALIDA<<'El paciente ...';
SINO     ¿El paciente no ..?;
        SALIDA<<'El paciente no ...';
SI_FIN
}
```

El operador asignación

Se puede utilizar el operador de asignación en cualquier expresión válida. La forma general del operador de asignación es:

Para tipo de dato numérico: *\$nombre de la variable\$ = expresión;*

Para tipo de dato cadena: *\$nombre de la variable\$ = 'cadena';*

Donde la expresión se refiere a un valor tan simple como una constante o tan compleja como se requiera, y ´cadena´ se refiere a una cadena de caracteres tan larga como se requiere. NIVEL9 utiliza solamente el signo "=" para indicar la asignación.

Ejemplo:

```
$variable de cadena$ = ' Bienvenido al mundo de NIVEL 9';  
$Variable numerica$ = 1;  
$Variable numerica2$ = 25*$variable numerica1$ + 1;
```

Entradas y salidas

La entrada y salida de valores se definen por las palabras claves: ENTRADA y SALIDA, ofreciendo de esta forma un mecanismo flexible y a la vez consistente para transferir datos entre dispositivos. Estas palabras claves controlan la entrada por el teclado y la salida a través de la pantalla del monitor.

La sintaxis que presentan estas palabras claves se muestra a continuación:

```
SALIDA << 'LISTA DE SALIDA' ['RET '], ['ESP ']
```

```
ENTRADA >> $ID$;
```

Instrucción SALIDA

Esta instrucción se utiliza para mostrar mensajes de texto, en la interfase de usuario de NIVEL9, específicamente en el área Sesión. A continuación se explica su utilidad y sus opciones.

Ejemplo de la instrucción SALIDA:

```
SALIDA<<'Tiene el paciente ...';  
SALIDA<<$variable numerica1$;  
SALIDA<<'El resultado de la variable numérica 1 es: '<<$variable numerica1$;
```

En el ejemplo se muestran las diferentes formas en que se puede utilizar la instrucción SALIDA dentro de un programa de la base de conocimiento.

En el caso que se muestra en la primera línea, solamente se presentará al usuario el mensaje que está encerrado entre las comillas simples, en este caso 'Tiene el paciente...'; mientras que en el segundo caso, el programa presentará al usuario el contenido de la variable de tipo numérica \$variable numerica1\$, es de hacer notar que el utilizar una variable de tipo numérica es necesario encerrarla entre los signos de \$. En el tercer caso, se hace una combinación de una cadena de texto, con el valor que contiene una variable, para este caso es de observar que la separación de variables se efectúa mediante el símbolo "<<".

Al utilizar esta instrucción, la cadena o valor de variable de salida que esta instrucción presentará al usuario lo escribirá en una fila. Si se utiliza otra instrucción SALIDA, el resultado se presentará en una nueva fila. De lo contrario, se escribirá a continuación del mensaje anterior, así como se puede ver en el tercer caso del ejemplo anterior.

Los parámetros opcionales que posee la instrucción SALIDA 'ret ' y 'esp ', pueden ser utilizados para dar formato al texto de salida; donde:

'ret ' es utilizado cuando se quiere dejar una fila en blanco entre dos bloques de texto.

Ejemplo:

```
SALIDA<<'BIENVENIDO AL SISTEMA EXPERTO';  
SALIDA<<'Por favor Responda a las siguientes preguntas';  
SALIDA<<'ret '  
SALIDA<<'¿Tiene el paciente ...?';
```

El resultado que se obtiene del código anterior es:

```
>>BIENVENIDO AL SISTEMA EXPERTO  
>>Por favor Responda a las siguientes preguntas  
  
>>¿Tiene el paciente ...?
```

'esp ' es utilizado cuando se quiere que el texto presentado deje espacios definidos simulando el formato de un tabulador.

Ejemplo:

```
SALIDA<<'BIENVENIDO AL SISTEMA EXPERTO';  
SALIDA<<'Por favor Responda a las siguientes preguntas';  
SALIDA<<'esp '  
SALIDA<<'¿Tiene el paciente ...?';
```

El resultado que se obtiene del código anterior es:

```
>>BIENVENIDO AL SISTEMA EXPERTO  
>>Por favor Responda a las siguientes preguntas  
>>¿Tiene el paciente ...?
```

Instrucción ENTRADA

Esta instrucción se utiliza para obtener del usuario, el valor de una variable previamente declarada de tipo numérico. Las variables de tipo cadena no pueden ser utilizadas con esta instrucción. Es de importancia recalcar que la instrucción ENTRADA siempre debe ubicarse al final de un grupo de instrucciones seguidas; esto se debe a que esta instrucción causa una interrupción en el sistema, lo cual impediría que se ejecuten las instrucciones que se encuentran después de ella.

Ejemplo de la instrucción ENTRADA:

```
SALIDA<<'¿Cuánto marca la presión de la línea 1?';  
RESP : '10', '20', '30', '40', '50', '60', '70', '80';  
ENTRADA>>'$La presión de la Línea 1$';
```

Sentencias especiales

Nivel9, tiene sentencias especiales que no se asemejan a otros lenguajes de programación tradicionales, ellas son:

IR_A_REGLA
ELIMINAR_REGLAS
MENSAJE
TERMINAR
IMAGEN
RESP

IR_A_REGLA

Instrucción que es utilizada para saltar una o más reglas, ya sea hacia delante o hacia atrás. Esta instrucción puede simular ciclos repetitivos en el código. Además permite darle seguimiento a la estructura de árbol que debe generarse en cualquier dominio de conocimiento que se esté gestionando.

La sintaxis para esta sentencia es:

IR_A_REGLA: número;

Donde:

IR_A_REGLA es la palabra clave que identifica la instrucción.
:
 separador de los términos que componen la instrucción.

número	valor entero que representa a que regla debe saltar.
;	símbolo terminal de la instrucción.

Ejemplo:

```

REGLA : 2
{
SI      ¿El paciente ha estado ...?;
      (
        SALIDA<<'¿Ha estado el paciente ...?';
        RESP : 'Si', 'No';
        IMAGEN : 'C:\\Nivel9_Samples\\Imágenes\\Imagen2.bmp';
      )
Y      ¿El paciente tiene ...?;
ENTONCES ¿Se sospecha una ...?;
        SALIDA<<'Se sospecha una ...';
SINO    ¿El paciente no tiene ...?;
        SALIDA<<'El paciente no tiene ...';
        IR_A_REGLA: 15;
SI_FIN
}

```

ELIMINAR_REGLAS

Esta instrucción elimina, de memoria, todos los datos, almacenados en la base de conocimiento, generados por las reglas que se encuentra en el rango especificado. Esto permite que la ejecución del sistema experto sea más eficiente, debido a que las búsquedas heurísticas realizadas por el motor de inferencia del software son más cortas.

Esta instrucción tiene similitud con la instrucción IR_A_REGLA, con la diferencia de que solamente puede ir de una regla a otra hacia delante, por lo tanto no pueden simularse ciclos repetitivos, ya que las reglas son eliminadas de memoria en la base de conocimiento.

La sintaxis utilizada para esta instrucción es:

ELIMINAR_REGLAS [numero: numero]

Donde:

ELIMINAR_REGLAS	palabra clave que identifica la instrucción
[]	símbolos utilizados para encerrar el rango de reglas que se quieren eliminar.

numero:numero valores que representan el rango de reglas que se eliminarán, el primer número representa la primera regla a eliminar dentro del rango, los dos puntos ":" es el símbolo de separación y en segundo número representa la última regla a eliminar dentro del rango.

Ejemplo:

```
REGLA : 1
{
SI      ¿La temperatura ...?;
(
    SALIDA<<'¿Tiene el paciente ...?';
    RESP : 'Sí', 'No';
    IMAGEN : 'E:\Denise15.bmp';
)
ENTONCES ¿El paciente tiene ...?;
    ELIMINAR_REGLAS[ 2: 3];
SI_FIN
}
```

MENSAJE

Instrucción utilizada para mostrar al usuario una ventana de diálogo modal, que notifique o informe algún mensaje importante. Las ventanas de tipo modal, se caracterizan por interrumpir el sistema hasta que el usuario oprima el botón aceptar.

La sintaxis de esta instrucción es:

MENSAJE: 'Cadena';

Donde:

MENSAJE	palabra clave que identifica la instrucción
:	separador de los términos que componen la instrucción
'Cadena'	Mensaje que se quiere presentar al usuario
;	símbolo terminal de la instrucción.

Ejemplo:

```
REGLA : 16
{
SI      ¿El paciente tiene ...?;
      (
        SALIDA<<'¿Tiene el paciente ...?';
        RESP : 'Sí', 'No';
        IMAGEN : 'E:\Denise14.bmp';
      )
Y      ¿Se sospecha una ...?;
      (
        SALIDA<<'¿Se sospecha que el paciente ...?';
        RESP : 'Sí', 'No';
        IMAGEN : 'E:\Denise13.bmp';
      )

ENTONCES ¿Se cree que el paciente ...?;
        SALIDA<<'CONCLUSIÓN : El paciente tiene ...';
        MENSAJE : 'Este es el mensaje al final';
SI_FIN
}
```

El resultado que se obtiene al ejecutar esta instrucción, al momento de ejecutarse la instrucción MENSAJE, es el que se muestra en la figura 42.



Fig. 42 Resultado de la ejecución de la instrucción MENSAJE

TERMINAR

Esta sentencia es utilizada para codificar la finalización del sistema experto. Generalmente se emplea cuando el sistema encuentra una solución al problema.

La sintaxis utilizada es:

TERMINAR;

Ejemplo:

```
REGLA: 9
{
```

```

SI          $Número de Personas Pequeñas$ == 0;
Y          $Número de Personas Medianas$ == 0;
Y          $Número de Personas Grandes$ == 0;
ENTONCES   ¿La tarea está completada?;
SALIDA<<'El número de personas a ordenar ha finalizado';
TERMINAR;
SINO       ¿La tarea es comenzar un nuevo carro?;
IR_A_REGLA : 1;
SI_FIN
}

```

IMAGEN

Instrucción que permite al programador presentar al usuario alguna imagen que sea alusiva o que aclare la pregunta en curso.

La sintaxis de esta instrucción es:

IMAGEN: “cadena”;

Donde:

IMAGEN	Corresponde a la palabra clave de la instrucción.
:	separador de los términos de la instrucción.
“cadena”	ubicación del folder donde se encuentra la imagen que se presenta.
;	símbolo terminal de la instrucción.

Ejemplo:

```

REGLA : 19
{
SI      ¿El paciente tiene ...?;
      (
        SALIDA<<'¿Tiene el paciente ...?';
        RESP : 'Si', 'No';
        IMAGEN : 'E:\Denise14.bmp';
      )
ENTONCES ¿Se cree que el paciente ...?;
      ...
SI_FIN
}

```

RESP

Instrucción que se utiliza para mostrar al usuario todas las posibles opciones de respuestas que se pueden dar para la pregunta en curso. La finalidad de esta instrucción es el evitar en lo posible errores de digitación y obligar al usuario a responder de una forma terminada.

La sintaxis que representa esta instrucción es:

RESP: (lista_cadena);

Donde:

RESP palabra clave que se utiliza para representar la instrucción.

: separador de términos de la instrucción.

(lista_cadena) lista de valores o de caracteres que se muestra en el área de respuesta para ser seleccionadas.

; símbolo termina de la instrucción.

Ejemplo:

```
SALIDA<<'Digite el valor que marca la presión de la línea 1';  
RESP : '10', '20', '30', '40', '50', '60', '70', '80';  
ENTRADA>>$La presión de la Línea 1$;
```

Estructuras de control

Estructura de Selección SI/ENTONCES/SINO/SI_FIN

Esta estructura es utilizada, muy comúnmente, para elegir entre una acción o una alternativa, de acuerdo a cierta condición. Aquí se toma una decisión, indicando el flujo del programa que se dirigirá hacia una ruta determinada o hacia otra, o en su forma más simple, si se cumple la condición se realizará una acción, sino, no se hará nada.

Toda instrucción SI debe contener un ENTONCES y terminar con un SI_FIN.

La sintaxis que presenta esta expresión es:

SI expresión lista_sentencias ENTONCES proposición lista_sentencias
[SINO PROPOSICIÓN LISTA_SENTENCIAS]
SI_FIN

Ejemplo:

```
SI      ¿El organismo tiene ...?;
(
    SALIDA<<'¿Tiene el organismo ...?.';
    RESP: 'Si', 'No';
    IMAGEN: 'C:\antenas.bmp';
)
Y      ¿El organismo podría ....?;
ENTONCES ¿El organismo podría ...?;
IR_A_REGLA: 32;
SINO   ¿El organismo podría ser ...?;
SI_FIN
```

Puede observarse que la expresión SINO puede ser opcional o utilizarse según sea el caso, de la condición que se pretenda analizar. La condición SI, puede estar asociada a un bloque de sentencias ejecutables. La sintaxis requiere que todas las instrucciones asociadas deben encerrar entre paréntesis "()", así como cada una de las instrucciones del bloque, debe terminar con un punto y coma ";", como se muestra en el ejemplo anterior.

Es importante notar que seguido de la palabra clave ENTONCES debe ir una proposición que complete el sentido de la regla. No pueden declararse solamente sentencias.

Ejemplo:

Forma correcta:

```
SI      ¿El organismo tiene ...?;
(
    SALIDA<<'¿Tiene el organismo ...?.';
    RESP: 'Si', 'No';
    IMAGEN: 'C:\antenas.bmp';
)
Y      ¿El organismo podría ....?;
ENTONCES ¿El organismo podría ...?;
IR_A_REGLA: 32;
SINO   ¿El organismo podría ser ...?;
SI_FIN
```

Forma incorrecta

