

45



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA EXPERTO DE DIAGNÓSTICO MÉDICO
VETERINARIO PARA AVES DE CORRAL

T E S I S

295848

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A :

JESÚS PINEDA GARCÍA

DIRECTOR DE TESIS: ING. JUAN MANUEL MARTÍNEZ VILLALOBOS



CIUDAD UNIVERSITARIA

SEPTIEMBRE 2001



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos:

Si bien es cierto; este trabajo de tesis es la formalización de la culminación de mi formación académica a nivel licenciatura, detrás de ella hubo y sigue habiendo todo un proceso de formación, en el que están participando diversos grupos sociales y personas.

Desde el principio y hasta siempre, gracias a mi Madre por darme la vida, por su guía, aliento y apoyo. Gracias por ella.

Por enseñarme "quebrados" y por impulsarme a superarme, por su constante presencia, por su apoyo y consejo, por su gran influencia, por ser mi hermana gracias a Georgina.

Gracias a todos los que fueron mis maestros y los que colaboraron de cualquier forma con mi formación académica y no académica.

Gracias a la Facultad de Ingeniería por permitirme ser parte de ella.

A la Universidad Nacional Autónoma de México; por permitirme acceder a ella, por creer en mi, por su pluralidad y por su grandeza.

A mi director de tesis, gracias por brindarme su experiencia y sus valiosos consejos.

A todos ustedes gracias.

Jesús Pineda García

Sistema Experto de Diagnóstico Médico Veterinario para aves de corral

Estructura temática

Introducción	i
1 Antecedentes de los Sistemas Expertos	1
1.1. Inteligencia artificial	1
1.1.1. Sistemas de Lenguaje Natural	2
1.1.2. Sistemas Reconocedores de Imágenes	2
1.1.3. Robótica	3
1.2. Sistemas Expertos	4
Campos de aplicación	4
Procedimiento de solución basado en conocimiento	4
Ventajas de los Sistemas Expertos	4
Desventajas de los Sistemas Expertos	6
1.2.1. Componentes de un Sistema Experto	6
1.2.2. Formas de Representación del Conocimiento	7
1.2.2.1. Reglas de producción	8
1.2.2.2. Redes semánticas	8
1.2.2.3. Frames	10
1.2.2.4. Cálculo de predicados	19
1.2.3. Estructuras de control de los Sistemas Expertos	20
1.2.4. Elección de una estrategia de control	22
2 Tipología y metodología de construcción de los Sistemas Expertos	25
2.1 Tipología	25
2.1.1 Sistemas Expertos de clasificación-interpretación	25
2.1.2 Sistemas Expertos de control (monitores)	25
2.1.2 Sistemas Expertos de previsión-concepción	26
2.2 Metodología	26
2.2.1 El modelo de ciclo de vida de un sistema convencional	26
2.2.2 Dificultades con el modelo de ciclo de vida tradicional	27
2.2.3 Ciclo de vida de desarrollo de un Sistema Experto	28
2.3 Proceso de desarrollo del Sistema Experto	28
2.3.1 Análisis del problema	29
2.3.2 Planificación del sistema	34
2.3.3 Desarrollo para alcanzar la madurez de la aplicación	39
2.3.4 Aplicación y mantenimiento	41
2.4 Herramientas de Software	42
3 Planteamiento del desarrollo del Sistema Experto	46
3.1 Análisis del problema	47
3.1.1 Forma de trabajo del experto	47
3.1.2 Tipo y alcance del conocimiento utilizado para la solución de problemas	47
3.1.3 Entorno de aplicación del Sistema Experto	48
3.1.4 Usuario del Sistema Experto	49
3.1.5 Costo y beneficio del Sistema Experto	50
3.1.5 Duración del análisis del problema	51
3.2 Planificación del sistema	52
3.2.1 Primer paso en la planificación del sistema: "Análisis de requerimientos"	52
3.2.2 Segundo paso en la planificación del sistema: "Proyecto"	58
3.2.3 Tercer paso en la planificación del sistema: "Realización parcial del proyecto"	62
3.2.4 Cuarto paso en la planificación del sistema: "Validación del proyecto"	69
3.2.5 Planificación del sistema "sin final"	70
3.3 Desarrollo para alcanzar la madurez de la aplicación	71
3.4 Aplicación y mantenimiento	73

Conclusiones	74
Apéndice A	75
Zonas del ave	75
Posibles síntomas por zona y su correspondiente enfermedad asociada	75
Apéndice B	77
NEWCASTLE	77
BRONQUITIS INFECCIOSA (BI)	77
LARINGOTRAQUEITIS	80
CORIZA INFECCIOSA	81
COLIBACILOSIS (INFECCIÓN POR E. COLI)	82
MYCOPLASMA GALLISEPTICUM (MG)	83
MYCOPLASMA SYNOVIAE (MS)	84
PASTEURELOSIS	86
ASPERGILOSIS PULMONAR	87
SIND CABEZA HINCHADA	88
INFLUENZA AVIAR (IA)	88
Apéndice C	90
Glosario	92
Bibliografía	95

Introducción

Los Sistemas Expertos son un tipo de software, que pretende emular el comportamiento de un experto en alguna área del conocimiento en la solución de problemas. No se trata de sustituir al experto, sino de apoyarlo en su trabajo, o en su caso poner al alcance de quien lo necesite esa capacidad para solucionar problemas.

Los Sistemas Expertos no están dotados de inteligencia, originalidad, sentido común, creatividad o conciencia, éstas son características exclusivas de los seres humanos, por tanto, no son otra cosa que un tipo de software que formaliza el conocimiento de un área específica y delimitada del saber humano, mediante técnicas de representación del conocimiento y a través de herramientas especializadas implementa dicha representación hasta obtener un producto, capaz de ser una extensión de un experto humano en la solución de problemas como una aplicación productiva.

Este es el objetivo del presente trabajo; conceptualizar en una forma de representación del conocimiento, el comportamiento de un experto veterinario en el diagnóstico de enfermedades de aves de corral, obteniendo un producto de software económico y portable, a través del cual esta capacidad de diagnosticar pueda ser utilizada por avicultores, o incluso por otros médicos veterinarios.

Este trabajo se divide en tres partes principales y una complementaria: en el capítulo uno **Antecedentes de los Sistemas Expertos**; se tocarán los puntos concernientes a la naturaleza de los Sistemas Expertos en su parte teórica ¿Qué son?, ¿Cuáles son sus partes y su interacción?, ventajas y desventajas, se hablará de las formas de representación a través de las cuales es posible formalizar el conocimiento humano (paso fundamental en el desarrollo de un Sistema Experto) y de las estructuras de control con las cuales se manipula el conocimiento una vez formalizado; sin estas estructuras no sería posible obtener provecho del conocimiento experto, por lo que son fundamentales en este tipo de desarrollos; también se presentará un breve esbozo de inteligencia artificial y sus campos de aplicación como referencia dado que los Sistemas Expertos están englobados en esta área.

En el capítulo dos **Tipología y Metodología de Construcción de los Sistemas Expertos**; se mencionarán sus diferentes tipos y su aplicación, se planteará la metodología utilizada en el desarrollo del Sistema Experto objeto de esta tesis, y se plantearán algunos elementos que nos auxiliarán en la elección de la herramienta de desarrollo del sistema llegado su momento.

En el capítulo tres **Planteamiento del Desarrollo del Sistema Experto**; después de haber establecido su fundamento teórico, se retomará la metodología de desarrollo elegida paso a paso, hasta completar el desarrollo de nuestro sistema, éste es el capítulo principal del presente trabajo. En él validaremos la utilidad de

esta técnica hasta obtener un producto de software capaz de poner al servicio de quien lo requiera, el conocimiento de un experto veterinario en el diagnóstico de aves de corral, específicamente de pollos.

Al final del trabajo aparecen tres apéndices: en **A** y **B**; se presenta información específica del dominio de la problemática objeto de estudio en este trabajo, dicha información es la parte medular del diseño de nuestro sistema ; por razones de agilidad en la lectura se presenta como apéndice.

Finalmente tenemos el apéndice **C** en el que se plantea a manera de ejemplo el análisis costo/beneficio del sistema desarrollado.

1 Antecedentes de los Sistemas Expertos

En los años sesentas se realizaban proyectos de inteligencia artificial que frecuentemente pretendían abarcar áreas muy extensas del conocimiento; con objetivos muy ambiciosos y dada la problemática que esto representaba, se optó por delimitar los dominios, surgiendo así el concepto de Sistemas Expertos.

A continuación y dado que ya se tocó el tema, hablaremos un poco a manera de preámbulo de la inteligencia artificial y de sus subcampos.

1.1. Inteligencia artificial

La inteligencia artificial es un término que pretende definir las técnicas de la lógica formal de los nuevos procedimientos y métodos de búsqueda de la representación del conocimiento en programas de ordenador. Por otro lado se describe como inteligente la presencia aparente de experiencia y conocimiento de causa en programas de ordenador; todos estos programas son resultado del estudio del comportamiento humano en la solución de problemas y de la investigación en el campo de la informática.

A continuación citaremos otras definiciones de inteligencia artificial:

Inteligencia artificial. Es la solución de problemas con el apoyo del computador, mediante la aplicación de procesos que son análogos al de razonamiento humano.

Los trabajos que buscan diseñar sistemas de cómputo con las características que se asocian con la inteligencia humana, se agrupan bajo el título de inteligencia artificial.

El objetivo de ésta es el desarrollo de paradigmas o algoritmos, que permitan a las máquinas realizar tareas que aparentemente requieren conocimiento o experiencia, cuando son realizadas por humanos. Bajo este concepto un sistema que se considere de inteligencia artificial debe ser capaz de hacer tres cosas:

- Almacenar conocimientos
- Aplicarlos en la solución de problemas
- Adquirir conocimientos a través de la experiencia

A este nivel dejaremos nuestro esbozo de inteligencia artificial, pasando ahora a hablar un poco de los campos de la misma; **Sistemas de Lenguaje Natural**, **Sistemas Reconocedores de Imágenes**, **Robótica** y por supuesto del tema que nos ocupa, **Sistemas Expertos**.

1.1.1. Sistemas de Lenguaje Natural

La tarea de la investigación de inteligencia artificial orientada al lenguaje, estriba en disponer en la máquina, de capacidades ligadas a la comprensión inteligente del lenguaje y a facilitar la comunicación entre ésta y el hombre.

Un sistema de procesamiento de lenguaje natural es aquél en que:

Parte de la información a procesar está codificada en dicho lenguaje.

Se aplican algoritmos para el análisis sintáctico, semántico y pragmático de la información y para su generación en lenguaje natural.

Los sistemas hasta ahora más desarrollados, son de diálogo, por ejemplo sistemas informativos; otros campos de investigación son, por ejemplo, la comprensión de textos, es decir que el sistema responda preguntas sobre un texto introducido anteriormente; otras áreas son: traducción automática y generación de extractos (resumen del texto).

1.1.2. Sistemas Reconocedores de Imágenes

En los sistemas reconocedores de imágenes debe identificarse el significado de imágenes con ayuda de procesos exactamente definidos. Dado que todavía no existen sistemas reconocedores de imágenes de aplicación general, los procesos desarrollados hasta la fecha sólo permiten aplicaciones especiales avanzadas (por ejemplo, robótica, automatismo industrial o análisis de fotografías aéreas).

Los subcampos generales del reconocimiento de imágenes son:

- División en procesos parciales apropiados.
- Determinación de las formas de representación para resultados intermedios y finales.
- Identificación y utilización de regularidades físicas.
- Identificación, representación y utilización de los conocimientos derivados de la experiencia.

Para ello deben resolverse muchos problemas distintos. Se buscan procesos capaces de distinguir cantos de iluminación, cantos de orientación y cantos de reflexión; es decir sombras, contornos y límites de color; también se busca desarrollar un hardware especial, con el que se puedan analizar posibles soluciones con gran cantidad de cálculo, por ejemplo, en la interpretación secuencial de imágenes.

Los dominios que lindan con el reconocimiento de imágenes, son el reconocimiento de patrones y tratamiento de imágenes.

El objetivo del reconocimiento de patrones es principalmente, reconocer las diferentes versiones de un mismo objeto, por ejemplo en la lectura de números o letras.

El tratamiento de imágenes abarca todos los procesos de cálculo para la manipulación de datos aportados por los mismos, en especial sólo de aquellos que tratan una imagen para obtener otra, por ejemplo: filtrado, alisado, aumento de contraste, corrección geométrica y eliminación de errores.

1.1.3. Robótica

Más allá de la fantasía que para la mayoría representan los robots, éstos desempeñan actividades monótonas y repetitivas en las industrias modernas, según un plan de acción previamente fijado con exactitud.

Con las técnicas de inteligencia artificial se intenta que el comportamiento de los robots sea más "inteligente". Por ejemplo tenemos la reacción ante fallos. Durante la ejecución del plan puede surgir un resultado imprevisto. Un objetivo de la investigación es modificar el plan en tiempo real; es decir durante el trabajo, e incorporar una reacción apropiada según el resultado.

Esta estrategia de crear un plan con sentido a partir de capacidades elementales puede utilizarse también para crear nuevos planes y no sólo para modificar los antiguos.

Además hay otras capacidades de necesaria aplicación si un robot debe realizar actividades no monótonas. Los robots especialmente los móviles necesitan una amplia capacidad sensora y visual.

Para poder convertir éstas y otras capacidades más avanzadas en realidad, se están empleando cada vez más, técnicas de inteligencia artificial en la programación de robots, entre ellas:

- Técnicas de solución espacial de problemas; espacios tridimensionales y geometría de objetos.
- Tiempo real; optimización de procesos de movimiento y coordinación.
- Tratamiento de señales sensoriales; elección de la información relevante de diferentes señales obtenidas por un sensor (datos visuales, electrónicos, etc.).
- Sistemas de planificación; que coordinan informaciones especiales, temporales y de sensores, para solucionar la tarea de forma óptima, o para coordinar varios robots en una producción en serie.

1.2. Sistemas Expertos

Bajo el término de Sistema Experto se entiende un nuevo tipo de software que imita el comportamiento de un experto humano en la solución de un problema. Pueden almacenar conocimientos de expertos para un campo determinado – y muy delimitado – y solucionar un problema mediante deducción lógica de conclusiones.

Campos de aplicación

Los campos de aplicación de los Sistemas Expertos son muy diversos, la utilización de este tipo de técnica se considera adecuada en problemas en los que la explosión combinatoria o el comportamiento dinámico impiden que una solución convencional mediante procesamiento de datos sea posible o rentable. Asimismo la complejidad del conocimiento en áreas estrechamente delimitadas, la carencia de algoritmos elaborados y de teorías completas son elementos que nos hacen optar por esta técnica.

Son en general problemas que implican soluciones en que interviene el conocimiento.

Procedimiento de solución basado en conocimiento

Este procedimiento comprende las siguientes capacidades:

- Utilización de normas u otras estructuras que contengan conocimientos y experiencia de expertos especializados.
- Deducción de lógica de conclusiones.
- Interpretación de datos ambiguos.
- Manipulación de conocimientos vagos, es decir, conocimientos afectados por valores de probabilidad.

Ventajas de los Sistemas Expertos

Es muy difícil cuantificar las ventajas de una nueva técnica, mas aún cuando se trata de una técnica mistificada como es el caso, lo que no hay que perder de vista es que se trata de programas de ordenador, los cuales nada tienen de humano ni de místico, lo único que hacen es almacenar conocimiento en forma de hechos, objetos, reglas y buscar respuestas dentro de éste cúmulo de conocimientos, de forma casi inteligente. En este sentido es en la mayor capacidad de memoria, velocidad de búsqueda y procesamiento en lo que superan al cerebro humano. A pesar de lo cual los Sistemas Expertos quedan supeditados al ser humano, ya que ni la creatividad, ni la originalidad, ni la conciencia de responsabilidad son implementables como cualidades del sistema.

Los Sistemas Expertos son sistemas auxiliares que pueden ofrecer una ayuda muy efectiva en los puestos de trabajo existentes, las rutinas se realizan con menos fallos, grandes masas de datos se analizan más rápidamente y las conclusiones complejas se captan con mayor claridad. Los Sistemas Expertos no son una amenaza, sólo por parecer técnicamente refinados y, por lo tanto, casi inteligentes. Por otro lado, la sustitución de profesiones por máquinas que requieren intuición o tacto, o alta cualificación resulta inimaginable, sin embargo tales profesiones si pueden verse apoyadas por buenos sistemas auxiliares, siendo este el principal objetivo de los Sistemas Expertos.

Con lo cual podemos concluir que tanto más responsabilidad, competencia e intuición requiera un puesto de trabajo, tanto más complejos deben ser los medios auxiliares para apoyar la labor, lo que nos hace pensar que el futuro de los Sistemas Expertos es bastante prometedor.

Veamos algunas ventajas:

Permanencia: Los Sistemas Expertos no olvidan, pero los humanos pueden hacerlo.

Reproductibilidad: Se pueden hacer muchas copias de un Sistema Experto, en cambio entrenar expertos humanos, consume mucho tiempo y tiene un alto costo.

Si bien los Sistemas Expertos son caros de construir y mantener, su operación no.

Los costos de desarrollo y mantenimiento pueden ser distribuidos entre muchos usuarios.

Los costos globales pueden ser considerados razonables; comparados con los caros y escasos que son los expertos humanos.

Consistencia: Con los Sistemas Expertos las transacciones similares se manejan del mismo modo; en cambio los expertos humanos pueden ser influenciados por eventos recientes o circunstancias de tipo personal.

Documentación: Un Sistema Experto puede suministrar documentación permanente de los procesos de decisión.

Totalidad: Un Sistema Experto puede revisar todo el universo, en cambio un experto humano únicamente puede revisar una muestra.

Oportunidad: Los fraudes o errores pueden prevenirse, dado que la información está disponible más rápido para la toma de decisiones.

Amplitud: El conocimiento de múltiples expertos humanos puede ser combinado para tener un sistema con más conocimiento del que una persona puede poseer.

Desventajas de los Sistemas Expertos

Sentido común: Aunado a un gran manejo de conocimiento técnico, los expertos humanos tienen sentido común. Y todavía no se sabe como dotar de éste a dichos sistemas.

Creatividad: Un individuo puede responder creativamente a una situación inusual; los Sistemas Expertos no.

Aprendizaje: Los expertos humanos se adaptan automáticamente a los cambios de ambiente; en cambio los Sistemas Expertos deben ser explícitamente actualizados.

Experiencia sensorial: El hombre tiene disponible un amplio rango de experiencia sensorial; los Sistemas Expertos son dependientes de entradas simbólicas.

Degradación: Los Sistemas Expertos no son buenos para reconocer cuando no existe respuesta; o bien, cuando el problema está fuera de su área de competencia.

1.2.1. Componentes de un Sistema Experto

Son los siguientes: base de conocimiento, mecanismo de inferencia, componente explicativo, interfaz del usuario y componente de adquisición.

- **Base de conocimiento:** La base de conocimiento de un Sistema Experto contiene el conocimiento de los hechos y de las experiencias de los expertos de un dominio determinado.

La base de conocimientos contiene todos los hechos, reglas y procedimientos del dominio de aplicación, que son importantes para la solución del problema. No solamente se almacenan datos individuales, sino que contiene también objetos complejos, sus cualidades, relaciones entre objetos y reglas para el procesamiento del conocimiento a partir del ya existente.

- **Mecanismo de inferencia:** El mecanismo de inferencia de un Sistema Experto, puede simular la estrategia de solución de un experto.

El mecanismo de inferencia es la unidad lógica con que se extraen conclusiones de la base de conocimientos, según un método fijo de solución de problemas, imitando el procedimiento de los expertos humanos para solucionar problemas.

- **Componente explicativo:** Comunica al usuario la estrategia de solución encontrada y el porqué de las decisiones tomadas.

- **Interfaz del usuario:** Sirve para que éste pueda realizar una consulta, en un lenguaje lo más natural posible y establece la forma en que el Sistema Experto se presentará al usuario.
- **Componente de Adquisición:** Ofrece ayuda a la estructuración e implementación del conocimiento, en la base de conocimiento.

1.2.2. Formas de Representación del Conocimiento

Es esencial la formalización y estructuración de éste si queremos procesarlo y manipularlo con técnicas de Sistemas Expertos.

Generalmente el conocimiento del experto está disponible mediante entrevistas en forma de descripciones de casos, o por su actividad.

Los métodos formales de representación del conocimiento son distintos aspectos de la lógica; por ejemplo, lógica de predicados, modal, multivaluada y difusa.

Tales métodos formales y matemáticos, no son auxiliares apropiados para comunicarse con expertos de los más diversos sectores especializados; tampoco estos métodos nos ofrecen formas generales de representación del saber.

Por ello se han desarrollado procedimientos de representación del conocimiento, que pueden ofrecer un apoyo eficiente a la estructuración y al procesamiento del saber. Tenemos para tal efecto:

Reglas de producción: Estas se basan en lógica de predicados, una descripción del saber en la forma "si .. , entonces...".

Redes semánticas: Una representación gráfica del saber sobre objetos y sus relaciones.

Frames: Estructuras de datos para la representación de objetos.

La base de estos procedimientos, sea en la representación o en el procesamiento del conocimiento, es en cierta manera el cálculo de predicados.

Cálculo de predicados: Deducción lógica de resultados; mediante el cumplimiento de determinadas condiciones puede extraerse una deducción lógica; la solución puede tener el valor de "verdadero" o "falso".

A continuación describiremos con mas detalle tales procedimientos:

- Reglas de producción.
- Redes semánticas.

- Frames.
- Cálculo de predicados.

1.2.2.1. Reglas de producción

La forma más comprensible de representación del conocimiento, se basa en las 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 por la cual en la actualidad, la mayoría de los Sistemas Expertos y los de mayor éxito, se basan en reglas de producción.

En la realización de Sistemas Expertos con reglas de producción se ha hecho patente la necesidad de procesar conocimientos vagos. Para la valoración de los resultados se utilizan ahora factores de certeza, que son factores arbitrarios de valoración que suelen encontrarse casi siempre dentro de los límites de -1 a $+1$. Donde -1 podría ser "seguro que no"; -0.5 "probablemente no"; 0 "desconocido"; $+0.5$ "probablemente si" y el $+1$ "seguro que si". El ámbito de valores dentro del espacio de solución elegido es continuo.

El componente más importante de un sistema de producción es el mecanismo de inferencia, con él se gobierna el procesamiento y la elección de reglas de producción; un buen mecanismo de inferencia destacará por sus eficientes métodos y estrategias de solución de conflictos y para la elección de una regla a partir de una serie.