```
SI      ¿El organismo tiene ...?;
(
    SALIDA<<'¿Tiene el organismo ...?.';
    RESP: 'Si', 'No';
    IMAGEN: 'C:\antenas.bmp';
)
Y      ¿El organismo podría ....?;
ENTONCES
```

```
IR_A_REGLA: 32;  
SINO ¿El organismo podría ser ...?;  
SI_FIN
```

Estructura de repetición o cíclica MIENTRAS/HACER/MIENTRAS_FIN

Esta estructura permite al programador especificar que un programa debe repetir una acción mientras alguna condición se cumpla, es decir, que se evalúa la condición de continuidad del ciclo antes de que el ciclo se ejecute.

Toda instrucción MIENTRAS debe contener un HACER y terminar con un MIENTRAS_FIN.

La sintaxis que presenta esta expresión es:

```
MIENTRAS expresión lista sentencias HACER proposición lista_sentencias  
MIENTRAS_FIN
```

Ejemplo:

```
REGLA :2 //ACOMODAR PERSONA GRANDE  
{  
MIENTRAS ¿La tarea es acomodar persona grande?;  
Y $Número de Personas Grandes$ > 0;  
Y $Peso de Personas Grandes$ + $Peso de Carro$ <= $Peso Máximo  
de Carro$;  
HACER ¿Acomodar la persona grande?;  
$Personas Grandes en Carro$ = $Personas Grandes en Carro$ + 1;  
$Peso de Carro$ = $Peso de Carro$ + $Peso de Personas Grandes$;  
$Número de Personas Grandes$ = $Número de Personas Grandes$ -1;  
MIENTRAS_FIN  
}
```

De igual forma que en la palabra clave ENTONCES de la estructura SI, en la palabra clave HACER, debe ir una proposición que complete el sentido de la regla. No pueden declararse solamente sentencias.

Expresiones Lógicas

Los operadores lógicos "Y" y "O", generan como resultado Cierto o Falso en función de la relación lógica existente entre sus argumentos. La manera más fácil de recordar el comportamiento de una operación lógica "Y" es especificando que una expresión con operador "Y" solo se cumple cuando la expresión relacional sea cierta, es decir, si ambos argumentos son ciertos. En lo

referente al operador "O" devolverá un falso si y solo si los dos operadores son falsos.

Expresiones relacionales

Las expresiones relacionales se utilizan para comparar valores entre sí. Es de notar que solamente pueden ser utilizados valores numéricos o variables de tipo numérico. Así como en otros lenguajes de programación, NIVEL9 utiliza los operadores relacionales que se mencionan a continuación:

<u>Operador</u>	<u>Acción</u>
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que
==	Igual
<>	Distinto

Estos operadores, por lo general, son utilizados dentro de las expresiones de las estructuras de control descritas anteriormente, ellas son: SI/ENTONCES/SINO/SI_FIN y MIENTRAS/HACER/MIESTRAS_FIN.

Expresiones Aritméticas

Los operadores utilizados en NIVEL9, son los mismos operadores básicos que se usan comúnmente en los lenguajes de programación conocidos. Ellos son: suma (+), resta (-), multiplicación (*), división (/). El operador potencia (^), es un operador que se considera en este lenguaje por su utilidad.

NIVEL9 utiliza para representar un número negativos el símbolo de cierre de admiración "!", Ejemplo: -2 se representa como !2.

Precedencia de los operadores

El orden de evaluación de una expresión en NIVEL9 viene determinado por el compilador. Este normalmente no altera el valor de la expresión a menos que se haya escrito con expresiones que modifican el valor de una variable mientras se está generando un valor que se utiliza en la expresión, como puede ocurrir con los operadores de incremento, decremento y asignación.

En la tabla 1, se muestra la precedencia de los operadores utilizados en el lenguaje NIVEL9. La precedencia de NIVEL9 es un poco diferente a la que utilizan los lenguajes tradicionales. La tabla está ordenada de forma

descendente, de mayor a menor precedencia. El nivel de precedencia indica cuál operador tiene mayor precedencia, entre menor es el número del nivel, mayor precedencia tiene el operador.

NIVEL DE PRECEDENCIA	OPERADORES	NOMBRES
Operadores Especiales		
1	!	Símbolo negativo
1	()	Paréntesis
Operadores Aritméticos		
2	^	Potencia
2	*	Multiplicación
2	/	División
3	+	Suma
3	-	Resta
Operadores Relacionales		
4	<	Menor
4	<=	Menor o Igual
4	>	Mayor
4	>=	Mayor o igual
4	==	Igual
4	<>	Diferente
Operadores Lógicos		
5	Y	Y (AND)
5	O	O (OR)

Tabla 1. Precedencia de los operadores del lenguaje NIVEL9.

G. CODIGO BASE DE CONOCIMIENTOS DE LAS FAMILIAS DE LOS ANELIDOS

INICIO

{

SALIDA<< 'Universidad de El Salvador. Instituto de Ciencias del Mar y Limnología de la Universidad de El Salvador. Departamento de Ecología Marina.';

/*Copyright © derechos reservados UES.

'América Central.'*/

SALIDA<<'Clave para la identificación de las Familias de Anélidos Poliquetos de la costa de El Salvador.';

REGLA: 0

{

SI ¿El organismo tiene el dorso cubierto por paleas en hileras transversas, setas finas y abundantes que semejan pelusa o por élitros?;

(

SALIDA<<'¿Tiene el organismo dorso cubierto por paleas en hileras transversas; setas finas y abundantes que semejan pelusa, o por élitros?.';

RESP: 'Si', 'No';

IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\pelusa.bmp';

)

ENTONCES ¿El organismo podría tener el dorso cubierto por paleas en hileras transversas?;

SINO ¿El organismo podría tener extremo posterior cubierto con un escudo quitinizado y con muchas branquias anales?;

IR_A_REGLA: 6;

SI_FIN

}

REGLA: 1

{

SI ¿El organismo tiene dorso cubierto por paleas en hileras transversas?;

(

SALIDA<<'¿Tiene el organismo el dorso cubierto por paleas en hileras transversas?.';

RESP: 'Si', 'No';

IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\paleas.bmp';

)

Y ¿El organismo podría tener el dorso cubierto por paleas en hileras transversas?;

ENTONCES ¿El organismo es de la Familia Chrysopetalidae?;

SALIDA<<' El organismo es de la Familia Chrysopetalidae (Ehlers 1864)';

IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\paleas.bmp';

TERMINAR;

SINO ¿El organismo podría tener dorso cubierto por pelusa?;

SI_FIN

}

REGLA: 2

{

SI ¿El organismo tiene dorso cubierto por pelusa?;

(

SALIDA<<'¿Tiene el organismo el dorso cubierto por pelusa?.';

RESP: 'Si', 'No';

IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\pelusa.bmp';

)

Y ¿El organismo podría tener dorso cubierto por pelusa?;

ENTONCES ¿El organismo es de la Familia Aphroditidae?;

SALIDA<<' El organismo es de la Familia Aphroditidae (Malmgren 1867).';

IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Aphroditidae.bmp';

TERMINAR;

SINO ¿El organismo podría tener neuráciculas distalmente en forma de T horizontal?;

SI_FIN

}

REGLA: 3

{

SI ¿El organismo tiene neuráciculas distalmente en forma de T horizontal?;

(

```

                SALIDA<<'¿Tiene el organismo neuráciculas distalmente en forma de T horizontal?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\000.bmp';
            )
Y            ¿El organismo podría tener neuráciculas distalmente en forma de T horizontal?;
ENTONCES    ¿El organismo es de la Familia Eulepethidae?;
                SALIDA<<' El organismo es de la Familia Eulepethidae.';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\000.bmp';
                TERMINAR;
SINO        ¿El organismo podría tener neurosetas compuestas y todos los segmentos posteriores con élitros?;
SL_FIN
}

REGLA: 4
{
SI          ¿El organismo tiene neurosetas compuestas y todos los segmentos posteriores con élitros?;
            (
                SALIDA<<'¿Tiene el organismo neurosetas compuestas y todos los segmentos posteriores con élitros?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\elitra.bmp';
            )
Y            ¿El organismo podría tener neurosetas compuestas y todos los segmentos posteriores con élitros?;
ENTONCES    ¿El organismo es de la Familia Sigalionidae?;
                SALIDA<<'El organismo es de la Familia Sigalionidae.';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\sigalionidae.bmp';
                TERMINAR;
SINO        ¿El organismo podría tener neurosetas simples; sin glándulas hiladoras en forma de espiral y notosetas generalmente presentes?;
SL_FIN
}

REGLA: 5
{
SI          ¿El organismo tiene neurosetas simples; sin glándulas hiladoras en forma de espiral y notosetas generalmente presentes?;
            (
                SALIDA<<'¿Tiene el organismo neurosetas simples, sin glándulas hiladoras en forma de espiral y notosetas generalmente
                presentes?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\neuroseta simple.bmp';
            )
Y            ¿El organismo podría tener neurosetas simples; sin glándulas hiladoras en forma de espiral y notosetas generalmente presentes?;
ENTONCES    ¿El organismo es de la Familia Polynoidae?;
                SALIDA<<'El organismo es de la Familia Polynoidae.';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\polynoidae.bmp';
                TERMINAR;
SINO        ¿No se encontró una solución?;
                SALIDA<<'No se encontró una solución. Se sugiere revisar de nuevo el organismo';
                TERMINAR;
SL_FIN
}

REGLA: 6
{
SI          ¿El organismo tiene extremo posterior cubierto con un escudo quitinizado y con muchas branquias anales?;
            (
                SALIDA<<'¿Tiene el organismo extremo posterior cubierto con un escudo quitinizado y con muchas branquias anales?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\quitinizado.bmp';
                IMAGEN: 'C:\\imagenes\\Imagen6.bmp';
            )
Y            ¿El organismo podría tener extremo posterior cubierto con un escudo quitinizado y con muchas branquias anales?;
ENTONCES    ¿El organismo es de la Familia Sternaspidae?;
                SALIDA<<'El organismo es de la Familia Sternaspidae.';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Sternaspidae.bmp';
                TERMINAR;
SINO        ¿El organismo podría tener los primeros parapodios que incluyen al prostomio?;
SL_FIN
}

REGLA: 7

```

```

{
SI      ¿El organismo tiene los primeros parapodios que incluyen al prostomio?;
      (
          SALIDA<<'¿En el organismo los primeros parapodios incluyen al prostomio?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\priparapodio.bmp';
      )
Y      ¿El organismo podría tener los primeros parapodios que incluyen al prostomio?;
ENTONCES      ¿El organismo podría tener 3 pares de cirros tentaculares sostenidos con acúculas y parapodios reducidos?;
SINO      ¿El organismo podría tener el extremo anterior con setas especializadas protectoras o con opérculos para el tubo?;
          IR_A_REGLA: 9;
SI_FIN
}

REGLA: 8
{
SI      ¿El organismo tiene 3 pares de cirros tentaculares sostenidos con acúculas y parapodios reducidos?;
      (
          SALIDA<<'¿Tiene el organismo 3 pares de cirros tentaculares sostenidos con acúculas y parapodios reducidos?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\000.bmp';
      )
Y      ¿El organismo podría tener 3 pares de cirros tentaculares sostenidos con acúculas y parapodios reducidos?;
ENTONCES      ¿El organismo es de la Familia Pisionidae?;
          SALIDA<<'El organismo es de la Familia Pisionidae.';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\000.bmp';
          TERMINAR;
SINO      ¿No se encontró una solución?;
          SALIDA<<'No se encontró una solución. Se sugiere revisar de nuevo el organismo';
          TERMINAR;
SI_FIN
}

REGLA: 9
{
SI      ¿El organismo tiene el extremo anterior con setas especializadas protectoras o con opérculo para el tubo?;
      (
          SALIDA<<'¿Tiene el organismo el extremo anterior con setas especializadas protectoras o con opérculo para el tubo?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\operculo.bmp';
      )
Y      ¿El organismo podría tener el extremo anterior con setas especializadas protectoras o con opérculos para el tubo?;
ENTONCES      ¿El organismo podría tener extremo anterior con opérculo calcáreo o setas fusionadas?;
SINO      ¿El organismo podría tener prostomio o peristomio con menos de 10 pares de apéndices o sin ellos?;
          IR_A_REGLA: 16;
SI_FIN
}

REGLA: 10
{
SI      ¿El organismo tiene extremo anterior con opérculo calcáreo o setas fusionadas?;
      (
          SALIDA<<'¿Tiene el organismo extremo anterior con opérculo calcáreo o setas fusionadas?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\calcareo.bmp';
      )
Y      ¿El organismo podría tener extremo anterior con opérculo calcáreo o setas fusionadas?;
ENTONCES      ¿El organismo podría tener tubo de grano de arena sementado y opérculo formado por 2 ó 3 hileras concéntricas de setas?;
          IR_A_REGLA: 14;
SINO      ¿El organismo podría tener setas especializadas tabicadas que forman una caja cefálica?;
SI_FIN
}

REGLA: 11
{
SI      ¿El organismo tiene setas especializadas tabicadas que forman una caja cefálica?;
      (
          SALIDA<<'¿Tiene el organismo setas especializadas tabicadas que forman una caja cefálica?.';
          RESP: 'Si', 'No';
      )

```

```

                IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\tabicadas.bmp';
            )
Y
ENTONCES ¿El organismo podría tener setas especializadas tabicadas que forman una caja cefálica?;
                ¿El organismo es de la Familia Flabelligeridae?;
                SALIDA<<'El organismo es de la Familia Flabelligeridae (Saint-Joseph 1894).';
                IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\Flabelligeridae.bmp';
                TERMINAR;
SINO ¿El organismo podría tener 7 antenas: 5 occipitales mayores y 2 frontales cortas y cónicas?;
SL_FIN
}

REGLA: 12
{
SI ¿El organismo tiene 7 antenas: 5 occipitales mayores y 2 frontales cortas y cónicas?;
(
                SALIDA<<'¿Tiene el organismo 7 antenas: 5 occipitales mayores y 2 frontales cortas y cónicas?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\antenas.bmp';
            )
Y
ENTONCES ¿El organismo podría tener 7 antenas: 5 occipitales mayores y 2 frontales cortas y cónicas?;
                ¿El organismo es de la Familia Onuphidae?;
                SALIDA<<'El organismo es de la Familia Onuphidae (Kinberg 1865).';
                IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\onuphidae.bmp';
                TERMINAR;
SINO ¿El organismo podría tener setas gruesas anteriores en hilera transversa?;
SL_FIN
}

REGLA: 13
{
SI ¿El organismo tiene setas gruesas anteriores en hilera transversa?;
(
                SALIDA<<'¿Tiene el organismo setas gruesas anteriores en hilera transversa?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\setagruesa.bmp';
            )
Y
ENTONCES ¿El organismo podría tener setas gruesas anteriores en hilera transversa?;
                ¿El organismo es de la Familia Pectinariidae?;
                SALIDA<<'El organismo es de la Familia Pectinariidae (Quatrefages 1865).';
                IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\Pectinariidae.bmp';
                TERMINAR;
SINO ¿El organismo es de la Familia Ampharetidae?;
                SALIDA<<'El organismo es de la Familia Ampharetidae (Malmgren 1867).';
                IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\Ampharethidae.bmp';
                TERMINAR;
SL_FIN
}

REGLA: 14
{
SI ¿El organismo tiene tubo de grano de arena sementado y opérculo formado por 2 ó 3 hileras concéntricas de setas?;
(
                SALIDA<<'¿Tiene el organismo tubo de grano de arena cementado y opérculo formado por 2 ó 3 hileras concéntricas de
                setas?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\tubo.bmp';
            )
Y
ENTONCES ¿El organismo podría tener tubo de grano de arena sementado y opérculo formado por 2 ó 3 hileras concéntricas de setas?;
                ¿El organismo es de la Familia Sabellaridae?;
                SALIDA<<'El organismo es de la Familia Sabellaridae.';
                IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\Sabellaridae.bmp';
                TERMINAR;
SINO ¿El organismo podría tener cuerpo simétrico con más de 4 setígeros torácicos y opérculo formado por un radiolo
modificado?;
SL_FIN
}

REGLA: 15
{
SI ¿El organismo tiene cuerpo simétrico con más de 4 setígeros torácicos y opérculo formado por un radiolo modificado?;

```

```

(
    SALIDA<<'¿Tiene el organismo cuerpo simétrico con más de 4 setíferos torácicos; opérculo formado por un radiolo
modificado?.';
    RESP: 'Si', 'No';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\simetrico.bmp';
)
Y
    ¿El organismo podría tener cuerpo simétrico con más de 4 setíferos torácicos y opérculo formado por un radiolo modificado?;
ENTONCES
    ¿El organismo es de la Familia Serpulidae?;
    SALIDA<<'El organismo es de la Familia Serpulidae.';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Serpulidae.bmp';
    TERMINAR;
SINO
    ¿No se encontró una solución?;
    SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente al organismo';
    TERMINAR;

SL_FIN
}

REGLA: 16
{
SI
    ¿El organismo tiene prostomio o peristomio con menos de 10 pares de apéndices o sin ellos?;
(
    SALIDA<<'¿Tiene el organismo prostomio o peristomio con menos de 10 pares de apéndices o sin ellos?.';
    RESP: 'Si', 'No';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\sinapendice.bmp';
)
Y
    ¿El organismo podría tener prostomio o peristomio con menos de 10 pares de apéndices o sin ellos?;
ENTONCES
    ¿El organismo podría tener setífero 4 con uno o varias espinas gruesas y tubo apergaminado?;
    IR_A_REGLA: 21;
SINO
    ¿El organismo podría tener cuerpo con branquias?;
SL_FIN
}

REGLA: 17
{
SI
    ¿El organismo tiene cuerpo con branquias?;
(
    SALIDA<<'¿Tiene el organismo cuerpo con branquias?.';
    RESP: 'Si', 'No';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\branquia.bmp';
)
Y
    ¿El organismo podría tener cuerpo con branquias?;
ENTONCES
    ¿El organismo podría tener branquias en hilera transversa y tentáculos bucales retráctiles?;
SINO
    ¿El organismo podría tener corona tentacular bien desarrollada y segmentos cortos?;
    IR_A_REGLA: 20;
SL_FIN
}

REGLA: 18
{
SI
    ¿El organismo tiene branquias en hilera transversa y tentáculos bucales retráctiles?;
(
    SALIDA<<'¿Tiene el organismo branquias en hilera transversa y tentáculos bucales retractiles?.';
    RESP: 'Si', 'No';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\transversa.bmp';
)
Y
    ¿El organismo podría tener branquias en hilera transversa y tentáculos bucales retráctiles?;
ENTONCES
    ¿El organismo es de la Familia Ampharetidae?;
    SALIDA<<'El organismo es de la Familia Ampharetidae. (Malmgren 1867).';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Ampharethidae.bmp';
    TERMINAR;
SINO
    ¿El organismo podría tener uncinos torácicos de manubrio largo y branquia pedunculada de 4 lóbulos?;
SL_FIN
}

REGLA: 19
{
SI
    ¿El organismo tiene uncinos torácicos de manubrio largo y branquia pedunculada de 4 lóbulos?;
(
    SALIDA<<'¿Tiene el organismo uncinos torácicos de manubrio largo y branquias pedunculada de 4 lóbulos?.';
    RESP: 'Si', 'No';

```

```

        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\uncinolargo.bmp';
    )
    Y
    ENTONCES
        ¿El organismo podría tener uncinos torácicos de manubrio largo y branquias pedunculada de 4 lóbulos?;
        ¿El organismo es de la Familia Trichobranchidae?;
        SALIDA<<'El organismo es de la Familia Trichobranchidae (Malmgren 1866).';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Trichobranchide.bmp';
        TERMINAR;
SINO
    ¿El organismo es de la Familia Terebellidae?;
    SALIDA<<'El organismo es de la Familia Terebellidae (Malmgren 1867).';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Terebellidae.bmp';
    TERMINAR;
SI_FIN
}

REGLA: 20
{
SI
    ¿El organismo tiene corona tentacular bien desarrollada y segmentos cortos?;
    (
        SALIDA<<'¿Tiene el organismo corona tentacular bien desarrollada y segmentos cortos?.';
        RESP: 'Si', 'No';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\corona.bmp';
    )
    Y
    ENTONCES
        ¿El organismo podría tener corona tentacular bien desarrollada y segmentos cortos?;
        ¿El organismo es de la Familia Sabellidae?;
        SALIDA<<'El organismo es de la Familia Sabellidae (Malmgren 1867).';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\corona.bmp';
        TERMINAR;
SINO
    ¿El organismo es de la Familia Owenidae?;
    SALIDA<<'El organismo es de la Familia Owenidae (Rioja 1917).';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\owenidae.bmp';
    TERMINAR;
SI_FIN
}

REGLA: 21
{
SI
    ¿El organismo tiene setífero 4 con una o varias espinas gruesas y tubo apergaminado?;
    (
        SALIDA<<'¿Tiene el organismo setífero 4 con uno o varias espinas gruesas y tubo apergaminado?.';
        RESP: 'Si', 'No';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\setifero4.bmp';
    )
    Y
    ENTONCES
        ¿El organismo podría tener setífero 4 con una o varias espinas gruesas y tubo apergaminado?;
        ¿El organismo es de la Familia Chaetopteridae?;
        SALIDA<<'El organismo es de la Familia Chaetopteridae (Malmgren 1867).';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\chaetopteridae.bmp';
        TERMINAR;
SINO
    ¿El organismo podría tener prostomio con 2 ó más antenas?;
SI_FIN
}

REGLA: 22
{
SI
    ¿El organismo tiene prostomio con 2 ó más antenas?;
    (
        SALIDA<<'¿Tiene el organismo prostomio con 2 ó más antenas?.';
        RESP: 'Si', 'No';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\antenas.bmp';
    )
    Y
    ENTONCES
        ¿El organismo podría tener prostomio con 2 ó más antenas?;
        ¿El organismo podría tener carúnculo?;
SINO
    ¿El organismo podría tener prostomio y peristomio sin apéndices?;
    IR_A_REGLA: 40;
SI_FIN
}

REGLA: 23
{
SI
    ¿El organismo tiene carúnculo?;
    (

```

```

                SALIDA<<'¿Tiene el organismo carúnculo?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\carunculo.bmp';
            )
Y                ¿El organismo podría tener carúnculo?;
ENTONCES        ¿El organismo es de la Familia Amphinomidae?;
                SALIDA<<'El organismo es de la Familia Amphinomidae (Savigny 1818).';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Amphinomidae.bmp';
                TERMINAR;
SINO            ¿El organismo podría tener palpos?;
SL_FIN
}

REGLA: 24
{
SI                ¿El organismo tiene palpos?;
                (
                    SALIDA<<'¿Tiene el organismo palpos?.';
                    RESP: 'Si', 'No';
                    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\antenas.bmp';
                )
Y                ¿El organismo podría tener palpos?;
ENTONCES        ¿El organismo podría tener palpos simples?;
                IR_A_REGLA: 32;
SINO            ¿El organismo podría ser holopelágico; sin setas y acículas limitadas a lóbulos aciculares del segmento 1 ó 2?;
SL_FIN
}

REGLA: 25
{
SI                ¿El organismo es holopelágico; sin setas y acículas limitadas a lóbulos aciculares del segmento 1 ó 2?;
                (
                    SALIDA<<'¿Es el organismo holopelágico; sin setas y acículas limitadas a lóbulos aciculares del segmento 1 ó 2?.';
                    RESP: 'Si', 'No';
                    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\?.bmp';
                )
Y                ¿El organismo podría ser holopelágico; sin setas y acículas limitadas a lóbulos aciculares del segmento 1 ó 2?;
ENTONCES        ¿El organismo es de la Familia Tomopteridae?;
                SALIDA<<'El organismo es de la Familia Tomopteridae (Grube 1848).';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\?.bmp';
                TERMINAR;
SINO            ¿El organismo podría tener prostomio largo, cónico y anulado?;
SL_FIN
}

REGLA: 26
{
SI                ¿El organismo tiene prostomio largo, cónico y anulado?;
                (
                    SALIDA<<'¿Tiene el organismo prostomio largo, cónico y anulado?.';
                    RESP: 'Si', 'No';
                    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\prostomiolargo.bmp';
                )
Y                ¿El organismo podría tener prostomio largo, cónico y anulado?;
ENTONCES        ¿El organismo podría tener probóscide con 4 mandíbulas en cruz?;
                IR_A_REGLA: 31;
SINO            ¿El organismo podría tener cirro dorsal muy desarrollado, folioso o digitado - lobular?;
SL_FIN
}

REGLA: 29
{
SI                ¿El organismo tiene cirro dorsal muy desarrollado, folioso o digitado - lobular?;
                (
                    SALIDA<<'¿Tiene el organismo cirro dorsal muy desarrollado, folioso o digitado - globular?.';
                    RESP: 'Si', 'No';
                    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\cirrodorsal.bmp';
                )
Y                ¿El organismo podría tener cirro dorsal muy desarrollado, folioso o digitado - lobular?;

```

```

ENTONCES      ¿El organismo es de la Familia Phyllocidae?;
              SALIDA<<'El organismo es de la Familia Phyllocidae (Williams 1851).';
              IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Phyllocidae.bmp';
              TERMINAR;
SINO          ¿El organismo podría tener branquia intermedia entre noto y neuropodio, y setas simples?;
SI_FIN
}

REGLA: 30
{
SI           ¿El organismo tiene branquia intermedia entre noto y neuropodio, y setas simples?;
            (
              SALIDA<<'¿Tiene el organismo branquia intermedia entre noto y neuropodio, y setas simples?.';
              RESP: 'Si', 'No';
              IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\branquiaintermedia.bmp';
            )
Y           ¿El organismo podría tener branquia intermedia entre noto y neuropodio, y setas simples?;
ENTONCES    ¿El organismo es de la Familia Nephthydae?;
            SALIDA<<'El organismo es de la Familia Nephthydae (Grube 1850).';
            IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Nephtyidae.bmp';
            TERMINAR;
SINO        ¿No se encontró una solución?;
            SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente el organismo';
            TERMINAR;
SI_FIN
}

REGLA: 31
{
SI           ¿El organismo tiene probóscide con 4 mandíbulas en cruz?;
            (
              SALIDA<<'¿Tiene el organismo probóscide con 4 mandíbulas en cruz?.';
              RESP: 'Si', 'No';
              IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\probóscide.bmp';
            )
Y           ¿El organismo podría tener probóscide con 4 mandíbulas en cruz?;
ENTONCES    ¿El organismo es de la Familia Glyceridae?;
            SALIDA<<'El organismo es de la Familia Glyceridae (Grube 1850).';
            IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\glyceridae.bmp';
            TERMINAR;
SINO        ¿El organismo es de la Familia Goniadidae?;
            SALIDA<<'El organismo es de la Familia Goniadidae (Kinberg 1866).';
            IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\goniadiidae.bmp';
            TERMINAR;
SI_FIN
}

REGLA: 32
{
SI           ¿El organismo tiene palpos simples?;
            (
              SALIDA<<'¿Tiene el organismo palpos simples?.';
              RESP: 'Si', 'No';
              IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\antenas.bmp';
            )
Y           ¿El organismo podría tener palpos simples?;
ENTONCES    ¿El organismo podría tener 7 antenas: 2 frontales cortas y 5 occipitales largas?;
SINO        ¿El organismo podría tener palpos bi o multiarticulados, probóscide con mandíbulas dentadas con paragnatos, papilas o ambos,
            notosetas compuestas y hasta 4 pares de cirros tentaculares?;
            IR_A_REGLA: 38;
SI_FIN
}