1.2.2.2. Redes semánticas

Las redes semánticas son un método de representación del conocimiento sobre las relaciones de los objetos, los nodos de tal red corresponden a los objetos y los arcos describen relaciones entre los objetos; así puede tomarse un arco con sus dos nodos como una sola unidad de conocimiento.

Pero en la red semántica no se da información sobre el procesamiento de la red, las reglas de inferencia deben estar expresadas de forma explícita.

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

En la red semántica pueden existir relaciones unidireccionales (dependencia de una relación entre objetos) y relaciones bidireccionales (relaciones entre objetos), las relaciones unidireccionales se representan con una flecha en la dirección del objeto.

Ejemplo: Dadas las relaciones siguientes:

Temperatura(sangre caliente, mamífero). La relación es temperatura; que describe la corporal y se da como hecho que los mamíferos son de sangre caliente.

Es_un(perro, mamífero). La relación Es_un; describe que el perro pertenece a los mamíferos, además de que es una relación unidireccional dado, que cada perro es un mamífero pero no cada mamífero es un perro.

Es_un(Lucas, perro). Esta relación describe que existe un Lucas que es un perro, en este caso tenemos una relación unidireccional, dado que no todos los perros se llaman Lucas; ni todo aquel que se llame Lucas es un perro.

Con estas relaciones podemos establecer una red semántica como la que se muestra en la figura 1.2.

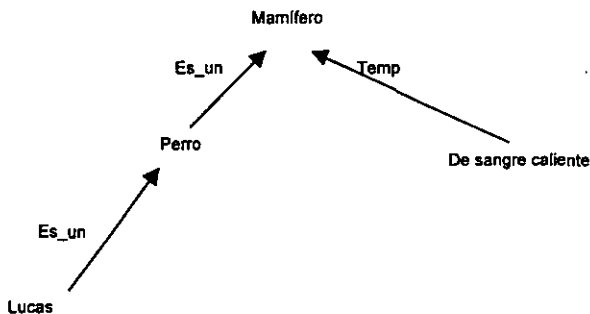


Figura 1.2

Además, como se mencionó, las reglas que posibilitan deducciones adicionales, se formulan fuera de la red.

Ejemplo: $Es_un(X,Y) \text{ AND } Es_un(Y,Z) \Rightarrow Es_un(X,Z)$

Esta regla nos indica; que si se cumple que X es Y y Y es Z entonces X será Z.

Para los datos del ejemplo: si Lucas es perro y perro es mamífero, entonces Lucas es mamífero.

1.2.2.3. Frames

Un frame es una estructura de datos que sirve para representar una situación estereotipada, una de las ideas intuitivas detrás de este tipo de representación, es que la memoria se basa mucho en estereotipos (propiedades típicas de los objetos). Añadido a cada frame hay varios tipos de información, parte de ésta hace referencia a cómo utilizar el frame; otra se refiere a lo que se espera que pueda suceder lugar y otra indica qué hacer si tales esperanzas no son confirmadas.

Un frame es por lo tanto, la división de objetos, o también de situaciones, en sus componentes, éstos son introducidos en los slots (ranuras) correspondientes del frame; los slots pueden estar a su vez subdivididos en facets.

Los sistemas de frames razonan acerca de clases de objetos usando representaciones prototípicas, pero que pueden modificarse para capturar las complejidades del mundo real. Se pueden capturar expectativas intentando tener una sola estructura de datos para poner el conocimiento relevante acerca de una clase de objetos, en lugar de tener el conocimiento distribuido en forma de reglas o fórmulas lógicas, lo cual nos permite construir conocimiento declarativo y procedural en un registro con slots y fillers o facets.

Los slots son atributos que tienen un valor y los fillers o facets son procedimientos que nos permiten manipular dichos valores tanto del frame al que pertenecen como de otros, en la figura 1.2.2.3.a se ejemplifica esta estructura.

Conferencia		
Fecha	30 de Agosto de 2001	Procedure1
Lugar	México, D. F.	Procedure2
Tema	Representación del conocimiento	Procedure3
Ponentes	M. I. Luis Calderón Acosta	Procedure4
Slot	Slot value	Facet

Figura 1.2.2.3.a

Antes de su utilización, el frame es una armazón preestructurada de datos. La configuración del frame y las definiciones de los slots están ya fijadas; a lo largo del procesamiento se van rellenando los slots con contenidos; en este proceso puede haber varios frames con la misma estructura pero diferente contenido.

Los valores de los slots (slot-value) son heredables; de esta forma no hace falta modificar más que el valor jerárquicamente superior en el slot y todas las instancias subordinadas del frame obtienen el nuevo valor.

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

Si se accede a un slot todavía vacío debe conseguirse el valor (if-needed procedure). En el procedimiento activado, el valor se calcula o se extrae de la base de conocimientos, o se solicita al usuario mediante un diálogo (user query) y se introduce en el slot.

El almacenamiento de un valor en el frame se realiza mediante un procedimiento llamado "if-added procedure" y el borrado, mediante un "if-removed procedure", estos serían los facets de sus correspondiente frames. Ambos procedimientos pueden influir sobre otros frames. La idea es que los cálculos para resolver problemas ocurren de efectos secundarios del flujo de información por el frame.

Los frames están puestos en una jerarquía en donde los frames de "abajo" pueden heredar los valores de los slots de los frames de "arriba". Normalmente la herencia se hace por medio de los arcos: is-a (al final instance-of), cuando un frame es atributo o slot de otro frame.

En general los frames de "arriba" tienen información típica (poco variable) mientras que los de "abajo" tienen información más específica. En ausencia de ésta, se utiliza la de los padres, lo que nos permite hacer deducciones a través de la jerarquía (se distinguen entre los frames clases o genéricos y los frames instancias).

Los slots pueden tener valores múltiples; si no encuentra un valor en el frame, busca por valores en los frames de arriba de la jerarquía (explora la relación transitiva de is-a), veamos esto a través del siguiente ejemplo; sea el frame genérico mostrado en la figura 1.2.2.3.b.

Conducto – Sanguíneo

Forma	Tubular	
Contiene	Sangre	
Slot	Slot value	Facet

Figura 1.2.2.3.b

Tenemos los siguiente frames instancias de las figuras 1.2.2.3.c, 1.2.2.3.d, 1.2.2.3.e y 1.2.2.3.f.

Arteria

Es-un	Conducto-sanguíneo	
Localización	{Brazo, Cabeza, Pierna, Tronco}	
Sangre	Rica-en-oxígeno	
Pared	Muscular	
Slot	Slot value	Facet

Figura 1.2.2.3.c

Vena		
Es-un	Conducto-sanguineo	
Pared	Muscular	
Slot	Slot value	Facet

Figura 1.2.2.3.d

Aorta		
Es-un	Arteria	
Localización	Tronco	
Diametro	Rica-en-oxígeno	
Slot	Slot value	Facet

Figura 1.2.2.3.e

Arteria-izquierda-X		
Es-un	Arteria	
Localización	Brazo	
Sangre	Rica-en-oxígeno	
Diámetro	Muscular	
Slot	Slot value	Facet

Figura 1.2.2.3.f

El permitir que un slot esté presente en más de un frame nos permite manejar excepciones (como sangre en el ejemplo anterior), de esta forma todos los frames instancias podrán heredar valores del frame genérico.

Defaults y Demons (facets/fillers)

Se puede tener información adicional, como: procedimientos para calcular el valor de un slot cuando no se tiene, procedimientos para actualizar valores de un slot, cuando un valor de otro slot es actualizado, restricciones en los valores que puede tener un slot, etc.

Datos, definiciones y procedimientos están agrupados en módulos que pueden compartir información y procedimientos por medio de mecanismos de herencia.

Los fillers o facets pueden tener varias formas de calcular un valor: value, default y demons.

Pegados a los slots pueden existir procedimientos que se activan cuando el slot es accesado o actualizado.

VALUE: (color (valor rojo))

DEFAULT: si no tiene un valor, toma el de default.

- Veamos esto a través de un ejemplo utilizando el frame de la figura 1.2.2.3.g:

Coche		
Color	Rojo	Procedure1
Llantas	¿?	Default 4
Slot	Slot value	Facet

Figura 1.2.2.3.g

Demons y métodos

IF-NEEDED: si no tiene un valor y se necesita, se invoca al procedimiento escrito en el facet if-needed (éste podría ser preguntarle al usuario, por ejemplo). En la figura 1.2.2.3.h se se muestra este método:

Bloque		
Largo	3	
Ancho	5	
Área	¿?	(if-needed (func-area (Ancho * Largo))))
Slot	Slot value	Facet

Figura 1.2.2.3.h

IF-ADDED: al añadir un valor en un slot se puede activar un procedimiento (el cual puede afectar el valor de otro slot). En la figura 1.2.2.3.i se ejemplificamos este método:

Bloque1		
Área	¿?	if-added (mult*2 area)
Ancho	5	
Área	¿?	(if-needed (func-area (Ancho * Largo))))
Slot	Slot value	Facet

Figura 1.2.2.3.h

IF-REMOVED: al quitar un valor de un slot se activa un procedimiento.

De igual forma se puede tener: before y after, los cuales se activan antes y después de obtener un valor, siendo posible realizar combinaciones con todos los procedimientos anteriores.

Estrategias

Como hemos visto los valores de un slot; pueden obtenerse de diversas formas, básicamente tenemos la herencia-Z y la herencia-N.

1. Valores, defaults, demons en un nivel y luego hacia arriba (herencia-Z). De esta forma los valores que se puedan obtener en un nivel son más confiables que los de sus niveles superiores.
2. Valores hacia arriba, defaults hacia arriba y demons hacia arriba (herencia-N). Bajo este esquema si se puede obtener un valor directamente es más confiable que lo que de por default o por medio de los demons.

Procedimiento de Herencia

Sea F un frame y S un slot

```
UNTIL se encuentre un valor para S o F = null
  IF F tiene un valor para S acaba
  ELSE sea F = superclase de F' por medio del slot IS-A
```

(Un frame F' será superclase de un frame F cuando F sea un slot de F').

Con default o demon sería:

Sea F un frame y S un slot

```
UNTIL se encuentre un valor para S o F = null
  IF F tiene un (demon/default) para S
  Then (ejecuta el demon/asigna el default) y acaba
  ELSE sea F = superclase de F' por medio del slot IS-A
```

Para combinar: valor, demon, default tenemos las herencias Z y N.

Herencia-Z:

Sea F un frame y S un slot

```
UNTIL se encuentre un valor para S o F = null
  IF F tiene un valor para S Then asigna el valor.
  ELSE IF F tiene un demon, Then ejecuta el demon
  ELSE IF F tiene un default para S, Then usa el default
  ELSE sea F = superclase de F' por medio del slot IS-A
```

Herencia-N:

Realiza: herencia con valor, herencia con demons y herencia con defaults.

Tipos

A los frames se les puede incluir restricciones: cardinalidad, rangos permisibles, tipo de datos, etc.

Ejemplo:

(intersección (enteros (intervalo 0 100)) (not.one.of 23 36))

Perspectivas

Un objeto puede verse desde varias perspectivas como se ejemplifica a continuación en la figura 1.2.2.3.i:

ladrillo -	Estructural	- función = soporte
ladrillo -	Juguete	- función = juego
ladrillo -	Regalo	- función = adorno
ladrillo -	Arma	- función = romper cabezas
ladrillo -	Cultural	- función = detener libros

Figura 1.2.2.3.i

Herencia Múltiple y Ambigüedad

A veces se quiere heredar información de más de un frame (la organización se vuelve más una red que un árbol), lo que puede provocar conflictos de información.

Con herencias múltiples no se tiene problemas mientras no exista conflicto en la información.

- En herencia múltiple se tiene que incluir un método que decida de donde heredar.
- Algunos sistemas no deciden (escépticos) por ser contradictorio.
- Algunos permiten varias conclusiones (crédulos).

Se puede usar información adicional para resolver la ambigüedad (por ejemplo con un demonio: IF-NEEDED).

A veces se cancelan líneas de herencia para eliminar ambigüedades, en algunos sistemas se permiten que los valores de slots apunten a otros frames.

Los frames se pueden utilizar para entender historias (idea de los Scripts). Veamos esto representado por los frames de las figuras 1.2.2.3j, 1.2.2.3k, 1.2.2.3l, 1.2.2.3m, 1.2.2.3n, 1.2.2.3o y 1.2.2.3p.

Evento

Lugar	L	Procedure1
Día	D	Procedure2
Tiempo	T	Procedure3
Slot	Slot value	Facet

Figura 1.2.2.3.j

Desastre

Es-un	Evento	Procedure1
Muertos	N	Procedure2
Heridos	M	Procedure3
Sin-casa	O	Procedure4
Daños	P	Procedure5
Slot	Slot value	Facet

Figura 1.2.2.3.k

Terremoto

Es-un	Desastre	Procedure1
Magnitud	X	Procedure2
Falla	Y	Procedure3
Slot	Slot value	Facet

Figura 1.2.2.3.l

Inundación

Es-un	Desastre	Procedure1
Río	R	Procedure2
Slot	Slot value	Facet

Figura 1.2.2.3.m

Huracán

Es-un	Desastre	Procedure1
Vel-viento	V	Procedure2
Nombre	N	Procedure3
Slot	Slot value	Facet

Figura 1.2.2.3.n

Evento-social		
Es-un	Evento	Procedure1
Invitados	L	Procedure2
Casa-se	M	Procedure3
Slot	Slot value	Facet

Figura 1.2.2.3.o

Boda		
Es-un	Evento-social	Procedure1
Novia	Na	Procedure2
Novio	No	Procedure3
Padres-Novia	Pna	Procedure4
Padres-Novio	Pno	Procedure5
Vestido-Novia	VN	Procedure6
Slot	Slot value	Facet

Figura 1.2.2.3.p

Representación del conocimiento utilizando frames

El objetivo de los frames es que al encontrar una situación nueva, se pueda seleccionar de memoria una estructura (frame) y adaptarla a la realidad cambiando detalles. Como ya vimos un frame es una estructura de datos para representar situaciones estereotípicas.

Enumeremos algunas características de los frames:

1. Tiene información asociada de cómo usar el frame, qué esperar que pase, qué hacer si las expectativas no se cumplen. Además están organizados en jerarquías: Los frames de arriba prácticamente no cambian y los terminales tienen slots que se llenan con nuevos datos.
2. Los diferentes frames comparten los frames terminales. Esto es crítico para poder coordinar la información desde diferentes puntos de vista.
3. Los frames terminales normalmente están llenos por asignaciones de default u omisión.
4. Estas pueden ser remplazadas fácilmente con nueva información.
5. Utilizan defaults y demons que facilitan el llenado de slots.
6. El sistema de frames tiene asociado un proceso de apareo controlado por la información en los frames y las metas a cumplir.
7. La lógica no es adecuada, por no poder representar aproximaciones (dadas por los defaults) imperfectas.
8. En el entendimiento de oraciones, si los niveles de arriba son satisfechos pero algunos de los terminales no lo son, entonces tenemos una oración sin sentido.

Si los niveles superiores son débiles pero los inferiores son "sólidos" tenemos una oración gramaticalmente errónea pero con sentido.

9. Se pueden describir escenarios.

10. Podemos pensar en frames sintácticos, semánticos, temáticos y narrativos.

11. Una frame puede verse como una colección de preguntas a hacer acerca de una situación hipotética: especifica puntos a tratar y métodos para manejarlos.

12. Una terminal debe de ser no sólo la pregunta que se quiere resolver, sino también sugerencias de cómo responderla. Los valores por default u omisión son los más simples.

La inferencia en los frames, puede realizarse de diversas formas algunas de las cuales son las siguientes:

Instanciación: dado un frame representando un concepto, podemos generar una instancia de ese concepto llenando sus slots.

"Criteriality": la posesión de atributos es suficiente y necesaria para que un objeto califique como instancia de un frame.

Apareamiento: una instancia de un concepto puede verse como instancia de otro.

Diferentes puntos de vista: un objeto puede tener propiedades aparentemente contradictorias si se le ve desde diferentes puntos de vista (traducción analógica, por ejemplo ver a un hombre como un cerdo).

Razonamiento por default: hacer ciertas deducciones (por default) y luego posiblemente eliminarlas al tener más información (razonamiento no-monotónico).

Razonamiento reflexivo: habilidad de hablar acerca de sí mismo (por ejemplo saber sobre su propio mecanismo de deducciones).

En principio, todos estos tipos de razonamiento se pueden hacer relativamente fácil usando lógica de primer orden, excepto el razonamiento por default y razonamiento reflexivo.

Implementación de reglas utilizando de frames

Una regla puede apoyarse en la estructura de los frames:

(Regla N

```
(IF (frame1 propiedad1,i valor1,i)
      (frame2 propiedad2,j valor2,j)
      ...)
(THEN (framen propiedadn,k valorn,k)
       (framem propiedadm,l valorm,l)
       ...))
```

La regla toma valores de frames en sus condiciones y modifica/genera frames en sus acciones. También se puede combinar con mecanismos de herencia para obtener valores de frames a partir de herencia de sus antecesores.

Por ejemplo:

```
(regla 37
  (if (clase ?X camión)
      (valor peso ?X ?P)
      (> ?P 10000)
      (min-cardinalidad llantas ?X 10))
  (THEN (clase ?X camion-grande)))
```

Al disparar la regla añadimos a una instancia de camión la clase de camión-grande y por lo tanto hereda todos los valores del frame camión-grande. También se pueden organizar módulos de reglas al asociar éstos con los *frames*. Por ejemplo, un frame camión asociado a un método que invoque reglas de diagnóstico (heredables a sus especializaciones), puede servir para guiar las reglas.

1.2.2.4. 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.

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:

Lenguaje natural	Cálculo de predicados
Tenemos los siguientes dos hechos:	
1. Alma y Hernán son matrimonio	matrimonio(Alma, Hernán)
Explicación: La relación "matrimonio" existe entre Alma y Hernán.	
2. Alma vive en Mazatlán	vive_en(Alma, Mazatlán)
Explicación: La relación "vive_en", existe entre Alma y Mazatlán	

Ahora veamos dos reglas generales de conclusiones:

3. Cuando dos personas, (X1,X2) matrimonio(X1,X2) => casados(X1,X2)
son matrimonio, están casadas.

Explicación: La relación "matrimonio" entre X1 y X2 tiene como consecuencia la relación "casados" de las personas X1 y X2.

4. Ahora bien cuando dos personas
X3 y X4 están casadas y la casados(X3,X4)AND vive_en(X4,X5)=>
persona X4 vive en el lugar X5, vive_en(X3,X5)
entonces la persona X3 también
vive en X5.

Basados en estos enunciados formales, puede demostrarse la suposición de que Hernán vive en Mazatlán, aunque ésta suposición hasta ahora no existía como hecho.

vive_en(Hernán, Mazatlán)

De esta forma tenemos que una 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 "Hernán" y "Mazatlán".
- b) Cuando en la parte de condiciones de la misma regla, se hayan cumplido todas las que llevan a esta conclusión.

1.2.3. Estructuras de control de los Sistemas Expertos

Como ya se vió es necesario formular un mecanismo de inferencia para el procesamiento del conocimiento, que evalúe las reglas y el conocimiento en hechos. Básicamente existen dos formas de evaluación de reglas:

- Encadenamiento hacia adelante (forward chaining, forward reasoning).
- Encadenamiento hacia atrás (backward chaining, backward reasoning).

El encadenamiento hacia adelante, se define también, como inferencia controlada por los datos o como método "if-added"; mientras que el encadenamiento hacia atrás, se conoce como inferencia controlada por el objetivo o método "if needed".

En ambos métodos pueden existir casos en los que haya varias reglas para elegir, por lo que debemos contar con criterios de solución de conflictos ante la presencia de varias reglas.

Encadenamiento hacia adelante

En el encadenamiento hacia adelante, se buscan en la base de conocimiento reglas para los hechos conocidos y se ejecuta su parte de acción hasta alcanzar el objetivo, o hasta que no quede ninguna regla más que pueda ser disparada.

Visto de forma algorítmica tenemos lo siguiente:

Encadenamiento hacia adelante:

CI: Conocimiento Inicial (hechos)

UNTIL alcanza el objetivo, o bien no hay reglas sin aplicarse

DO

1. Determinar el dominio K de las reglas, cuyas premisas se cumplen con CI (aquellas que sean susceptibles de ser ejecutadas).
2. Elegir de K una regla R de acuerdo a la estrategia de solución de conflictos.
3. $CI := CI \cup$ el resultado de la evaluación de R aplicada a CI

Encadenamiento hacia atrás

En el encadenamiento hacia atrás se parte de un objetivo (una hipótesis), de una conclusión. Todas las reglas que contienen esta conclusión con comprobadas, para saber si cumplen sus condiciones.

Veamos su representación algorítmica:

Encadenamiento hacia atrás:

CI: Conocimiento Inicial (hechos)

UNITIL hipótesis (objetivo) confirmada, o no hay más reglas que disparar

DO

1. Determinar el dominio K de las reglas, cuya conclusión pueda unificarse con la hipótesis.
2. Elegir de K una regla R de acuerdo a la estrategia de solución de conflictos.
3. En el caso de que la premisa de R no se encuentre en CI, se realiza encadenamiento hacia atrás; esto es dicha premisa se tomaría como hipótesis y se trataría de probarla.

Razonamiento mixto

Cabe mencionar que existe un encadenamiento bidireccional, en el que se plantea el encadenamiento hacia adelante y el encadenamiento hacia atrás en forma alternada, que no es lo mismo que encadenamiento mixto. En éste se combinan ambos tipos de razonamiento; las reglas llaman simultáneamente a la parte premisa; es decir, a los hechos establecidos (encadenamiento hacia adelante) y a los hechos por establecer (encadenamiento hacia atrás).

En relación al encadenamiento bidireccional, existen varias posibilidades; generalmente, el Sistema Experto procede primero al encadenamiento hacia adelante, para determinar las conclusiones parciales y después éstas hipótesis se verifican mediante un encadenamiento hacia atrás.

El motor de inferencia puede igualmente utilizar en forma alterna los encadenamientos hacia adelante y hacia atrás, examinando en cada ciclo si estos dos razonamientos confluyen, lo cual permite recorrer la mitad del camino y pararse en el medio o punto de conjunción.

Otra posibilidad es realizar primero un encadenamiento hacia atrás sobre los objetivos o los subobjetivos del sistema, con una propagación hacia adelante de los objetivos verificados.