REGLA: 33
{
SI           ¿El organismo tiene 7 antenas: 2 frontales cortas y 5 occipitales largas?;
            (
              SALIDA<<'¿Tiene el organismo 7 antenas: 2 frontales cortas y 5 occipitales largas?.';
              RESP: 'Si', 'No';
              IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\antenas.bmp';
            )

```

```

)
Y      ¿El organismo podría tener 7 antenas: 2 frontales cortas y 5 occipitales largas?;
ENTONCES      ¿El organismo es de la Familia Onuphidae?;
                SALIDA<<'El organismo es de la Familia Onuphidae (Kinberg 1865).';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\onuphidae.bmp';
                TERMINAR;
SINO      ¿El organismo podría tener palpos como proyecciones y ventro laterales libres?;
SI_FIN
}

REGLA: 34
{
SI      ¿El organismo tiene palpos como proyecciones y ventro laterales libres?;
        (
                SALIDA<<'¿Tiene El organismo palpos como proyecciones y ventro laterales libres?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\palposlibres.bmp';
        )
Y      ¿El organismo podría tener palpos como proyecciones y ventro laterales libres?;
ENTONCES      ¿El organismo podría tener prostomio más largo que ancho?;
SINO      ¿El organismo podría tener probóscide inerme, papilada o con 2 mandíbulas hasta 3 antenas?;
                IR_A_REGLA: 37;
SI_FIN
}

REGLA: 35
{
SI      ¿El organismo tiene prostomio más largo que ancho?;
        (
                SALIDA<<'¿Tiene el organismo prostomio más largo que ancho?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\000.bmp';
        )
Y      ¿El organismo podría tener prostomio más largo que ancho?;
ENTONCES      ¿El organismo es de la Familia Pisionidae?;
                SALIDA<<'El organismo es de la Familia Pisionidae (Southern 1914).';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\000.bmp';
                TERMINAR;
SINO      ¿El organismo podría tener parapodios poco desarrollados o reducidos con acículas cortas y con proventrículo?;
SI_FIN
}

REGLA: 36
{
SI      ¿El organismo tiene parapodios poco desarrollados o reducidos con acículas cortas y con proventrículo?;
        (
                SALIDA<<'¿Tiene el organismo parapodios poco desarrollados o reducidos con acículas cortas y con proventrículo?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\proventrículo.bmp';
        )
Y      ¿El organismo podría tener parapodios poco desarrollados o reducidos con acículas cortas y con proventrículo?;
ENTONCES      ¿El organismo es de la Familia Syllidae?;
                SALIDA<<'El organismo es de la Familia Syllidae (Grube 1850).';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\palposlibres.bmp';
                TERMINAR;
SINO      ¿No se encontro una solución?;
                SALIDA<<'No se encontró una solución.';
                TERMINAR;
SI_FIN
}

REGLA: 37
{
SI      ¿El organismo tiene probóscide inerme, papilada o con 2 mandíbulas hasta 3 antenas?;
        (
                SALIDA<<'¿Tiene el organismo probóscidae inerme, papilada o con 2 mandíbulas hasta 3 antenas?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\proboscideinerme.bmp';
        )
}

```

Y ¿El organismo podría tener probóscide inerme, papilada o con 2 mandíbulas hasta 3 antenas?
ENTONCES ¿El organismo es de la Familia Pilargidae?
SALIDA<<'El organismo es de la Familia Pilargidae (Saint-Joseph 1899).';
IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\pilargidae.bmp';
TERMINAR;

SINO ¿El organismo es de la Familia Eunicidae?
SALIDA<<'El organismo es de la Familia Eunicidae (Savigny 1818).';
IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Eunicidae.bmp';
TERMINAR;

SI_FIN
}

REGLA: 38
{

SI ¿El organismo tiene palpos bi o multiarticulados, probóscide con mandíbulas dentadas con paragnatos, papilas o ambos, notosetas compuestas y hasta 4 pares de cirros tentaculares?
(
SALIDA<<'¿Tiene el organismo palpos bi o multiarticulados, probóscide con mandíbulas dentadas con paragnatos, papilas o ambos, notosetas compuestas y hasta 4 pares de cirros tentaculares?.';
RESP: 'Si', 'No';
IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\paragnato.bmp';
)

Y ¿El organismo podría tener palpos bi o multiarticulados, probóscide con mandíbulas dentadas con paragnatos, papilas o ambos, notosetas compuestas y hasta 4 pares de cirros tentaculares?
ENTONCES ¿El organismo es de la Familia Nereidae?
SALIDA<<'El organismo es de la Familia Nereidae (Johnston 1845).';
IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Nereidae.bmp';
TERMINAR;

SINO ¿El organismo podría tener probóscide con mandíbulas lisas o sin ellas, sin paragnatos ni papilas, notosetas simples o ausentes y hasta 8 pares de cirros tentaculares?
SI_FIN
}

REGLA: 39
{

SI ¿El organismo tiene probóscide con mandíbulas lisas o sin ellas, sin paragnatos ni papilas, notosetas simples o ausentes y hasta 8 pares de cirros tentaculares?
(
SALIDA<<'¿Tiene el organismo probóscide con mandíbulas lisas o sin ellas, sin paragnatos ni papilas, notosetas simples o ausentes y hasta 8 pares de cirros tentaculares?.';
RESP: 'Si', 'No';
IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\notoseta.bmp';
)

Y ¿El organismo podría tener probóscide con mandíbulas lisas o sin ellas, sin paragnatos ni papilas, notosetas simples o ausentes y hasta 8 pares de cirros tentaculares?
ENTONCES ¿El organismo es de la Familia Hesionidae?
SALIDA<<'El organismo es de la Familia Hesionidae (Sars 1862).';
IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Hesionidae.bmp';
TERMINAR;

SINO ¿No se encontro una solución?
SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente el organismo.';
TERMINAR;

SI_FIN
}

REGLA: 40
{

SI ¿El organismo tiene prostomio y peristomio sin apéndices?
(
SALIDA<<'¿Tiene el organismo prostomio y peristomio sin apéndices?.';
RESP: 'Si', 'No';
IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\sinapendice.bmp';
)

Y ¿El organismo podría tener prostomio y peristomio sin apéndices?
ENTONCES ¿El organismo podría tener segmentos post-perisomiales con palpos pares?
SINO ¿El organismo podría tener prostomio con una antena media?
IR_A_REGLA: 61;

SI_FIN
}

```

REGLA: 41
{
SI      ¿El organismo tiene segmentos post-peristomiales con palpos pares?;
      (
          SALIDA<<'¿Tiene el organismo segmentos post-peristomiales con palpos pares?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\41.bmp';
      )
Y      ¿El organismo podría tener segmentos post-peristomiales con palpos pares?;
ENTONCES      ¿El organismo podría tener parapodios con setas simples y falcigeros compuestos, y tegumento liso?;
SINO      ¿El organismo podría tener una sola branquia medio dorsal en un setífero anterior?;
          IR_A_REGLA: 43;
SI_FIN
}

REGLA: 42
{
SI      ¿El organismo tiene parapodios con setas simples y falcigeros compuestos, y tegumento liso?;
      (
          SALIDA<<'¿Tiene el organismo parapodios con setas simples y falcigeros compuestos, y tegumento liso?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\42.bmp';
      )
Y      ¿El organismo podría tener parapodios con setas simples y falcigeros compuestos, y tegumento liso?;
ENTONCES      ¿El organismo es de la Familia Cirratulidae?;
          SALIDA<<'El organismo es de la Familia Cirratulidae (Carus 1863) .';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\cirratulidae.bmp';
          TERMINAR;
SINO      ¿No se encontro una solución?;
          SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente el organismo.';
          TERMINAR;
SI_FIN
}

REGLA: 43
{
SI      ¿El organismo tiene una sola branquia medio dorsal en un setífero anterior?;
      (
          SALIDA<<'¿Tiene el organismo una sola branquia medio dorsal en un setífero anterior?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\43.bmp';
      )
Y      ¿El organismo podría tener una sola branquia medio dorsal en un setífero anterior?;
ENTONCES      ¿El organismo es de la Familia Cossuridae?;
          SALIDA<<'El organismo es de la Familia Cossuridae (Day 1963).';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Cossuridae.bmp';
          TERMINAR;
SINO      ¿El organismo podría tener parapodios bien desarrollados o como rebordes bajos?;
SI_FIN
}

REGLA: 44
{
SI      ¿El organismo tiene parapodios bien desarrollados o como rebordes bajos?;
      (
          SALIDA<<'¿Tiene el organismo parapodios bien desarrollados o como rebordes bajos?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\44.bmp';
      )
Y      ¿El organismo podría tener parapodios bien desarrollados o como rebordes bajos?;
ENTONCES      ¿El organismo podría tener prostomio aguzado, romo o redondeado?;
          IR_A_REGLA: 46;
SINO      ¿El organismo podría tener parapodios con algunas setas multitabicadas?;
SI_FIN
}

REGLA: 45
{
SI      ¿El organismo tiene parapodios con algunas setas multitabicadas?;

```

```

(
    SALIDA<<'¿Tiene el organismo parapodios con algunas setas multibacadas?.';
    RESP: 'Si', 'No';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\tabicadas.bmp';
)
Y
¿El organismo podría tener parapodios con algunas setas multibacadas?;
ENTONCES
    ¿El organismo es de la Familia Flabelligeridae?;
    SALIDA<<'El organismo es de la Familia Flabelligeridae.';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Flabelligeridae.bmp';
    TERMINAR;
SINO
    ¿No se encontró una solución?;
    SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente el organismo.';
    TERMINAR;
SI_FIN
}

REGLA: 46
{
SI
    ¿El organismo tiene prostomio aguzado, romo o redondeado?;
    (
        SALIDA<<'¿Tiene el organismo prostomio aguzado, romo o redondeado?.';
        RESP: 'Si', 'No';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Orbiniidae.bmp';
    )
Y
    ¿El organismo podría tener prostomio aguzado, romo o redondeado?;
ENTONCES
    ¿El organismo podría tener cuerpo separable en tórax y abdomen?;
    IR_A_REGLA: 48;
SINO
    ¿El organismo podría tener segmentos alargados y cilíndricos, sin setas bilimbadas ni setas acompañantes y sin grandes espinas
    posteriores?;
SI_FIN
}

REGLA: 47
{
SI
    ¿El organismo tiene segmentos alargados y cilíndricos, sin setas bilimbadas ni setas acompañantes y sin grandes espinas posteriores?;
    (
        SALIDA<<'¿Tiene el organismo segmentos alargados y cilíndricos, sin setas bilimbadas ni setas acompañantes y sin grandes
        espinas posteriores?.';
        RESP: 'Si', 'No';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Maldanidae.bmp';
    )
Y
    ¿El organismo podría tener segmentos alargados y cilíndricos, sin setas bilimbadas ni setas acompañantes y sin grandes espinas
    posteriores?;
ENTONCES
    ¿El organismo es de la Familia Maldanidae?;
    SALIDA<<'El organismo es de la Familia Maldanidae (Malmgren 1867).';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Maldanidae.bmp';
    TERMINAR;
SINO
    ¿No se encontró una solución?;
    SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente el organismo.';
    TERMINAR;
SI_FIN
}

REGLA: 48
{
SI
    ¿El organismo tiene cuerpo separable en tórax y abdomen?;
    (
        SALIDA<<'¿Tiene el organismo cuerpo separable en tórax y abdomen?.';
        RESP: 'Si', 'No';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\48.bmp';
    )
Y
    ¿El organismo podría tener cuerpo separable en tórax y abdomen?;
ENTONCES
    ¿El organismo podría tener parapodios laterales en tórax, dorsales en abdomen?;
    IR_A_REGLA: 58;
SINO
    ¿El organismo podría tener probóscido con aparato mandibular complejo y mandíbula de una sola pieza?;
SI_FIN
}

REGLA: 49
{

```

```

SI      ¿El organismo tiene probóscide con aparato mandibular complejo y mandíbula de una sola pieza?;
      (
          SALIDA<<'¿Tiene el organismo probóscide con aparato mandibular complejo y mandíbula de una sola pieza?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\49.bmp';
      )
Y      ¿El organismo podría tener probóscide con aparato mandibular complejo y mandíbula de una sola pieza?;
ENTONCES ¿El organismo podría tener segmentos anteriores con ganchos cubiertos y con 2 portadores maxilares; generalmente sin
ojos?;
      IR_A_REGLA: 57;
SINO    ¿El organismo podría tener branquias en 15 - 20 segmentos desde los setígeros 4 - 10?;
SI_FIN  }

REGLA: 50
{
SI      ¿El organismo tiene branquias en 15 - 20 segmentos desde los setígeros 4 - 10?;
      (
          SALIDA<<'¿Tiene el organismo branquias en 15 - 20 segmentos desde los setígeros 4 - 10?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\50.bmp';
      )
Y      ¿El organismo podría tener branquias en 15 - 20 segmentos desde los setígeros 4 - 10?;
ENTONCES ¿El organismo es de la Familia Paraonidae?;
          SALIDA<<'El organismo es de la Familia Paraonidae (Cerruti 1909).';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\paraonidae.bmp';
          TERMINAR;
SINO    ¿El organismo podría tener branquias en otra disposición y segmentos alargados y cilíndricos?;
SI_FIN  }

REGLA: 51
{
SI      ¿El organismo tiene branquias en otra disposición y segmentos alargados y cilíndricos?;
      (
          SALIDA<<'¿Tiene el organismo branquias en otra disposición y segmentos alargados y cilíndricos?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\Maldanidae.bmp';
      )
Y      ¿El organismo podría tener branquias en otra disposición y segmentos alargados y cilíndricos?;
ENTONCES ¿El organismo es de la Familia Maldanidae?;
          SALIDA<<'El organismo es de la Familia Maldanidae.';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\Maldanidae.bmp';
          TERMINAR;
SINO    ¿El organismo podría tener prostomio como un cono alargado, generalmente anulado?;
SI_FIN  }

REGLA: 52
{
SI      ¿El organismo tiene prostomio como un cono alargado, generalmente anulado?;
      (
          SALIDA<<'¿Tiene el organismo prostomio como un cono alargado, generalmente anulado?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\prostomiolargo.bmp';
      )
Y      ¿El organismo podría tener prostomio como un cono alargado, generalmente anulado?;
ENTONCES ¿El organismo podría tener prostomio con 4 mandíbulas y parapodios todos uni o birrámeos?;
SINO    ¿El organismo podría tener prostomio entero, aguzado o redondeado y branquias cirríformes, pectinadas o ausentes?;
      IR_A_REGLA: 55;
SI_FIN  }

REGLA: 53
{
SI      ¿El organismo tiene prostomio con 4 mandíbulas y parapodios todos uni o birrámeos?;
      (
          SALIDA<<'¿Tiene el organismo prostomio con 4 mandíbulas y parapodios todos uni o birrámeos?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imagenes\Especies\ANELIDOS\probóscide.bmp';
      )

```

```

)
Y      ¿El organismo podría tener prostomio con 4 mandíbulas y parapodios todos uni o birrámeos?;
ENTONCES      ¿El organismo es de la Familia Glyceriadae?;
                SALIDA<<'El organismo es de la Familia Glyceriadae (Grube 1850).';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\glyceriadae.bmp';
                TERMINAR;
SINO      ¿El organismo podría tener más de 4 mandíbulas y parapodios anteriores unirrámeos, posteriores birrámeos?;
SI_FIN
}

REGLA: 54
{
SI      ¿El organismo tiene más de 4 mandíbulas y parapodios anteriores unirrámeos y posteriores birrámeos?;
        (
                SALIDA<<'¿Tiene el organismo más de 4 mandíbulas y parapodios anteriores unirrámeos y posteriores birrámeos?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\goniadiidae.bmp';
        )
Y      ¿El organismo podría tener más de 4 mandíbulas y parapodios anteriores unirrámeos y posteriores birrámeos?;
ENTONCES      ¿El organismo es de la Familia Goniadiidae?;
                SALIDA<<'El organismo es de la Familia Goniadiidae.';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\goniadiidae.bmp';
                TERMINAR;
SINO      ¿No se encontró una solución?;
                SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente el organismo.';
                TERMINAR;
SI_FIN
}

REGLA: 55
{
SI      ¿El organismo tiene prostomio entero, aguzado o redondeado y branquias cirriformes, pectinadas o ausentes?;
        (
                SALIDA<<'¿Tiene el organismo prostomio entero, aguzado o redondeado y branquias cirriformes, pectinadas o ausentes?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\branquia.bmp';
        )
Y      ¿El organismo podría tener prostomio entero, aguzado o redondeado y branquias cirriformes, pectinadas o ausentes?;
ENTONCES      ¿El organismo es de la Familia Opheliidae?;
                SALIDA<<'El organismo es de la Familia Opheliidae (Malmgren 1867).';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Opheliidae.bmp';
                TERMINAR;
SINO      ¿No se encontró una solución?;
                SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente el organismo.';
                TERMINAR;
SI_FIN
}

REGLA: 57
{
SI      ¿El organismo tiene segmentos anteriores con ganchos cubiertos y con 2 portadores maxilares y generalmente sin ojos?;
        (
                SALIDA<<'¿Tiene el organismo segmentos anteriores con ganchos cubiertos y con 2 portadores maxilares y generalmente
sin ojos?.';
                RESP: 'Si', 'No';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\57.bmp';
        )
Y      ¿El organismo podría tener segmentos anteriores con ganchos cubiertos y con 2 portadores maxilares y generalmente sin ojos?;
ENTONCES      ¿El organismo es de la Familia Lumbrineridae?;
                SALIDA<<'El organismo es de la Familia Lumbrineridae (Malmgren 1867).';
                IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Lumbrineridae.bmp';
                TERMINAR;
SINO      ¿No se encontró una solución?;
                SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente el organismo.';
                TERMINAR;
SI_FIN
}

REGLA: 58
{

```

```

SI      ¿El organismo tiene parapodios laterales en tórax, dorsales en abdomen?;
      (
          SALIDA<<'¿Tiene el organismo parapodios laterales en tórax, dorsales en abdomen?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Orbiniidae.bmp';
      )
Y      ¿El organismo podría tener parapodios laterales en tórax, dorsales en abdomen?;
ENTONCES ¿El organismo es de la Familia Orbiniidae?;
          SALIDA<<'El organismo es de la Familia Orbiniidae (Hartman 1942).';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Orbiniidae.bmp';
          TERMINAR;
SINO    ¿El organismo podría tener parapodios laterales en todo el cuerpo?;
SI_FIN }

REGLA: 59
{
SI      ¿El organismo tiene parapodios laterales en todo el cuerpo?;
      (
          SALIDA<<'¿Tiene el organismo parapodios laterales en todo el cuerpo?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Capitellidae.bmp';
      )
Y      ¿El organismo podría tener parapodios laterales en todo el cuerpo?;
ENTONCES ¿El organismo podría tener setas capilares restringidas al tórax, o hasta los primeros setíferos abdominales y cuerpo blando?;
SINO    ¿No se encontró una solución?;
          SALIDA<<'No se encontró una solución.';
          TERMINAR;
SI_FIN }

REGLA: 60
{
SI      ¿El organismo tiene setas capilares restringidas al tórax, o hasta los primeros setíferos abdominales y cuerpo blando?;
      (
          SALIDA<<'¿Tiene el organismo setas capilares restringidas al tórax, o hasta los primeros setíferos abdominales y cuerpo blando?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Capitellidae.bmp';
      )
Y      ¿El organismo podría tener setas capilares restringidas al tórax, o hasta los primeros setíferos abdominales y cuerpo blando?;
ENTONCES ¿El organismo es de la Familia Capitellidae?;
          SALIDA<<'El organismo es de la Familia Capitellidae (Grube 1862).';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\Capitellidae.bmp';
          TERMINAR;
SINO    ¿No se encontro una solución?;
          SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente al organismo.';
          TERMINAR;
SI_FIN }

REGLA: 61
{
SI      ¿El organismo tiene prostomio con una antena media?;
      (
          SALIDA<<'¿Tiene el organismo prostomio con una antena media?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\paraonidae.bmp';
      )
Y      ¿El organismo podría tener prostomio con una antena media?;
ENTONCES ¿El organismo podría tener branquias con 15 - 20 segmentos desde los setíferos 4 - 10?;
SINO    ¿El organismo podría tener prostomio aplanado y espatulado, tan ancho como la porción más ancha del cuerpo?;
          IR_A_REGLA: 63;
SI_FIN }

REGLA: 62
{
SI      ¿El organismo tiene branquias con 15 - 20 segmentos desde los setíferos 4 - 10?;
      (

```

```

        SALIDA<<'¿Tiene el organismo branquias con 15 - 20 segmentos desde los setíferos 4 - 10?.';
        RESP: 'Si', 'No';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\50.bmp';
    )
Y      ¿El organismo podría tener branquias con 15 - 20 segmentos desde los setíferos 4 - 10?;
ENTONCES ¿El organismo es de la Familia Paraonidae?;
        SALIDA<<'El organismo es de la Familia Paraonidae (Cerruti 1909).';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\paraonidae.bmp';
        TERMINAR;
SINO    ¿No se encontro una solución?;
        SALIDA<<'No se encontró una solución. Se sugiere revisar nuevamente al organismo.';
        TERMINAR;

SL_FIN
}

REGLA: 63
{
SI      ¿El organismo tiene prostomio aplanado y espatulado, tan ancho como la porción más ancha del cuerpo?;
    (
        SALIDA<<'¿Tiene el organismo prostomio aplanado y espatulado, tan ancho como la porción más ancha del cuerpo?.';
        RESP: 'Si', 'No';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\63.bmp';
    )
Y      ¿El organismo podría tener prostomio aplanado y espatulado, tan ancho como la porción más ancha del cuerpo?;
ENTONCES ¿El organismo es de la Familia Magelonidae?;
        SALIDA<<'El organismo es de la Familia Magelonidae (Cunningham y Ramage 1888).';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\000.bmp';
        TERMINAR;
SINO    ¿El organismo podría tener parapodios inconspicuos; segmentos abdominales prolongados y las setas forman cinturas
completas alrededor del cuerpo?;
SL_FIN
}

REGLA: 64
{
SI      ¿El organismo tiene parapodios inconspicuos; segmentos abdominales prolongados y las setas forman cinturas completas alrededor del
cuerpo?;
    (
        SALIDA<<'¿Tiene el organismo parapodios inconspicuos; segmentos abdominales prolongados y las setas forman cinturas
completas alrededor del cuerpo?.';
        RESP: 'Si', 'No';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\heterospionidae.bmp';
    )
Y      ¿El organismo podría tener parapodios inconspicuos; segmentos abdominales prolongados y las setas forman cinturas completas
alrededor del cuerpo?;
ENTONCES ¿El organismo es de la Familia Heterospionidae?;
        SALIDA<<'El organismo es de la Familia Heterospionidae (Hartman 1963).';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\heterospionidae.bmp';
        TERMINAR;
SINO    ¿El organismo es de la Familia Spionidae?;
        SALIDA<<'El organismo es de la Familia Spionidae (Grube 1850).';
        IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ANELIDOS\pisionidae.bmp';
        TERMINAR;

SL_FIN
}
}
FIN

```

H. CODIGO BASE DE CONOCIMIENTOS DE LOS ONUFIDOS

```
INICIO
{
SALIDA<<'Esquema de la familia Onuphidae.';

REGLA: 64
{
SI      ¿El organismo tiene parapodios prolongados?;
      (
          SALIDA<<'¿Tiene el organismo parapodios prolongados?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Parapodio prolongado.bmp';
      )
ENTONCES ¿El organismo podría tener ganchos anteriores pseudocompuestos bidentados?;
SINO     ¿El organismo podría tener bránquias en espiral?;
      IR_A_REGLA: 66;
SI_FIN
}

REGLA: 65
{
SI      ¿El organismo tiene ganchos anteriores pseudocompuestos bidentados?;
      (
          SALIDA<<'¿Tiene el organismo ganchos anteriores pseudocompuestos bidentados?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Gancho bidentado.bmp';
      )
Y      ¿El organismo podría tener ganchos anteriores pseudocompuestos bidentados?;
ENTONCES ¿El organismo es del genero Americonuphis reesei?;
      SALIDA<<' El organismo es del genero Americonuphis reesei.';
      IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Americonuphis reesei.bmp';
      TERMINAR;

SINO     ¿No se encontró una solución?;
      SALIDA<<'No se encontró una solución';
      IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\28.bmp';

TERMINAR;
SI_FIN
}

REGLA: 66
{
SI      ¿El organismo tiene bránquias en espiral?;
      (
          SALIDA<<'¿Tiene el organismo bránquias en espiral?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Branquia espiralada.bmp';
      )
Y      ¿El organismo podría tener bránquias en espiral?;
ENTONCES ¿El organismo podría tener ganchos pseudocompuestos total o parcialmente bidentados?;
SINO     ¿El organismo podría tener bránquias pectinadas o simples?;
      IR_A_REGLA: 76;
SI_FIN
}

REGLA: 67
{
SI      ¿El organismo tiene ganchos pseudocompuestos total o parcialmente bidentados?;
      (
          SALIDA<<'¿Tiene el organismo ganchos pseudocompuestos total o parcialmente bidentados?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Gancho bidentado.bmp';
      )
Y      ¿El organismo podría tener ganchos pseudocompuestos total o parcialmente bidentados?;
ENTONCES ¿El organismo podría tener ganchos pseudocompuestos claramente bidentados?;
SINO     ¿El organismo podría tener todos los ganchos anteriores tridentados, lóbulo presetal del primer setígero con el pliegue transverso?;
      IR_A_REGLA: 75;
}
```

```

SI_FIN
}

REGLA: 68
{
SI      ¿El organismo tiene ganchos pseudocompuestos claramente bidentados?;
      (
          SALIDA<<'¿Tiene el organismo ganchos pseudocompuestos claramente bidentados?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Gancho bidentado.bmp';
      )
Y      ¿El organismo podría tener ganchos pseudocompuestos claramente bidentados?;
ENTONCES ¿El organismo podría tener seta pectinada oblícua?;
SINO    ¿El organismo podría tener ganchos anteriores bi y tridentados?;
      IR_A_REGLA: 74;
SI_FIN
}

REGLA: 69
{
SI      ¿El organismo tiene seta pectinada oblícua?;
      (
          SALIDA<<'¿Tiene el organismo seta pectinada oblícua?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Oblicua.bmp';
      )
Y      ¿El organismo podría tener seta pectinada oblícua?;
ENTONCES ¿El organismo podría tener lóbulo presetal del primer setífero con pliegue transversal distalmente entero?;
SINO    ¿El organismo podría tener seta pectinada con dientes gruesos poco numerosos?;
      IR_A_REGLA: 72;
SI_FIN
}