1.2.4. Elección de una estrategia de control

No importando el tipo de razonamiento que se elija, la construcción de un motor de inferencia comprende dos etapas: La definición de los métodos de selección de las reglas candidatas y la elección de la regla a activar.

En la etapa "definición de los métodos de selección" de las reglas candidatas, se confronta cada una con el conjunto de hechos de la base o con el objetivo. Esta operación de búsqueda por comparación se denomina filtrado y puede preceder a una fase de restricción.

En la etapa "elección de la regla a activar" se tiene influencia en el rendimiento del sistema, dado que el experto sabe qué conocimiento debe utilizar en una situación dada, pero no el Sistema Experto. Es muy común la elección de una estrategia prefijada, optar por ejemplo por la primera, o por la aplicada más recientemente. Un camino muy interesante es la utilización de meta-reglas para guiar la elección del sistema, aquí intervienen los datos bajo forma de reglas de producción las cuales expresan estrategias de elección de reglas.

La calidad de la elección realizada para construir estos dos elementos, afecta de forma determinante a la eficacia del motor de inferencia.

Se emplea mucho el control dirigido por los datos, pues cualquier programa puede efectuar sus deducciones a partir de las informaciones suministradas por el usuario, sin que esto entrañe precisar su solución.

Este enfoque es fácilmente programable y parece el más natural cuando el sistema dispone de un cierto número de hechos, efectuándose la resolución de forma ciega, sin saber a priori, qué objetivo hay que alcanzar.

Sin embargo ésta solución es ineficaz para problemas complejos.

Por el contrario si son varios los objetivos a alcanzar o verificar, es más interesante utilizar el control dirigido por los objetivos con el fin de evitar la explosión combinatoria.

Parece lógico que existan diferentes enfoques para diferentes problemas y aconsejable la existencia de los dos tipos de encadenamiento en un motor de inferencia, sobre todo considerando que la mayor parte de las veces, el conocimiento que se tiene es empírico, pero en otras teórico.

El encadenamiento hacia delante está generalmente, indicado para el tratamiento de conocimientos empíricos, que juegan el papel de meta-conocimientos y el encadenamiento hacia atrás se adapta mejor a la confirmación de objetivos, precedentemente evocados, por medio del conocimiento cierto.

También es recomendable éste último en el caso de informaciones incompletas donde debe producirse diálogo con el usuario.

Deben considerarse también otros criterios, además de los modos de razonamiento, para diferenciar los motores de inferencia.

Estrategias irrevocables o por tentativas

Se dice que una estrategia es irrevocable cuando el camino tomado no obliga nunca a volver a la causa; contrariamente, una estrategia por tentativas puede remitir a la causa de la aplicación de una regla si la elección desemboca en un rechazo; el motor de inferencia opera entonces con vuelta atrás (encadenamiento hacia atrás) para intentar una regla descartada anteriormente.

En general, los motores de inferencia con encadenamiento hacia atrás, utilizan frecuentemente una estrategia por tentativas, mientras que los motores de inferencia con encadenamiento hacia adelante operan con estrategias irrevocables.

Estrategia en profundidad o búsqueda en amplitud

Con una estrategia en profundidad, el motor de inferencia comienza, explorando inicialmente el árbol de soluciones, se realiza la búsqueda bajando de nodo en nodo por la izquierda hasta llegar al último, después se sube un nodo para

continuar por el que se encuentra a la derecha, bajando por la izquierda y así sucesivamente hasta encontrar la solución.

Por el contrario, con búsqueda en amplitud el motor de inferencia explora el árbol en anchura (estudiando todas las ramas posibles), se realiza la búsqueda bajando un nodo, subiendo y bajando al que se encuentra a la derecha y así sucesivamente hasta que se acaba el nivel, después se baja al nodo de la extrema izquierda para descender al siguiente nivel y continuar con la búsqueda en todo ese nivel, tal como se describió para el nivel anterior y se continúa así hasta encontrar la solución.

La estrategia puede ser en profundidad e irrevocable, o por tentativas, incluso en amplitud.

De hecho, las estrategias por tentativas funcionan generalmente primero en profundidad y después en amplitud, donde explora todo el grafo, ya que el retorno hacia atrás es inútil.

En la figura 1.2.4.1.a se muestra el grafo de la búsqueda en profundidad, mientras que en la figura 1.2.4.1.b se considera la búsqueda en amplitud.

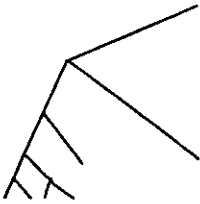


Figura 1.2.4.1.a

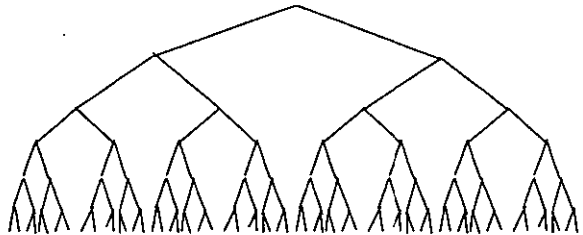


Figura 1.2.4.1.b

Tipología y metodología de construcción de los Sistemas Expertos

2 Tipología y metodología de construcción de los Sistemas Expertos

2.1 Tipología

Es muy difícil agrupar a los Sistemas Expertos en clases homogéneas, dado que son muy diversos; sin embargo si puede hacerse una diferenciación, en función de la naturaleza de la tarea que realizan y en consecuencia de su uso, en tres grandes categorías: Sistemas Expertos de clasificación-interpretación, Sistemas Expertos de control (monitores) y Sistemas Expertos de previsión-concepción.

2.1.1 Sistemas Expertos de clasificación-interpretación

Éstos realizan por ejemplo, una clasificación de las causas posibles de mal funcionamiento en función de verosimilitudes o mediante análisis de datos (cuantitativos, cualitativos, visuales, etc.), con el objeto de obtener su significado, este tipo de Sistemas Expertos es el más antiguo (diagnóstico, prospección, etc.) y el más habitual, distinguiéndose por un conjunto de datos estáticos que se corresponden a observaciones o a determinadas manifestaciones.

2.1.2 Sistemas Expertos de control (monitores)

Se caracterizan por una noción del tiempo que es primordial y por una tarea de verificación de la correcta ejecución de un proceso: control de procesos, vigilancia de hechos, etc.

Estos Sistemas Expertos supervisan la evolución de los datos o de las señales procedentes de los procesos a controlar; tratan, por consiguiente, datos continuos en una lógica no monótona; si A es cierto en el instante T, se puede tomar A en consideración en no importa que instante $T + Q$; a diferencia de la lógica monótona de los Sistemas Expertos de la primera categoría, en donde una proposición lógica, no puede tomar más de un valor en un tiempo.

De esta forma, observemos las diferencias entre un problema de diagnóstico y un problema de control de proceso. El primero consiste en interpretar un conjunto de datos que caracterizan una situación, en la que se busca la explicación más coherente: el diagnóstico; en la segunda categoría, se deben interpretar medidas efectuadas a intervalos regulares para desencadenar inmediatamente una acción, cuando se verifican determinadas proposiciones.

Hace falta una estrategia de control que permita seguir la evolución de los datos, ya que en el caso clásico de los diagnósticos, la interpretación es fundamentalmente puntual.

2.1.2 Sistemas Expertos de previsión-concepción.

Estos Sistemas Expertos se caracterizan por una concesión de recursos bajo limitaciones (al mejor). Los mecanismos de inferencia de éste tipos de sistemas, se anticipan a los resultados, permitiendo modificarlo sobre previsiones, con lo que se tiene la indicación de un futuro previsible a partir de datos presentes, pasados y bajo modelos limitados.

Las anteriores clasificaciones no están cerradas y determinadas aplicaciones pueden pertenecer a varias clases. La importancia de tales clasificaciones es su utilidad en la elección de la metodología y la herramienta de desarrollo.

2.2 Metodología

En un proceso de desarrollo se abarcan todos los pasos de planificación y las medidas durante la creación del software, para tal efecto nace el término: "Técnica de Procesos", la cual surgió como una respuesta a la llamada crisis de software en los años 60. En aquel entonces dicha disciplina no parecía estar en posibilidad de conjugar debidamente la calidad, el costo y los plazos con los deseos del cliente; sólo con la introducción de una técnica de procesos de software con una lista de obligaciones para clientes y para fabricantes de software, se pudo lograr resolver tal situación.

En los primeros años de expansión de los Sistemas Expertos, se acostumbraban dejar de lado los principios básicos de la técnica de procesos, ya que se trabajaba mas aprisa sin dedicar mucho tiempo a las planificaciones; auxiliándose de herramientas cómodas, se desarrollaban versiones de demostración en poco tiempo, para luego ir las ajustando paulatinamente, pasando de ésta técnica a lo que hoy conocemos como ingeniería de software.

Dadas las nuevas posibilidades en software y hardware, buscando aprovecharlas al máximo, no podemos aplicar la técnica de procesos convencional, sino que debemos modificar ésta última convenientemente. Hablaremos en primera instancia del modelo de ciclo de vida de un sistema; ya que veremos que las metodologías de desarrollo de Sistemas Expertos, no son sino una variación de este modelo.

2.2.1 El modelo de ciclo de vida de un sistema convencional

Este modelo define las fases básicas de un proyecto de desarrollo de software, describiendo la salida de cada fase. Estrechamente asociado con el modelo de ciclo de vida, está una técnica de administración de acompañamiento (dependencia), que depende de una revisión al completar cada fase, para establecer el status verdadero del proyecto de desarrollo.

El ciclo de vida puede ser descrito idealmente de la siguiente manera: comienza con la manifestación de una necesidad de un cliente, un analista de sistemas

trabaja con éste para analizar la tarea y definir los requerimientos funcionales del sistema propuesto; estos son descritos formalmente en una especificación de requerimientos; siguiendo una revisión y aprobación de especificaciones, un diseño, el cual es descrito en una especificación de diseño desarrollado para satisfacer los requerimientos descritos en tal especificación. Las estrategias de pruebas también son incluidas en este punto.

El diseño es implementado usando una representación (por ejemplo en lenguaje de programación) entendible por una máquina. Se verifica a través de la ejecución de pruebas siguiendo el plan diseñado para tal efecto. Después de la fase de prueba, el sistema es liberado al cliente iniciándose un largo período de mantenimiento.

Sin embargo este modelo presenta algunas dificultades.

2.2.2 Dificultades con el modelo de ciclo de vida tradicional

El uso de este modelo ayudó a reducir la crisis de software, pero todavía persisten varios problemas fundamentales.

1. El modelo de ciclo de vida es lineal.

El uso de un modelo de ciclo de vida tradicional, asume que cada etapa puede ser completa y correctamente implementada antes de moverse a la próxima y podría asumirse que:

- Un completo entendimiento de los requerimientos de sistemas puede ser derivado y descrito al principio del proyecto.
- Es posible definir pasos intermedios discretos, cada uno de los cuales puede realizarse y verificarse independientemente, antes de ir al siguiente paso.
- La salida de cada etapa puede ser alineada exactamente con los requerimientos originales.

Desafortunadamente; si bien estas suposiciones sirven para simplificar la totalidad del problema, son invariablemente incorrectas de acuerdo al siguiente principio:

2. La iteración es inevitable en cualquier proyecto mayor de software.

Dado que es difícil controlarla, el desarrollo se basa en percepciones de modelos lineales, cualquier cambio en los resultados de la fase anterior, retarda el desarrollo, originando una carencia de disciplina.

En realidad, usualmente no existe alguien quien realmente entienda el problema al comenzar el proyecto.

En la mayoría de los casos, las primeras fases del desarrollo del sistema deben completarse antes de que alguien realmente entienda lo que necesita realizarse, lo cual conduce a una elegante solución para un problema equivocado, de tal forma que el sistema final cubra la especificación de requerimientos; pero no las

expectativas del usuario. Los expertos en ingeniería de software tienen mucho tiempo de haber identificado esas debilidades y una amplia investigación está en curso para modificar el modelo tradicional. Sin embargo este modelo es ampliamente usado en la industria.

Veamos ahora el tipo de ciclo de vida que nos ocupa.

2.2.3 Ciclo de vida de desarrollo de un Sistema Experto

El desarrollo de un Sistema Experto generalmente sigue un modelo que es una versión iterativa del modelo de ciclo de vida tradicional. El de un Sistema Experto está basado en el reconocimiento de la naturaleza evolutiva del software. Esto es especialmente cierto en el desarrollo de Sistema Expertos, porque por definición, tratan con áreas de problemas que inicialmente están pobremente definidas y no bien entendidas. Existen varias características únicas del modelo de Sistemas Expertos:

- Cliente (y dominio experto) se involucran durante el proceso completo; esto contrasta con el sistema tradicional, en el cual el usuario especifica los requerimientos y se aleja, esperando el producto final.
- Demostraciones frecuentes de trabajo para fortalecer el diseño. Estas permiten al cliente y al experto visualizar la funcionalidad del sistema completo y solicitar cambios.
- Los cambios son vistos como saludables – de hecho, este es el concepto central. Específicamente, se enfatizan en la fase en que son más fáciles de implementar. Las herramientas de desarrollo de Sistemas Expertos apoyan en la administración de los efectos del cambio.

2.3 Proceso de desarrollo del Sistema Experto

A continuación se describe el método de desarrollo de un Sistema Experto utilizada por la empresa SIEMENS NIXDORF; líder en ingeniería eléctrica y electrónica con mas de 150 años de historia, la cual ofrece soluciones tecnológicas en: industria, energía, informática y comunicaciones, salud, transporte e iluminación, SIEMENS es pionera en el desarrollo de Sistemas Expertos y el presente trabajo está basado en su metodología de desarrollo. En general los libros consultados ofrecen metodologías con muchas semejanzas y algunas diferencias; estas últimas mas de forma que de fondo y la razón por la que se eligió es porque está precedida de desarrollos exitosos en diversas áreas del conocimiento; tales como: medicina, comunicaciones, lingüística, y en particular han desarrollado Sistemas Expertos de diagnóstico clínico, como es el caso del que nos ocupamos en esta ocasión.

Como se mencionó anteriormente la técnica del desarrollo de Sistemas Expertos es una variación del modelo de ciclo de vida de un sistema; de tal suerte que al desarrollar un proyecto se distinguen cuatro subprocesos: **análisis del problema**,

planificación del sistema, desarrollo para alcanzar la madurez de la aplicación y aplicación y mantenimiento.

Las funciones de los dos primeros procesos se deben cumplir en su totalidad antes de pasar al siguiente proceso; mientras que los dos siguientes pueden no serlo del todo, ya que la aplicación del Sistema Experto ha de ser viable incluso con una base de conocimiento incompleta.

Enseguida abordaremos los subprocesos y su delimitación recíproca.

2.3.1 Análisis del problema

En esta etapa suponemos que la técnica de Sistemas Expertos es la adecuada para solucionar el problema planteado, por lo tanto el objetivo es confirmar esta suposición o rebatirla, ya se han separado las funciones parciales que no pueden ser resueltas dentro del proyecto general, con ayuda de un Sistema Experto; tales funciones parciales se trabajarán según la tecnología de procesos habitual, las cuestiones existentes sobre cada uno de los temas no pueden responderse siempre con exactitud al principio.

Cuanto menor sea el riesgo para el cliente en el subproceso de planificación del sistema mayor será el costo durante el análisis del problema.

En el primer subproceso, la importancia se centra en la viabilidad y necesidad como principio de un Sistema Experto, para lo cual se deben definir los siguientes puntos:

Forma de trabajo del experto

Tipo y alcance del conocimiento utilizado para la resolución del problema

Entorno de aplicación

Usuarios

Costos y rentabilidad

Al finalizar este subproceso, se evalúa la idoneidad de la técnica de Sistemas Expertos para solucionar el sistema planteado, sin embargo en este punto quedarán algunas preguntas sin respuesta, por lo que en el siguiente subproceso de planificación puede modificarse la decisión en cuanto a utilizar ésta técnica.

Forma de trabajo del experto

El experto dispone del conocimiento necesario para resolver determinados problemas, por experiencia, éste sólo está parcialmente documentado por escrito, la parte principal se encuentra en la cabeza del experto y es a través de entrevistas, que el ingeniero del conocimiento captará y documentará este conocimiento, en estrecha colaboración con el experto.

En este punto antes de continuar con el análisis se deben aplicar los llamados "Criterios Críticos", si uno de ellos se cumple el proyecto del Sistema Experto no puede llevarse a cabo.

Criterios Críticos

No existe experto.

No tiene tiempo; es decir, es imprescindible para otras labores.

No está motivado.

¿Existe el experto?

Ante esta pregunta han fracasado ya algunos proyectos de Sistemas Expertos. El desarrollo de éstos implica la existencia de un experto, con conocimiento profundo en la resolución de problemas del dominio de la aplicación.

¿Está el disponible?

Dado que el experto es una parte fundamental del desarrollo de un Sistema Experto, no debe ser una persona imprescindible durante el desarrollo del proceso, por lo cual se deberá liberar para esta función.

¿Está motivado y se muestra cooperativo?

No sirve el mejor de los expertos si no está convencido de lo positivo de un Sistema Experto. Un beneficio tangible para el experto, podría ser la descarga de obligaciones rutinarias que le permitiera dedicarse a fondo a funciones de mayor importancia.

Adicionalmente hay que poner especial atención en la elección del experto, debe considerarse:

- Elegir a una persona que esté en mejor situación de comunicar su conocimiento.
- Se preferirá aquella que tenga conocimientos de informática, dado que esto permitirá una mejor comunicación entre el experto y el ingeniero del conocimiento.
- La experiencia ha demostrado que son mejores dos expertos normales, que uno con mayores conocimientos pero que no esté en disponibilidad de comunicar su conocimiento por sus responsabilidades.

Tipo y alcance del conocimiento utilizado para la solución de problemas

El tipo y volumen del conocimiento del experto decidirán si el problema puede procesarse con ayuda de técnicas de Sistemas Expertos; los problemas numéricos, o la pura utilización de bases de datos no son, sin duda, temas a tratar con esta tecnología, ya que se pueden solucionar mejor, más rápido y más barato, con técnicas de programación convencional.

A continuación se presenta una serie de preguntas, que servirá de apoyo al ingeniero de conocimiento, para obtener una visión del conocimiento del experto.

¿Se limitará el nivel de aplicación?

En un Sistema Experto las funciones a resolver no deben ser demasiado voluminosas en la primera etapa. Hay que aclarar qué resultados puede ofrecer el sistema.

¿Cómo se describe el nivel de aplicación?

La aplicación a solucionar deberá describirse con la máxima precisión, debiendo comprobarse si el conocimiento puede ser descrito en su totalidad, dado que esto será necesario para reproducir las vías de solución.

¿Qué proporción de conocimientos generales se requiere para la solución del problema?

Cuanto más conocimiento general se halle en el análisis de un campo, menos recomendable es el desarrollo de un Sistema Experto.

¿Se trata de conocimiento difuso?

Hay muy pocos temas que disponen de teorías elaboradas para el tratamiento de probabilidades. El experto a menudo documenta declaraciones con probabilidades, que finalmente nos llevan a manifestaciones "curiosas" luego de los encadenamientos. Por lo tanto el conocimiento difuso debe ser reducido.

¿Existen soluciones o intentos de soluciones convencionales para el problema planteado?

Esto es algo que debe conocerse para evaluar la conveniencia de optar por un Sistema Experto.

¿Con qué frecuencia se modifica el conocimiento?

Los parámetros más importantes que se refieren a la modificación de la base de conocimientos, son: cantidad, frecuencia, volumen y si pueden llevarse a cabo de manera sencilla y segura.

¿Se procesarán datos confidenciales?

Si esto es cierto debe crearse un elemento de adquisición que permita al experto introducir esta información confidencial.

Entorno de aplicación del Sistema Experto

Hay que descubrir en qué forma puede introducirse en todo el proceso.

Existen al menos tres posibilidades distintas:

- El Sistema Experto es una solución stand-alone.
- Es parte de un sistema más grande y/o interactúa con otros sistemas.
- Se aplica en tiempo real, hay acoplamiento de procesos.

¿Qué requisitos se plantea respecto a la calidad de la respuesta?

A este respecto existen dos aspectos fundamentales: el tiempo y la exactitud. Por lo que se requiere saber en primer instancia ¿En qué plazo se dispondrá de la respuesta? y ¿Qué consecuencia tendrá una solución errónea?

¿En qué hardware debe funcionar el sistema?

Siempre será preferible usar el ya existente que adquirir uno nuevo, además se deben considerar las interfaces de usuario y con otros sistemas.

Para este punto también se tienen tres "Criterios Críticos"

1. No puede alcanzarse la calidad requerida de la solución.
2. La incorporación en el proceso de trabajo resulta imposible.
3. La creación de un entorno de trabajo, resulta económicamente muy costosa.

El cumplimiento de estos criterios críticos deberá decidirse por la dirección de la empresa o cliente.

Usuario del Sistema Experto

Tiene que integrarse lo antes posible al proyecto, para lo cual es necesario responder algunas cuestiones:

¿Quién es el típico usuario de este sistema?

Es importante conocer el papel que juega el usuario típico del sistema dentro de la organización; su nivel de formación y su grado de motivación respecto al desarrollo del Sistema Experto, asimismo debe determinarse si habrá diferentes tipos de usuarios y sus características.

Experto y usuario deben dominar un mismo léxico para poderse comunicar a través del Sistema Experto.

¿Qué funcionalidad se requiere para la interfaz de usuario?

En este punto se deben determinar cuáles requisitos debe cumplir el componente explicativo; los gráficos que debe llevar y el grado de comodidad del mismo.

¿Pueden ponerse a disposición de cada usuario los medios necesarios para la solución del problema?

En esta interrogante se tienen los siguientes "Criterios Críticos":

1. La diferencia respecto a conocimiento especializado entre experto y usuario es demasiado grande.
2. Los medios necesarios para la solución del problema no pueden ponerse a disposición de cada usuario.

Costo y beneficio del Sistema Experto

Durante la fase de análisis del problema, no puede establecerse una clara relación costo/beneficio, ya que no se dispone de suficientes parámetros cuantificables para ello, sin embargo ya se pueden tratar algunos puntos.

¿Qué costos surgen hasta la aplicación del Sistema Experto?

En este punto se tienen los siguientes factores principales:

Tiempo de trabajo del desarrollador del software.
Tiempo de trabajo del experto
Costos de hardware y software
Formación del encargado a largo plazo del sistema.

¿Qué costos surgen durante la aplicación del Sistema Experto?

En este punto los costos principales son la actualización y el uso de software y hardware.

¿Qué importancia tiene este tema a largo plazo?

Dado que en la actualidad no se piensa en aplicación de Sistemas Expertos de bajo costo, lo que se debe cuestionar es la importancia de los temas que se ataquen a largo plazo.

¿Qué ahorros de tiempo y material se esperan?

En este punto se deben valorar, por un lado, el tiempo que el experto podría dedicar a otras actividades y el del usuario que resolverá los problemas con mayor rapidez.

¿Con qué frecuencia se aplica un Sistema Experto?

Por un lado se debe considerar el número de usuarios y por otro la cantidad de utilizaciones del sistema por usuario.

¿Se afectará capital intelectual de la empresa?

Los Sistemas Expertos tienen en su base de conocimientos, el aseguramiento del capital intelectual de la empresa.

¿Hay otros efectos secundarios positivos?

Estos efectos pueden ser muy diversos desde la documentación del conocimiento hasta la mejora de la imagen de la empresa.

Duración del análisis del problema

El tiempo de análisis puede variar desde una semana hasta varios meses, como se mencionó anteriormente esto dependerá de que tanto desee arriesgar el cliente en el subproceso de Planificación del Sistema y desde luego, de la experiencia de los desarrolladores, además de la naturaleza de la problemática.

2.3.2 Planificación del sistema

El objetivo de la planificación del sistema, es el proyecto técnico de procesamiento de datos del Sistema Experto. El subproceso de planificación se subdivide en cuatro pasos: **analizar los requisitos, proyectar el sistema, realizar parte del sistema y validar el proyecto del sistema.**

En el primer paso: "analizar requisitos", se crea un modelo lógico, que describe el punto de vista orientado a la aplicación de la función a solucionar y se limita a las actividades imprescindibles a dicho fin.

En el siguiente paso: "proyectar el sistema", se crea un proyecto técnico de procesamiento de datos, teniendo en cuenta las limitaciones de implementación descubiertas en el modelo lógico.

La aplicación de entornos de desarrollo de software permite, en el tercer paso: "realización parcial del proyecto del sistema", una rápida conversión del proyecto del sistema, en la implementación de un prototipo.

El paso: "validar el proyecto del sistema", proporciona valiosos resultados para una mejora del proyecto.

Estas cuatro fases crean un ciclo que puede repetirse varias veces durante la planificación, el proyecto del sistema se acerca así, poco a poco, a las necesidades del usuario. Esta forma iterativa de proceder es adecuada para el desarrollo de Sistemas Expertos, ya que las funciones a resolver son complejas y están mal estructuradas, dado que nadie puede formular todos los requisitos al inicio de la planificación del sistema.

Sinopsis

Una vez finalizado el análisis del problema, se realiza el proyecto del sistema en el segundo subproceso del desarrollo del Sistema Experto. A continuación se presenta un esbozo de dicho subproceso en la figura 2.3.2 a.

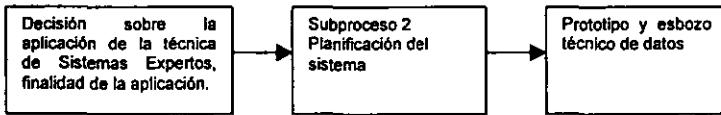


Figura 2.3.2 a

Como se observa en la figura 2.3.2 b; en primer instancia está el análisis de los requisitos, que nos da una primera comprensión del problema, ésta se transforma en un esbozo aproximado en el paso "proyectar el sistema", en la siguiente etapa "Realizar parcialmente el proyecto del sistema", se crea un prototipo que se valida en el cuarto paso.

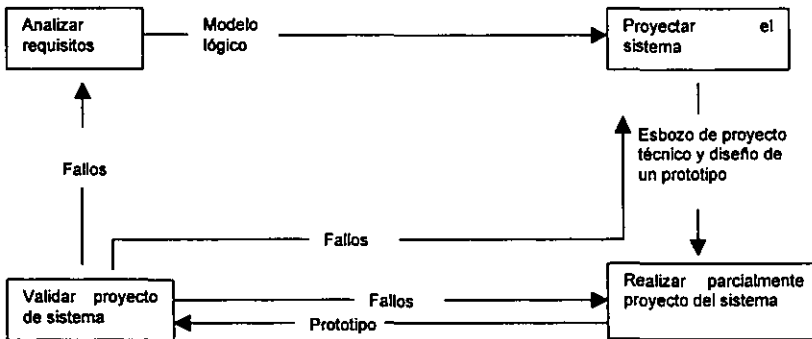


Figura 2.3.2 b

Implementar un prototipo significa realizar provisionalmente parte del sistema con este esbozo del proyecto, valorando y mejorando el resultado. Estos pasos no tienen que darse necesariamente en orden.

Cuando se inicia el proyecto, el peso principal recae en los dos primeros ciclos de la planificación del sistema, en el análisis de requisitos y a medida que avanza, se van trasladando a la validación.

Los resultados de la planificación del sistema se mejoran paso a paso, hasta que se dispone de los siguientes resultados:

- Esbozo de proyecto técnico
- Estructura de la base de conocimientos
- La estrategia de inferencia
- Posibilidades explicativas para resultados deducidos
- La interfaz de usuario
- Evaluación y concepto de prueba
- Base del sistema (hardware y software utilizados)

Primer paso en la planificación del sistema: "Análisis de requerimientos"

Los desarrolladores deben obtener una primer visión del problema, para ello deben dirigirse algunas preguntas clave a los expertos en la solución de la problemática:

- ¿Qué funciones existen?
- ¿Qué relación existe entre ellas?
- ¿En qué momento se dispone de qué información?
- ¿De dónde proceden las informaciones necesarias? (usuarios, bases de datos, otros sistemas)
- ¿Qué funciones no deben ser solucionadas por el sistema?
- ¿Qué información de detalle hace falta?
- ¿Cómo se estructura esta información?
- ¿Qué suposiciones hace el experto?
- ¿Qué condiciones marco están disponibles?
- ¿Existen varias soluciones?

Después se tendrán que hacer entrevistas con el experto en un ámbito mas estrecho, incluso llegando a la ejemplificación de casos concretos, o la misma observación del experto en su medio real.

El modelo lógico

El modelo lógico es una descripción de los requisitos del problema, que reúne toda la información extraída del experto, en él se modelan sólo las funciones básicas.

En un Sistema Experto se intenta reproducir el comportamiento del experto en la solución de problemas, por lo cual se recomienda hacer el análisis de los requisitos en dos etapas; en la primera se plantea el entorno del problema y el cómo lo ha solucionado hasta ese momento el experto (estado ES), a partir de estos requisitos puede plantearse el modelo lógico del (estado DEBE) en la segunda etapa.

Veamos una descripción del modelo lógico a través del siguiente método de modelado.

1. Objetos y sus relaciones de comunicación

Los pilares del modelo lógico son los llamados objetos, dentro de los cuales se procesan las informaciones, el intercambio de información entre éstos, tiene lugar a través de "vías de comunicación", en la representación de las relaciones de información entre objetos se trata fundamentalmente: "Qué objeto procesa o produce informaciones".

2. Contenido y estructura de las informaciones

La información intercambiada entre los objetos, se describe respecto al contenido y estructura, no se trata de la representación material de esta información, sino del contenido de la misma.

3. Comportamiento de los objetos

En el procesamiento de la información por los objetos hay que observar generalmente unas cuantas reglas determinadas. Se muestra lo que debe suceder para transformar la información presente en otra que sea nueva.

Segundo paso en la planificación del sistema: "Proyecto"

El modelo lógico creado en el paso "Análisis de requisitos", no tiene en cuenta de qué forma se realiza el comportamiento de los objetos, este es el contenido del segundo paso.

Hay que trabajar en las siguientes funciones:

- Clasificación de las funciones lógicas en los componentes de software; también hace falta un modelado para las funciones parciales realizadas sobre una base de conocimientos. Son mejores varios mecanismos de inferencia especializados con bases de conocimiento locales, que un módulo de inferencia universal, con una sola base global de conocimientos.
- Estructurar el desarrollo de las funciones parciales. Este es el paso de la activación de objetos accionada por datos, en el modelo lógico, a una jerarquía de llamada de los componentes de software.
- Proyecto de las estrategias de inferencia y estructuras del conocimiento.

Como en la metodología clásica de desarrollo de proyectos, estas actividades tienen como resultado un modelo de realización, las distintas funciones del modelo lógico se clasifican sobre objetos reales (por ejemplo módulos de software), un objeto real puede tomar varias funciones del modelo lógico y viceversa.

Para un proyecto completo deben definirse:

- El comportamiento dinámico de los objetos
- La memoria interna de los objetos
- La estructura y contenido de las informaciones

Este es el grado de detalle que se desea alcanzar en el análisis clásico del sistema, pero en el desarrollo de Sistemas Expertos deben asegurarse, ya en esta fase algunos aspectos de la futura implementación mediante un prototipo.

Un prototipo permite

- Una verificación de las decisiones centrales del proyecto, como estrategia de la inferencia o la representación del conocimiento.
- Un análisis de principios de solución técnica de software para la implementación final.
- Una determinación aproximada del diálogo necesario entre el usuario y de las explicaciones necesarias.

Para el diseño de un prototipo válido no hace falta tener en cuenta todos los aspectos, bastan aquellas funciones que tienen influencia hacia fuera y se precisa de componentes de software, cuya ejecución de procesamiento de datos parezca incierta. De esta forma, el prototipo refleja el Sistema Experto futuro, mediante soluciones parciales características.

Tercer paso en la planificación del sistema: "Realización parcial del proyecto"

La implementación de prototipos se reduce en principio, a la conversión del modelo de diseño en una forma de representación apoyada por la herramienta de desarrollo; sin embargo pueden aparecer algunas dificultades; el modelo puede contener conceptos que no pueden ser cubiertos por los Shells disponibles en el mercado.

En este caso existen dos posibilidades:

1. Ampliar un Shell ya existente.
2. Implementar un software propio para un desarrollo.

La mayoría de los Shells ofrecen una interfaz para funciones externas, de esta forma pueden realizarse conexiones con otros sistemas de software, tales como bases de datos. Pero éste mecanismo no basta para ampliaciones reales de Shell; el desarrollador del Sistema Experto, necesita tener acceso a los elementos básicos del componente de resolución de problemas y de representación del conocimiento; a este efecto los módulos de software y sus interfaces, deben estar abiertos. Desafortunadamente muy pocos Shells cumplen estos requisitos.

No hay que perder de vista que en este punto ya contamos con un primer prototipo.

Cuarto paso en la planificación del sistema: "Validación del proyecto"

El prototipo se prueba en estrecha colaboración entre experto e ingeniero del conocimiento, esto se lleva a cabo principalmente con ejemplos, se prefiere no utilizar aquellos anteriormente empleados como base para la adquisición del conocimiento, en este punto lo principal es verificar si:

¿El sistema se comporta tal como el experto lo esperaba?

Esta pregunta debe plantearse, respecto a todos los criterios definidos en la fase de análisis de requisitos.

Planificación del sistema "sin final"

El ciclo de planificación del sistema puede repetirse varias veces, mejorando el prototipo. Un elevado número de ciclos es un signo negativo de la planificación; hay que procurar realizar el mínimo número posible. La planificación del sistema queda cerrada cuando el prototipo llegó a un estado estable; para este efecto se cuenta con tres criterios críticos:

- El mecanismo de inferencia y la representación del conocimiento funcionan.
- De cada círculo de problemas hay al menos un caso ejemplar implementado.
- El gasto para la realización del sistema de aplicación puede valorarse como justificado.

2.3.3 Desarrollo para alcanzar la madurez de la aplicación

En este subproceso se crea el Sistema Experto para su aplicación productiva. La base para la implementación, está formada por las experiencias obtenidas en la planificación del sistema. El desarrollo del núcleo y de las posibilidades de explicación, la realización de la interfaz de usuario y la anexión y ampliación de otros componentes tienen lugar en esta fase. Se captó ya gran parte del conocimiento especializado necesario integrándolo al sistema. Se hicieron también las pruebas necesarias para dar el visto bueno.

Una vez acabados los trabajos en el prototipo, empieza el desarrollo del sistema de aplicación; para ello se establece primero un plan de proyecto que tenga los siguientes paquetes de trabajo:

- Desarrollo de un sistema central (Mecanismo de inferencia, Representación del Conocimiento).
- Ampliación de la base de conocimientos.

- Realización de una interfaz de usuario, para consultas y adquisición del conocimiento.
- Posibilidades de explicación del Sistema Experto.
- Pruebas.

En este punto deben valorarse dos variantes:

1. Reimplementación del prototipo

No se utiliza el código del programa, pero se conservan los paradigmas de programación y se toman las estructuras de los datos. La base para la reimplementación es el concepto inicial de la planificación del sistema.

2. Postdesarrollo del prototipo hacia el sistema de aplicación

El principal punto se encuentra en la optimización del código. El entorno de desarrollo para el prototipo y sistema de aplicación es el mismo.

Adquisición del conocimiento

La base de conocimientos debe ampliarse como punto fundamental para la realización del sistema, dado que el ingeniero del conocimiento actúa como intermediario entre el conocimiento experto y el sistema, éste refleja la comprensión de aquel, no la del experto, por lo cual resultará muy útil un componente de adquisición del conocimiento configurado para tales efectos. Dicho componente cubre los detalles de implementación del Sistema Experto y posibilita una descripción de las ampliaciones o modificaciones.

Estrategia de ampliación de la base de conocimientos

Las pruebas necesarias del Sistema Experto corren a cargo de los expertos y los desarrolladores. El experto es el único responsable de un contenido correcto de la base de conocimientos y ante la necesidad de incorporar nuevo conocimiento, debe implementarse un mecanismo que garantice la consistencia del mismo.

Tras la ampliación de la base de conocimientos existente se verifica primero el conocimiento nuevo; durante esta verificación surgen casos que se almacenan en un archivo, constan principalmente de los datos de introducción de una consulta del Sistema Experto. Tras la aceptación del nuevo conocimiento, se repasan los casos antiguos. No deben aparecer nuevos fallos; la repetición de los casos es muy costosa en tiempo, por lo que se recomienda automatizar este proceso.

Prueba y ejecución del programa piloto

Contrariamente con lo que sucede con el desarrollo de sistemas convencionales, los Sistemas Expertos no pueden ser completamente probados. En el programa piloto participan tres grupos:

- **Los expertos:** Quienes conocen el campo de aplicación, y son responsables del contenido de la base de conocimientos y del tipo de conclusiones.
- **Los ingenieros del conocimiento:** Quienes poseen el tocante a la realización técnica, siendo responsables de las posibilidades y limitaciones de la técnica informática.
- **Los usuarios:** Quienes utilizarán el Sistema Experto y han de verificar la mejora cualitativa y cuantitativa de su trabajo, en la práctica.

La prueba del programa se desarrolla en tres grupos:

1. **Casos de pruebas definidos:** En esta etapa trabajan estrechamente los expertos y el ingeniero del conocimiento.
2. **Primera aplicación práctica:** Aquí trabajan los expertos y un usuario cooperativo y voluntarioso. Se prueban casos prácticos en un entorno de aplicación real.
3. **Aceptación:** Esta es la etapa final de la prueba y dado que se implica la aceptación por parte del usuario, el ingeniero del conocimiento no participa de forma activa.

2.3.4 Aplicación y mantenimiento

El paso a aplicación productiva debería tener lugar lo antes posible, para establecer rápidamente relaciones con la experiencia de la aplicación práctica; por ello, la entrega de un Sistema Experto se realiza frecuentemente por partes.

El mantenimiento a la base de conocimientos se deja en manos del usuario, ya que sólo él dispondrá del conocimiento especializado, necesario a largo plazo. Durante el tiempo de realización del proyecto se trabaja con, al menos, un colaborador por parte del usuario, que conozca ya la representación del conocimiento y la estructura de la base de conocimientos, a largo plazo estará en posibilidad de mantener la base de conocimientos.

El último subproceso del desarrollo del Sistema Experto, se desarrolla bajo la dirección del cliente. El ingeniero del conocimiento es informado sobre problemas y fallos.

La aplicación

La aplicación del Sistema Experto debería realizarse lo antes posible, buscando:

- Obtener una forma rápida de retroceder (para una posterior ampliación) y
- obtener lo antes posible el beneficio del sistema.

Esto puede lograrse poniendo en aplicación sólo una parte del conocimiento experto, sin embargo, no se deben olvidar los riesgos de una aplicación demasiado precipitada. Por un lado si el usuario obtiene mensajes del sistema antes de que la solución de un problema esté bien definida e implementada, puede perder prematuramente la confianza en el desarrollo. Por otro lado también puede ser decepcionante el que las respuestas del sistema sean demasiado generales o ambiguas y el usuario hubiera podido llegar a ellas por sí mismo. En el momento de la aplicación, el sistema debe ser tan potente, que el usuario se convenza del beneficio personal del sistema, a pesar de las dificultades inherentes al arranque.

El mantenimiento

El mantenimiento del sistema se refiere a la ampliación regular del mismo. Esto es, adaptaciones que no modifican "demasiado" al sistema.

Considera:

1. Actualización de la base de conocimientos.
2. Determinación de fallos y limitaciones.
3. Realización de cambios por 1 y 2.

2.4 Herramientas de Software

Existen una gran variedad de herramientas de software (lenguajes y sistemas de programación), mediante los cuales pueden desarrollarse los Sistemas Expertos, sin embargo se han desarrollado lenguajes especialmente adecuados para dicho fin. En la mayoría de los casos están subordinados a los diferentes paradigmas de programación.

Veamos un esbozo de los principales paradigmas de programación:

Paradigma de procedimiento: Bajo esta definición se entiende la programación algorítmica. Un programa consta de la descripción del desarrollo del procesamiento.

Paradigma declarativo de programación: También es conocido como paradigma basado en reglas. Es la descripción de un problema con hechos y reglas. En un cumplimiento estricto del paradigma, no puede formularse desarrollo de procedimiento.

Paradigma funcional de programación: La solución de un problema se consigue con la consecuente aplicación de funciones. Una función ejecutada (suministra) como resultado un valor, que es reprocesado por la función anterior. En el cumplimiento consecuente del paradigma funcional de programación, no se requerirá ninguna asignación de valor en un programa.

Programación orientada al objeto: En la programación orientada al objeto se describen objetos, éstos envían información a otros objetos y aplican métodos a la recepción de información. Éstos métodos pueden enviar a su vez informaciones, o provocar también modificaciones en el estado del objeto.

Los lenguajes que han encontrado mayor aplicación en el desarrollo de Sistemas Expertos y Shells son PROLOG y LISP.

PROLOG es un lenguaje declarativo, en el que el conocimiento puede representarse fácilmente como reglas de producción.

LISP es un lenguaje funcional, que al igual que PROLOG presenta las siguientes ventajas:

Su principal orientación se centra más en el procesamiento de símbolos y estructuras que en el cálculo de cifras, además no hay diferenciación entre estructuras de datos y de programas.

Otra ayuda procede de los Shells, que pueden encontrarse en el mercado con distintas capacidades, estos se basan alternativamente en PROLOG o en LISP.

Como entornos de hardware para el desarrollo de Sistemas Expertos se utilizan a menudo máquinas "dedicadas". Su microcódigo se orienta al lenguaje de programación.

Shells

A lo largo de los capítulos anteriores hemos utilizado la palabra Shell, sin haberla definido. Veamos qué es.

Es una herramienta utilizada por el Ingeniero del conocimiento en su trabajo; los Shells contienen ya componentes de un Sistema Experto, excepto la base de conocimientos, algunos son sólo herramientas para la estructuración del saber, es decir programas para representar las relaciones del conocimiento, otros disponen ya de uno o más mecanismos de representación del conocimiento y, en algunos

casos, un mecanismo de inferencia, otros ofrecen, a su vez, posibilidades de control más amplias o ayudan en la configuración de la interfaz.

Los primeros Shells surgen como una consecuencia de la creación de un Sistema Experto, al extraer toda la información referente al caso de aplicación en particular para el cual fue escrito. Este resto recibe el nombre de Shell, es decir un caparazón o concha para un Sistema Experto, de tal suerte que contiene entornos de programación y otros métodos auxiliares, como por ejemplo, estructuras para la representación del conocimiento, mecanismos de inferencia, o apoyo para el componente explicativo; en ocasiones incluso se tienen estructuras de control que permiten influir en el procesamiento del conocimiento.

Los componentes más deseados en un Shell son:

- Formalismo para la representación del conocimiento.
- Medios de estructuración para la base de conocimientos.
- Mecanismo de inferencia.
- Interfaz de usuario, adaptado a los requisitos en la confección del sistema, en la estructuración y ampliación de la base de conocimientos y en el uso final.
- Apoyo en la creación de un componente explicativo.
- Mecanismos para la comprobación de consistencia, búsqueda de errores y ampliación de la base de conocimiento.
- Ayudas en la adquisición de conocimientos.

El uso de un Shell con estas características, permite al encargado del desarrollo concentrarse plenamente en la confección de la base de conocimientos, con lo que se reduce el esfuerzo de desarrollo, además de que el uso de un Shell no requiere profundo dominio de un lenguaje de programación.

Sin embargo el uso de Shells también conlleva algunas desventajas:

- A pesar de la flexibilidad que los Shells ofrecen, el encargado del desarrollo está ligado a los conceptos básicos del Shell, esto es, el Shell determinado no puede ser la base de cualquier Sistema Experto.
- El encargado del desarrollo debe aprender el lenguaje y la semántica del Shell elegido, por lo que la ventaja de que no hacen falta más que conocimientos de programación, es relativa.
- La introducción de funciones nuevas o la modificación de antiguas sólo es posible con Shells que ofrezcan una interfaz para un determinado lenguaje de programación.

A pesar de estas desventajas, para una estructuración rápida de un Sistema Experto, es recomendable comenzar con un Shell, para poder crear un prototipo de demostración.

La gama de los Shells disponibles va desde los más sencillos, que sólo ofrecen ayuda en la representación del conocimiento, hasta los Shells más complejos,

que ponen a disposición distintas formas de representación del conocimiento y diferentes estrategias de inferencia.

La potencia y el radio de acción de un Shell, dependerá en gran medida también del hardware utilizado, PC, estación de trabajo, ordenador de gran tamaño, etc. Además de que deberá considerarse la portabilidad del Shell elegido para correr en una u otra plataforma.

3 Planteamiento del desarrollo del Sistema Experto

Análisis del problema: En este punto nuestra atención estará centrada en validar la viabilidad de la realización de este proyecto, con una técnica de Sistemas Expertos. A lo largo del mismo se aplicarán algunos criterios críticos y se comenzará a esbozar el dominio de aplicación del sistema.