REGLA: 70
{
SI      ¿El organismo tiene lóbulo presetal del primer setífero con pliegue transversal distalmente entero?;
      (
          SALIDA<<'¿Tiene el organismo lóbulo presetal del primer setífero con pliegue transversal distalmente entero?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Pliegue entero.bmp';
      )
Y      ¿El organismo podría tener lóbulo presetal del primer setífero con pliegue transversal distalmente entero?;
ENTONCES ¿El organismo es del género Diopatra farallonensis?;
      SALIDA<<' El organismo es del género Diopatra farallonensis.';
      IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Diopatra farallonensis.bmp';
      TERMINAR;
SINO    ¿El organismo podría tener lóbulo presetal del primer setífero distalmente bilobulado?;
SI_FIN
}

REGLA: 71
{
SI      ¿El organismo tiene lóbulo presetal del primer setífero distalmente bilobulado?;
      (
          SALIDA<<'¿Tiene el organismo lóbulo presetal del primer setífero distalmente bilobulado?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\1-2.bmp';
      )
Y      ¿El organismo podría tener lóbulo presetal del primer setífero distalmente bilobulado?;
ENTONCES ¿El organismo es del género Diopatra oblicua?;
      SALIDA<<' El organismo es del género Diopatra oblicua.';
      IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\aphroditidae1.bmp';
      TERMINAR;
SINO    ¿No se encontró una solución?;
      SALIDA<<'No se encontró una solución.';
      IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\28.bmp';
TERMINAR;
SI_FIN
}

```

```

REGLA: 72
{
SI      ¿El organismo tiene seta pectinada con dientes gruesos poco numerosos?;
      (
          SALIDA<<'¿Tiene el organismo seta pectinada con dientes gruesos poco numerosos?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Dientes gruesos.bmp';
      )
Y      ¿El organismo podría tener seta pectinada con dientes gruesos poco numerosos?;
ENTONCES ¿El organismo es del genero Diopatra splendidissima?;
          SALIDA<<' El organismo es del genero Diopatra splendidissima.';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\aphroditidae1.bmp';
          TERMINAR;
SINO    ¿El organismo podría tener seta pectinada con numerosos dientes finos?;
SL_FIN
}

REGLA: 73
{
SI      ¿El organismo tiene seta pectinada con numerosos dientes finos?;
      (
          SALIDA<<'¿Tiene el organismo seta pectinada con numerosos dientes finos?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\1-2.bmp';
      )
Y      ¿El organismo podría tener seta pectinada con numerosos dientes finos?;
ENTONCES ¿El organismo es del genero Diopatra ornata?;
          SALIDA<<' El organismo es del genero Diopatra ornata.';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Diopatra ornata.bmp';
          TERMINAR;
SINO    ¿No se encontró una solución?;
          SALIDA<<'No se encontró una solución.';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\28.bmp';
TERMINAR;
SL_FIN
}

REGLA: 74
{
SI      ¿El organismo tiene ganchos anteriores bi y tridentados?;
      (
          SALIDA<<'¿Tiene el organismo ganchos anteriores bi y tridentados?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\bi y tri dentado.bmp';
      )
Y      ¿El organismo podría tener ganchos anteriores bi y tridentados?;
ENTONCES ¿El organismo es del genero Diopatra neotridens?;
          SALIDA<<' El organismo es del genero Diopatra neotridens.';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\aphroditidae1.bmp';
          TERMINAR;
SINO    ¿No se encontró una solución?;
          SALIDA<<'No se encontró una solución.';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\28.bmp';
TERMINAR;
SL_FIN
}

REGLA: 75
{
SI      ¿El organismo tiene todos los ganchos anteriores tridentados, lóbulo presetal del primer setígero con el pliegue
transverso?;
      (
          SALIDA<<'¿Tiene el organismo todos los ganchos anteriores tridentados, lóbulo presetal del primer setígero con el pliegue
transverso?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Pliegue transverso.bmp';
      )
Y      ¿El organismo podría tener todos los ganchos anteriores tridentados, lóbulo presetal del primer setígero con el pliegue
transverso?;
ENTONCES ¿El organismo es del genero Diopatra tridentada?;
          SALIDA<<' El organismo es del genero Diopatra tridentada.';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\aphroditidae1.bmp';
}

```

```

TERMINAR;
SINO      ¿No se encontró una solución?;
          SALIDA<<'No se encontró una solución';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\28.bmp';

TERMINAR;
SI_FIN
}

REGLA: 76
{
SI      ¿El organismo tiene bránquias pectinadas o simples?;
      (
          SALIDA<<'¿Tiene el organismo bránquias pectinadas o simples?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Branquia pectinada.bmp';
      )
Y      ¿El organismo podría tener bránquias pectinadas o simples?;
ENTONCES ¿El organismo podría tener ganchos pseudocompuestos con cubiertas largas y truncadas?;
SINO      ¿No se encontró una solución?;
          SALIDA<<'No se encontró una solución';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\28.bmp';

TERMINAR;
SI_FIN
}

REGLA: 77
{
SI      ¿El organismo tiene ganchos pseudocompuestos con cubiertas largas y truncadas?;
      (
          SALIDA<<'¿Tiene el organismo ganchos pseudocompuestos con cubiertas largas y truncadas?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Cubierta larga.bmp';
      )
Y      ¿El organismo podría tener ganchos pseudocompuestos con cubiertas largas y truncadas?;
ENTONCES ¿El organismo es del genero Paradioapatra barrazai?;
          SALIDA<<' El organismo es del genero Paradioapatra barrazai.';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Paradiopatra barrazai.bmp';
          TERMINAR;
SINO      ¿El organismo podría tener ganchos pseudocompuestos con cubiertas cortas y truncadas con ceratóforo occipital más largo
que el largo del prostomio?;
SI_FIN
}

REGLA: 78
{
SI      ¿El organismo tiene ganchos pseudocompuestos con cubiertas cortas y truncadas con ceratóforo occipital más largo que el largo del
prostomio?;
      (
          SALIDA<<'¿Tiene el organismo ganchos pseudocompuestos con cubiertas cortas y truncadas con ceratóforo occipital más
largo que el largo del prostomio?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\Cubiertas cortas.bmp';
      )
Y      ¿El organismo podría tener ganchos pseudocompuestos con cubiertas cortas y truncadas con ceratóforo occipital más largo que el largo
del prostomio?;
ENTONCES ¿El organismo es del genero Onuphis brevicirris?;
          SALIDA<<' El organismo es del genero Onuphis brevicirris.';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\aphroditidae1.bmp';
          TERMINAR;
SINO      ¿El organismo podría tener ceratóforo occipital más corto que el prostomio?;
SI_FIN
}

REGLA: 79
{
SI      ¿El organismo tiene ceratóforo occipital más corto que el prostomio?;
      (
          SALIDA<<'¿Tiene el organismo ceratóforo occipital más corto que el prostomio?.';
          RESP: 'Si', 'No';
          IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\1-2.bmp';
      )

```

```

)
Y           ¿El organismo podría tener ceratóforo occipital más corto que el prostomio?;
ENTONCES   ¿El organismo podría tener ganchos tridentados simples presentes en algunos setíferos anteriores?;

SINO       ¿No se encontró una solución?;
           SALIDA<<'No se encontró una solución';
           IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\28.bmp';

TERMINAR;
SI_FIN
}

REGLA: 80
{
SI         ¿El organismo tiene ganchos tridentados simples presentes en algunos setíferos anteriores?;
           (
           SALIDA<<'¿Tiene el organismo ganchos tridentados simples presentes en algunos setíferos anteriores?.';
           RESP: 'Si', 'No';
           IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\1-2.bmp';
           )
Y           ¿El organismo podría tener ganchos tridentados simples presentes en algunos setíferos anteriores?;
ENTONCES   ¿El organismo podría tener algunos ganchos pseudocompuestos bidentados y cirro ventral cirriforme en los primeros dos
setíferos?;
SINO       ¿El organismo podría tener todos los ganchos pseudocompuestos tridentados y cirro ventral cirriforme en los primeros nueve setíferos?;
           IR_A_REGLA: 83;
SI_FIN
}

REGLA: 81
{
SI         ¿El organismo tiene algunos ganchos pseudocompuestos bidentados y cirro ventral cirriforme en los primeros dos setíferos?;
           (
           SALIDA<<'¿Tiene el organismo algunos ganchos pseudocompuestos bidentados y cirro ventral cirriforme en los primeros
dos setíferos?.';
           RESP: 'Si', 'No';
           IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\1-2.bmp';
           )
Y           ¿El organismo podría tener algunos ganchos pseudocompuestos bidentados y cirro ventral cirriforme en los primeros dos setíferos?;
ENTONCES   ¿El organismo es del genero Kinbergonuphis pigmentata?;
           SALIDA<<' El organismo es del genero Kinbergonuphis pigmentata.';
           IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\aphroditidae1.bmp';
           TERMINAR;
SINO       ¿El organismo podría tener ganchos tridentados simples ausentes y cirro ventral cirriforme presente en los primeros cuatro setíferos?;
SI_FIN
}

REGLA: 82
{
SI         ¿El organismo tiene ganchos tridentados simples ausentes y cirro ventral cirriforme presente en los primeros cuatro setíferos?;
           (
           SALIDA<<'¿Tiene el organismo ganchos tridentados simples ausentes y cirro ventral cirriforme presente en los primeros
cuatro setíferos?.';
           RESP: 'Si', 'No';
           IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\1-2.bmp';
           )
Y           ¿El organismo podría tener ganchos tridentados simples ausentes y cirro ventral cirriforme presente en los primeros cuatro setíferos?;
ENTONCES   ¿El organismo es del genero Kinbergonuphis cedroensis?;
           SALIDA<<' El organismo es del genero Kinbergonuphis cedroensis.';
           IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\aphroditidae1.bmp';
           TERMINAR;
SINO       ¿No se encontró una solución?;
           SALIDA<<'No se encontró una solución';
           IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\28.bmp';

TERMINAR;
SI_FIN
}

REGLA: 83
{
SI         ¿El organismo tiene todos los ganchos pseudocompuestos tridentados y cirro ventral cirriforme en los primeros nueve setíferos?;
           (

```

```

nueve setígeros?.';
    SALIDA<<'¿Tiene el organismo todos los ganchos pseudocompuestos tridentados y cirro ventral cirriforme en los primeros
    RESP: 'Si', 'No';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\1-2.bmp';
)
Y
ENTONCES ¿El organismo podría tener todos los ganchos pseudocompuestos tridentados y cirro ventral cirriforme en los primeros nueve setígeros?;
    ¿El organismo es del genero Kinbergonuphis vermillionensis?;
    SALIDA<<' El organismo es del genero Kinbergonuphis vermillionensis.';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\aphroditidae1.bmp';
    TERMINAR;
SINO ¿El organismo podría tener cirro ventral digitiforme en los primeros siete setígeros?;
SI_FIN
}

REGLA: 84
{
SI ¿El organismo tiene cirro ventral digitiforme en los primeros siete setígeros?;
(
    SALIDA<<'¿Tiene el organismo cirro ventral digitiforme en los primeros siete setígeros?.';
    RESP: 'Si', 'No';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\1-2.bmp';
)
Y
ENTONCES ¿El organismo podría tener cirro ventral digitiforme en los primeros siete setígeros?;
    ¿El organismo es del genero Kinbergonuphis kristiani?;
    SALIDA<<' El organismo es del genero Kinbergonuphis kristiani.';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\aphroditidae1.bmp';
    TERMINAR;
SINO ¿No se encontró una solución?;
    SALIDA<<'No se encontró una solución.';
    IMAGEN: 'C:\Nivel9_Samples\Imágenes\Especies\ONUPHIDAE\28.bmp';
TERMINAR;
SI_FIN
}
}
FIN

```

I. FUNDAMENTO TEORICO DE LOS ONUFIDOS

Los onúfidios se distinguen por presentar 2 antenas frontales y 5 antenas occipitales, aparatos maxilares cortos, notopodio conformado por la base de la branquia y el cirro dorsal, acícula interna, ganchos, setas limbadas, setas pectinadas y ganchos Subaciculares. En general, la identificación a nivel específica depende principalmente de la distribución de las branquias, diferentes tipos de setas y tamaño y forma de algunas estructuras⁴.

Es importante señalar que los onúfidios presentan formas de vida tubícula, construyen tubos conspicuos, fuertes y membranosos pero, son capaces de desplazarse durante períodos de estrés o peligro⁵. Asimismo, pueden aparecer en elevado número en zonas intermareales⁶. Lo anterior, reviste importancia ecológica en la dinámica marina, pues se ha demostrado que los poliquetos tubícolas reducen la habilidad de penetración de las larvas de otras especies a la superficie del sedimento⁷.

Rivera en el año 2000 desarrollo la clasificación de los especímenes para el reconocimiento de estos.

Considerando que estudios ecológicos firmes deben estar respaldados por la identificación positiva de las especies que componen los ambientes marinos, se ha diseñado esta clave ilustrada para que sea utilizada por investigadores interesados en estudiar como grupo objetivo el macrozoobentos, los anélidos poliquetos o los onúfidios de la costa de El Salvador.

⁴ Fauchald, 1968.

⁵ Uebelacker & Johnson, 1984; Fauchald, 1977.

⁶ Ruppert & Barnes, 1996.

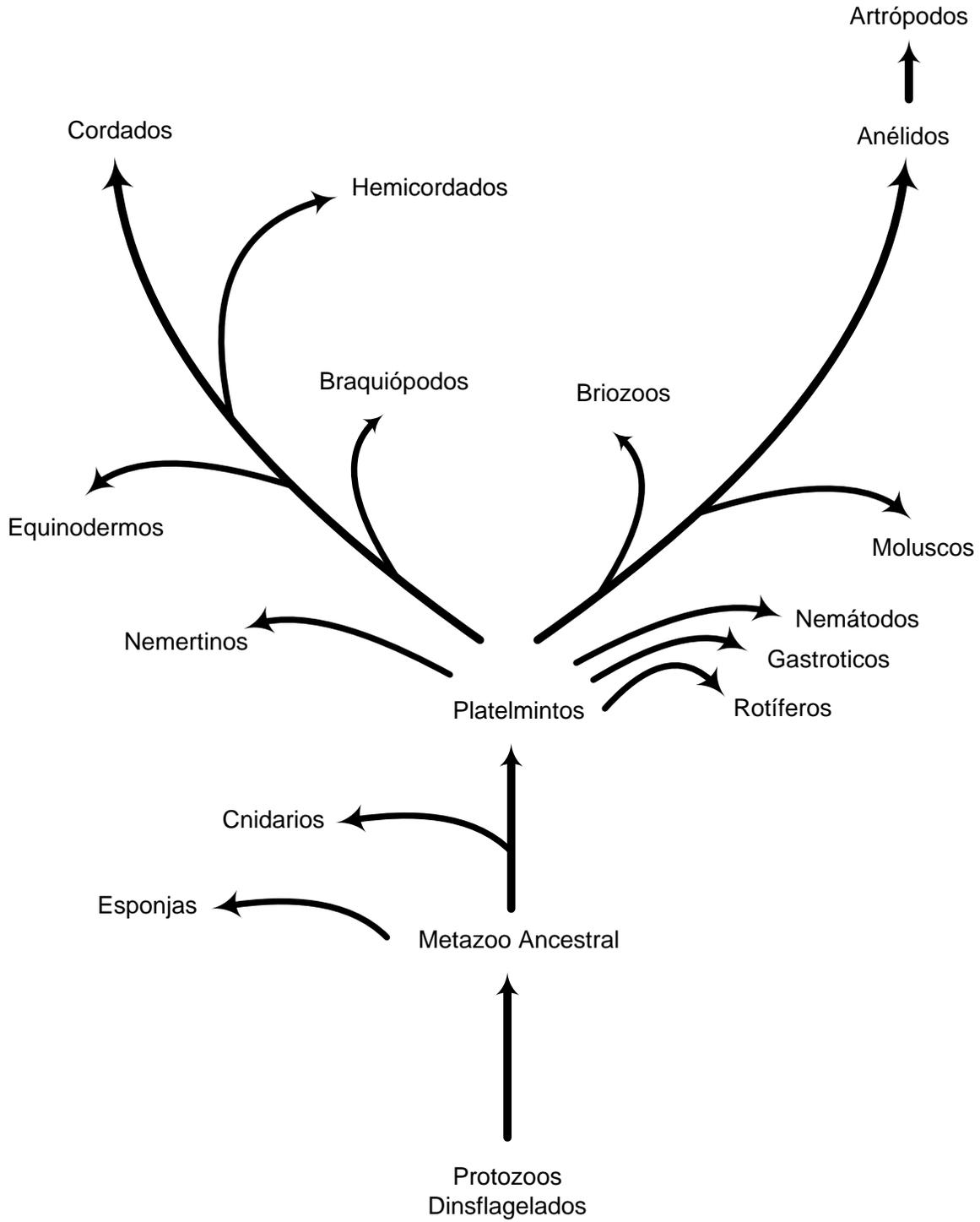
⁷ Woodin, 1974.

Cuadro1. Aspectos ecológicos básicos de los onúfidos de El Salvador.

Especie	Distribución mundial y Regional	Distribución local	Hábitat
<i>Americanuphis reesei</i> Fauchal, 1973	Océano Pacífico Oriental Tropical El Salvador al Golfo de Panamá	• Al Sur de la Bahía de Jiquilisco	Entre 20 y 50 m de profundidad en fondos lodosos y arenosos
<i>Diopatra farallonensis</i> Fauchald, 1968	Océano Pacífico Oriental Tropical Golfo de California a El Salvador	• Frente a Ahuachapán • Al Sur El Pimental	Entre 20 y 50 m de profundidad en fondos lodosos
<i>Diopatra neotridens</i> Hartman, 1944	Océano Pacífico Oriental Tropical, Baja California (México) a Panamá	• Frente a Ahuachapán • Al Sur del Icacal • Al Sur de la Bahía de Jiquilisco • Al Sur El Pimental • Al Sur de Barra Salada	Entre 20 y 125 m de profundidad en fondos lodosos, areno-lodosos con restos de bivalvos
<i>Diopatra obliqua</i> Hartman, 1944	Océano Pacífico Oriental Tropical, Oeste de México a Perú	• Frente a Ahuachapán • Al Sur El Pimental	Entre 20 y 30 m de profundidad en fondos lodosos
<i>Diopatra ornata</i> Moore, 1911	Océano Pacífico Oriental Tropical, Canadá a El Salvador	• Al Sur de Barra salada • Barra de Santiago • Bahía de Jiquilisco • Golfo de Fonseca • Estero de San Diego	Zonas estuarinas y áreas oceánicas a 33 m de profundidad en fondos lodosos y arenosos
<i>Diopatra splendidissima</i> Kinberg, 1857	Océano Pacífico Oriental Tropical, California a Ecuador	• Frente a Usulután • Al Sur Isla Meanguera • Al Sur de La Libertad	Entre 35 y 55 m de profundidad en fondos lodosos
<i>Diopatra tridentata</i> Hartman, 1944	Amfiamericano, en el Océano Pacífico de California a Colombia	• Frente a Usulután • Al Sur Isla Meanguera • Al Sur de La Libertad • Al Sur El Pimental • Al Sur Jucuarán • Al Sur de la Bahía de Jiquilisco • Al Sur El Icacal • Al Sur Barra Salada	Entre 15 y 50 m de profundidad en fondos lodosos, areno-lodosos con restos de bivalvos
<i>Kinbergonuphis cedroensis</i> Fauchald, 1982	Océano Pacífico Oriental Tropical, Sur de California a Ecuador	• Al Sur de los Cóbanos	A 20 m de profundidad en fondos areno-lodosos
<i>Kinbergonuphis</i>	Océano Pacífico	• Al Sur de La	Entre 20 y 55 m de

Espece	Distribución mundial y Regional	Distribución local	Hábitat
<i>kristiani</i> De León <i>et al</i> 2004	Oriental Tropical, Solamente conocida en El Salvador	Libertad <ul style="list-style-type: none"> • Al Sur El Icacal • Al Sur Los Cóbano • Al Sur Costa Azul • Frente a Usulután 	profundidad en fondos lodoso, areno-lodoso, areno-lodoso con restos de bivalvo
<i>Kinbergonuphis pigmentata</i> Fauchald, 1982	Océano Pacífico Oriental Tropical, Bahía San Quintín, Baja California (México) a El Salvador	<ul style="list-style-type: none"> • Al Sur El Pimental 	A 20 m de profundidad en fondos lodoso
<i>Kinbergonuphis vermillonensis</i> Fauchald, 1982	Océano Pacífico Oriental Tropical, Golfo de California a El Salvador	<ul style="list-style-type: none"> • Al Sur de los Cóbano 	A 20 m de profundidad en fondos areno-lodoso
<i>Onuphis brevicirris</i> Hartmann-Shröder, 1959	Océano Pacífico Oriental Tropical, El Salvador	<ul style="list-style-type: none"> • Estero de Jaltepeque 	En fondo lodoso y arenoso
<i>Paradiopatra barrazai</i> De León <i>et al</i> 2004	Océano Pacífico Oriental Tropical, solamente conocida en El Salvador	<ul style="list-style-type: none"> • Al Sur de la Bahía de Jiquilisco 	A 20 m de profundidad en fondo lodoso

MATRIZ FILOGENETICA



J. IMPORTANCIA DE LOS ANÉLIDOS POLIQUETOS

Abundancia. Los anélidos poliquetos presentan gran abundancia tanto en número de especies como de individuos, generalmente es uno de los grupos de mayor abundancia en fondos blandos y duros.

Modificación del Sustrato. Los poliquetos modifican el entorno que habitan por sus modos de vida y sus patrones de alimentación (Fauchald & Jumars, 1979). La modificación del sustrato duro es ocasionada por formación de tubos calcáreos que transforman la tersura superficial, aumentan la fricción relativa y pueden contribuir a la corrosión del sustrato metálico (Salazar *et al* 1989). Los más comunes son los perpulidos, son segundos en importancia a los balanos (ten Hove, 1979) con los que también se presentan en cascos de embarcaciones, portes de plataformas petroleras y otros sustratos donde reciben, junto con otros organismos, el calificativo de incrustantes, bio-incrustantes o foulers porque se relacionan, como se anotó, con la corrosión (Gunter y Geyer, 1955). También crecen masivamente en los túneles que conducen agua de mar para enfriamiento restringen el flujo del agua, deben removerse periódicamente.

Los poliquetos modifican los fondos blandos al formar galerías que cambian las condiciones de oxigenación, acarrear partículas hacia la superficie y viceversa Knox, 1977). También modifican la consistencia del sedimento por la formación de capsulas fecales de alta resistencia a la acción mecánica y bacteriológica (Cadée, 1979). En algunas playas mixtas, con rocas y arena, se presentan los poliquetos sabeláridos; dichos poliquetos forman tubos cementando granos de arena y son gregarios. El resultado son arrecifes de arena cementada que contribuyen a estabilizar la línea de costa al reducir la erosión, y dada la selectividad con la que actúan al construir sus tubos, modifican la composición granulométrica de las playas adyacentes (Kirtley y Tanner, 1968).

Una modificación especial es ocasionada por algunos nereidos cultivadores de algas (Woodin, 1977) ya que facilitan la colonización temporal y espacial del sustrato arenoso por las mismas, como resultado de la humedad retenida, reducen el impacto de la salinidad y desecación en su microambiente de la zona litoral.

Contaminación. Los poliquetos son el grupo más abundante en algunos medios contaminados incluso cuando se trata de descargas térmicas (Crema y Bonvicini-Pagliai, 1980). Cuando se discute sobre poliquetos-polucion, es obligado comentar sobre el famoso capitélido *Capitella capitata*. Esta especie ha sido considerada como indicadora de contaminación por materia orgánica porque alcanza grandes densidades en los sitios afectados (Reish, 1957; Bellan, 1964). También se le ha considerado como una especie colonizadora u oportunista por su rápido arribo a algunas zonas perturbadas o defaunadas (Grassle y Grassle, 1977). Sin embargo, esta especie euricora es en realidad un complejo de por lo menos seis especies sibilinas, muy difíciles de distinguir por su morfología (Grassle, 1980). Este panorama precautorio también se presenta en otra especie indicadora; el espiónido *Polydora ligni* presenta diferencias conspicuas entre distintas poblaciones porque también constituye un complejo de especies (Rice y Simon, 1980). Por ende, debemos ser cuidadosos al identificar nuestras muestras y aceptar que en caso de serio problema o de gran duda, designar al ejemplar problema como sp. A o sp. 1 y ser consistente de manera que no hagamos de cada gusano una línea más de nuestro reporte. Cuando nuestro ejemplar es muy cercanos a alguna especie conocida, se puede anteponer el críca (=casi) abreviado a ca. antes del nombre específico.

El nereido *Neanthes arenaceodentata* que habita el Pacífico Oriental templado y llega a las aguas mexicanas (y a veces se ha confundido con *N. caudata* o *N. succionea*) es tolerante a la contaminación por hidrocarburos y ha sido propuesto como modelo para la toxicología citogenética marina (Pesch y Pesch, 1980). En el glicérido *Glycera alba*, se han realizado algunos métodos bioquímicos para estudiar los efectos de la contaminación orgánica (Blackstock, 1980). Recientemente se realizó un ensayo para determinar métodos biológicos para el tratamiento de aguas de desecho. Kurihara

(1983) estudio y demostró la eficiencia y factibilidad para que se emplearan los nereidos sedimentívoros *Neanthes japonica* y *Perinereis nuntia vallata*. Encontró que digieren una gran parte de la materia orgánica presente en el agua del drenaje y son capaces de perpetuarse en base a esa alimentación.