Tema propuesto: Sistema Experto de Diagnóstico Médico Veterinario para aves de corral.

Objetivo: Desarrollar un sistema económico y portable que auxilie el diagnóstico de enfermedades para aves de corral (pollos), en apoyo a los avicultores que por alguna circunstancia no tengan acceso a un experto veterinario, e incluso que auxilie a médicos veterinarios en esta misma labor.

Definición del problema:

Existen criadores de aves de corral (pollos) que por diversas razones no pueden utilizar los servicios de un médico veterinario, con la prontitud y eficacia que desearían.

Método

Para la resolución de esta problemática se plantea el desarrollo de un sistema, que provea a los citados avicultores con una herramienta que les permita determinar el estado de salud de sus aves y de esta forma asegurar la supervivencia y reproducción de las mismas.

Resultados esperados

- Conceptualizar en una forma de representación del conocimiento, la experiencia de un experto veterinario en el diagnóstico de enfermedades de aves de corral.
- Obtener un producto de software que se constituya en una herramienta útil y accesible, para avicultores e incluso para médicos veterinarios.

3.1 Análisis del problema

Se desea desarrollar una herramienta que apoye en el diagnóstico de enfermedades de aves de corral, particularmente pollos, se requiere con una interfaz de usuario fácil de utilizar, también se requiere de manera ideal que posea un módulo de mantenimiento a la base de conocimiento, que el propio usuario o en su defecto el experto, pueda alimentar y con un componente explicativo claro.

3.1.1 Forma de trabajo del experto

Antes de continuar con el análisis aplicaremos los llamados "Criterios críticos", si uno de ellos se cumple el proyecto del Sistema Experto no puede llevarse a cabo.

No existe ningún experto.

No tiene tiempo; es decir, es imprescindible para otras labores.

No está motivado.

No se cumple ninguno de los criterios anteriores, dado que se cuenta con un experto veterinario, el cual tiene gran interés en que se desarrolle el Sistema Experto, además de que conoce y confía en la técnica de Sistemas Expertos.

El experto veterinario, fundamenta su diagnóstico en el análisis clínico; esto es, a partir de una observación (inspección ocular) del paciente, en este caso de los pollos; tanto de forma colectiva (mediante un muestreo) como individual (dentro de la misma muestra); determina la posibilidad de que una de las aves, varias de ellas o incluso la parvada completa pueda estar infectada con alguna enfermedad, para posteriormente realizar análisis más rigurosos, como estudios de laboratorio por ejemplo y de acuerdo a las características del padecimiento, proceder con un tratamiento, o llevar a cabo alguna acción como aislar a las aves infectadas, o en un caso extremo aniquilar a la parvada entera.

3.1.2 Tipo y alcance del conocimiento utilizado para la solución de problemas

La función primordial del Sistema Experto, es el diagnosticar a través del análisis clínico. Al mismo tiempo se desea que el sistema sea capaz de justificar el diagnóstico, a través de su componente explicativo y ofrecer alternativas de solución (tratamiento) en caso de encontrarse algún padecimiento en las aves auscultadas.

El dominio de la aplicación es finito, se tienen contempladas las enfermedades de mayor importancia que atacan a los pollos, además de un cuadro sintomático para cada padecimiento, e identificadas las zonas del ave en las que se manifiestan dichos síntomas. Más adelante se hablará del dominio de aplicación con más detalle.

También se cuenta con criterios de solución de conflictos, dado que existen síntomas comunes en algunas enfermedades.

En el manejo del conocimiento no se involucra el cálculo matemático y tampoco se procesa información de carácter confidencial.

3.1.3 Entorno de aplicación del Sistema Experto

El sistema a desarrollarse no interactuará con otros sistemas; esto es, no habrá ningún flujo de información desde o hacia el sistema con otros sistemas.

Se desea que realice un diagnóstico inmediato, luego del análisis clínico, pues la calidad de la respuesta es vital para el aseguramiento de la permanencia de la parvada.

No se desean inversiones en equipos adicionales, pretendiéndose que la inversión en software sea también mínima, por lo que se desea que el sistema trabaje sobre la plataforma actual; en estos momentos se cuenta con computadoras personales basadas en tecnología intel, cuyo sistema operativo es Windows 95 y tienen instalada la suite de productividad de Microsoft Office, dichas computadoras personales no son parte de una red local.

Para este punto también se tienen tres "Criterios críticos"

1. No puede alcanzarse la calidad requerida de la solución.
2. La incorporación en el proceso de trabajo resulta imposible.
3. La creación de un entorno de trabajo, resulta económicamente costosa.

Se considera que no se cumple ninguno de los criterios anteriores; dado que, la calidad de la solución estará sustentada por el apego a la metodología de desarrollo de Sistemas Expertos que hemos elegido, asimismo se buscará en todo momento satisfacer los requerimientos del usuario, apoyándonos en una relación muy estrecha con éste último, se requiere también que el producto le sea útil por mucho tiempo, para lo cual será necesario un excelente servicio de mantenimiento, una buena documentación, un confiable sistema de recuperación (a este respecto no se tiene definido un esquema de respaldo periódico, sino hasta que la base de conocimientos del sistema cambie, además de contar con al menos dos copias del sistema) y, desde luego, un impecable soporte en el momento que se requiera.

En esta aplicación, nos aseguraremos de que lo que se entregue esté completo, sea oportuno, no tengo errores, sea confiable, útil y por supuesto estable.

Por otro lado el desarrollo estará orientado a ser un proceso similar a la naturaleza del diagnóstico, como el experto lo realiza y se buscará que la solución que proveeremos se incorpore al entorno de trabajo con una inversión mínima.

3.1.4 Usuario del Sistema Experto

Inicialmente será nuestro experto veterinario y su equipo de trabajo, el cual desde luego tiene que estar familiarizado con el lenguaje necesario para comunicarse con el sistema, sin embargo se pretende desarrollar una aplicación que pueda ser ejecutada y utilizada por todo aquel al que le resulte útil, para lo cual se desea desarrollar una interfaz de usuario amigable, en un entorno lo más similar posible a la naturaleza de la problemática y su solución.

Dado que la solución que proveerá nuestro sistema está basada en primera instancia por el análisis clínico; será necesario que el sistema cuente con una interfaz de usuario, que permita la incorporación a este proceso en un ambiente cercano a la naturaleza misma de tal proceso; esto es, se requiere de una interfaz que permite emular la forma de trabajo del experto, por lo tanto en primera instancia dado que basa su diagnóstico en el análisis clínico, requerimos de una interfaz, que permita al usuario visualizar al ave auscultada, como lo haría el experto veterinario en su mente, distinguiendo las diferentes zonas del cuerpo y que al tiempo que las vaya recorriendo, pueda detectar los síntomas que se presenten y almacenándolos en un elemento tal que permita visualizarlos, como ayuda para que el usuario los tenga presentes y no los elija mas de una vez, o en su caso elimine aquellos síntomas que elija por error.

Luego de este análisis clínico, la interfaz de usuario deberá proveer de un elemento, que permita al usuario activar la máquina de inferencia, la cual deberá hacer una manipulación de la base de conocimientos que de como resultado un diagnóstico.

También se desea una interfaz que permita al usuario alimentar y mantener la base de conocimientos, sin ayuda del ingeniero de conocimiento de manera ideal.

En este punto se tienen los siguientes "Criterios críticos":

1. La diferencia respecto a conocimiento especializado entre experto y usuario, es demasiado grande.
2. Los medios necesarios para la solución del problema, no pueden ponerse a disposición de cada usuario.

De acuerdo a lo que se planea, la aplicación a desarrollar proveerá al usuario de un entorno tal que éste será capaz de obtener por sí mismo, la información necesaria para asegurar la supervivencia de su parvada; además tal aplicación deberá ejecutarse con requerimientos mínimos de software y hardware, por lo que el sistema será relativamente fácil de adquirir. Más adelante se tocarán estos puntos con más detalle.

Por lo anterior se considera que no se cumple ninguno de los criterios críticos, en este punto.

3.1.5 Costo y beneficio del Sistema Experto

Si bien es cierto en este punto aún no se cuenta con la suficiente información para hacer un análisis costo / beneficio preciso; si es posible hacer algunas estimaciones basados en la información que se tiene.

Costos:

- Durante el desarrollo:
 - Ingeniero del conocimiento.
 - Programador.
 - Experto.
 - Equipo de desarrollo (mobiliario, energía eléctrica, hardware, papelería, inmueble, licencia de desarrollo, software, etc.).

Después del desarrollo:

- Mantenimiento del Sistema.
- Software y hardware sobre la que correrá la aplicación.
- Licencias y actualizaciones.

Beneficios:

- No siempre se cuenta con un experto veterinario a la mano cuando se necesita, sobre todo en el caso de comunidades alejadas, por lo que el beneficio para este sistema se estimará, por un lado basándonos en los ahorros al no tener que consultar a un experto en primera instancia y después de los beneficios de asegurar la supervivencia de la parvada.
- Traslado desde la comunidad en la que se encuentra la granja hasta donde se encuentre el médico veterinario, o en su caso el costo de una visita a domicilio, además del costo que implicaría la disponibilidad las 24 horas del día los 365 días del año.
- El valor de contar con el conocimiento del experto almacenado y el aseguramiento de la permanencia y reproductibilidad del mismo.

Por el momento dejaremos a este nivel nuestro análisis costo / beneficio, nos hemos concentrado en identificar las variables involucradas en este análisis, sin atender los montos en dinero de tales variables; esto con el fin de mantener la vigencia del presente trabajo. En el apéndice C se retoma esta información planteándose una estimación, basada en costos reales a manera de ejemplo.

3.1.5 Duración del análisis del problema

Una vez finalizada la primera parte del desarrollo del Sistema Experto, se considera que las características de la problemática son adecuadas para ser tratadas con una técnica de Sistemas Expertos, por lo cual se decide continuar con el desarrollo.

Como parte complementaria del paso **análisis del problema** presentamos un cronograma del proceso de desarrollo completo en la figura 3.1.6.

Tarea	Duración	Responsable
Análisis del problema	7 días	Ingeniero del conocimiento / Experto veterinario
• Forma de trabajo del experto	2 días	Ingeniero del conocimiento / Experto veterinario
• Tipo y alcance del conocimiento utilizado para la solución de problemas	2 días	Ingeniero del conocimiento / Experto veterinario
• Entorno de aplicación del Sistema Experto	1 día	Ingeniero del conocimiento / Experto veterinario
• Usuario del Sistema Experto	1 día	Ingeniero del conocimiento / Experto veterinario
• Costo y beneficio del Sistema Experto	1 días	Ingeniero del conocimiento
Planificación del sistema	23 días	Ingeniero del conocimiento / programador / Experto veterinario
• Primer paso en la planificación del sistema: "Análisis de requerimientos"	6 días	Ingeniero del conocimiento / Experto veterinario
• Segundo paso en la planificación del sistema: "Proyecto"	5 días	Ingeniero del conocimiento / Experto veterinario
• Tercer paso en la planificación del sistema: "Realización parcial del proyecto"	8 días	Ingeniero del conocimiento / programador / Experto veterinario
• Cuarto paso en la planificación del sistema: "Validación del proyecto"	1 día	Ingeniero del conocimiento / programador / Experto veterinario
• Planificación del sistema "sin final"	3 días	Ingeniero del conocimiento
Desarrollo para alcanzar la madurez de la aplicación	5 días	Ingeniero del conocimiento / programador / Experto veterinario / Usuario voluntarioso
• Adquisición de conocimientos	2 días	Ingeniero del conocimiento / Experto veterinario
• Estrategia de ampliación de la base de conocimientos	1 día	Ingeniero del conocimiento / programador / Experto veterinario
• Prueba y ejecución del programa piloto	2 días	Programador /Experto veterinario /Usuario voluntarioso
Aplicación y mantenimiento	7 días	
• Aplicación	7 día	Experto veterinario / Usuario voluntarioso
• Mantenimiento		Dependiendo de la complejidad

Figura 3.1.6

3.2 Planificación del sistema

En este punto la atención estará centrada en obtener un modelo lógico del sistema; dado que en el paso anterior **análisis del problema**, ya se ha decidido atacar esta problemática con la técnica de Sistemas Expertos.

De igual forma en esta etapa se pretende obtener la estructura de nuestra base de conocimientos, en una forma de representación del conocimiento, la estrategia de inferencia, la interfaz de usuario y el componente explicativo; esto es, prácticamente toda la estructura de nuestro Sistema Experto, aunque únicamente a un nivel de prototipo.

3.2.1 Primer paso en la planificación del sistema: "Análisis de requerimientos"

De acuerdo a nuestro análisis del problema hecho en el punto anterior; la función principal a solucionar es la del diagnóstico; veamos a manera de diagrama de bloques en la figura 3.2.1.a en qué consiste este proceso.

El proceso de diagnóstico consiste de tres subprocesos; en primera instancia lo que se conoce como **análisis clínico**: observación del paciente, a través de la cual se identifica la sintomatología; el siguiente subproceso es lo que hemos llamado **identificación del padecimiento o enfermedad** la cual se realiza en base a la sintomatología obtenida en el subproceso anterior; finalmente como último subproceso, los **resultados**: toda la información que se obtiene luego de aplicar la estrategia de inferencia, y los criterios de solución de conflictos.

Diagnóstico

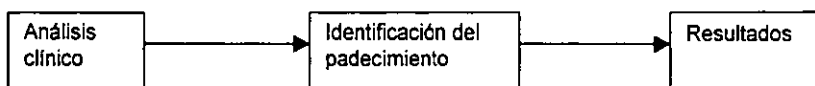


Figura 3.2.1. a

Modelo lógico

Con el objeto de definir el nuestro veamos como se está realizando en este momento la tarea del diagnóstico (estado ES), para después plantear la solución con nuestro Sistema Experto (estado DEBE).

Estado "ES"

En este momento el experto veterinario, basa su diagnóstico en un análisis clínico a través del cual identifica el padecimiento, para posteriormente emitir un resultado; esto es, aplica su proceso de diagnóstico. Veámoslo con mas detalle:

En el subproceso de **análisis clínico**: El experto observa al ave en cuestión, revisándola minuciosamente, analiza las distintas partes de su cuerpo y su comportamiento, identificando en base a su experiencia, la sintomatología que se presenta en cada una. El experto va almacenando en su memoria tales síntomas.

En el subproceso **identificación del padecimiento**: El experto busca que enfermedades se caracterizan por los síntomas que presenta el ave en cuestión y en base a su criterio de solución de conflictos y su mecanismo de inferencia, identifica el o los padecimientos que aquejan a dicho animal.

Finalmente en el subproceso **resultados**: El experto emite su diagnóstico, tratamiento, sistemas del ave afectados, etc.

Veamos ahora como resolverá el problema nuestro Sistema Experto.

Estado "DEBE"

Nuestro Sistema Experto también aplicará el proceso de diagnóstico. La primera diferencia que se observa en este proceso es la obvia presencia del mismo, ahora tenemos al usuario, al Sistema Experto y al paciente.

En el subproceso de **análisis clínico**: el usuario ejecutará la aplicación, la cual debe proveerle de un medio que le permita visualizar al ave zona por zona, con un elemento que le permita fácilmente seleccionar y al mismo tiempo visualizar su selección de síntomas hasta que ésta concluya, luego de lo cual el subproceso se dará por completado.

En el subproceso **identificación del padecimiento**: el usuario solicitará al sistema que procese la información, por lo cual el sistema deberá contar con un elemento de programa, que le permita emular el mecanismo de inferencia y el solucionador de conflictos, apoyándose en la base de conocimientos y las reglas involucradas, para determinar el o los padecimientos presentes en el ave.

Finalmente en el subproceso **resultados**: el sistema mostrará al usuario todo el cúmulo de información propio del diagnóstico, a través de elementos de programa que le permitan visualizar fácilmente tal información.

Como se observa las formas de proceder del experto y del sistema son muy similares, que es precisamente lo que se busca con la aplicación de una técnica de Sistemas Expertos; las diferencias principales son: capacidad de memoria, velocidad de búsqueda y procesamiento, la posibilidad de que cualquier usuario pueda obtener un diagnóstico, poder multiplicar fácilmente al sistema lo que no ocurre con el experto.

El modelo lógico de nuestro sistema en realidad es muy sencillo; como se muestra en la figura 3.2.1.b, se trata de una relación entre las zonas del cuerpo del ave,

con las enfermedades que pueden atacarlos, en base a los síntomas que presentan.

Modelo lógico

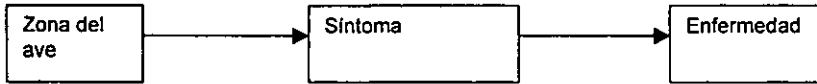


Figura 3.2.1.b

En este punto hemos hablado de las zonas del ave, los síntomas que puede presentar y las enfermedades que puede padecer; sin embargo no hemos hablado de manera particular de tales cuestiones; en los apéndices A y B encontraremos tal información en forma detallada.

Forma de Representación del Conocimiento

En este punto, y en base a las características de la información que estaremos manejando, hemos elegido a los frames como forma de representación del conocimiento, de tal suerte que nuestro sistema puede caracterizarse por la siguiente representación:

Primeramente tenemos al frame genérico Enfermedad mostrado en la figura 3.2.1.c.

Enfermedad		
Nombre enfermedad	Valor	
Síntomas_presentes	Valor	

Figura 3.2.1.c

Los frames instancias correspondientes al frame anterior, tendrán los slots que los caracterizan y el slot Es_una mediante el cual serán identificadas como instancias o subclases con el valor "Enfermedad" en la figura 3.2.1.d se muestra esta caracterización.

Enfermedad _i		
Es una	Enfermedad	
Nombre enfermedad	Valor	
Descripción	Valor	
Edad de Presentación	Valor	
Causa	Valor	
Transmisión	Valor	
Mortalidad	Valor	
Diagnóstico	Valor	
Diagnóstico diferencial	Valor	
Tratamiento	Valor	
Aparato o sistema afectado	Valor	

Figura 3.2.1.d

Para el caso de estos frames instancias los valores que puede tomar Enfermedad_i son lo correspondientes a las enfermedades que se muestran en el apéndice B y que son: Newcastle, Bronquitis infecciosa (BI), Laringotraqueitis, Coriza infecciosa, Colibacilosis (infección por e. Coli), Mycoplasma Gallisepticum (MG), Mycoplasma Synoviae (MS), Pasteurelosis, Aspergilosis pulmonar, Sind cabeza hinchada e Influenza aviar (IA). Para el caso de todas las características de las enfermedades también se muestran en dicho apéndice.

En segundo lugar tenemos en la figura 3.2.1.e a las Clases Zona_j; que representa a las distintas partes del cuerpo en que dividimos al ave para fines del análisis clínico, las cuales son frames genéricos.

Zona _j		
Nombre zona	Valor	
Síntoma _k	Valor _k	
Síntoma _i	Valor _i	
...		
...		
Síntoma _r	Valor _r	
slot	slot value	facet

Figura 3.2.1.c

Para el caso de los frames instancias, los valores que puede tomar Zona_j son los siguientes: Zona torax, Zona cabeza, Zona cola, Zona alas, Zona cuello, Zona patas, Zona abdomen, Zona cola y Zona dorso. Los frames instancias se muestran en la figura 3.2.1.d.

Zona _j		
Es_una	Zona _j	
Síntoma _k	Valor _k	
Síntoma _i	Valor _i	
...		
...		
Síntoma _r	Valor _r	
slot	slot value	facet

Figura 3.2.1.d

Finalmente tenemos el frame genérico correspondiente a los síntomas que pueden estar presentes en el ave se muestra en la figura 3.2.1.e.

Síntoma _k		
Nombre	Valor	
Enfermedad asociada	Valor	
slot	slot value	facet

Figura 3.2.1.e

Los frames instancias del frame anterior son los correspondientes a todas los posibles síntomas (los cuales se muestran en el apéndice A) y su correspondiente enfermedad asociada, su estructura es la mostrada en la figura 3.2.1.f.

Síntoma _k		
Es_un	Síntoma _k	
Nombre	Valor	
Enfermedad asociada	Valor	Enfermedad (Síntomas_presentes= Síntomas_presentes +1)
slot	slot value	facet

Figura 3.2.1.f

Mecanismo de inferencia

El mecanismo de inferencia funcionará de la siguiente manera: en primera instancia, mediante la interfaz de usuario, éste identificará los síntomas presentes en el ave, lo que generará un conjunto de síntomas, relacionados con una o más enfermedades, de tal suerte que el conjunto de síntomas nos generará un conjunto de enfermedades; para el caso de que este contenga un solo elemento será inmediata la generación del resultado; sin embargo para el caso de que dicho conjunto contenga más de un elemento, entrarán en acción los criterios de solución de conflictos.

Criterios de solución de conflictos

Para nuestro Sistema Experto, tenemos dos criterios de solución de conflictos; como se mencionó, cuando la máquina de inferencia genera un conjunto de enfermedades con más de un elemento, debe aplicarse un criterio que nos permita decidir sobre la elección de la enfermedad.

El primer criterio es el grado de mortalidad de las enfermedades involucradas; nos es más benéfico, determinar que una ave padece de una enfermedad mortal aislándola y/o aplicándole un tratamiento que ayude a salvar al resto de la parvada, que elegir de dicho conjunto la enfermedad que no es mortal; con lo que no tomaríamos las precauciones necesarias y pondríamos en peligro a nuestra parvada, si el diagnóstico es erróneo el impacto no es comparable.

Si después de aplicar el primer criterio, todavía tenemos un conjunto de enfermedades de más de un elemento aplicaremos el segundo; el cual es muy simple, dado que elige aquella enfermedad que tenga mas síntomas presentes en el conjunto de síntomas.

¿Y si con todo y esto aún tenemos un conjunto con más de un elemento?

De acuerdo a la información proporcionada por el experto; estos dos criterios son suficientes para resolver cualquier caso posible, por el momento continuaremos con nuestro desarrollo aceptando ésta suposición, la cual se validará a lo largo del mismo desarrollo y en su caso se buscará un criterio extra que nos permita solucionar tal conflicto si éste existiera.

Veamos esto de forma gráfica en la figura 3.2.1.g.

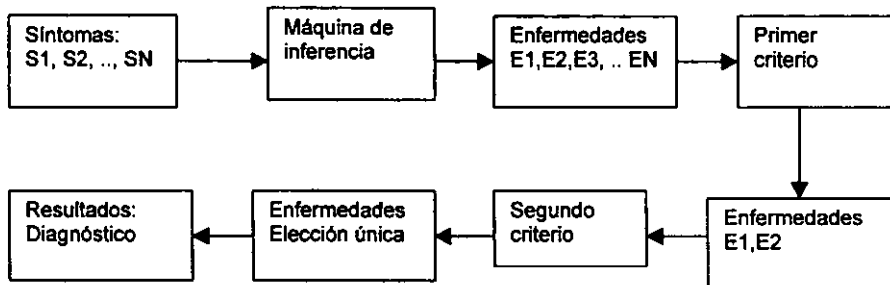


Figura 3.2.1.g

3.2.2 Segundo paso en la planificación del sistema: "Proyecto"

En este punto nos ocuparemos en definir los elementos del sistema y su función dentro del mismo, por ello en primer lugar los listaremos para posteriormente ir definiendo su función.

En todo Sistema Experto tenemos: base de conocimientos, mecanismo de inferencia, componente explicativo, interfaz de usuario y componente de adquisición de conocimiento.

Base de conocimientos

La base de conocimientos es el elemento más importante de nuestro Sistema Experto, su función será la de almacenar el conocimiento de nuestro experto; los hechos, las reglas, los procedimientos y los métodos necesarios para resolver los problemas como lo haría él mismo; de tal suerte que en este momento anticipamos la necesidad de una herramienta de desarrollo que soporte el manejo de frames, que es la forma de representación del conocimiento que se ha elegido, además de soportar el manejo de reglas, procedimientos y métodos propios del sistema.

Los hechos: Estos los vemos reflejados en los apéndices A y B, las zonas del ave, los síntomas por zona y las enfermedades. Veamos los tipos de datos que requeriremos para almacenar esta información: En las figuras 3.2.2.a, 3.2.2.b, 3.2.2.c, 3.2.2.d, 3.2.2.e y 3.2.2.f.

Enfermedad	
Nombre enfermedad	Alfanumérico de longitud variable
Síntomas presentes	Entero

Figura 3.2.2.a

Enfermedad _i	
Es una	Alfanumérico de longitud variable
Nombre enfermedad	Alfanumérico de longitud variable
Descripción	Alfanumérico de longitud variable
Edad de Presentación	Alfanumérico de longitud variable
Causa	Alfanumérico de longitud variable
Transmisión	Alfanumérico de longitud variable
Mortalidad	Alfanumérico de longitud variable
Diagnóstico	Alfanumérico de longitud variable
Diagnóstico diferencial	Alfanumérico de longitud variable
Tratamiento	Alfanumérico de longitud variable
Aparato o sistema afectado	Alfanumérico de longitud variable

Figura 3.2.2.b

Zona_j

Nombre zona	Alfanumérico de longitud variable
Síntoma _k	Alfanumérico de longitud variable
Síntoma _l	Alfanumérico de longitud variable
...	
...	
Síntoma _r	Alfanumérico de longitud variable

Figura 3.2.2.c

Zona_j

Es_una	Alfanumérico de longitud variable
Síntoma _k	Alfanumérico de longitud variable
Síntoma _l	Alfanumérico de longitud variable
...	
...	
Síntoma _r	Alfanumérico de longitud variable

Figura 3.2.2.d

Síntoma_j

Nombre	Alfanumérico de longitud variable
Enfermedad asociada	Alfanumérico de longitud variable

Figura 3.2.2.e

Síntoma_j

Es_un	Alfanumérico de longitud variable
Nombre	Alfanumérico de longitud variable
Enfermedad asociada	Alfanumérico de longitud variable

Figura 3.2.2.f

Los métodos y procedimientos: Serán los elementos del sistema que nos permitirán elegir la zona del ave y los síntomas por región de la misma.

Las reglas: Una vez elegidos los síntomas por región, se aplicarán las reglas que nos ayudarán a determinar el diagnóstico; la elección de la sintomatología nos generará - como es obvio- una lista de hechos, los cuáles serán mostrados al usuario en una caja de diálogo y al mismo tiempo estarán presentes en la memoria de trabajo de nuestro Sistema Experto. A partir de que el usuario pida al sistema que procese la información se aplicarán las reglas siguientes:

Regla # 1:

Si existe al menos un síntoma en la memoria de trabajo entonces aplica Regla # 2 si no aplica regla # 3.

Regla # 2

Si síntoma n está presente entonces incrementa contador de síntomas presentes en el frame "Enfermedad" y elimina el síntoma n de la memoria de trabajo del sistema, aplica regla # 1.

Regla # 3 Una vez generado el universo de síntomas, aplica los criterios de solución de conflictos y diagnóstica.

Mecanismo de inferencia

Como se ha mencionado, es el elemento del sistema que nos permite manipular la base de conocimientos, para obtener los resultados de la aplicación del sistema de tal forma que de acuerdo al funcionamiento de nuestra máquina de inferencia, también podemos anticipar que requeriremos de una herramienta de desarrollo que soporte el encadenamiento hacia adelante, dado que es el mecanismo de inferencia que utiliza nuestro sistema.

Cabe mencionar que a pesar de que las reglas son parte de la base de conocimientos, se aplican a través del mecanismo de inferencia.

Componente explicativo

Como se ha mencionado, el componente explicativo es el elemento del sistema que nos permite comunicar al usuario, la estrategia de solución y el porqué de la decisión tomada, como se mencionó en el primer proceso de desarrollo; "análisis del problema", se desea que este componente sea claro y de fácil manejo para el usuario, por lo que se constituye en otro elemento a considerar para la elección de la herramienta de desarrollo; específicamente se requieren cajas de texto, que incorporen barras de desplazamiento, y que permitan visualizar el total de la información que resulte de las consultas, en un ambiente amigable.

Interfaz de usuario

La interfaz de usuario, es el componente a través del cual el usuario interactuará con el sistema, estando en ella los elementos necesarios para que la interacción sea posible. Como se mencionó en el primer proceso de desarrollo del sistema "análisis del problema", se desea una interfaz que sea tan amigable como lo permitan las herramientas de desarrollo actuales, buscando que la interacción del usuario con el sistema sea de una forma natural.

Dado que para emular el proceso de diagnóstico, se requiere contar con un elemento de programa que permita visualizar al ave de forma gráfica, así como las regiones de su cuerpo, a través de las cuales el usuario pueda recorrer y

seleccionar los síntomas que la citada ave presente; se plantea la necesidad de contar con una caja de diálogo, la cual indicará de qué región se trata y el universo de posibles síntomas para alguna región, al seleccionar tales síntomas se asertarán en forma de hechos en otra caja de diálogo, con el fin que el usuario tenga presentes qué síntomas ha seleccionado.

Se observa que específicamente se requiere de un ambiente, gráfico; que cuente con manejo de imágenes, botones de comando, cajas de diálogo, etc. Para el caso del manejo de imágenes es deseable tener como característica el concepto de hiper-región. Éstos son otros elementos a considerar para la elección de la herramienta de desarrollo.

Componente de adquisición de conocimiento

Es el elemento que nos permite dar mantenimiento a la base de conocimiento, por lo que se buscará que permita al usuario interactuar de una manera fácil y natural con dicha base; esto es, se requiere de cajas de diálogo y de texto, a través de las cuales se pueda alimentar la base de conocimiento de manera transparente para el usuario.

Una vez que hemos definido los componentes básicos de nuestro sistema, estamos listos para plantear el prototipo. Antes de realizarlo plantearemos el diseño de las pantallas del sistema; como se mencionó en el primer paso en la planificación del sistema "Análisis de requerimientos"; el proceso de diagnóstico consta de tres subprocesos: análisis clínico, identificación del padecimiento y resultados.

De tal suerte que utilizaremos una pantalla para el análisis clínico, la identificación del padecimiento será consecuencia de la activación del mecanismo de inferencia, usando una segunda pantalla que mostrará los resultados.

Pantalla para el análisis clínico

En la pantalla para el análisis clínico; se desplegará la imagen de un pollo, la cual tendrá definidas las regiones de su cuerpo, de un modo tal que el usuario pueda fácilmente identificarlas y al mismo tiempo debe existir un elemento de software que muestre el nombre de la zona y los posibles síntomas asociados, por ejemplo una caja de texto, de un modo tal que el usuario pueda elegir los síntomas que observe presentes en el ave que esté auscultando; los síntomas seleccionados deberán aparecer como otro elemento de la pantalla por ejemplo una caja de texto, que tenga una etiqueta "síntomas" y muestre tales síntomas, también debemos proveer de un elemento que permita al usuario eliminar los ya seleccionados; o bien retornar al inicio o incluso terminar la aplicación, una vez que el usuario termine de elegir los síntomas, deberá existir otro elemento en la pantalla que permita accionar el mecanismo de inferencia, por ejemplo un botón de comando.

Pantalla de resultados

En la pantalla de resultados debe mostrarse toda la información luego del diagnóstico: nombre de la enfermedad, descripción, edad de presentación, causa, transmisión, mortalidad, diagnóstico, diagnóstico diferencial, tratamiento y aparato o sistema afectado; tal información puede mostrarse en cajas de texto con su respectiva etiqueta, buscando una presentación agradable y sobre todo, que se muestre la información completa, en la misma pantalla debe contarse con un elemento que permita regresar a la pantalla anterior, para realizar otro diagnóstico.

3.2.3 Tercer paso en la planificación del sistema: "Realización parcial del proyecto"

En el punto anterior observamos la definición de los componentes de nuestro Sistema Experto y su función general a nivel de diseño; nos auxiliaremos de dichas observaciones para elegir la herramienta de desarrollo en primera instancia, de nuestro prototipo del sistema, a través del cual podrán verse más específicamente tales elementos; así como su funcionamiento e interacción.

Preferentemente se buscará utilizar un Shell ya existente, antes que crear uno propio; también buscará que dicho Shell posea las características adecuadas a las necesidades de desarrollo de nuestro sistema.

Elección de la herramienta de desarrollo

Antes de elegirla es importante tener como referencia, los criterios necesarios para hacer esta elección, para lo cual nos auxiliaremos de los siguiente aspectos:

Representación del conocimiento: Esta característica se refiere a la capacidad de la herramienta de soportar diferentes paradigmas de programación; basado en reglas, orientado a objetos, procedural, etc.

Portabilidad: Esta alude a la capacidad de la herramienta de poder instalarse en diferentes computadoras, sin necesidad de modificar el código y de su capacidad de poder generar aplicaciones que también puedan ejecutarse sobres diferentes plataformas.

Integración y extensibilidad: Esta característica se refiere a la capacidad de la herramienta, de poderse comunicar con otros lenguajes a través de protocolos y subrutinas.

Desarrollo interactivo: Toca lo referente a las facilidades que ofrece la herramienta para el desarrollo como son: ayudas para la depuración de programas, ayuda en línea, editor integrado, interfaces, menús despleables, múltiples ventanas, soporte para desarrollo modular, posibilidad de particionar la base de conocimientos, mecanismos de validación dinámicos y estáticos para

valores de slots y argumentos de funciones, además de análisis semántico de patrones de reglas.

Documentación: Lo relativo a la disponibilidad de manuales de referencia y guías de usuario.

Por otro lado también han de considerarse las necesidades específicas que ya se han estado mencionando en puntos anteriores:

- Manejo de frames como forma de representación del conocimiento incorporada.
- Manejo de encadenamiento hacia adelante.
- Orientación a objetos.
- Portable (que corra en el hardware y software actuales).
- Que permita implementación rápida y fácil.
- Operación de imágenes y ambiente gráfico.
- Manejo de hiper-regiones.
- Experiencia en la herramienta.

Estas son las características, que se desea posea la herramienta de desarrollo a utilizar en la implementación del presente sistema, prefiriendo siempre una herramienta que ya posea incorporadas tales características, tratando de evitar inversión en tiempo de desarrollo al dotar a una herramienta ya existente de tales características. Desde luego que como último recurso se crearía un Shell propio.

A continuación se presenta una tabla comparativa en la figura 3.2.3.a; con las herramientas consideradas en nuestra evaluación; evidentemente no son todas las existentes en el mercado, pero se considera que es una muestra representativa.

Herramienta	Amzi Prolog	Common Lisp	Clips	Level 5
Representación del conocimiento	si	si	si	si
Portabilidad	si	si	si	si
Integración y extensibilidad	si	si	si	si
Desarrollo interactivo	si	si	si	si
Documentación	si	si	si	si
Soporte en línea	si	si	si	si
Manejo de frames	si	si	si	si
Encadenamiento hacia adelante	si	si	si	si
Orientación a objetos	si	si	si	si
Implementación rápida y fácil	si	si	si	si
Manejo de imágenes y ambiente gráfico	si	si	si	si
Procesos demonio	si	si	si	si
Manejo de hiper-regiones	si	si	si	si
Experiencia con la herramienta	no	no	no	si

Figura 3.2.3.a

Está claro que con cualquiera de las herramientas consideradas en la tabla anterior se podría desarrollar nuestro Sistema Experto, incluso otra que no fuera dirigida a este tipo de desarrollos; sin embargo, en nuestra elección se le da gran peso a la experiencia en el manejo de la herramienta, a la facilidad para desarrollar rápidamente un prototipo; y sobre todo que satisfaga las necesidades de nuestro desarrollo, por lo tanto se toma la decisión de realizar el prototipo con la herramienta Level 5 Object.

A continuación mostramos nuestro prototipo del sistema: como se mencionó en el segundo paso "proyecto", nuestro sistema constará de dos pantallas: una para el análisis clínico y otra para los resultados mostradas en las figuras 3.2.3.b y 3.2.3.c respectivamente, veamos dichas pantallas así como sus componentes y función.

Pantalla para el análisis clínico:

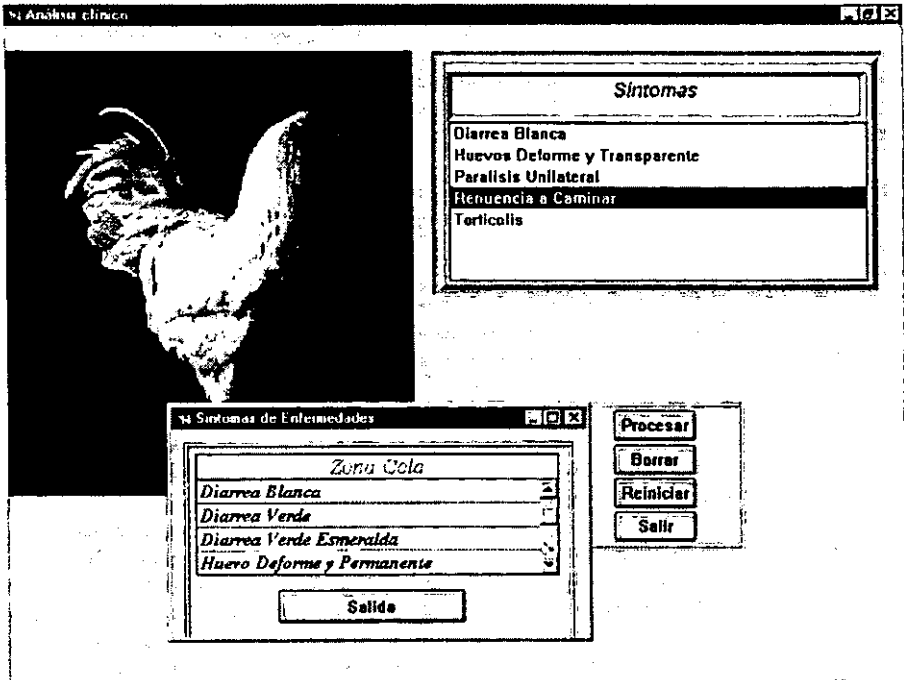


Figura 3.2.3.b

Componentes

Imagen del ave: Este componente además de mostrar un pollo, define las "hiper-regiones" dentro de su área (estas son las partes del cuerpo, elegidas para el análisis clínico), que responden al evento "click del botón izquierdo del mouse", mostrando el segundo componente de esta pantalla.

Caja de diálogo de síntomas de enfermedades: Este elemento permite visualizar los posibles síntomas por región del ave, indicando cual trata, con una etiqueta en la parte superior, que dice "Zona X", este elemento además cuenta con un "combo box", el cual permite visualizar los síntomas relacionados con la zona en cuestión, los cuales pueden elegirse posicionando el puntero del mouse y presionando el botón izquierdo; los síntomas al seleccionarse son mostrados en el tercer componente de esta pantalla.

Caja de texto de síntomas seleccionados: Como su nombre lo indica, básicamente es una caja de texto, con una barra de desplazamiento, a través de la cual el usuario puede visualizar los síntomas ya seleccionados, mientras realiza el análisis clínico; finalmente el cuarto componente es una agrupación de botones de comando.

Botón procesar: Este botón responde al click izquierdo del mouse al igual que los restantes tres y será el encargado de activar la máquina de inferencia mostrando la pantalla de resultados.

Botón borrar: Permitirá al usuario eliminar los síntomas que desee omitir, de los ya seleccionados antes de solicitar el diagnóstico; tales síntomas deberán elegirse utilizando el puntero de mouse, para después activar el citado botón borrar.

Botón reiniciar: Permite eliminar los síntomas ya seleccionados y comenzar desde el principio, con el subproceso de análisis clínico.

Botón salir: Este botón permitirá al usuario terminar el proceso, una vez que ya no desee solicitar más diagnósticos.

Pantalla de resultados

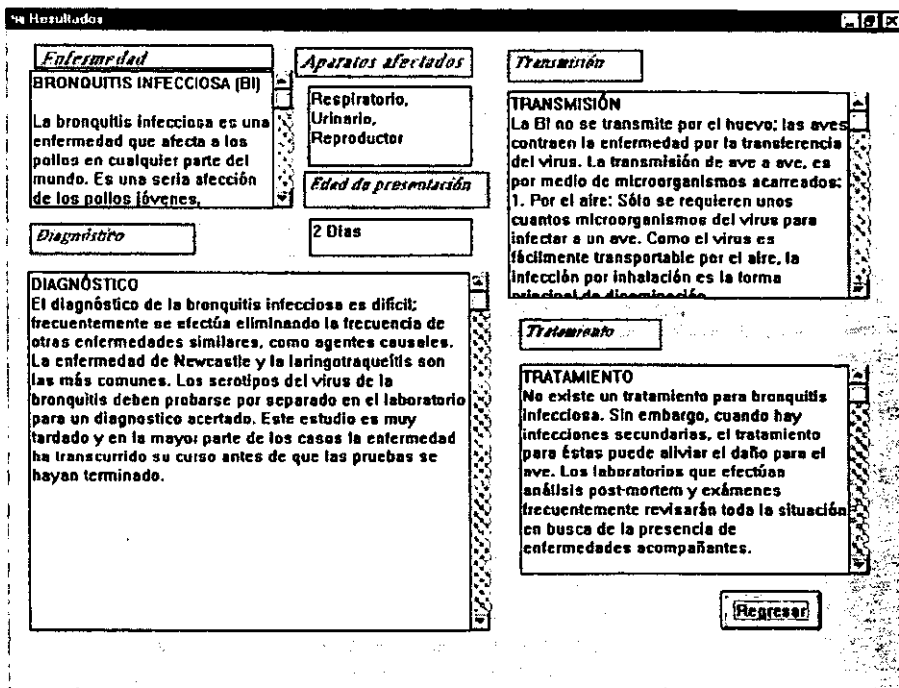


Figura 3.2.3.c

Esta pantalla cuenta con seis elementos, de los cuales cinco son cajas de texto, dado que es de carácter informativo, las cinco cajas de texto contarán con una etiqueta que indique que información están mostrando y barras de desplazamiento, para el caso en que la información no pueda desplegarse completa por cuestiones de espacio.

El otro componente con que cuenta esta pantalla, es un botón de comando con la etiqueta "Regresar", el cual responde al click izquierdo del mouse y muestra la pantalla "análisis clínico".

Las pantallas de la aplicación son el componente con el que el usuario interactúa, pero entre la del análisis clínico y la de resultados, existe el elemento más importante de todo sistema experto, el cual permite manipular la información de la base de conocimiento. Naturalmente el elemento al que nos referimos es el mecanismo de inferencia.

El mecanismo de inferencia

En la pantalla de análisis clínico, cuando el usuario elige los síntomas que observa en el ave que ausculta, está interactuando ya con el mecanismo de inferencia; al traer en primera instancia la información almacenada en la base de conocimientos: nombres de las zonas del ave y síntomas asociados a dichas zonas; tales elementos son accedidos a través de la máquina de inferencia, mientras el usuario elige la sintomatología, se va generando un conjunto de hechos que son asertados en la memoria de trabajo de nuestro Sistema Experto; para posteriormente al concluir esta fase y pedir al sistema que ejecute el mecanismo de inferencia aplicar las reglas y los criterios de solución de conflictos, mencionados con anterioridad.

La memoria de trabajo del sistema tiene un aspecto como el siguiente:

Síntoma 1

Síntoma 2

.

.

.

Síntoma n

De tal forma que el funcionamiento de la máquina de inferencia es el siguiente: Se aplica la regla # 1, la cual dispara la regla # 2, dado que hay más de un síntoma en la memoria de trabajo, al aplicarse la regla #2 internamente, el sistema toma de la base de conocimientos, el frame de síntomas que a continuación muestra la figura 3.2.3.d:

Nombre sintoma	Síntoma ₁	Enfermedad ₁ (Síntomas_presentes ₁ = Síntomas_presentes ₁ +1)
Nombre sintoma	Síntoma ₂	Enfermedad ₂ (Síntomas_presentes ₂ = Síntomas_presentes ₂ +1)
.	.	.
Nombre sintoma	Síntoma _n	Enfermedad _n (Síntomas_presentes _n = Síntomas_presentes _n +1)
slot	slot value	facet

Figura 3.2.3.d

Al encontrar el síntoma ejecuta el método asociado, el cual genera una una instancias "Enfermedad_n" del frame "Enfermedad", igualando el slot "Síntomas_presentes" a uno e incrementándolo en uno cada vez que el procedimiento del facet del frame "Síntoma_n" es ejecutado al coincidir la "Enfermedad asociada" con "Enfermedad_n".

Se incrementa el contador de la enfermedad correspondiente y se elimina dicho mal de la memoria de trabajo del sistema, regresando a aplicar la regla #1, lo cual se repite hasta eliminar todos los síntomas de la memoria de trabajo del sistema.

Para este momento en el sistema se han generado tantas instancias "Enfermedad_n", como enfermedades asociados a los síntomas elegidos, a continuación se muestra dichas instancias en las figuras 3.2.3.e, 3.2.3.f y 3.2.3.g

Enfermedad₁

Nombre enfermedad	Enfermedad ₁	
Síntomas presentes	Valor ₁	

Figura 3.2.3.e

Enfermedad₂

Nombre enfermedad	Enfermedad ₂	
Síntomas presentes	Valor ₂	

Figura 3.2.3.f

...
...
...