Al margen de lo anterior, los poliquetos alcanzan considerable importancia para los estudios de ecología del bentos marino porque proporcionan la respuesta más rápida de la fauna a cualquier perturbación ambiental (Simon y Dauer, 1977), aunque sea moderada y de origen orgánico (Dauer y Conner, 1980).

Red Trópica y Parasitismo Otra consecuencia de la abundancia de estos anélidos es su importancia en las redes trópicas. Algunos peces demersales consumen abundantemente algunas especies del néftido *Nephtys* (Alheit, 1979). El nereído *Nereis virens* y el glicérido *Glycera dibranchiata* se explotan comercialmente como carnada para pesca deportiva en las costas del Atlántico de los Estados Unidos (Pettibone, 1963).

En ocasiones, como resultado de la transformación del cuerpo de algunos poliquetos para la reproducción (epitokia), y de su sincronización para salir en masa a la columna de agua (actividad en enjambre), se hacen tremendamente abundantes en algunos ambientes restringidos (atolones). Así, se convierten en un verdadero festín para peces, aves y humanos. El caso mejor documentado es el del eunícido *Eunice viridis*, el palolo de Samoa, que es capturado y degustado vivo o frito, en medio de celebraciones por los aborígenes samoyanos (Francé, 1967).

El parasitismo como forma de alimentación entre los poliquetos es más bien raro. De las más de 80 familias solo tres se consignan como totalmente parásitas que viven sobre su hospedero: Ichthyotomidae (peces), Spintheridae (esponjas) e Histriobdellidae (cangrejos dulceacuícolas y langostas). La primera parece restringirse al Mediterráneo, la segunda es latitudes frías, y la tercera es de amplia distribución pero limitada a latitudes templado-frías y las formas de agua dulce se han hallado en

Sudamérica, pero ninguna para México. Por lo menos una de las especies de esta última familia, *Histriobdella homari*, no es parásita sino una microfaga que controla las poblaciones de bacterias de la cámara branquial de las langostas *Homarus vulgaris* y *H. americanus* (Gelmer, 1980). El endoparasitismo se restringe a algunos miembros de la familia Arbellidae, la mayoría de los casos se restringe el parasitismo a las etapas juveniles del parásito.

Bioquímica. Algunos cirratúlidos (*Tharsyx marioni*) son notables por su capacidad de concentrar elementos tóxicos como el arsénico en sus palpos (Gibas et al., 1983); aunque se desconoce la función de dicha acumulación, quizás actué en los mecanismos de defensa del mismo modo que la concentración de vanadio en algunos tunicados (Stoecker, 1978). Otros poliquetos ganaron cierta fama por producir compuestos químicos de utilidad potencial como insecticidas, como agentes anti-cáncer, o como bactericidas.

El líquido celómico de ciertos glicéridos como *Glycera dibranchiata*, exhibe actividad antibacteriana porque contiene una proteína termolábil no relacionada a la limosina, que constituyó un nuevo tipo de proteína antimicrobiana (Anderson y Chain, 1982).

K. Clave ilustrada para la identificación de los onúfidos (Polychaeta: Onuphidae) de El Salvador.

Carlos Giovanni Rivera

Área Ecología del Bentos

Instituto de Ciencias del Mar y Limnología de El Salvador (ICMARES)

Facultad de Ciencias Naturales y Matemática

Universidad de El Salvador

icmares.benthosecology@gmail.com

Resumen

Se presenta una guía de identificación taxonómica de 13 especies de anélidos de la Familia Onuphidae (Kinberg, 1865), reportados para la costa y áreas oceánicas de El Salvador. La clave es presentada de forma dicotómica y reversible, con esquemas de las principales estructuras para la identificación definitiva de las especies. Las ilustraciones han sido tomadas o modificadas de las enlistadas en las referencias.

Los Onúfidos en El Salvador

La mayoría de las investigaciones sobre anélidos poliquetos han sido conducidas en estuarios, mientras que las zonas oceánicas se han estudiado en menor medida (Hartmann-Schröder, 1959; Molina, 1992; Calles *et al.* 1994; Barraza, 1994; Rivera & Ibarra, 1995; Canjura, 1996; Vasconcelos & Fuentes, 1997; Rivera & Romero, 2002).

Las investigaciones realizadas el siglo pasado reportaron únicamente los onúfidos *Diopatra ornata* y *Onuphis brevicirris* (Hartmann-Schröder, 1959; Molina, 1992; Calles *et al.* 1994; Barraza, 1994; Rivera & Ibarra, 1995; Canjura, 1996; Vasconcelos & Fuentes, 1997). En contraste, Rivera & Romero (2002) encontraron que los onúfidos

representaron la Familia más diversa y mejor distribuida en la costa salvadoreña hasta los 200 metros de profundidad, incrementando los registros a 13 especies (cuadro 1).

El género mejor representado es *Diopatra* con 6 especies, seguido de *Kinbergonuphis* con 4 especies, los géneros *Americanuphis*, *Onuphis* y *Paradiopatra* tienen un representante cada uno.

Es importante destacar el hallazgo de dos especies nuevas para la ciencia *Kinbergonuphis kristiani* y *Paradiopatra barrazai* en las costas salvadoreñas (De León *et al* 2004) y el substancial incremento de registros taxonómicos nuevos para El Salvador (Rivera & Romero, 2002).

Asimismo, considerando que estudios ecológicos firmes deben estar respaldados por la identificación positiva de las especies que componen los ambientes marinos, se ha diseñado esta clave ilustrada para que sea utilizada por investigadores interesados en estudiar como grupo objetivo el macrozoobentos, los anélidos poliquetos o los onúfidos de la costa de El Salvador.

Clave para las especies de Onúfidos de la Costa de El Salvador

(La clave para las especies es una modificación de la propuesta por Fauchald (1968))

- | | | |
|---|--|---|
| 1 | Setígeros anteriores con parapodios prolongados, ganchos anteriores pseudocompuestos bidentados..... | |
| | <i>Americanuphis reesei</i> Fauchald, 1973 (Figura 1 a) | |
| | - Parapodios anteriores no prolongados..... | 2 |
| 2 | (1) Branquias en espiral..... <i>Diopatra</i> | 3 |
| | - Branquias de otro tipo, pectinadas o simples..... | 8 |
| 3 | (2) Ganchos pseudocompuestos total o parcialmente bidentados..... | 4 |
| | - Todos los ganchos anteriores tridentados, lóbulo presetal del primer setígero con un pliegue transversal, el lóbulo distal en forma de collar..... <i>Diopatra tridentata</i> Hartman, 1944 (Figura 1 c) | |
| 4 | (3) Ganchos pseudocompuestos claramente | |

bidentados.....	5
- Ganchos anteriores bi y tridentados.....	
..... <i>Diopatra neotridens</i> Hartman, 1944 (Figura 1 b)	
5 (4) Seta pectinada oblicua.....	6
- Seta pectinada recta.....	7
6 (5) Lóbulo presetal del primer setífero con pliegue transverso, distalmente entero.....	
..... <i>Diopatra farallonensis</i> Fauchald, 1968 (Figura 1 d)	
- Lóbulo presetal del primer setífero distalmente bilobulado.....	
<i>Diopatra obliqua</i> Hartman, 1944 (Figura 1 e)	
7 (6) Seta pectinada con dientes gruesos poco numerosos.....	
..... <i>Diopatra splendidissima</i> Kinberg, 1857 (Figura 1 f)	
- Seta pectinada con numerosos dientes finos.....	
..... <i>Diopatra ornata</i> Moore, 1911 (Figura 1 g)	
8 (2) Ganchos pseudocompuestos con cubiertas largas y truncadas, branquias desde el setífero 6 con hasta 8 filamentos, primer setífero con doble lóbulo presetal, el proximal transverso, el distal corto, seta pectinada asimétrica con 12 dientes cortos y un diente lateral largo.....	
<i>Paradiopatra barrazai</i> De León <i>et al</i> 2004 (Figura 2 f).	
- Ganchos pseudocompuestos con cubiertas cortas y truncadas.....	9
9 (8) Ceratóforo occipital más largo que el largo del prostomio.....	
..... <i>Onuphis brevicirris</i> Hartmann-Shröder, 1959 (Figura 2 a)	
- Ceratóforo occipital más corto que el prostomio.....	10
<i>Kinbergonuphis</i>	
10 (9) Ganchos tridentados simples presentes en algunos setíferos anteriores.....	11
- Ganchos tridentados simples ausentes, cirro ventral cirriforme presente en los primeros cuatro setíferos.....	
<i>Kinbergonuphis cedroensis</i> Fauchald, 1982 (Figura 2 b)	
11 (10) Con algunos ganchos pseudocompuestos bidentados, cirro ventral cirriforme en los primeros dos setíferos...	
<i>Kinbergonuphis pigmentata</i> Fauchald, 1982 (Figura 2 d)	
- Todos los ganchos pseudocompuestos tridentados.....	12
12 (11) Cirro ventral cirriforme en los primeros nueve setíferos.....	
..... <i>Kinbergonuphis vermillonensis</i> Fauchald, 1982 (Figura 2 e)	
- Cirro ventral digitiforme en los primeros siete setíferos.....	
..... <i>Kinbergonuphis kristiani</i> De León <i>et al</i> 2004 (Figura 2 c)	

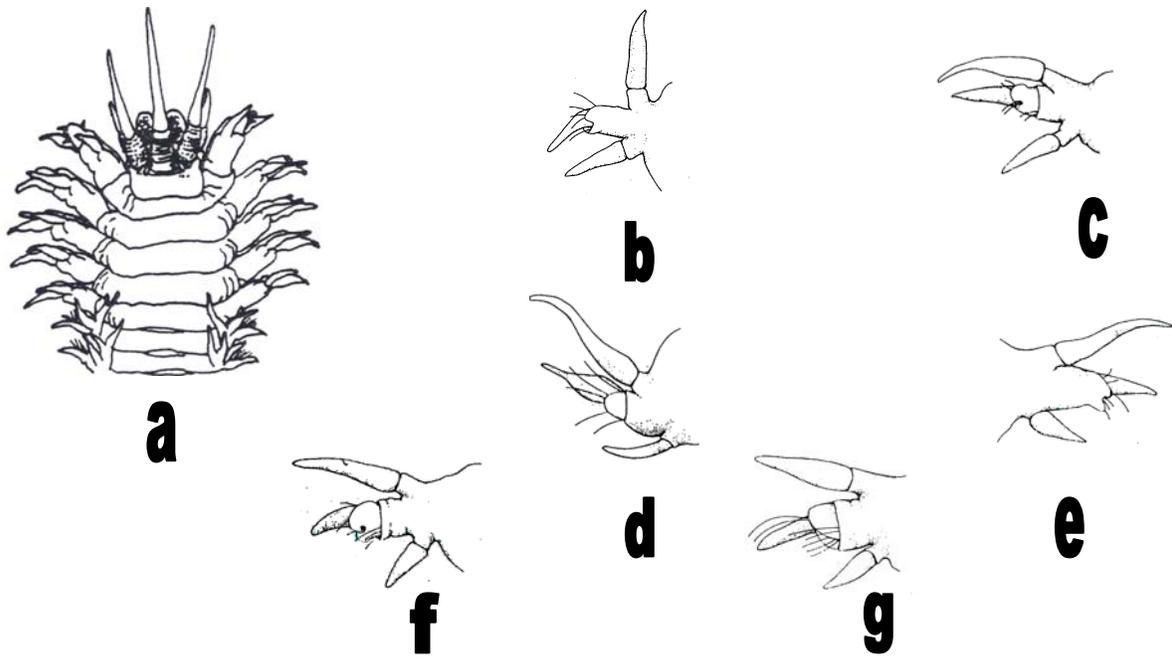


Figura 1. a Vista dorsal anterior *Americanuphis reesei* (Fauchald, 1973); b Vista anterior parapodio 1 *Diopatra neotridens* (Hartman, 1944); c Vista anterior parapodio 1 *Diopatra tridentata* (Hartman, 1944); d Vista anterior parapodio 1 *Diopatra farallonensis* (Fauchald, 1968); e Vista anterior parapodio 1 *Diopatra obliqua* (Hartman, 1944); f Vista anterior parapodio 1 *Diopatra splendidissima* (Fauchald, 1968); g Vista anterior parapodio 1 *Diopatra ornata* (Fauchald, 1968).

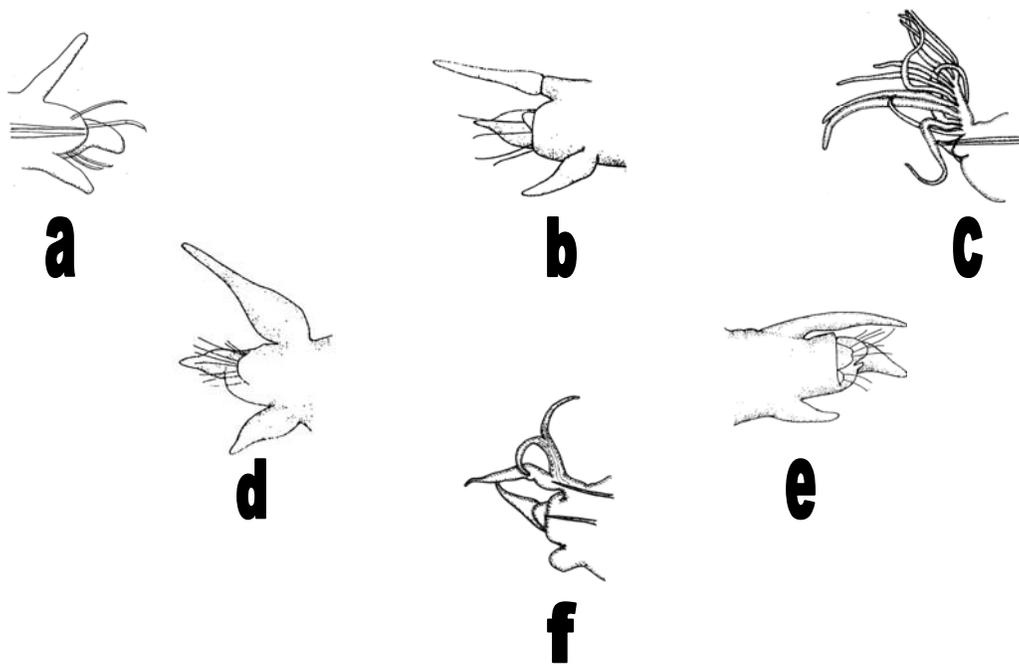


Figura 2. a Vista anterior parapodio 1 *Onuphis brevicirris* (Hartmann-Shröder, 1959); b Vista anterior parapodio 1 *Kinbergonuphis cedroensis* (Fauchald, 1982); c Vista anterior parapodio 25 *Kinbergonuphis*

kristiani (De León *et al* 2004); d Vista anterior parapodio 1 *Kinbergonuphis pigmentata* (Fauchald, 1982); e Vista anterior parapodio 1 *Kinbergonuphis vermillonensis* (Fauchald, 1982); f Vista anterior parapodio 7 *Paradiopatra barrazai* (De León *et al* 2004).

Conclusiones

La costa de El Salvador como parte del Océano Pacífico Oriental Tropical contiene una impresionante biodiversidad marina, principalmente de macroinvertebrados. En tal sentido, el incremento del esfuerzo de los investigadores sobre este grupo, tendrá como consecuencia registros taxonómicos nuevos para la costa y probablemente para la ciencia. Estudios detallados y monitoreo de las comunidades bénticas determinarán si la adición de registros obedece a cambios ambientales y sucesiones comunitarias o simplemente al resultado del aumento en el esfuerzo investigativo.