Enfermedad_n

Nombre enfermedad	Enfermedad _n	
Síntomas presentes	Valor _n	

Figura 3.2.3.g

Y es en este momento cuando se aplica la regla # 3, la cual indica que se apliquen los criterios de solución de conflictos y que se diagnostique; lo primero que hace la máquina de inferencia es observar si existe más de una instancia del frame "Enfermedad_n", si el elemento es único el diagnóstico es inmediato, sino se aplica el primer criterio de solución de conflictos, nuevamente se accede a la base de conocimientos, para validar si alguna de las enfermedades presentes es mortal; esto nos genera un nuevo conjunto de enfermedades; si este tiene un único elemento, el diagnóstico es presentado, sino, se aplica el segundo criterio de solución de conflictos, por lo que la máquina de inferencia valida cual de las enfermedades aún presentes en el conjunto generado tiene un mayor número de síntomas presentes, después de esto se genera un conjunto de un elemento único y se procede a diagnosticar, con lo que nuevamente, a través del mecanismo de inferencia, se manda visualizar la pantalla de resultados. Después de generar el diagnóstico todas las instancias del frame "Enfermedad_n" son destruidas, quedando así el sistema listo para un nuevo diagnóstico.

3.2.4 Cuarto paso en la planificación del sistema: "Validación del proyecto"

En este punto lo fundamental es, asegurar que el prototipo se ajusta a los requerimientos identificados en el paso "análisis del problema" tanto de forma como de fondo. Esta labor se realizará conjuntamente entre el experto veterinario y el ingeniero del conocimiento.

Puntos a validar:

Lo fundamental es que el sistema diagnostique; sin embargo, comenzaremos por partes: como se ha mencionado a lo largo del desarrollo nuestro sistema consta de tres partes fundamentales: la pantalla de análisis clínico, la máquina de inferencia y la pantalla de resultados.

En la tabla mostrada en la figura 3.2.4 se listan los elementos indispensables para validar la funcionalidad del prototipo.

Pantalla de análisis clínico	
Visualización del ave sometida a revisión	Si
Fácil identificación de las zonas de su cuerpo	Si
Se muestran los síntomas asociados a las distintas zonas del ave	Si
Facilidad de seleccionar los síntomas identificados en el ave auscultada	Si
Se tienen presentes los síntomas ya seleccionados	Si
Se permite eliminar síntomas elegidos por error	Si
Se permite regresar a elegir un síntoma de una zona ya revisada	Si
Se permite volver a comenzar con el análisis clínico si así se desea	Si
Permite finalizar la aplicación en cualquier momento	Si
Permite accionar la máquina de inferencia, una vez seleccionados todos los síntomas	Si
Máquina de inferencia	
Se ejecuta cuando el usuario lo solicita	Si
Están implementadas las reglas necesarias	No
Están presentes los hechos en la memoria de trabajo	No
Están incorporados los métodos necesarios	No
Están disponibles los procedimientos suficientes	No
Interactúa con la base de conocimientos, las reglas, los métodos, los hechos y los procedimientos para realizar el diagnóstico	Parcialmente
Pantalla de resultados	
Muestra la información requerida por el usuario	Si
Nombre Enfermedad	Si
Descripción	Si
Edad de Presentación	Si
Causa	Si
Transmisión	Si
Diagnóstico	Si
Diagnóstico diferencial cuando existe	Si
Tratamiento	Si
Aparato o sistema afectado	Si
Del sistema	
La base de conocimientos está totalmente implementada	No
Se probaron todos los casos posibles con éxito	No
La máquina de inferencia es del todo funcional	No
El sistema se ha completado	No

Figura 3.2.4

Como se observa en la figura 3.2.4, aún queda mucho trabajo por realizar, sin embargo, en este punto se ha alcanzado un buen avance sobre todo en la parte de conceptualización, en el siguiente punto se interactuará con el experto veterinario, hasta llegar a un prototipo que se aproxime mucho al sistema final. Se trabajará principalmente en la máquina de inferencia.

3.2.5 Planificación del sistema "sin final"

En este punto el objetivo es hacer que el mecanismo de inferencia y la forma de representación del conocimiento funcionen. Para tal efecto se trabajará en la implementación de las reglas, métodos y procedimientos necesarios para que la máquina de inferencia pueda interactuar con la base de conocimiento, apoyados en la forma de representación del conocimiento elegida.

En este punto es necesario retomar nuevamente el ciclo completo de la planificación del sistema.

De acuerdo a la tabla de la figura 3.2.4, en este punto se trabajó fundamentalmente en la máquina de inferencia, y en conjunto con el experto veterinario se validó la funcionalidad de la misma; luego de lo cual se aplicaron los siguientes criterios:

1. El mecanismo de inferencia y la representación del conocimiento funcionan.
2. De cada círculo de problemas hay al menos un caso ejemplar implementado.
3. El gasto para la realización del sistema de aplicación puede valorarse como justificado.

Se considera que se cumplen los criterios anteriores, dado que se han realizado pruebas de funcionalidad con el experto veterinario y el mecanismo de inferencia está funcionando correctamente, ya han sido implementadas las reglas descritas con anterioridad, así como los métodos y procedimientos asociados necesarios para la interacción con la base de conocimientos; la forma de representación del conocimiento está más que validada con las pruebas de interacción que han corrido entre ésta y el mecanismo de inferencia; el sistema es capaz de realizar el proceso completo del diagnóstico, aunque sólo para algunos casos específicos, y observa una estabilidad tal, que nos permite concluir con esta etapa en forma satisfactoria.

Finalmente la valoración del gasto para la realización del sistema de aplicación también se considera justificada, más adelante se retomará este punto junto con todo lo referente al análisis costo / beneficio, a manera de ejemplo, por lo pronto dejaremos a este nivel la justificación del citado gasto.

3.3 Desarrollo para alcanzar la madurez de la aplicación

En este punto el objetivo es obtener un producto de software listo para su aplicación productiva.

Aquí se tienen dos alternativas:

- Tomar como modelo el prototipo, partiendo nuevamente del concepto inicial de la planificación del sistema, para implementar un nuevo prototipo, tomando del primero las estructuras y paradigmas de programación, sin reutilizar el código.
- Y la alternativa dos: "el postdesarrollo del prototipo", utilizando el código ya escrito y trabajando para optimizarlo.

El camino que decidimos tomar es el segundo, por lo cual el trabajo estará centrado en la ampliación de la base de conocimientos y la refinación del mecanismo de inferencia, dado que la interfaz con el usuario y el componente explicativo están prácticamente terminados, a menos que en esta fase resultaran necesarias nuevas adecuaciones.

Adquisición de conocimiento

En conjunto con el experto veterinario, se complementó la base de conocimiento; el sistema no cuenta con una interfaz de usuario para realizar esta labor, la decisión se tomó, dado que se considera que el grado de modificación del conocimiento no será alto y el trabajo de conformar esta interfaz si sería complejo.

Estrategia de ampliación de la base de conocimientos

Como se mencionó en el punto anterior, en este momento el sistema ya cuenta con todo el cúmulo de conocimiento de nuestro experto veterinario, si se ha de incorporar nuevo conocimiento, esto sería hecho por el experto que posea el nuevo conocimiento, en conjunto con un desarrollador, mediante la herramienta con que se desarrollo el sistema.

Prueba y ejecución del programa piloto

En este punto se validará la funcionalidad y operabilidad de nuestro sistema y con base en los resultados obtenidos, se evaluará si el mismo está listo para incorporarse como una aplicación productiva. Para tal efecto nos apoyamos nuevamente de la tabla de la figura 3.2.4, que para efectos de distinguirla denominaremos figura 3.3.

Pantalla de análisis clínico	
Visualización del ave sometida a revisión	Si
Fácil identificación de las zonas su cuerpo	Si
Se muestran los síntomas asociados a las distintas zonas del ave	Si
Facilidad de seleccionar los síntomas identificados en el ave auscultada	Si
Se tienen presentes los síntomas ya seleccionados	Si
Se permite eliminar síntomas elegidos por error	Si
Se permite regresar a elegir un síntoma de una zona ya revisada	Si
Se permite volver a comenzar con el análisis clínico si así se desea	Si
Permite terminar la aplicación en cualquier momento	Si
Permite accionar la máquina de inferencia una vez seleccionados todos los síntomas	Si
Máquina de inferencia	
Se ejecuta cuando el usuario lo solicita	Si
Están implementadas las reglas necesarias	Si
Están presentes los hechos en la memoria de trabajo	Si
Están incorporados los métodos necesarios	Si
Están disponibles los procedimientos suficientes	Si
Interactúa con la base de conocimientos, las reglas, los métodos, los hechos y los procedimientos para realizar el diagnóstico	Si
Pantalla de resultados	
Muestra la información requerida por el usuario	Si
Nombre Enfermedad	Si
Descripción	Si
Edad de Presentación	Si
Causa	Si
Transmisión	Si
Diagnóstico	Si
Diagnóstico Diferencial cuando existe	Si
Tratamiento	Si
Aparato o sistema Afectado	Si
Del sistema	
La base de conocimientos está totalmente implementada	No
Se han probado todos los casos posibles con éxito	Si
La máquina de inferencia es totalmente funcional	Si
El sistema está terminado	No

Figura 3.3

Como se observa en la tabla anterior, aún existen algunos puntos que dicen "No", esto obedece a que dentro del concepto de Sistemas Expertos una base de conocimientos nunca está completamente implementada, dado que el conocimiento es de naturaleza evolutiva y siempre surge nuevo.

Por otro lado, bajo este mismo concepto, un sistema experto nunca se termina del todo, siempre está en un ciclo infinito de desarrollo, ya sea por mejoras, actualizaciones, fallas, etc.

Sin embargo, se considera que en este momento se tiene un sistema estable y listo para ser lanzado como una aplicación productiva.

3.4 Aplicación y mantenimiento

Como se mencionó en el punto anterior, el sistema está listo para introducirse como una aplicación productiva, por lo que comenzará un período de observación del desempeño de la aplicación; en este proceso intervendrán el experto veterinario y su equipo de trabajo, dado que ellos son los usuarios iniciales del sistema.

El mantenimiento se basa en el siguiente esquema:

1. Actualización de la base de conocimientos.
2. Determinación de fallos y limitaciones.
3. Realización de cambios por 1 y 2.

Conclusiones

El desarrollo de sistemas de información, presupone para la mayoría una sustitución de los hombres y sus habilidades por máquinas, lo cual toma mayor fuerza cuando se habla de Sistemas Expertos. Ya se ha visto a lo largo del presente trabajo, que el desplazamiento de hombres por máquinas no es el objetivo del desarrollo de Sistemas Expertos; lo que se persigue es dotar al hombre de herramientas que le permitan apoyarse en todas las funciones, tareas, labores y en general en cualquier actividad que éste desarrolle.

Los Sistemas Expertos son en todo caso extensiones del saber humano, al igual que los sistemas tradicionales, son a fin de cuentas, desarrollo de sistemas de información: un tipo "especial" de software.

Como se vio a través de esta tesis, el desarrollo de un Sistema Experto se apoya en gran medida en la metodología de desarrollo de software tradicional y lo que los hace diferentes el uno del otro, son las técnicas y los métodos aplicados en cada una de sus fases. Fundamentalmente la etapa "Planificación del sistema", en la cual se realiza gran parte del desarrollo, se caracteriza por recorrer las diferentes fases del desarrollo tradicional de manera iterativa, por la estrecha colaboración con el usuario en la cual se destaca no sólo el apego a los requerimientos; sino la búsqueda de la satisfacción del mismo y la utilidad y vigencia del sistema a través del tiempo; otra característica muy destacable es el desarrollo de prototipos.

El objetivo principal en un desarrollo de este tipo, es conceptualizar en una forma de representación del conocimiento; la experiencia y habilidad de un experto humano en alguna área del conocimiento, para su posterior implementación mediante alguna herramienta de desarrollo de software y su liberación como una aplicación productiva. En este sentido se considera cubierto este objetivo, con el desarrollo logrado en el presente trabajo. El sistema desarrollado se usa al momento de escribir estas conclusiones con objetivos académicos y de investigación, pero se considera que se tienen mayores alcances.

Finalmente quiero comentar que la realización del presente trabajo, fué una excelente oportunidad de poner en práctica, los conocimientos adquiridos durante mi formación académica e incluso laboral, es de mencionarse que se mantuvo un alto apego a la metodología elegida, lo que fué fundamental para llevar a buen término la conclusión de esta tesis. Hubo un largo período de madurez de la idea y también considero, que este hecho fue muy benéfico en el presente desarrollo.

Apéndice A

Zonas del ave

Abdomen, alas, cabeza, cola, cuello, dorso, patas y torax.

Posibles síntomas por zona y su correspondiente enfermedad asociada

Zona Abdomen	Enfermedad Asociada
Tics	Newcastle
Parálisis unilateral	Newcastle
Diarrea verde	Newcastle

Figura A.1

Zona Alas	Enfermedad Asociada
Ala manchada amarillo	Laringotraqueitis
Parálisis unilateral	Newcastle
Tics	Newcastle

Figura A.2

Zona Cabeza	Enfermedad Asociada
Opacidad de córnea	Newcastle
Tics	Newcastle
Parálisis unilateral	Newcastle
Quejido	Newcastle
Temblo de cabeza	Newcastle
Estornudo generalizado	Newcastle
Ronquera general	Bronquitis infecciosa
Grito corto	Bronquitis infecciosa
Cara hinchada	Sínd cabeza hinchada
Exudo nasal	Bronquitis infecciosa
Grito chifido	Bronquitis infecciosa
Secreción lacrimal	Bronquitis infecciosa
Blefarconjuntivitis	Laringotraqueitis
Grito largo	Laringotraqueitis
Bloqueo pronunciado	Laringotraqueitis
Estornudo	Newcastle
Secreción nasal y sinusitis	Laringotraqueitis
Bajo consumo de alimento	Laringotraqueitis
Secreción clara nasal	Laringotraqueitis
Grito	Micoplasmosis MG
Cresta pálida y encogida	Micoplasmosis MG
Estornudo y grito	Micoplasmosis MG
Cabeza hinchada	Sínd cabeza hinchada
Hinchazón de párpados en la Region periorcular	Sínd cabeza hinchada
Hinchazón de senos infraorbitales	Sínd cabeza hinchada
Hinchazón de mandíbula inferior y nuca	Sínd cabeza hinchada
Conjuntivitis inflamación de las glándulas oculares	Sínd cabeza hinchada
Baja de postura de pico acompañada de signos respiratorios	Sínd cabeza hinchada

Figura A.3

Zona cola	Enfermedad asociada
Baja de postura	Bronquitis infecciosa
Diarrea blanca	Colibaciosis
Diarrea verde	Colibaciosis
Diarrea verde esmeralda	Newcastle
Huevo deforme y permanente	Bronquitis infecciosa
Huevos deforme y transparente	Newcastle
Parálisis unilateral en cola	Newcastle
Postura aves pingüino	Bronquitis infecciosa
Tics en cola	Newcastle

Figura A.4

Zona Cuello	Enfermedad Asociada
Estertores traqueales bronquiales	Micoplasmosis MS
Parálisis unilateral	Newcastle
Tics	Newcastle
Torticolis	Newcastle

Figura A.5

Zona Dorso	Enfermedad Asociada
Parálisis unilateral	Newcastle
Tics	Newcastle

Figura A.6

Zona Patas	Enfermedad Asociada
Flacidez de piernas	Newcastle
Parálisis unilateral	Newcastle
Renuencia a caminar	Micoplasmosis MS
Tics	Newcastle

Figura A.7

Zona torax	Enfermedad asociada
bloqueo y disnea	Aspergilis pulmonar
complicacion bacteriana	Micoplamosis MG
estertores	Estomudo generalizado
parálisis unilateral	Newcastle
postraciones y cianosis	Coriza infecciosa
tics	Newcastle

Figura A.8

Apéndice B

NEWCASTLE

También conocida como neumoencefalitis aviar, pseudopeste y peste aviar es una enfermedad viral causada por un paramixovirus, que puede producir signos respiratorios digestivos y nerviosos en la mayoría de las aves a cualquier edad y en el hombre llega a producir conjuntivitis.

Aparato o sistema afectado:

Respiratorio, Nervioso, Digestivo, Reproductor, Locomotor.

Edad de Presentacion:

2 semanas

Transmisión:

Se transmite principalmente por aerosoles (vía aérea) y contacto directo, pero también son importantes el agua y el alimento contaminado, así como el personal de la granja y el equipo. Igualmente existe la transmisión a partir de portadores sanos.

Mortalidad: alta

Diagnóstico diferencial

Por el cuadro respiratorio, la enfermedad de Newcastle clínicamente se puede confundir con:

- Bronquitis infecciosa (en caso de enfermedad de Newcastle tipo lentogénico).
- Laringotraqueítis (en caso de enfermedad de Newcastle tipo velogénico)
- Coriza infecciosa.
- Enfermedad respiratoria crónica.

Por el cuadro nervioso puede confundirse con:

- Encefalomiелitis aviar
- Encefalomalacia
- Enfermedad de Marek

Por la lesión en el proventrículo puede confundirse con:

- Aflatoxicosis
- Infección de la Bolsa de Fabricio

- Síndrome de la mala absorción
- Enfermedad de Marek

Tratamiento

No existe. Se recomienda el uso de antibióticos de amplio espectro a fin de evitar complicaciones bacterianas.

BRONQUITIS INFECCIOSA (BI)

La bronquitis infecciosa afecta a los pollos en cualquier parte del mundo. Es una seria afección de los ejemplares jóvenes, causando una alta mortalidad. En las aves de postura conlleva una gran pérdida económica debido a la reducida producción de huevo y la baja calidad de cascarón, el pollo es la única ave susceptible que se conoce.

Causa

Esta enfermedad la provoca un virus filtrable, existen 20 o más serotipos reconocidos; en algunos casos, unos producen inmunidad cruzada; los serotipos más conocidos son:

Massachusetts
Connecticut
Holland

Los serotipos menos conocidos son:

Arkansas-99
Florida
JMK
SE-17
T australiano
Cepa DPI tipo Arkansas(3168)

Propiedades de los tipos importantes:

1. La cepa Massachusetts produce un tipo de enfermedad grave.
2. La cepa Massachusetts produce inmunidad cruzada con la cepa Connecticut.
3. La cepa Connecticut produce una escasa inmunidad, cruzada con la Massachusetts.
4. La cepa Holland produce inmunidad cruzada con las Massachusetts, Connecticut, JMK, Florida y la SE-17.
5. En algunas regiones predomina la cepa JMK; en las otras la Arkansas-99.
6. Salvo la vacuna específica, ninguna otra afecta la cepa DPI tipo Arkansas(3168).

7. Muchas de las cepas producen daño uterino en pollas jóvenes, lo que reduce la posterior producción de huevos.

Aparato o sistema afectado:

Respiratorio, Urinario, Reproductor.

Edad de Presentación:

2 Dias

Transmisión:

La BI no se transmite por el huevo; las aves contraen la enfermedad por la transferencia del virus; la transmisión de ave a ave, es por medio de microorganismos acarreados:

1. Por el aire: Sólo se requieren unos cuantos microorganismos del virus para infectar a un ave. Como el virus es fácilmente transportable por el aire, la infección por inhalación es la forma principal de diseminación.
2. Por medio de persona, aves y animales: Esta representa una de las principales formas de diseminación, de una caseta a otra y de una granja a otra.
3. Por equipo, etcetera.
4. Dentro o sobre el alimento
5. Por aves portadoras: Las aves pueden transmitir el virus hasta cuatro semanas después de la recuperación.

Mortalidad: alta

Diagnóstico

El diagnóstico de la bronquitis infecciosa es difícil; a menudo se lleva a cabo eliminando la frecuencia de otras enfermedades similares, como agentes causales. La enfermedad de Newcastle y la laringotraqueítis son las más comunes. Los serotipos del virus de la bronquitis deben probarse por separado en el laboratorio para un diagnóstico acertado. El estudio es muy tardado y en la mayor parte de los casos, la enfermedad ha completado su curso antes de que las pruebas se hayan terminado.

Tratamiento

No existe para bronquitis infecciosa; sin embargo, cuando hay infecciones secundarias, el tratamiento para éstas puede aliviar el daño para el ave. Los laboratorios que efectúan análisis post-mortem, exámenes frecuentemente revisarán toda la situación en busca de la presencia de enfermedades acompañantes.

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

LARINGOTRAQUEITIS

También conocida como Laringotraqueítis infecciosa o difteria aviaria, es una enfermedad respiratoria causada por un virus herpes, que se caracteriza por hemorragias en la tráquea, expectoración de exudado sanguinolento y formación de corpúsculos intranucleares en el epitelio respiratorio al inicio de la enfermedad, afecta a la gallina doméstica, al faisán, al pavo y al pavo real de cualquier edad.

Aparato o sistema afectado:

Respiratorio.

Edad de Presentación:

3-4 semanas.

Transmisión:

La enfermedad se transmite en forma horizontal, por aerosoles de aves enfermas o portadores asintomáticos o por personal y equipo contaminado.

Mortalidad: alta

Diagnóstico diferencial

La laringotraqueítis puede confundirse con:

- Bronquitis Infecciosa
- Enfermedad de Newcastle velogénica viscerotrópica
- Viruela diftérica
- Avitaminosis A
- Enfermedad respiratoria crónica complicada

Tratamiento

Aún no se ha encontrado un tratamiento. Se recomienda el empleo de antibióticos de amplio espectro para prevenir infecciones secundarias y como tratamiento sintomático, expectorantes en el agua de bebida tales como:

- Iodine (yodo orgánico)
- Sulfatiazol sódico

Vacunación:

- Solo en zonas donde existe la laringotraqueítis
- Sobre brote, empezando con las parvadas sanas

La vacuna que se utiliza es de virus vivo atenuado y la vía más adecuada es la ocular.

Se puede aplicar en agua de bebida, con dos gramos de leche en polvo por litro como protector, pero nunca por aspersion, ya que aún los virus atenuados pueden producir reacciones graves.

CORIZA INFECCIOSA

También conocida como coriza bacilar, es una enfermedad producida por *Haemophilus gallinarum*, que afecta principalmente las vías respiratorias altas, aunque puede llegar a producir acrosaculitis cuando se complica con *Mycoplasma gallisepticum*. Se caracteriza por producir inflamación de la cara y exudado nasal mucoso de mal olor, afecta a la gallina doméstica a partir de los siete días de edad y ocasionalmente se puede presentar en faisanes y en gallinas de Guinea.

Aparato o sistema afectado:

Respiratorio.

Edad de presentación:

3-4 semanas.

Transmisión:

Se transmite principalmente por agua contaminada o directamente de un ave a otra; son muy importantes los portadores asintomáticos (aves recuperadas).

Mortalidad: Baja

Diagnóstico diferencial

La Coriza infecciosa se puede confundir con:

- Viruela aviaria
- Enfermedad respiratoria crónica
- Avitaminosis A
- Enfermedad de Newcastle
- Bronquitis infecciosa
- Pasteurelisis

Tratamiento

En agua y alimento: sulfadimetoxina

Intramuscular: tetraciclinas, critromicina, cloramfenicol o tilosina.

Cuando las gallinas han estado infectadas poco tiempo pueden recaer al suspenderse el tratamiento, ya que no desarrollan inmunidad contra *Haemophilus gallinarum*. Cabe recordar que todas aquellas que hayan enfermado quedan como portadoras sanas.

COLIBACILOSIS (INFECCIÓN POR E. COLI)

Los microorganismos coliformes ocasionan diversas enfermedades avícolas, con amplia variedad de manifestaciones. *Escherichia coli* (*E. coli*) es una bacteria que representa uno de los muchos microorganismos del grupo de los coliformes que habitan la parte inferior de las vías intestinales, la mayor parte son inofensivos y se conocen como saprofitos, ayudan en el proceso de la digestión; otros son patógenos y producen ciertas enfermedades avícolas. Aunque la mayor parte no lo son, los pocos que si lo son tienen la capacidad de producir una alta morbilidad y mortalidad con serias pérdidas económicas.

Aparato o sistema afectado:

Respiratorio.

Edad de presentación:

7-14 Días.

Transmisión:

Existen varias formas de transmisión de *E. coli*:

- 1.- Fecal: Los microorganismos en vías intestinales se eliminan en forma continua a través de la materia fecal, de esta manera las bacterias se secan y flotan en el aire, pudiendo llegar a individuos no infectados a través del aparato respiratorio.
- 2.- Contaminación del cascarón: Como el huevo completo se asienta en la cloaca antes de eclusión se contamina con el excremento en vías intestinales.
- 3.- Respiratoria: Como la infección del saco aéreo por *E. coli* puede resultar en infección a través del aparato respiratorio, el polvo contaminado en la caseta avícola puede ser causa directa de la transmisión.
- 4.- Ovárica: La transmisión a través del ovario puede ser posible si las aves se reproducen cuando por una infección del útero. De esta manera las gallinas reproductoras infectadas, transmiten la enfermedad al pollito recién nacido.
- 5.- Alimento: Aunque no es una ruta de infección primaria, los coliformes pueden tener oportunidad de entrar al cuerpo, a través de alimento contaminado.

Mortalidad: alta

lenta 7-15 días

Diagnóstico

Una prueba de laboratorio es la única forma satisfactoria para un diagnóstico acertado.

Tratamiento

Todo tratamiento debe comenzar con una campaña de limpieza, ya que gran parte de las infecciones por *E. coli* empiezan en instalaciones sucias. LA sulfadimetoxia más ormetoprim es el único tratamiento en el alimento recomendado para colibacilosis; Se han utilizado otros fármacos incluyendo tetraciclinas, sulfas, novobiocina y gentamicina. Las pruebas de sensibilidad a los antibióticos pueden ser útiles para determinar medicamentos eficaces.

MYCOPLASMA GALLISEPTICUM (MG)

Esta enfermedad es muy conocida y extremadamente importante tanto para las aves de engorda; como para aves productoras de huevo. La infección de los sacos aéreos en las aves de engorda, es una causa de decomiso de las canales enfermas, por no ser aptas para consumo humano. En parvadas de postura positivas a MG se ha demostrado que se producen hasta 20 huevos menos por año que las parvadas negativas.

Aparato o sistema afectado:

Respiratorio, Reproductor.

Edad de presentación:

2 semanas en adelante

Transmisión:

Existen varios canales por donde el *Mycoplasma gallisepticum*, el microorganismo serotipo S-6 puede transmitirse:

- 1.- A través del huevo incubable: Este es el principal medio de difusión; el microorganismo pasa de una generación a la siguiente.
- 2.- El microorganismo se puede mover a cortas distancias vía aérea.
- 3.- En ropa, bolsas de alimento, equipo avícola y camiones; entre el personal se encuentran portadores importantes.
- 4.- Por aves infectadas: La eliminación completa de la población de las casetas y de las granjas es esencial para prevenir la exposición de las nuevas parvadas a dicho mal.

Mortalidad: alta

Diagnóstico

MG puede diagnosticarse fácilmente por la tos y el gorgoreo involucrados, las pruebas de laboratorio utilizadas en el diagnóstico son:

- 1.- Prueba rápida de suero en placa, o de aglutinación en tubo
- 2.- Prueba de inhibición de hemoaglutinación
- 3.- Examen embrionario: Los embriones infectados tienen lesiones en los sacos aéreos.

Tratamiento

La tilosina es el antibiótico específico para el tratamiento de la enfermedad en aves con MG; también pueden emplearse fármacos como aureomicina, eritromicina, espectinomina, doxiclina o estreptomina, para varios rangos de edad.

El sumergimiento del huevo, usando tilosina y gentamicina en solución, ofrece otro método de ayuda, pero no de control.

MYCOPLASMA SYNOVIAE (MS)

Es una enfermedad que ha predominado por muchos años; la identificación del microorganismo causante, ha provocado el entendimiento de las complicaciones detrás de MS.

Causa

El agente causal de la enfermedad es *Mycoplasma synoviae*, similar a *Mycoplasma Gallisepticum*; existe un solo serotipo; es pequeño, delicado y sin pared celular rígida.

Aparato o sistema afectado:

Respiratorio, Locomotor, Reprodutor.

Edad de presentación:

4 semanas en adelante

Transmisión:

Existen varias formas de transmisión y probablemente otras que no se han aclarado todavía.

1. A través del huevo: La enfermedad es definitivamente transmitida por el huevo y aunque el porcentaje de los infectados es bajo, estos originan pollitos infectados, por lo que al cabo del tiempo un gran porcentaje de la parvada se infecta.

Después de la infección, las reproductoras eliminarán MS, por un periodo de 14 a 40 días.

2. Por aire: Los microorganismos se transmiten con facilidad de un ave a otra dentro del corral por medio del aire, pero probablemente no de una caseta a otra.

3. En ropa, camiones, equipo, etc.: Las formas mecánicas de transferencia son de mucha importancia, en el transporte de los microorganismos a grandes distancias.

Mortalidad: baja

Diagnóstico

La inflamación de las articulaciones del corvejón y del cojinete plantar son un síntomas de MS, pero no necesariamente un diagnóstico, existen demasiadas enfermedades que producen estados similares. Las aves deben enviarse al laboratorio en donde se deben efectuar dos pruebas:

1. Prueba de aglutinación en placa: El suero de la sangre se mezcla con el antígeno de MS. como una prueba de anticuerpo, al obtener suero de la aves que tienen la enfermedad se agruparán con el antígeno. Esto también se puede llevar a cabo en la caseta avícola. Nota: Existe un poco de aglutinación cruzada entre MG y los microorganismos de MS. Los anticuerpos de MG pueden cruzarse-aglutinarse con el antígeno MS; lo opuesto no es verdadero.

2. Inoculación a un ave: El material se obtiene de las articulaciones sospechosas; se muele, tiñe e inyecta en las articulaciones del cojinete plantar a pollitos de cuatro semanas de edad; si el donador es un ave infectada con MS o MG, se presentará definitivamente una inflamación en los cojinetes plantares en los pollitos receptores, en alrededor de una semana. Debe diferenciarse MS de MG, en forma serológica.

Tratamiento

Algunos antibióticos de amplio espectro son valiosos:

1. Clorotetraciclina. Adicionada al alimento en un grado de 100 a 200 g por tonelada(200lb) aproximadamente durante una semana. No darlo a gallinas ponedoras.

2. Oxitetraciclina: Adicionar al alimento en un grado de 200 g por tonelada (2000 lb) más o menos por una semana; Reanude el tratamiento con medicamentos si la enfermedad no disminuye; el tratamiento puede continuar por largo tiempo si la enfermedad no es llevada bajo control en el primer tratamiento.

3. Inyecciones individuales: Los brotes serios en aves adultas pueden tratarse mediante la inyección de oxitetraciclina o eritromicina, el tratamiento individual en esta forma, no es practico con los pollos de engorda por el costo de mano de obra a pagar y el estrés del manejo.

PASTEURELOSIS

También conocida como cólera aviario, es una enfermedad producida por *Pasteurella multocida*, que puede presentar signos respiratorios y un cuadro septicémico; se presenta tanto en aves como en mamíferos y afecta con mayor severidad a los pavos jóvenes y a las gallinas adultas. Es poco frecuente en las explotaciones modernas.

Aparato o sistema afectado:

Respiratorio.

Edad de presentación:

5-6 semanas.

Transmisión:

Los portadores asintomáticos son muy importantes y constituyen la principal forma de transmisión de la enfermedad, se lleva a la parvada principalmente por agua y alimentos contaminados por excreciones nasales y bucales, así como por aerosoles y canibalismo.

Mortalidad: baja

Diagnóstico diferencial

La Pasteurelisis se puede confundir con:

- Enfermedad de Newcastle
- Coriza infecciosa
- Intoxicaciones
- Salmonelosis
- Golpe de calor
- Sinovitis
- Plaga aviaria
- Plaga de los patos

Tratamiento

Se utiliza la sulfaquinoxalina en agua de bebida, así como penicilina, triptomocina o clortetraciclina por vía intramuscular.

Vacunación:

- Se recomienda la producción de autobacterinas por las diferencias antigénicas en las diferentes cepas.
- Vacunas con bacterias vivas atenuadas.
- Aplicarlas a aves de seis a ocho semanas de edad y revacunar 10 semanas después
- Bacterinas múltiples emulsionadas.

ASPERGILOSIS PULMONAR

También conocida como neumonía de la nacedoras, neumonía enzoótica, neumonía silenciosa o neumonía micótica, es una enfermedad respiratoria producida principalmente por el hongo *Aspergillus Fumigatus*; se caracteriza por la ausencia de ruidos respiratorios durante el curso de la enfermedad; en algunas ocasiones se vuelve sistémica y llega a producir ceguera y signos nerviosos. Afecta tanto a las aves como a los mamíferos.

Aparato o sistema afectado:

Respiratorio.

Edad de presentación:

1-10 Dias.

Transmisión:

La transmisión es aerógena por esporas del hongo. La principal fuente de infección son las nacedoras, las cajas de transporte y la cama contaminada; las aves adultas inhalan las esporas a partir de la cama y el alimento contaminado; sin embargo, las gallinas y los pavos adultos son muy resistentes, por lo que esta forma de transmisión tiene poca importancia en estas especies.

Mortalidad: baja

Diagnóstico diferencial

La Aspergilosis debe diferenciarse de:

- Bronquitis Infecciosa
- Enfermedad de Newcastle
- Encefalomiелitis aviar
- Encefalomalacia
- Deficiencia de vitamina A
- Purolosis

Tratamiento

En pollitos no se aplica ningún tratamiento, ya que los animales con signos difícilmente se recuperarán; por otro lado rápidamente desarrollan resistencia y no aparecen otros animales enfermos después de los quince días de edad; en aves de ornato y compañía, el tratamiento es empírico y se emplea nistatina y anfotericina B, pero difícilmente resulta útil porque cuando se llega a diagnosticar, las lesiones que produjo la Aspergilosis ya son muy severas.

SIND CABEZA HINCHADA

Aparato o sistema afectado:

Respiratorio

Edad de presentación:

Todas las edades.

Transmisión:

rápida 2-3 días.

Mortalidad: baja

INFLUENZA AVIAR (IA)

La influenza aviar es una enfermedad viral de muchas especies de aves afecta el aparato respiratorio y digestivo y al sistema nervioso. En su forma más virulenta, presenta una mortalidad extremadamente alta.

Causa

El virus de la Influenza aviar está clasificado como un ortomixovirus tipo A; también existen numerosos subtipos que se identifican por la clase de hemaglutinina (H) y neuraminidasa(N). Cada uno difiere en su patogenicidad, capacidad para infectar diferentes especies y transmisibilidad.

Aparato o sistema afectado:

Respiratorio.

Edad de presentación:

No se sabe.

Transmisión:

Las aves acuáticas son los hospedadores naturales de la IA; se ha aislado el virus de especies domésticas y silvestres y del agua de los lagos y estanques que utilizan las aves acuáticas. La enfermedad se transporta con facilidad de una granja a otra en ropa contaminada, canastos y equipo.

Mortalidad: muy alta

Diagnóstico

Se puede utilizar la evidencia serológica para determinar la exposición a IA, pero la confirmación positiva requiere de aislamiento e identificación del virus.

Tratamiento

No se conoce algún tratamiento para esta enfermedad; el uso de antibióticos de amplio espectro puede reducir las complicaciones con otras infecciones. Se considera que el único remedio es eliminar la población infectada.

Apéndice C

Aquí retomaremos los aspectos mencionados en el punto 3.1.5. Costo y beneficio del Sistema Experto y a manera de ejemplo pondremos cifras en pesos a los conceptos en este punto plasmados. Tales costos fueron obtenidos de empresas dedicadas al desarrollo de software, por razones de confidencialidad no se mencionan los nombres de dichas empresas. Los costos son el promedio del universo muestra considerado.

Costos:

Durante el desarrollo:

Concepto	Costo por hora en \$	Total horas	Total en pesos
Ingeniero del conocimiento	545	264	143,880.00
Programador	380	96	36,480.00
Experto	500	296	148,000.00
Equipo de desarrollo			6,500.00
Licencia de desarrollo			9,273.00
Total			344,133.00

Figura C.1

Por equipo de desarrollo se entiende: mobiliario, energía eléctrica, hardware, papelería, inmueble, software, etc. Considerando un costo de \$5000.00 al mes para este concepto.

- Después del desarrollo:
 - Mantenimiento del sistema
 - Software y hardware sobre la que correrá la aplicación
 - Licencias y actualizaciones

Mantenimiento del sistema

El esquema de mantenimiento que se plantea es el siguiente:

El software se garantiza por 6 meses contra cualquier fallo atribuible al desarrollo, sin costo para el cliente y en caso de que la falla sea atribuible a problemas de definición por parte del cliente, la visita se cobrará a razón de \$545.00 por hora, al igual que la hora subsecuente, en tanto se resuelva la problemática.

Para el caso de mejoras o nuevos requerimientos se hará un nuevo análisis costo/beneficio.

Software y hardware sobre la que correrá la aplicación

Licencias y actualizaciones de software y hardware

Este costo se desprecia dado que el sistema correrá sobre la plataforma actual, lo que no implica gasto adicional.

Beneficios:

Se mencionaron los siguientes:

- No siempre se cuenta con un experto veterinario a la mano, cuando se necesita, sobre todo en el caso de comunidades alejadas, por lo que el beneficio para este sistema se estimará por un lado, basándonos en los ahorros al no tener que consultar a un experto en primera instancia y después de los beneficios de asegurar la supervivencia de la parvada.

Básicamente se podría decir que el beneficio consiste en ahorrar el costo de una consulta, sin embargo, no es sólo una consulta, sino tantas como aves se tengan o incluso tantas consultas como situaciones que requieran la presencia de un medico veterinario se presenten. Por otro lado al asegurar la supervivencia de la parvada, el beneficio puede estimarse desde el valor de un solo pollo hasta el valor de la parvada completa; es muy difícil de cuantificar sin tener un caso particular, por lo que dejaremos a este nivel nuestra estimación.

- Traslado desde la comunidad en la que se encuentra la granja hasta donde se encuentre el médico veterinario, o en su caso el costo de una visita a domicilio, además del que implicaría de la disponibilidad las 24 horas del día los 365 días del año.

Este concepto encierra gastos de transporte, hospedaje, alimentación y sobre todo disponibilidad, nuevamente tales cantidades en dinero requieren ejemplificarse para poder cuantificarse.

- El valor de contar con el conocimiento del experto almacenado y el aseguramiento de la permanencia y reproductibilidad del mismo.

Este es un concepto que tampoco se puede cuantificar; sin embargo es indudable que aporta mucho valor.

Glosario

Análisis clínico

EL análisis clínico consiste en una observación (inspección ocular) del paciente en este caso de los pollos tanto de forma colectiva (mediante un muestreo) como de forma individual (dentro de la misma muestra); a través de la cual se determina la posibilidad de que una de las aves, varias de ellas o incluso la parvada completa, pueda estar infectada de alguna enfermedad.

Aplicación productiva

Como es sabido una aplicación tiene varias etapas: concepción, análisis, desarrollo, etc. Una aplicación estará en su etapa productiva cuando esté lista para ser utilizada por un usuario final.

Asertar

Dentro de la terminología de Sistemas Expertos, el proceso de elegir un hecho y colocarlo en la memoria de trabajo del sistema.

Base de conocimiento

La base de conocimiento de un Sistema Experto, contiene lo concerniente a los hechos y experiencias de los expertos en un dominio determinado.

Cálculo de predicados

Lenguaje formal con sintaxis y gramática propias, capaz de valorar enunciados lógicos y extraer conclusiones para la creación de nuevos enunciados.

Componente de Adquisición

Ofrece ayuda a la estructuración e implementación del conocimiento en la base de conocimiento.

Componente explicativo

Comunica al usuario la estrategia de solución encontrada y el porqué de las decisiones tomadas.

Criterios Críticos

Elemento que establece las condiciones para evaluar una situación crítica.

Criterio de solución de conflictos

Elemento que nos permite tomar una decisión ante una situación particular.

Dominio

Las relaciones y dependencias de un área de conocimientos.

Hecho

Una declaración válida; en un Sistema Experto existe un hecho, cuando es leído por el mecanismo de inferencia y se reconoce como valido.

Ingeniero del conocimiento

Especialista que reúne el conocimiento especializado de un determinado tema y lo transfiere en una forma procesable por el ordenador; pregunta al experto para obtener el conocimiento, lo documenta y lo estructura, construyendo la base de conocimientos.

Inteligencia

Definirla es una problema filosófico y/o biológico, para fines del presente trabajo se describirá como inteligente: la presencia aparente de experiencia y conocimiento de causa en programas de ordenador.

Interfaz del usuario

Sirve para que éste pueda realizar una consulta en un lenguaje lo más natural posible, estableciendo la forma en que el Sistema Experto se presentará al usuario. La interfaz del usuario es el intermediario o enlace, entre el sistema y el usuario.

Mecanismo de inferencia

Es la unidad lógica con que se extraen conclusiones de la base de conocimientos, según un método fijo de solución de problemas que está configurado; imita el procedimiento de los expertos humanos para solucionar problemas. Componente de software que permite extraer conclusiones del contenido de las base de conocimientos; por ejemplo reglas y hechos.

Mecanismo de valoración

Componente de software que valora la información representada en la base de conocimientos; contiene, por ejemplo, al igual que componentes que extraen conclusiones y hechos, también componentes que valoran y aplican el conocimiento de control.

Memoria de trabajo

Dentro de la terminología de Sistemas Expertos así se llama al espacio de memoria en donde se almacenan todos los hechos elegidos por el usuario; las reglas se aplican sobre los los hechos presentes en la memoria de trabajo, hasta que no hay más por aplicar o bien no hay hechos que validar.

Motor de inferencia

Ver Mecanismo de inferencia

Portabilidad

Esta característica se refiere a la capacidad de la herramienta de software para instalarse en diferentes computadoras, sin necesidad de modificar ni el código y a su capacidad de generar aplicaciones que también puedan ejecutarse sobres diferentes plataformas.

Procedimiento de solución basado en conocimiento

Este procedimiento comprende las siguientes capacidades:

- Utilización de normas u otras estructuras que contengan conocimientos y experiencia de expertos especializados.
- Deducción lógica de conclusiones.
- Interpretación de datos ambiguos.
- Manipulación de conocimientos vagos; es decir, conocimientos afectados por valores de probabilidad.

Regla

Vease regla de producción.

Regla de producción

Dentro del concepto de Sistemas Expertos se utilizan como una forma de representación del conocimiento, a través de reglas de la forma si-entonces.

Shell

Herramienta utilizada por el Ingeniero del conocimiento en su trabajo. Los Shells incluyen componentes de un Sistema Experto, excepto la base de conocimientos; algunos sólo son herramientas para la estructuración del saber, es decir, programas para representar las relaciones del conocimiento. Otros disponen ya de uno o más mecanismos de representación del conocimiento y, en algunos casos, un mecanismo de inferencia, otros ofrecen, a su vez, posibilidades de control más amplias o ayudan en la configuración de la interfaz.

Sistemas Expertos

Bajo el término de Sistema Experto se entiende un nuevo tipo de software que imita el comportamiento de un experto humano en la solución de un problema. Pueden almacenar conocimientos de expertos para un campo determinado – y muy delimitado – y solucionar un problema mediante deducción lógica de conclusiones.

Slot

Componente de frame; a través de ellos pueden describirse las cualidades de un objeto.

Software portable

Ver portabilidad.

Bibliografía

David W. Rolston. Principles of Artificial Intelligence and expert systems development. Mc Graw Hill. USA 1988.

Dieter Nebendahl. Sistemas Expertos Parte 1. Marcombo. Barcelona España 1991.

Dieter Nebendahl. Sistemas Expertos Parte 2. Marcombo. Barcelona España 1991.

Corina Schmelkes. Manual para la presentación de anteproyectos e informes de investigación (Tesis). Colección Textos Universitarios en Ciencias Sociales. Editorial Harla. México 1988.

Clarence E. Bundy. La Producción avícola. Editorial Continental. México 1991.

Ivan R. Balconi. Topics of present and future interest for poultry industry. Editorial Midia Relaciones. Mexico 1995.

Tom Coleman. Expert Systems for the Data Processing Professional. NCC Publications. England 1989.

Enrique Castillo , Elena Alvarez. Sistemas Expertos aprendizaje e incertidumbre. Editorial Paraninfo. España 1989.

Brachman, Ronald J. and Levesque, Hector, J. Readings in Knowledge Representation, Editorial Morgan Kaufmann. Los Altos, California. USA 1985.

Hayes, Patrick J. The logic of frames. In Frame Conceptions and Text Understanding. Editorial D. Metzling, Walter de Gruyter and Co., Berlin 1979.